

Quel est le meilleur langage de programmation ?

Répondre à cette question est... impossible.

Chaque langage a ses propres caractéristiques qui vont le rendre performant sur un type de projet. À l'inverse, certaines technologies sont peu efficaces, voire incompatibles avec une application. Alors, comment choisir ? En optant pour les langages les plus populaires et utilisés par les développeurs. Cette notoriété signifie qu'ils sont suffisamment flexibles et puissants pour gérer les demandes et besoins les plus courants.

Un langage de programmation est mis en œuvre par un traducteur automatique

- Un compilateur
- Un interprète

Notion de compilateur

Un compilateur est un programme informatique qui traduit le code source écrit dans un langage de programmation en langage machine compréhensible par l'ordinateur. Le langage source est généralement un langage de haut niveau conçu pour être facile à lire et à écrire pour les humains, tandis que le langage machine est un langage de bas niveau conçu pour être exécuté par l'unité centrale de traitement (CPU) de l'ordinateur.

Le processus de compilation est composé de plusieurs étapes, notamment :

- **Analyse lexicale** : Cette étape divise le code source en jetons, qui sont les éléments constitutifs de base du langage.
- **Analyse syntaxique** : Cette étape vérifie si les jetons sont disposés dans l'ordre correct selon la grammaire du langage source.
- **Analyse sémantique** : Cette étape détermine la signification des jetons.
- **Génération de code intermédiaire** : Cette étape génère un code qui est indépendant de l'architecture de l'ordinateur cible.
- **Optimisation** : Cette étape améliore les performances du code intermédiaire.
- **Génération de code machine** : Cette étape génère le code machine final qui peut être exécuté par le processeur de l'ordinateur cible.

Les compilateurs sont utilisés pour compiler une grande variété de langages de programmation, notamment C, C++, Java, Python et Fortran. Les compilateurs sont un outil essentiel pour le développement de logiciels, car ils permettent aux programmeurs d'écrire du code dans des langages de haut niveau qui sont portables et efficaces.

Voici quelques-uns des avantages de l'utilisation de compilateurs :

- **Portabilité** : Le code compilé peut s'exécuter sur n'importe quel ordinateur doté du processeur approprié.
- **Efficacité** : Le code compilé peut s'exécuter beaucoup plus rapidement que le code **interprété**.
- **Sécurité** : Les compilateurs peuvent détecter et signaler les erreurs dans le code source.

- **Facilité d'utilisation** : Les compilateurs peuvent rendre le développement de logiciels plus facile et plus efficace.

Voici quelques-uns des inconvénients de l'utilisation de compilateurs :

- **Complexité** : Les compilateurs peuvent être des programmes très complexes.
- **Coût** : Les compilateurs commerciaux peuvent être coûteux.
- **Dépendance** : Les compilateurs peuvent dépendre d'un système d'exploitation ou d'une architecture de matériel spécifique.

Dans l'ensemble, les compilateurs sont un outil essentiel pour le développement de logiciels. Ils offrent de nombreux avantages, tels que la portabilité, l'efficacité et la sécurité. Cependant, ils peuvent également être complexes et coûteux.

Notion d'interprete

Un interprète, tout comme un compilateur, joue un rôle clé dans l'exécution de programmes informatiques, mais il le fait de manière différente. Voici un résumé des points principaux concernant les interprètes :

Qu'est-ce qu'un interprète ?

Un interprète est un programme qui traduit et exécute directement le code source écrit dans un langage de programmation, ligne par ligne. Contrairement à un compilateur qui traduit le code source entier en langage machine à l'avance, l'interprète effectue cette traduction et l'exécution de manière simultanée.

Comment fonctionne un interprète ?

- **Lecture du code source**: L'interprète lit le code source ligne par ligne.
- **Traduction**: Chaque ligne est traduite dans un format compréhensible par le processeur (généralement une représentation intermédiaire).
- **Exécution**: L'instruction traduite est immédiatement exécutée par le processeur.
- **Boucle**: L'interprète passe à la ligne suivante et répète les étapes 1 à 3 jusqu'à la fin du code.

Exemples de langages interprétés:

- Python
- Ruby
- JavaScript (dans les navigateurs web)
- PHP
-

Avantages de l'utilisation des interprètes:

- **Développement rapide et interactif**: L'exécution instantanée permet un développement et un test plus rapides.
- **Portabilité**: Le code interprété est généralement portable entre différents systèmes d'exploitation, car l'interprète lui-même s'adapte à la machine cible.
- **Flexibilité**: L'interprétation permet de modifier le code en cours d'exécution, ce qui est utile pour le scripting et les environnements dynamiques.

Inconvénients de l'utilisation des interprètes:

- **Efficacité**: Le code interprété peut être plus lent que le code compilé, car la traduction est effectuée à chaque exécution.
- **Dépendance à l'interprète**: Le programme dépend de la présence et de la compatibilité de l'interprète pour fonctionner.

- **Débogage:** Le débogage peut être plus difficile avec les interprètes, car il nécessite de suivre le code en temps réel.

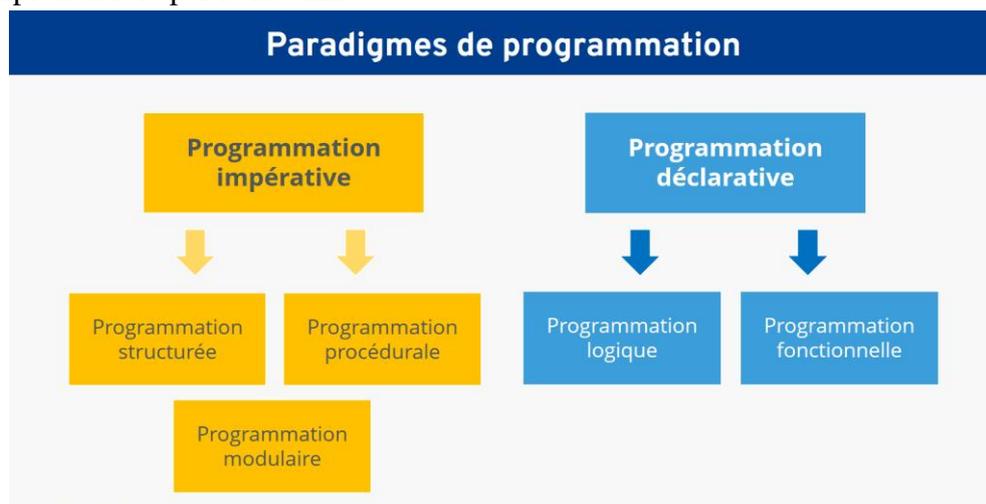
Comparatif compilateur vs interprète:

Caractéristique	Compilateur	Interprète
Traduction	Une fois pour toutes	Ligne par ligne
Exécution	Code machine direct	Code traduit en temps réel
Portabilité	Code compilé spécifique	

Un interprète, tout comme un compilateur, joue un rôle clé dans l'exécution de programmes informatiques, mais il le fait de manière différente. Voici un résumé des points principaux concernant les interprètes :

1 – Introduction : type de langage de programmation

Un **langage de programmation** est une notation conventionnelle destinée à formuler des algorithmes et produire des programmes informatiques qui les appliquent. D'une manière similaire à une langue naturelle, un langage de programmation est composé d'un alphabet, d'un vocabulaire, de règles de grammaire, de significations, mais aussi d'un environnement de traduction censé rendre sa syntaxe compréhensible par la machine



1 – 1- programmation imperative

En informatique, la **programmation impérative** est une méthode de pensée-(paradigme) qui décrit les opérations en séquences d'instructions exécutées par l'ordinateur pour modifier l'état du programme. Ce type de programmation est le plus répandu parmi l'ensemble des langages de programmation existants, et se différencie de la programmation déclarative (dont la programmation logique ou encore la programmation fonctionnelle sont des sous-ensembles

La plupart des langages de haut niveau comporte cinq types d'instructions de base principales :

1. la séquence d'instructions
2. l'assignation ou affectation
3. l'instruction conditionnelle
4. la boucle
5. les branchements

- **Le style de programmation structuré** étend le principe impératif de base à des **structures de contrôle** concrètes : séquences, sélection et itération (répétition). La maîtrise ou l'évitement complet des instructions de saut, qui rendent le code impératif inutilement compliqué, se font en arrière-plan.
- **L'approche procédurale** divise la tâche à exécuter en plusieurs sous-tâches, que le code doit alors décrire séparément. Les blocs de programmation ainsi créés peuvent ensuite être réutilisés au sein d'autres programmes. Le **modèle de programmation modulaire** va encore plus loin en concevant, développant et testant les différents composants du programme indépendamment les uns des autres. Ce n'est qu'à la fin que la combinaison des différents modules forme le logiciel proprement dit.

1 – 2 – programmation déclarative

C'est un paradigme qui consiste à créer des programmes sur la base de composants indépendants du contexte et sans état. Cette forme de programmation cherche à réduire les effets de bord en décrivant la tâche que le programme doit accomplir au lieu de décrire comment le programme doit accomplir cette tâche. Ainsi, on peut trouver une correspondance claire entre ce type de langage et la logique mathématique
exemple : SQL

1 – 3 – programmation fonctionnelle

C'est un dérivé de la programmation déclarative, ainsi on n'y retrouve pas d'état. Les données ne peuvent être manipulées que par des évaluations de fonctions mathématiques.

Le manque d'état induit une absence d'opération d'affectation. Cela permet aussi d'éviter tout effet de bord et permet de voir le programme comme une véritable application mathématique. Si un bug apparaît, il est très simple à repérer car il se trouve **forcément** dans la fonction ne présentant pas le résultat voulu. En effet, il n'y a pas de variables partagées ou d'état global.

Autrement dit, un langage fonctionnel offre la possibilité de passer par des fonctions pour atteindre le résultat voulu. Ces fonctions renverront **toujours** le même résultat pour les mêmes données en entrées sans modifier l'état potentiel qui se trouve à l'extérieur de ces fonction

Nota : Meta langage – ML

ML (contraction de *Meta Language*) est un langage de programmation généraliste **fonctionnel**.

ML a été standardisé en 1983, puis révisé en 1997 ; le langage résultant s'appelle Standard ML (SML), et a été notamment implémenté dans Standard ML of New Jersey (SML/NJ). En parallèle, des équipes françaises ont développé Caml, dont la popularité dans la communauté ML internationale est maintenant très grande..

OCaml (INRIA) ne suit pas le standard SML mais est plutôt une évolution orientée objet de ML. Ce langage se caractérise par un compilateur produisant des programmes disposant d'une très grande rapidité d'exécution ainsi qu'une bibliothèque importante. Les développeurs et les utilisateurs forment une communauté très active pour un langage issu de la recherche. OCaml a servi de base pour le développement du langage **F#** sur la plate-forme **.NET** de Microsoft.

1 – 4 – Programmation objet

La **programmation orientée objet (POO)**, ou **programmation par objet**, est un paradigme de programmation informatique. Elle consiste en la définition et l'interaction de briques logicielles appelées **objets** ; un objet représente un concept, une idée ou toute entité du monde physique, comme une voiture, une personne ou encore une page d'un livre. Il possède une structure interne et un comportement, et il sait interagir avec ses pairs. Il s'agit donc de représenter ces objets et leurs relations ; l'interaction entre les objets via leurs relations permet de concevoir et réaliser les fonctionnalités attendues, de mieux résoudre le ou les problèmes. Dès lors, l'étape de modélisation revêt une importance majeure et nécessaire pour la POO. C'est elle qui permet de transcrire les éléments du réel sous forme virtuelle.

On distingue dans les langages objets deux mécanismes du typage :

- **le typage dynamique** : le type des objets est déterminé à l'exécution lors de la création desdits objets (Smalltalk, Common Lisp, Python, PHP...) ;
- **le typage statique** : le type des objets est vérifié à la compilation et est soit explicitement indiqué par le développeur lors de leur déclaration (C++, Java, C#, Pascal...), soit déterminé par le compilateur à partir du contexte (Scala, OCaml...).

1 – 5 – programmation logique

La programmation logique est un type de paradigme de programmation qui est en grande partie basée sur la logique formelle. Tout programme écrit dans un langage de programmation logique est un ensemble de phrases sous forme logique, exprimant des faits et des règles concernant un domaine problématique.

Les principales familles de langages de programmation logique comprennent Prolog, la programmation d'ensembles de réponses (ASP) et Datalog. Dans toutes ces langues, les règles sont écrites sous la forme de clauses :

1 – 6 – Programmation graphique

Dans la programmation graphique, les éléments du langage de programmation sont disponibles sous la forme d'**éléments graphiques**, d'où le nom de programmation graphique. L'apparence et le nom de ces éléments permettent d'identifier d'emblée la **tâche qui sera ainsi réalisée lors de l'exécution du programme**. Les pictogrammes servent à aiguiller les programmeurs et permettent d'éviter les structures trop complexes ou les concepts trop abstraits. Le **style de programmation** fondamental – également appelé paradigme de programmation – présente notamment les propriétés suivantes :

- Visuel : les éléments sont insérés dans le cycle du programme par glisser-déposer.
- Orienté événement : chaque étape du programme commence lorsqu'un événement défini au préalable survient.
- Impératif : la programmation est composée d'une suite de commandes.
- Orienté objet : les différents objets se voient attribuer des tâches définies.

1 – 7 – langage de script

Un **langage de script** est un **langage** de programmation interprété qui permet de manipuler les fonctionnalités d'un système informatique configuré pour fournir à l'interpréteur de ce **langage** un environnement et une interface qui déterminent les possibilités de celui-ci.

Les langages compilés sont convertis en permanence en fichiers exécutables avant leur exécution. Au contraire, les langages de script sont généralement convertis en code machine à la volée, lors de l'exécution, par un programme nommé **interpréteur**. Bien que cette approche puisse donner lieu à des problèmes de performances, les instructions n'étant pas gérées uniquement par le processeur, elle facilite la compatibilité des scripts avec les programmes écrits dans d'autres langages. Les langages compilés sont convertis en permanence en fichiers exécutables avant leur exécution. Au contraire, les langages de script sont généralement convertis en code machine à la volée, lors de l'exécution, par un programme nommé interpréteur. Bien que cette approche puisse donner lieu à des problèmes de performances, les instructions n'étant pas gérées uniquement par le processeur, elle facilite la compatibilité des scripts avec les programmes écrits dans d'autres langages.

L'action de scripter consiste essentiellement à écrire une série de commandes qui sont interprétées une à une par une application ou un moteur de script. Même si le script guide la plateforme dans ses actions (en lui donnant un script à lire et à interpréter), l'exécution est effectuée par l'environnement d'exécution et non par le langage de script lui-même.

C'est ainsi que les langages de script sont différents des langages de programmation tels que Java, que l'on peut « écrire une fois, exécuter n'importe où » (slogan officiel de Java signifiant que les programmes Java peuvent s'exécuter en tant qu'applications autonomes dans n'importe quel environnement; depuis son invention, il est également devenu le principe WORA qui fait référence aux capacités multi-plateformes). WORA, qui est abrégé en Write Once Run Anywhere, est la fonctionnalité applicable aux programmes qui ont la capacité de s'exécuter sur n'importe quel système d'exploitation ou n'importe quelle machine. Sun **Microsystems a donné cette terminologie pour leur langage de programmation – Java**

Les principales différences entre les langages de script et de programmation sont donc les suivantes :

Langages de script	Langages de programmation
Spécifique à la plateforme	Plateforme agnostique (multi-plateforme)
Interprété (Le plus souvent)	Compilé
Plus rapide à l'exécution	Plus lent à l'exécution
Plus de codes	Moins de codes
Crée des applications autonomes	Crée des applications faisant partie d'une pile

Certains langage script ont évolué et offre des add-ons permettant de les considérer comme langage de programmation. Par exemple les langages ; Python ,Apple script, PHP, Autolt, Perl, Lisp peuvent etre compilés

1 – 8 – langage de balisage

Un langage de balisage est un type de langage utilisé pour annoter du texte et incorporer des balises dans des documents électroniques de style précis, quelle que soit la plate-forme informatique, le système d'exploitation, l'application ou le programme.

Le terme langage de balisage est dérivé du balisage des manuscrits, où les annotations manuscrites étaient annotées sous la forme d'instructions de l'imprimante. Les langages de balisage sont également utilisés dans les listes de lecture, les graphiques vectoriels, les services Web et les interfaces utilisateur. **HTML** est le langage de balisage le plus utilisé.

Il existe trois types de langage de balisage électronique:

- **Balisage de présentation** : utilisé par les systèmes de traitement de texte traditionnels avec WYSIWYG ; il est caché aux utilisateurs humains.
- **Marquage procédural** : intégré au texte pour fournir des instructions de traitement de texte aux programmes. Un tel texte est visiblement manipulé par l'auteur. Les systèmes de balisage procédural incluent des constructions de programmation, où des macros ou des sous-programmes sont définis et invoqués par leur nom.
- **Balisage descriptif** : utilisé pour étiqueter des parties d'un document sur la façon dont elles doivent être traitées. Par exemple, la balise HTML est utilisée pour étiqueter les citations dans le texte.

Les langages de balisage entrelacent généralement le texte du document avec les instructions de balisage dans le même flux de données ou de fichiers. Les codes entre crochets (<>) sont des instructions de balisage (également appelées balises) et le texte entre ces instructions est le texte du document réel. Les codes qui apparaissent près du début et de la fin de la première instruction sont appelés balisage sémantique et décrivent le texte inclus. En revanche, le balisage de présentation spécifie une caractéristique de texte particulière sans description

Quelques langages spécifiques de balisage

1 – 8 – 1 - Gencode

Gencode est un langage de balisage générique développé par IBM dans les années 1960. Il s'agissait de la première version de langage de balisage proposé par IBM et il a permis à l'entreprise de construire un système de gestion de toute sa documentation. 90 % des documents techniques d'IBM étaient écrits en Gencode.

Gencode est l'ancêtre du SGML et du HTML. Il a jeté les bases des langages de balisage modernes que nous utilisons aujourd'hui.

Voici quelques points clés à retenir sur Gencode :

Il s'agissait d'un langage de balisage hiérarchique. Les balises étaient imbriquées les unes dans les autres pour créer une structure arborescente.

Les balises étaient définies par des DTD (Document Type Definitions). Les DTD définissaient la structure et le contenu des balises.

Gencode était utilisé pour baliser des documents textuels. Il n'était pas conçu pour baliser des documents graphiques ou multimédias.

Gencode a été un langage de balisage important et influent. Il a permis de faire progresser le développement des langages de balisage et a contribué à jeter les bases du Web moderne.

1 – 8 – 2 - Latex

LaTeX est un puissant langage de balisage et un système de composition de documents largement utilisé dans le milieu scientifique et académique. Il s'agit d'un ensemble de macrocommandes qui facilitent l'utilisation du logiciel de composition de documents TeX, créé par Donald Knuth.

Voici quelques caractéristiques clés de LaTeX :

- **Séparation du fond et de la forme:** Contrairement aux traitements de texte classiques où la mise en forme est effectuée en même temps que l'écriture, LaTeX sépare le contenu (texte et formules) de la présentation visuelle. Cela permet une meilleure cohérence et une qualité d'impression supérieure.
- **Focus sur les aspects techniques:** LaTeX est particulièrement adapté à la composition de documents techniques et scientifiques, incluant des formules mathématiques complexes et des structures de document bien définies.
- **Utilisation de commandes:** Au lieu d'un éditeur WYSIWYG (ce que vous voyez est ce que vous obtenez), LaTeX nécessite l'écriture de commandes pour définir la structure du document, le style de texte, et l'insertion d'éléments comme les figures et les tableaux.
- **Flexibilité et personnalisation:** LaTeX offre un large éventail de paquets et de styles qui permettent de personnaliser l'apparence du document et d'ajouter des fonctionnalités spécifiques.

1 – 8 – 3 - Langage de balisage extensible (XML)

XML (Extensible Markup Language) est un langage de balisage qui définit un ensemble de règles pour coder des documents dans un format à la fois lisible par l'homme et lisible par machine. Les objectifs de conception de XML se concentrent sur la simplicité, la généralité et la convivialité sur Internet. C'est un format de données textuelles avec un fort support via Unicode pour différentes langues humaines. Bien que la conception de XML se concentre sur les documents, le langage est largement utilisé pour la représentation de structures de données arbitraires telles que celles utilisées dans les services Web.

- XML signifie **extensible Markup Language** (langage de balisage extensible)
- XML est un langage de balisage comme [HTML](#)
- XML est conçu pour stocker et transporter des données
- XML est conçu pour être autodéscriptif

1 – 8 – 4 - Langage de balisage généralisé (GML)

le Géographie Markup Language (GML) est un langage dérivé du XML pour encoder, manipuler et échanger des données géographiques. C'est un standard développé par l'Open Geospatial Consortium pour garantir l'interopérabilité des données dans le domaine de l'information géographique et de la géomatique.

Le GML consiste en un ensemble de schémas XML qui définissent un format ouvert pour l'échange de données géographiques et permettent de construire des modèles de données spécifiques pour des domaines spécialisés, comme l'urbanisme, l'hydrologie ou la géologie. ce langage ne doit pas être confondu avec **GML (generalizer markup language)** qui est un langage de script développé par IBM

Remarque : **GML (Generalized Markup Language)** **Ancêtre** du SGML et HTML

Développé par IBM dans les années 1960, il s'agissait de la première version de langage de balisage proposé. GML a permis à IBM de construire un système de gestion de toute sa documentation. 90 % des documents techniques d'IBM étaient écrits en GML..

1 – 8 - 5 -Langage de balisage généralisé standard (SGML)-HTML

SGML (Standard Generalized Markup Language) est un langage de balisage générique et normalisé, défini par l'ISO (norme ISO 8879:1986). Il permet de décrire la structure et le contenu de documents textuels de manière indépendante de leur présentation.

HTML (HyperText Markup Language) est un langage de balisage dérivé de SGML, conçu pour la création de pages web. Il permet de définir la structure et le contenu des pages web, ainsi que d'ajouter des liens hypertextes et des éléments multimédias.

Voici quelques points clés à retenir sur la relation entre SGML et HTML :

- **HTML est un sous-ensemble de SGML.** Cela signifie que tous les documents HTML sont valides selon les règles de SGML, mais tous les documents SGML ne sont pas valides selon les règles de HTML.
- **SGML est plus complexe que HTML.** Il offre plus de fonctionnalités et de flexibilité, mais il est également plus difficile à apprendre et à utiliser.
- **HTML est plus largement utilisé que SGML.** La plupart des pages web sont créées en HTML, tandis que SGML est utilisé pour des applications plus spécialisées.

Avantages de SGML:

- **Flexibilité et puissance:** Permet de créer des langages de balisage spécifiques à des domaines d'application précis.
- **Standardisation:** Norme internationale garantissant la compatibilité entre différents systèmes.
- **Pérennité:** Structure indépendante de la présentation, facilitant l'adaptation aux nouvelles technologies.

Avantages de HTML:

- **Simplicité:** Langage facile à apprendre et à utiliser.
- **Large adoption:** Largement utilisé pour la création de pages web.
- **Richesse fonctionnelle:** Permet d'ajouter des liens hypertextes, des images, des vidéos, etc.

Conclusion:

SGML et HTML sont deux langages de balisage importants. SGML est un langage générique et puissant, tandis que HTML est un langage plus simple et plus largement utilisé pour la création de pages web. Le choix du langage dépend des besoins spécifiques de l'application.

1 – 8 – 6- Initiative TEI

La Text Encoding Initiative (abrégé en TEI, en français « initiative pour l'encodage du texte ») est un format de balisage et une communauté académique internationale dans le champ des [humanités numériques](#) visant à définir des recommandations pour l'encodage de ressources numériques, et plus particulièrement de [documents textuels](#). Depuis 1987, le modèle théorique s'est adapté à différentes technologies, d'abord sous la forme d'une [DTD SGML](#), puis [XML](#). Dans sa version P5 (2007), le schéma TEI est représenté dans plusieurs langages, et notamment, [Relax NG](#)

1 – 9 – Langage des feuilles de style

En [typographie](#), une **feuille de style** est l'ensemble d'[attributs de caractères](#) et de formats de paragraphes pouvant être appliqués en une seule opération à un ou plusieurs paragraphes, à un livre, etc. De façon extensive, c'est l'ensemble des contraintes typographiques, de maquette, couleur, reliure, etc. qui définissent l'unité d'une [collection littéraire](#), d'une [revue](#) périodique, etc.

C'est aussi un document permettant de mettre en forme un autre document, comme un document rédigé dans un [langage de balisage](#), comme [SGML](#), [HTML](#) ou [XML](#). Pour ce faire, les feuilles de style sont rédigées dans un langage spécifique. L'objectif principal des feuilles de style est de séparer autant que possible la présentation et le contenu d'une page. Elles peuvent aussi permettre d'adapter automatiquement la présentation d'une page en fonction du média servant à l'afficher. On peut citer les avantages suivants :

- La structure du document et la présentation peuvent être gérées dans des fichiers séparés.
 - La conception d'un document peut se faire sans se soucier de la présentation
1. Dans le cas d'un site web, la présentation est uniformisée : les documents (pages [HTML](#)) font référence aux mêmes feuilles de style. Cette caractéristique permet, de plus, une remise en forme rapide de l'aspect visuel.
 - Pour un même document, on peut donner le choix entre plusieurs feuilles de style, par exemple une pour l'impression et une pour la lecture à l'écran.

Les langages de feuilles de style les plus courants sont :

1– 9 -1 - CSS (Cascading Style Sheets)

CSS est un langage informatique qui permet de définir la présentation des documents HTML et XML. Il s'agit d'un élément essentiel du Web moderne, car il permet de contrôler l'apparence des pages web, y compris la police de caractères, la couleur, la taille des éléments, la disposition, etc.

Voici quelques points clés à retenir sur CSS :

- **Fonctionnement** : Les feuilles de style CSS sont des fichiers externes qui sont liés aux documents HTML. Elles définissent des règles de style qui sont appliquées aux éléments du document en fonction de leur sélecteur.
- **Avantages** : Permet de séparer le contenu (HTML) de la présentation (CSS), ce qui facilite la maintenance et la mise à jour des pages web.
- **Flexibilité** : Offre un large éventail de propriétés et de valeurs pour contrôler l'apparence des éléments.
- **Compatibilité** : Largement pris en charge par tous les navigateurs web modernes.

Utilisations de CSS :

- **Mise en forme de texte** : Définir la police de caractères, la taille, la couleur, le style (gras, italique, etc.)
- **Mise en page** : Définir les marges, le padding, l'alignement, le positionnement des éléments
- **Ajout d'effets visuels** : Définir des bordures, des arrière-plans, des ombres, des animations
- **Responsive design** : Adapter la mise en page des pages web aux différents supports (ordinateurs, tablettes, smartphones)

Conclusion:

CSS est un langage essentiel pour tout développeur web. Il permet de créer des pages web attractives et fonctionnelles, et de les adapter aux différents supports.

1- 9 – 2 - XSL (Extensible Stylesheet Language)

XSL est un langage de programmation qui permet de transformer des documents XML en d'autres formats, tels que HTML, PDF ou texte brut. Il est composé de deux parties principales :

- **XSLT (XSL Transformations)**: Permet de transformer la structure et le contenu d'un document XML.
- **XSL-FO (XSL Formatting Objects)**: Permet de définir la mise en page du document final.

Voici quelques points clés à retenir sur XSL :

- **Fonctionnement** : Les feuilles de style XSL sont des fichiers externes qui sont liés aux documents XML. Elles définissent des règles de transformation qui sont appliquées aux éléments du document en fonction de leur sélecteur.
- **Avantages** : Permet de séparer le contenu (XML) de la présentation (XSL), ce qui facilite la maintenance et la mise à jour des documents.
- **Flexibilité** : Offre un large éventail de fonctionnalités pour transformer et mettre en forme les documents XML.
- **Compatibilité** : Largement pris en charge par les navigateurs web modernes et les logiciels de traitement de documents XML.

Utilisations de XSL :

- **Transformation de documents XML en HTML** : Permet de créer des pages web dynamiques à partir de données XML.
- **Création de rapports et de documents PDF** : Permet de générer des documents formatés à partir de données XML.
- **Conversion de données XML entre différents formats** : Permet de convertir des données XML d'un format à un autre.

Outils pour travailler avec XSL :

- **Transformateurs XSLT** : De nombreux logiciels et bibliothèques permettent de transformer des documents XML à l'aide de XSLT, comme Saxon ou Xalan.
- **Éditeurs de code** : De nombreux éditeurs de code proposent des fonctionnalités spécifiques pour faciliter le développement XSL, comme la coloration syntaxique et l'auto-complétion.

Conclusion :

XSL est un langage puissant et polyvalent pour transformer et mettre en forme des documents XML. Il est utilisé dans de nombreux domaines, tels que la publication web, la gestion de documents et l'échange de données.

1– 9 – 3 - DSSSL : Langage de feuille de style pour SGML

DSSSL signifie Document Style Semantics and Specification Language. Il s'agit d'un langage de feuille de style utilisé pour formater et transformer des documents SGML (Standard Generalized Markup Language).

Points clés à retenir :

- **Langage de transformation puissant**: DSSSL est un langage de programmation complet, capable de transformations complexes sur les documents SGML.
- **Antécédent de XSL**: DSSSL a été un précurseur important de XSL (Extensible Stylesheet Language), le langage de feuille de style standard pour XML.
- **Moins populaire aujourd'hui**: XSL a largement supplanté DSSSL, mais il reste utile pour certains cas spécifiques.

Fonctionnalités de DSSSL :

- **Formatage de documents:** Définition de la mise en page, des polices, des couleurs, etc.
- **Transformation de documents:** Conversion de documents SGML en d'autres formats, comme HTML ou PDF.
- **Génération de contenu:** Ajout de contenu dynamique aux documents SGML.

Avantages de DSSSL :

- **Puissant et flexible:** Permet de réaliser des transformations complexes.
- **Précis:** Offre un contrôle précis sur la mise en forme et le contenu.
- **Langage de programmation complet:** Offre des fonctionnalités avancées pour les développeurs.

Inconvénients de DSSSL :

- **Syntaxe complexe:** Le langage peut être difficile à apprendre et à utiliser.
- **Moins populaire que XSL:** La communauté d'utilisateurs et de développeurs est plus petite.
- **Moins de support:** Les outils et les bibliothèques DSSSL sont moins nombreux que pour XSL.

Cas d'utilisation de DSSSL :

- Formats de documents complexes : Documents avec des structures et des exigences de mise en forme complexes.
- Transformation de données : Conversion de données SGML vers d'autres formats.
- Génération de rapports : Création de rapports dynamiques à partir de données SGML.

1– 10 – langage de requête

Un **langage de requête** est un langage informatique utilisé pour accéder aux données d'une base de données ou d'autres systèmes d'information. Il permet d'obtenir les données vérifiant certaines conditions (on parle de critères de sélection), comme toutes les personnes qui habitent une ville donnée. Les données peuvent être triées, elles peuvent également être regroupées suivant les valeurs d'une donnée particulière (par exemple on va regrouper toutes les personnes qui habitent la même rue).

La grammaire d'un langage de requête est adaptée à la structure des données interrogées. Le langage de requête le plus connu et le plus utilisé est SQL

Par ordre alphabétique :

- **Datalog** pour les bases de données déductives
- **DMX** pour les modèles d'exploration de données (Data Mining)
- **MDX** pour les bases de données multidimensionnelles OLAP
- **OQL** pour les bases de données orientées objet
- **Prolog**
- **SPARQL** pour les graphes RDF
- **SQL** pour les bases de données relationnelles
- **XQuery** pour les données XML
- **XPath** pour parcourir le DOM
- **QBE** (Query By Example)- langage IBM

1 -10- 1- Datalog : Langage de requête et de règles pour les bases de données déductives

Datalog est un langage de requête déclaratif et logique utilisé pour interroger et manipuler des données dans des bases de données déductives. Il s'agit d'un sous-ensemble de Prolog, un langage de programmation logique.

Points clés à retenir :

- **Déclaratif**: Exprime les requêtes en termes de relations et de règles logiques, sans spécifier l'ordre d'exécution.
- **Logique**: Permet de déduire de nouvelles informations à partir de faits connus.
- **Bases de données déductives**: Permet de stocker des faits et des règles logiques pour dériver de nouvelles connaissances.

Fonctionnalités de Datalog :

- **Requêtes de base**: Accéder aux données stockées dans la base de données.
- **Règles déductives**: Définir des règles pour déduire de nouvelles informations à partir de faits connus.
- **Récursivité**: Définir des règles qui se réfèrent à elles-mêmes, permettant des requêtes complexes.

Avantages de Datalog :

- **Concision** : Permet d'exprimer des requêtes complexes de manière concise et lisible.
- **Déclaratif** : Facilite la compréhension et la maintenance des requêtes.
- **Logique** : Permet de raisonner sur les données et de déduire de nouvelles informations.

Inconvénients de Datalog :

- **Moins puissant que SQL**: Ne peut pas gérer toutes les requêtes SQL complexes.
- **Pas de prise en charge directe des transactions**: Nécessite des solutions de contournement pour gérer les modifications de données.
- **Moins populaire que SQL**: La communauté d'utilisateurs et de développeurs est plus petite.

Cas d'utilisation de Datalog :

- **Bases de données de connaissances**: Stocker et interroger des connaissances dans des domaines comme la médecine, la finance ou la biologie.
- **Systèmes d'inférence de règles**: Détecter des fraudes, recommander des produits ou prédire des événements.
- **Vérification de programmes**: Analyser des programmes informatiques pour détecter des erreurs..

1 – 10 – 2 -Le langage de requête DMX

DMX (Data Mining Extensions) est un langage utilisé pour créer et exploiter des modèles d'exploration de données dans Microsoft SQL Server Analysis Services. Il offre un ensemble d'instructions, de fonctions et d'opérateurs pour :

1. Définir et manipuler des structures et modèles d'exploration de données:

Créer et modifier des structures de données (tables, cubes, etc.)
Importer et exporter des modèles et des structures
Définir des algorithmes d'exploration de données
Supprimer des modèles existants

2. Extraire des données des modèles d'exploration de données:

Requêtes de prédiction: Prédire des valeurs pour de nouveaux cas
Requêtes de découverte de connaissances: Identifier des patterns et des tendances dans les données
Requêtes de classification: Classer des cas dans des catégories prédéfinies
Requêtes d'association: Déterminer des relations entre les attributs

3. Analyser les résultats d'exploration de données:

Explorer les modèles et les prédictions
Visualiser les résultats
Interpréter les données

Syntaxe DMX:

La syntaxe de DMX est similaire à celle du langage SQL. Elle est composée d'instructions, de clauses et d'expressions.

1 – 10 – 3 -Le langage de requête MDX

MDX (Multidimensional Expressions) est un langage de requête utilisé pour interroger et manipuler des données multidimensionnelles stockées dans des cubes OLAP (Online Analytical Processing). Il est similaire au langage SQL pour les bases de données relationnelles, mais avec des fonctionnalités spécifiques pour gérer les structures et les concepts propres aux cubes OLAP.

Voici quelques-unes des principales fonctionnalités de MDX :

- **Sélection et filtrage de données:** Vous pouvez sélectionner des cellules spécifiques, des membres de dimension ou des ensembles de cellules à partir d'un cube OLAP. Vous pouvez également filtrer les données en fonction de critères spécifiques.
- **Agrégation de données:** Vous pouvez calculer des agrégats tels que des sommes, des moyennes, des médianes, etc., sur des ensembles de données multidimensionnelles.
- **Navigation dans les dimensions:** Vous pouvez naviguer dans les différentes dimensions d'un cube OLAP pour explorer les données selon différents points de vue.
- **Calculs et expressions:** Vous pouvez créer des expressions complexes pour effectuer des calculs sur les données du cube OLAP.
- **Création de rapports et de tableaux de bord:** Vous pouvez utiliser MDX pour créer des rapports et des tableaux de bord dynamiques qui présentent les données du cube OLAP de manière interactive.

1 – 10 – 4 -Le langage de requête OOL

OOL (Object-Oriented Language) n'est pas un langage de requête standardisé et unique. Il peut faire référence à plusieurs langages de requête qui adoptent une approche orientée objet pour la manipulation de données. Voici quelques exemples de langages de requête OOL :

1. OQL (Object Query Language):

Développé par l'Object Data Management Group (ODMG).
Utilisé pour interroger des bases de données orientées objet.
Permet de naviguer dans les relations entre les objets et d'extraire des données.

2. XQuery:

Langage de requête standardisé pour XML.
Permet de sélectionner, de filtrer et de transformer des données XML.
Offre des fonctionnalités pour manipuler des structures hiérarchiques et imbriquées.

3. LINQ (Language Integrated Query):

Intégré dans le framework .NET de Microsoft.

Permet d'interroger des collections d'objets de manière simple et concise.

Offre une syntaxe expressive et familière aux développeurs .NET

4. M (Power Query Language):

Langage de requête utilisé dans Microsoft Power BI et Power Query.

Permet de transformer et de charger des données à partir de sources variées.

Offre une interface graphique intuitive pour la création de requêtes.

Points importants à retenir :

Il n'y a pas de langage de requête OOL unique et universel.

Chaque langage OOL possède ses propres caractéristiques et fonctionnalités.

Le choix du langage OOL dépend du contexte et des besoins spécifiques.

En plus des exemples cités ci-dessus, d'autres langages de requête peuvent être considérés comme OOL, tels que :

- FQL (Facebook Query Language)
- HiveQL (Hive Query Language)
- Pig Latin (Apache Pig)

1 – 10 – 5 - Le langage de requête Prolog

Prolog (Programmation Logique) est un langage de requête déclaratif utilisé pour l'intelligence artificielle et le traitement du langage naturel. Il se base sur la logique formelle pour décrire les relations entre les faits et les règles, permettant ainsi de déduire de nouvelles connaissances à partir d'une base de connaissances existante.

Voici quelques-unes des principales fonctionnalités de Prolog :

- **Déclaration de faits:** Vous pouvez définir des faits qui décrivent des informations statiques du monde réel.
- **Définition de règles:** Vous pouvez créer des règles qui décrivent des relations logiques entre les faits.
- **Interrogation de la base de connaissances:** Vous pouvez poser des questions sur les faits et les règles pour obtenir des déductions et des conclusions.
- **Programmation logique:** Vous pouvez écrire des programmes qui implémentent des algorithmes de recherche et de résolution de problèmes.

1 – 10 – 6 -Le langage de requête SPARQL

SPARQL (SPARQL Protocol and RDF Query Language) est un langage de requête standardisé pour interroger et manipuler des données stockées dans le format RDF (Resource Description Framework). RDF est un modèle de données utilisé pour représenter des informations sur le Web sémantique.

Voici quelques-unes des principales fonctionnalités de SPARQL :

- **Sélection de données:** Vous pouvez sélectionner des triplets RDF spécifiques ou des ensembles de triplets à partir d'un graphe RDF.

- **Filtrage de données:** Vous pouvez filtrer les données en fonction de critères spécifiques.
- **Agrégation de données:** Vous pouvez calculer des agrégats tels que des sommes, des moyennes, des médianes, etc., sur des ensembles de données RDF.
- **Construction de graphes RDF:** Vous pouvez créer de nouveaux graphes RDF à partir de données existantes.
- **Manipulation de vocabulaires RDF:** Vous pouvez définir et utiliser des vocabulaires RDF pour organiser et enrichir les données.

Points importants à retenir :

SPARQL est un langage de requête puissant pour le Web sémantique.

Il permet d'interroger et de manipuler des données RDF.

La syntaxe de SPARQL est basée sur des expressions booléennes et des motifs de graphes.

1 – 10 – 7 - Le langage de requête SQL

SQL (Structured Query Language) est un langage de requête standardisé utilisé pour interroger et manipuler des données stockées dans des bases de données relationnelles. Il est largement utilisé dans le monde entier par les développeurs, les analystes de données et les scientifiques des données.

Voici quelques-unes des principales fonctionnalités de SQL :

- **Sélection de données:** Vous pouvez sélectionner des colonnes spécifiques ou des ensembles de lignes à partir d'une table.
- **Filtrage de données:** Vous pouvez filtrer les données en fonction de critères spécifiques.
- **Agrégation de données:** Vous pouvez calculer des agrégats tels que des sommes, des moyennes, des médianes, etc., sur des ensembles de données.
- **Tri des données:** Vous pouvez trier les données par une ou plusieurs colonnes.
- **Modification de données:** Vous pouvez ajouter, modifier et supprimer des données dans une table.
- **Création de tables:** Vous pouvez créer de nouvelles tables pour stocker des données.
- **Définition de relations entre les tables:** Vous pouvez définir des relations entre les tables pour organiser les données.

Points importants à retenir :

SQL est un langage de requête puissant et universel.

Il permet d'interroger et de manipuler des données dans des bases de données relationnelles.

La syntaxe de SQL est simple et intuitive.

1 – 10 – 8 Le langage de requête XQuery

XQuery est un langage de requête standardisé pour interroger et manipuler des données XML. Il est conçu pour exploiter la structure hiérarchique et arborescente des documents XML, et offre des fonctionnalités puissantes pour :

1. Extraire des informations :

- **Sélection de nœuds:** Vous pouvez sélectionner des éléments, des attributs, des commentaires et des espaces blancs spécifiques à partir d'un document XML.

- **Filtrage de données:** Vous pouvez filtrer les données en fonction de critères spécifiques.
- **Navigation dans les documents:** Vous pouvez naviguer dans la structure arborescente du document XML pour explorer les données.

2. Transformer des données:

- **Tri des données:** Vous pouvez trier les données par une ou plusieurs valeurs.
- **Agrégation de données:** Vous pouvez calculer des agrégats tels que des sommes, des moyennes, des médianes, etc., sur des ensembles de données XML.
- **Construction de nouveaux documents:** Vous pouvez créer de nouveaux documents XML à partir de données existantes.

3. Manipuler des structures XML:

- **Ajout de nœuds:** Vous pouvez ajouter de nouveaux éléments, attributs et commentaires à un document XML.
- **Suppression de nœuds:** Vous pouvez supprimer des éléments, attributs et commentaires d'un document XML.
- **Modification de nœuds:** Vous pouvez modifier les valeurs des éléments et attributs d'un document XML.

Points importants à retenir :

- XQuery est un langage de requête puissant et flexible pour les données XML.
- Il permet d'extraire, de transformer et de manipuler des données XML.
- La syntaxe de XQuery est basée sur **XPath** et est donc familière aux développeurs XML

1 –10 – 9 - Le langage de requête XPath

XPath (XML Path Language) est un langage de requête standardisé pour localiser des portions d'un document XML. Il est utilisé conjointement avec d'autres langages comme XSLT et XQuery pour naviguer et extraire des données spécifiques d'un document XML.

Voici quelques-unes des principales fonctionnalités de XPath :

- **Sélection de nœuds:** Vous pouvez sélectionner des éléments, des attributs, des commentaires et des espaces blancs spécifiques à partir d'un document XML.
- **Filtrage de données:** Vous pouvez filtrer les données en fonction de critères spécifiques.
- **Navigation dans les documents:** Vous pouvez naviguer dans la structure arborescente du document XML pour explorer les données.
- **Tests logiques:** Vous pouvez effectuer des tests logiques pour comparer des valeurs et prendre des décisions.
- **Manipulation de chaînes:** Vous pouvez manipuler des chaînes de caractères pour extraire des informations spécifiques.

Points importants à retenir :

XPath est un langage de requête simple et puissant pour les documents XML.

Il est utilisé pour naviguer et extraire des données spécifiques d'un document XML.

La syntaxe de XPath est concise et facile à apprendre.

De nombreuses ressources sont disponibles pour apprendre XPath

1 – 10 – 10 - Le langage de requête QBE

QBE (Query By Example) est un langage de requête graphique utilisé pour interroger des bases de données relationnelles. Il permet de construire des requêtes en remplissant un tableau avec des exemples de données, ce qui le rend accessible aux utilisateurs non techniciens.

Voici quelques-unes des principales fonctionnalités de QBE :

- **Construction de requêtes par l'exemple:** Vous pouvez créer des requêtes en fournissant des exemples des données que vous souhaitez obtenir.
- **Sélection de champs:** Vous pouvez sélectionner les champs que vous souhaitez afficher dans les résultats de la requête.
- **Filtrage de données:** Vous pouvez filtrer les données en fonction de critères spécifiques.
- **Tri des données:** Vous pouvez trier les données par une ou plusieurs colonnes.
- **Agrégation de données:** Vous pouvez calculer des agrégats tels que des

Avantages de QBE :

- **Facile à utiliser:** QBE est un langage de requête intuitif et accessible aux utilisateurs non techniciens.
- **Visuel:** La construction de requêtes par l'exemple permet de visualiser facilement les données et les relations entre les tables.
- **Productif:** QBE permet de créer des requêtes complexes rapidement et facilement.

Inconvénients de QBE :

- **Moins puissant que SQL:** QBE ne permet pas de créer des requêtes aussi complexes que SQL.
- **Peu flexible:** QBE peut être moins flexible que SQL pour certains types de requêtes.

QBE est un langage de requête utile pour les utilisateurs qui ont besoin d'un moyen simple et intuitif d'interroger des bases de données relationnelles.

Points importants à retenir :

QBE est un langage de requête graphique pour les bases de données relationnelles.

Il permet de construire des requêtes en remplissant un tableau avec des exemples de données.

QBE est facile à utiliser et accessible aux utilisateurs non techniciens.

QBE est moins puissant et flexible que SQL pour certains types de requêtes..

2 – Classement des meilleurs langages de programmation (étude YeePLY)

2 – 1 - Python

Python est l'un des langages de programmation les plus utilisés aujourd'hui et la tendance continue d'augmenter. Il a tout pour plaire : c'est un logiciel libre, une syntaxe simple et facile à comprendre, ce qui permet **d'économiser du temps et des ressources**.

C'est l'un des meilleurs pour commencer dans le monde de la programmation. Python est un langage polyvalent qui admet de multiples applications. L'une d'entre elles est l'Intelligence Artificielle, grâce à des bibliothèques telles que Keras ou TensorFlow. Il peut également être utile

pour les applications Big Data, grâce aux bibliothèques de traitement de données. Ce langage de programmation est également utilisé dans le développement web, notamment grâce à ses frameworks Django ou Flask. Par exemple, les sites web SemRush ou Reddit sont développés avec Python.

Voici quelques-unes des principales fonctionnalités de Python :

- **Syntaxe simple et claire:** Le code Python est facile à lire et à écrire, ce qui le rend accessible aux débutants.
- **Interprété:** Python n'a pas besoin d'être compilé avant d'être exécuté, ce qui permet un développement rapide et interactif.
- **Typage dynamique:** Le type des variables n'est pas déclaré à l'avance, ce qui rend le code plus flexible.
- **Bibliothèques standard:** Python dispose d'une vaste collection de bibliothèques standard pour de nombreuses tâches courantes.
- **Grande communauté:** Python a une communauté active et solidaire qui offre de nombreuses ressources et du soutien.

Avantages de Python :

- **Facile à apprendre:** Python est l'un des langages de programmation les plus faciles à apprendre, ce qui le rend idéal pour les débutants.
- **Polyvalent:** Python peut être utilisé pour une grande variété de tâches, ce qui en fait un langage de programmation très utile.
- **Puissant:** Python est un langage de programmation puissant qui peut être utilisé pour des projets complexes.
- **Gratuit et open source:** Python est un langage de programmation gratuit et open source, ce qui signifie qu'il peut être utilisé par tout le monde.

Inconvénients de Python :

- **Peut être lent:** Python peut être plus lent que d'autres langages de programmation pour certaines tâches.
- **Pas toujours le meilleur choix pour les applications mobiles:** Python n'est pas toujours le meilleur choix pour les applications mobiles, car il peut être plus gourmand en ressources que d'autres langages.
- **Python est un langage de programmation excellent** pour les débutants et les développeurs expérimentés. Il est polyvalent, puissant et facile à apprendre. Si vous recherchez un langage de programmation pour apprendre ou pour un nouveau projet, Python est une excellente option.

2 – 2 - C C : un langage de programmation puissant et polyvalent

C'est un langage de programmation généraliste, impératif et de bas niveau. Inventé au début des années 1970 pour réécrire Unix, C est devenu l'un des langages les plus populaires, encore de nos jours.

Points forts du C :

- **Puissant :** C offre un contrôle précis sur le matériel et la mémoire, ce qui le rend idéal pour le développement de logiciels système et d'applications qui nécessitent des performances élevées.
- **Polyvalent :** C peut être utilisé pour créer une grande variété d'applications, des systèmes d'exploitation aux jeux vidéo.
- **Efficace :** Le code C est généralement compilé en code machine très efficace, ce qui permet d'obtenir des performances élevées.

- **Portable** : Le code C est portable sur la plupart des plateformes,
- **Largement utilisé** : C est l'un des langages les plus populaires au monde, ce qui signifie qu'il existe une grande communauté de développeurs et de nombreuses ressources disponibles.

Conclusion :

C est un langage de programmation puissant et polyvalent qui peut être utilisé pour créer une grande variété d'applications. Si vous recherchez un langage qui vous donne un contrôle précis sur le matériel et la mémoire, et qui est capable de produire des logiciels très performants, C est un excellent choix.

2 – 3 – JAVA

Java est un **langage de programmation orienté objet à usage général**, conçu pour avoir le moins de dépendances d'implémentation possible. Avec ce langage de programmation, vous pouvez créer des **applications et des processus sur plusieurs appareils**.

Son champ d'application est très large, ce qui vous permet de créer des logiciels pour les **appareils mobiles**, les terminaux de vente, les distributeurs automatiques de billets, les **IoT** (Internet des objets), ainsi que les **pages web**.

Java était le langage par défaut pour développer des **applications pour Android** et, bien que **Kotlin** gagne du terrain jour après jour, Java est toujours très nécessaire.

En ce qui concerne les **jeux mobiles**, les premiers ont été développés avec Java. Aujourd'hui, Java continue à jouer un rôle important dans le monde des jeux vidéo, grâce à des Framework tels que **JMonkeyEngine** ou **Ardor 3D**.

Caractéristiques principales :

- **Orienté objet**: Java organise le code en objets, ce qui facilite la réutilisation et la maintenance du code.
- **Multiplateforme**: Le code Java peut être exécuté sur n'importe quelle plateforme disposant d'une machine virtuelle Java (JVM), ce qui le rend très portable.
- **Robuste**: Java est conçu pour être fiable et sécurisé, ce qui en fait un choix populaire pour les applications critiques.
- **Performant**: Java est un langage performant qui peut être utilisé pour développer des applications à grande échelle.
- **Largement utilisé**: Java est l'un des langages de programmation les plus populaires au monde, avec des millions de développeurs et d'utilisateurs.

Utilisations :

Java est utilisé pour développer une grande variété d'applications, notamment :

- **Applications web**: Java est un langage populaire pour le développement d'applications web back-end et d'applets.
- **Applications mobiles**: Java est le langage principal pour le développement d'applications Android.
- **Logiciels d'entreprise**: Java est largement utilisé pour développer des logiciels d'entreprise, tels que des systèmes de gestion de contenu (CMS) et des logiciels de gestion intégrés (ERP).
- **Jeux**: Java est utilisé pour développer des jeux vidéo, en particulier des jeux mobiles.

- **Internet des objets (IoT):** Java est de plus en plus utilisé pour développer des applications pour l'IoT.

Avantages :

- **Portable:** Le code Java peut être exécuté sur n'importe quelle plateforme disposant d'une JVM.
- **Robuste:** Java est un langage fiable et sécurisé.
- **Performant:** Java est un langage performant.
- **Largement utilisé:** Java est un langage populaire avec une grande communauté de développeurs.
- **Grand nombre de bibliothèques:** Il existe un grand nombre de bibliothèques Java disponibles pour une variété de tâches.

Inconvénients :

- : Java est un langage complexe à apprendre.
- Pas aussi performant Peut être complexe que certains autres langages: Java n'est pas toujours le langage le plus performant pour certaines tâches.
- **Consommation de mémoire:** Les applications Java peuvent consommer beaucoup de mémoire.

2 – 4– JAVASCRIPT

JavaScript est un langage de programmation de scripts principalement utilisé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript est au cœur des langages utilisés par les développeurs web.

Caractéristiques principales :

- **Interprété :** Le code JavaScript est interprété par le navigateur web, ce qui signifie qu'il n'a pas besoin d'être compilé avant d'être exécuté.
- **Dynamique :** JavaScript est un langage à typage dynamique, ce qui signifie que le type d'une variable n'est pas défini à l'avance.
- **Orienté objet :** JavaScript est un langage orienté objet, ce qui permet de créer des objets et des classes.
- **Multiplateforme :** JavaScript peut être utilisé sur n'importe quelle plateforme disposant d'un navigateur web.
- **Largement utilisé :** JavaScript est l'un des langages de programmation les plus populaires au monde.

Utilisations :

JavaScript est utilisé pour :

- **Ajouter de l'interactivité aux pages web:** JavaScript permet de créer des animations, des jeux, des formulaires interactifs, etc.
- **Développer des applications web côté client:** JavaScript peut être utilisé pour développer des applications web complètes qui s'exécutent dans le navigateur web.
- **Développer des applications mobiles :** JavaScript peut être utilisé pour développer des applications mobiles hybrides avec des frameworks comme React Native.

- **Développer des serveurs web** : JavaScript peut être utilisé pour développer des serveurs web avec Node.js.
- **Développer des jeux vidéo** : JavaScript peut être utilisé pour développer des jeux vidéo simples ou plus complexes.

Avantages:

- **Facile à apprendre**: JavaScript est un langage relativement facile à apprendre, surtout par rapport à d'autres langages de programmation.
- **Polyvalent** : JavaScript peut être utilisé pour une grande variété de tâches.
- **Largement utilisé** : JavaScript est un langage populaire avec une grande communauté de développeurs.
- **Grand nombre de bibliothèques** : Il existe un grand nombre de bibliothèques JavaScript disponibles pour une variété de tâches.

Inconvénients :

- **Peut-être lent**: JavaScript peut être lent par rapport à d'autres langages de programmation.
- **Difficile à déboguer**: Le code JavaScript peut être difficile à déboguer.
- **Problèmes de sécurité**: JavaScript peut être utilisé pour créer des attaques de sécurité.

Conclusion :

JavaScript est un langage de programmation puissant et polyvalent qui peut être utilisé pour une grande variété de tâches. Il est facile à apprendre, largement utilisé et dispose d'une grande communauté de développeurs. Si vous souhaitez apprendre un langage de programmation pour le développement web, JavaScript est un excellent choix.

2-5- C#

C# est un langage qui continue d'apparaître dans les listes de langages de programmation les plus couramment utilisés. **Créé par Microsoft**, il est présent aussi bien dans le milieu des affaires que dans les institutions gouvernementales, les banques ou les institutions médicales.

Cependant, il est très polyvalent et a d'autres utilisations que les anciennes applications : il est également utilisé dans des domaines tels que **l'internet des objets ou le développement de jeux vidéo, de sites web ou d'applications mobiles**. Quand il s'agit de développement de sites web, vous pouvez utiliser le Framework **ASP.NET** ou l'outil **Xamarin** pour développer des **applications natives pour Android et iOS**.

Mais si les jeux vidéo sont votre spécialité, **c#** est le langage que vous devez maîtriser. Rappelons qu'**Unity** – l'un des outils les plus puissants actuellement disponibles – supporte nativement plusieurs langues. Et **c#** est l'un d'eux.

Une grande variété d'applications, notamment :

- Applications de bureau avec Windows Forms ou WPF
- Applications web avec ASP.NET Core
- Jeux avec Unity
- Applications mobiles avec Xamarin
- Développement cloud avec Azure

!

Points forts de C#:

- **Facile à apprendre** : La syntaxe de C# est claire et concise, ce qui le rend facile à apprendre pour les débutants.
- **Polyvalent** : C# peut être utilisé pour créer une grande variété d'applications.
- **Orienté objet** : C# encourage la création de code modulaire et réutilisable.
- **Typé statiquement** : Le typage statique permet de détecter les erreurs de programmation dès la phase de compilation, ce qui rend le code plus fiable.
- **.NET Framework** : C# s'appuie sur le .NET Framework, qui offre un large éventail de bibliothèques et d'outils de développement.

Conclusion:

C# est un langage de programmation puissant et polyvalent qui peut être utilisé pour créer une grande variété d'applications. Si vous recherchez un langage facile à apprendre et qui offre de nombreuses opportunités, C# est un excellent choix.

2 –6– PHP

PHP (Hypertext Preprocesseur) est un langage de programmation de code à usage général côté serveur. **Ce langage assure une bonne communication entre le web et le serveur**, de sorte que les pages web développées avec ce langage soient stables et fonctionnent bien. Actuellement, de **nombreuses pages web sont conçues avec WordPress, qui fonctionne avec PHP**. Par conséquent, ce langage vous permettra de développer à la fois des projets et des plugins pour cette plateforme.

Points forts de PHP:

- **Facile à apprendre**: La syntaxe de PHP est simple et accessible, ce qui le rend facile à apprendre pour les débutants.
- **Gratuit et open source**: PHP est un logiciel libre et open source, ce qui signifie qu'il est gratuit à utiliser et à distribuer.
- **Largement utilisé**: PHP est l'un des langages de programmation les plus populaires pour le développement web, utilisé par des millions de sites web dans le monde.
- **Polyvalent**: PHP peut être utilisé pour créer une grande variété de sites web et d'applications web, y compris des sites web e-commerce, des blogs, des forums, des systèmes de gestion de contenu (CMS) et bien plus encore.
- **Nombreuses bibliothèques et Framework**: Il existe un large éventail de bibliothèques et de frameworks disponibles pour PHP, ce qui facilite le développement d'applications web complexes.

Conclusion :

PHP est un langage de programmation puissant et flexible qui est idéal pour le développement web. Si vous recherchez un langage facile à apprendre et qui offre de nombreuses opportunités, PHP est un excellent choix.

2 – 7 – C++

C++ est un langage de programmation compilé, orienté objet et générique. Développé par Bjarne Stroustrup en tant qu'extension du langage C, il est largement utilisé pour créer une grande variété d'applications, notamment :

- Applications de bureau avec Windows Forms ou WPF
- Applications web avec ASP.NET Core
- Jeux avec Unity

- Applications mobiles avec Xamarin
- Développement cloud avec Azure

Et bien plus encore !

Points forts de C++:

- **Puissant et flexible** : C++ offre un contrôle précis sur le matériel et la mémoire, ce qui le rend idéal pour le développement de logiciels système et d'applications qui nécessitent des performances élevées.
- **Orienté objet** : C++ encourage la création de code modulaire et réutilisable.
- **Générique** : C++ permet de créer des classes et des fonctions génériques qui peuvent être utilisées avec différents types de données.
- **Performant** : Le code C++ est généralement compilé en code machine très efficace, ce qui permet d'obtenir des performances élevées.
- **Largement utilisé** : C++ est l'un des langages de programmation les plus populaires au monde, ce qui signifie qu'il existe une grande communauté de développeurs et de nombreuses ressources disponibles.

Conclusion :

C++ est un langage de programmation puissant et flexible qui peut être utilisé pour créer une grande variété d'applications. Si vous recherchez un langage qui vous donne un contrôle précis sur le matériel et la mémoire, qui est capable de produire des logiciels très performants et qui vous permet de profiter des avantages de la **programmation orientée objet**, **C++ est un excellent choix.**

C++ est aujourd'hui normalisé par l'ISO – ISO 14882

2 – 8 – R

R est un langage de programmation et un logiciel libre spécialement conçu pour les statistiques et la science des données. Développé par Ross Ihaka et Robert Gentleman, il est devenu un outil incontournable dans de nombreux domaines, **notamment** :

- **Recherche scientifique** : analyse de données expérimentales, modélisation statistique, etc.
- **Finance** : analyse de marché, gestion des risques, etc.
- **Marketing** : analyse de la clientèle, ciblage publicitaire, etc.
- **Santé publique** : analyse de données épidémiologiques, études cliniques, etc.

Points forts de R:

- Spécialisé en statistiques et en science des données: R intègre de nombreuses fonctions et outils statistiques directement dans son environnement, facilitant ainsi l'analyse et la manipulation des données.
- **Gratuit et open source** : R est accessible à tous et bénéficie d'une large communauté active qui développe constamment de nouveaux packages et extensions.
- **Graphiques de haute qualité** : R propose une large bibliothèque de fonctions pour créer des graphiques de qualité professionnelle, facilitant la visualisation et la compréhension des données.
- **Langage interprété** : R permet d'exécuter et d'observer les résultats du code ligne par ligne, ce qui facilite l'apprentissage et le prototypage.

- **Nombreux packages et extensions** : Un large éventail de packages et d'extensions est disponible pour étendre les fonctionnalités de R et s'adapter à des besoins spécifiques.

Conclusion :

Si vous êtes impliqué dans l'analyse de données, la modélisation et la visualisation, R est un langage incontournable. Son accessibilité, sa communauté active et ses fonctionnalités dédiées en font un excellent choix pour les débutants comme pour les experts en science des données.

2 – 9– Objectif C

Objective-C est un langage de programmation orienté objet, dérivé du langage C, et utilisé principalement pour le développement d'applications sur les plateformes macOS et iOS d'Apple. Créé par Brad Cox en 1984, il a été intégré à l'environnement de développement d'Apple (Xcode) et est devenu un pilier du développement Apple pendant de nombreuses années.

Points forts d'Objective-C:

- **Orienté objet** : Objective-C permet de créer des applications modulaires et réutilisables en utilisant des concepts tels que les classes, les objets et les méthodes.
- **Puissant et flexible** : Offre un contrôle précis sur le fonctionnement des applications et permet d'exploiter les fonctionnalités spécifiques des plateformes Apple.
- **Largement utilisé** : Un grand nombre d'applications populaires pour macOS et iOS ont été créées avec Objective-C, ce qui signifie qu'il existe une grande communauté de développeurs et de ressources disponibles.
- **Intégration avec Cocoa et Cocoa Touch** : S'intègre parfaitement aux Framework Cocoa et Cocoa Touch d'Apple, qui fournissent des composants et des outils pour le développement d'applications macOS et iOS.

Conclusion :

Objective-C est un langage de programmation puissant et flexible qui a joué un rôle majeur dans le développement d'applications pour macOS et iOS. Bien qu'Apple ait introduit Swift comme langage de programmation privilégié pour ses plateformes, Objective-C reste une technologie importante pour de nombreuses applications existantes et offre une base solide pour apprendre le développement Apple.

2 – 10 – Swift

Ce langage de programmation reste dans le TOP 10 des plus utilisés. **Créé par Apple**, il se concentre sur le développement **d'applications pour iOS ,Ipad ,tvOset macOS.**

Il est apparu en 2014, mais a, malgré sa courte durée de vie, une importante communauté de développeurs derrière lui, qui s'est développée surtout depuis qu'il est devenu open source.

Swift est connu pour être :

- **Facile à apprendre** : Sa syntaxe est claire et concise, ce qui le rend accessible même aux débutants en programmation.
- **Sûr** : Il intègre des fonctionnalités de sécurité avancées qui aident à prévenir les erreurs et les bugs courants.

- **Performant** : Il génère du code efficace qui s'exécute rapidement sur les appareils Apple.
- **Open source** : Cela signifie que le code source de Swift est accessible à tous et que n'importe qui peut contribuer à son développement.

Si vous êtes intéressé par le développement d'applications pour les appareils Apple, Swift est un excellent choix à apprendre.

2 - 11 – Matlab : Langage de programmation pour le calcul numérique

MATLAB (abréviation de Matrix Laboratory) est un environnement de programmation et un langage de programmation orienté calcul numérique. Développé par MathWorks, il est utilisé par des millions d'ingénieurs et de scientifiques à travers le monde pour :

- Analyser des données
- Développer des algorithmes
- Créer des modèles
- Visualiser des résultats

Points forts de MATLAB :

- **Manipulation de matrices:** MATLAB est particulièrement efficace pour manipuler des matrices et des tableaux, ce qui en fait un outil puissant pour le calcul scientifique et l'ingénierie.
- **Environnement interactif:** MATLAB offre un environnement interactif qui permet d'exécuter des commandes et d'explorer des concepts en temps réel.
- **Bibliothèques et outils:** MATLAB propose une vaste collection de bibliothèques et d'outils pour une grande variété de domaines, tels que l'apprentissage automatique, le traitement du signal, la finance et la robotique.
- **Facilité d'apprentissage:** Le langage de programmation de MATLAB est relativement simple à apprendre, ce qui le rend accessible aux débutants.

Utilisations de MATLAB :

- **Recherche et développement:** MATLAB est largement utilisé dans les domaines de la recherche et du développement, notamment dans les universités et les laboratoires de recherche.
- **Industrie:** MATLAB est utilisé dans divers secteurs industriels, tels que l'automobile, l'aéronautique, l'électronique et l'énergie.
- **Enseignement:** MATLAB est un outil pédagogique populaire pour l'enseignement des mathématiques, de l'ingénierie et des sciences.

Conclusion:

MATLAB est un outil puissant et polyvalent pour le calcul numérique. Il est largement utilisé dans les domaines de la recherche, du développement et de l'enseignement. Si vous recherchez un langage de programmation pour manipuler des matrices, développer des algorithmes et créer des modèles, MATLAB est une excellente option à considérer.

2 – 12 - Rust

Rust est un langage de programmation **multi-paradigme compilé** qui met l'accent sur la performance, la sûreté des types et la concurrence. Il est souvent considéré comme un successeur potentiel du C et du C++ en raison de sa rapidité et de sa sécurité.

Points forts de Rust :

- **Performance:** Rust est extrêmement rapide et économe en mémoire. Il peut s'exécuter sur des systèmes embarqués et s'intégrer facilement à d'autres langages.
- **Fiabilité :** Le système de typage de Rust garantit la sécurité de la mémoire et permet d'éviter les erreurs courantes.
- **Productivité :** Rust dispose d'une excellente documentation, d'un compilateur convivial et d'outils de pointe.
- **Communauté active :** Rust possède une communauté active et grandissante qui est toujours prête à aider.

Utilisations de Rust :

- **Systèmes embarqués:** Rust est idéal pour le développement de logiciels embarqués où la performance et la fiabilité sont critiques.
- **Outils de ligne de commande:** Rust est souvent utilisé pour créer des outils de ligne de commande rapides et efficaces.
- **Serveurs web:** Rust peut être utilisé pour créer des serveurs web performants et évolutifs.
- **Applications de bureau:** Rust peut être utilisé pour créer des applications de bureau avec une interface utilisateur graphique.

Alternatives à Rust : C ,C++,GO

Conclusion :

Rust est un langage de programmation puissant et polyvalent qui offre de nombreux avantages par rapport aux langages traditionnels comme le C et le C++. Si vous recherchez un langage performant, fiable et moderne, Rust est une excellente option à considérer.

2 – 13 – TypeScript : Un langage de programmation puissant et flexible

TypeScript est un langage de programmation **open source** développé par **Microsoft** qui s'appuie sur **JavaScript** pour offrir une expérience de développement plus robuste et sécurisée.

Points forts de TypeScript :

- **Typage statique optionnel :** TypeScript permet de définir des types pour les variables, les fonctions et les classes, ce qui permet de détecter les erreurs de programmation dès la phase de compilation et d'améliorer la lisibilité du code.
- **Compilation en JavaScript :** Le code TypeScript est compilé en JavaScript standard, ce qui signifie qu'il peut être utilisé dans n'importe quel environnement où JavaScript est pris en charge.
- **Meilleure prise en charge des outils :** TypeScript offre une meilleure prise en charge des outils de développement tels que les éditeurs de code, les IDE et les frameworks de test, ce qui permet d'améliorer la productivité des développeurs.

- **Grandes bibliothèques et frameworks** : TypeScript est compatible avec les bibliothèques et frameworks JavaScript populaires, tels que Angular, React et Vue.js.

Utilisations de TypeScript :

- **Applications web complexes** : TypeScript est particulièrement adapté au développement d'applications web complexes, car il permet de créer du code plus robuste et plus facile à maintenir.
- **Applications mobiles** : TypeScript peut être utilisé pour développer des applications mobiles natives à l'aide de frameworks tels que React Native ou Ionic.
- **Applications de bureau** : TypeScript peut être utilisé pour développer des applications de bureau à l'aide de frameworks tels que Electron.
- **Jeux** : TypeScript peut être utilisé pour développer des jeux à l'aide de frameworks tels que Phaser.

Conclusion :

TypeScript est un langage de programmation puissant et flexible qui offre de nombreux avantages aux développeurs JavaScript. Si vous recherchez un moyen d'améliorer la qualité et la robustesse de votre code, TypeScript est une excellente option à considérer.

2 – 14 – GO - Un langage de programmation simple et efficace

Go est un langage de programmation **open source** créé par **Google en 2009**. Il est compilé, statiquement typé et s'inspire de C, Pascal et CSP. Go est conçu pour être simple, efficace et performant, ce qui en fait un choix populaire pour le développement de logiciels système, d'applications web et de microservices.

Points forts de Go :

- **Simplicité** : Go est un langage simple à apprendre et à utiliser, avec une syntaxe claire et concise.
- **Efficacité** : Go est un langage très performant, capable de rivaliser avec C et C++ en termes de vitesse d'exécution.
- **Concurrence** : Go intègre nativement la gestion de la concurrence, ce qui facilite la création d'applications concurrentes et évolutives.
- **Fiabilité** : Go est un langage robuste et fiable, avec un garbage collector intégré et une faible probabilité d'erreurs de segmentation.
- **Grandes bibliothèques et Framework** : Go possède un écosystème riche en bibliothèques et frameworks pour différents domaines d'application, tels que le réseau, la base de données, la manipulation de fichiers, etc.

Utilisations de Go :

- **Outils de ligne de commande** : Go est un excellent choix pour développer des outils de ligne de commande simples et performants.
- **Serveurs web** : Go est largement utilisé pour développer des serveurs web et des API RESTful.
- **Microservices** : Go est un choix idéal pour le développement de microservices, grâce à sa légèreté et sa capacité à gérer la concurrence.

- **Cloud computing** : Go est souvent utilisé pour développer des applications cloud en raison de sa scalabilité et de sa fiabilité.
- **Outils de développement** : Go est utilisé pour développer des outils de

Conclusion :

Go est un langage de programmation puissant et polyvalent qui offre de nombreux avantages aux développeurs. Si vous recherchez un langage simple, efficace et performant pour développer des logiciels de qualité, Go est une excellente option à considérer.

2 – 15 – Kotlin -Un langage de programmation moderne et concis

Kotlin est un langage de programmation **open source** développé par **JetBrains** en 2011. Il est concis, expressif et entièrement interopérable avec Java. Kotlin est rapidement devenu un langage de choix pour le développement Android, et il est également utilisé pour créer des applications web, des applications de bureau et des jeux.

Points forts de Kotlin :

- **Concision** : Kotlin permet d'écrire du code plus concis que Java, ce qui améliore la lisibilité et la maintenabilité du code.
- **Expressivité** : Kotlin offre des fonctionnalités modernes telles que les fonctions lambda, les expressions de portée et les extensions de fonctions, ce qui permet d'écrire du code plus expressif et plus élégant.
- **Interopérabilité avec Java** : Kotlin est entièrement interopérable avec Java, ce qui signifie que le code Kotlin peut être utilisé avec du code Java existant et vice versa.
- **Sécurité du code** : Kotlin offre un certain nombre de fonctionnalités qui contribuent à la sécurité du code, telles que le typage statique et les vérifications nulles.
- **Grandes bibliothèques et Framework** : Kotlin possède un écosystème riche en bibliothèques et frameworks pour différents domaines d'application, tels que le réseau, la base de données, l'interface utilisateur, etc.

Utilisations de Kotlin :

:

- **Applications Android** : Kotlin est le langage officiel pour le développement Android, et il est utilisé par plus de 60% des développeurs Android professionnels.
- **Applications web** : Kotlin peut être utilisé pour développer des applications web back-end et front-end à l'aide de frameworks tels que Spring Boot et React.
- **Applications de bureau** : Kotlin peut être utilisé pour développer des applications de bureau à l'aide de frameworks tels que JavaFX et Kotlin Desktop.
- **Jeux** : Kotlin peut être utilisé pour développer des jeux à l'aide de frameworks tels que LibGDX et Korge.

Conclusion :

Kotlin est un langage de programmation moderne et puissant qui offre de nombreux avantages aux développeurs. Si vous recherchez un langage concis, expressif et interopérable avec Java, Kotlin est une excellente option à considérer.

2 – 16 – Ruby - Un langage de programmation convivial et productif

Ruby est un langage de programmation **open source**, dynamique et orienté objet, créé par **Yukihiro Matsumoto en 1993**. Il est connu pour sa syntaxe élégante et intuitive, sa nature flexible et sa communauté active. Ruby est utilisé pour développer une grande variété d'applications, des sites web aux applications mobiles en passant par les outils de ligne de commande.

Points forts de Ruby :

- **Convivialité** : Ruby est un langage facile à apprendre et à utiliser, même pour les débutants en programmation. Sa syntaxe est proche du langage naturel, ce qui rend le code Ruby facile à lire et à comprendre.
- **Productivité** : Ruby permet aux développeurs de créer du code rapidement et efficacement. Il offre de nombreuses fonctionnalités qui permettent d'automatiser des tâches répétitives et d'écrire du code plus concis.
- **Flexibilité** : Ruby est un langage flexible qui peut être utilisé pour développer différents types d'applications. Il est compatible avec de nombreux frameworks et bibliothèques, ce qui permet aux développeurs de choisir les outils les mieux adaptés à leurs besoins.
- **Orienté objet** : Ruby est un langage orienté objet, ce qui permet de créer du code plus modulaire et réutilisable.
- **Communauté active** : Ruby a une communauté active et solidaire qui fournit de nombreuses ressources aux développeurs, tels que des tutoriels, des bibliothèques et des frameworks.

Utilisations de Ruby :

- **Sites web** : Ruby est un langage populaire pour le développement de sites web, notamment grâce à des frameworks tels que Rails.
- **Applications web** : Ruby peut être utilisé pour développer des applications web back-end et front-end à l'aide de frameworks tels que Rails et Sinatra.
- **Applications mobiles** : Ruby peut être utilisé pour développer des applications mobiles natives à l'aide de frameworks tels que React Native et RubyMotion.
- **Outils de ligne de commande** : Ruby est un excellent choix pour développer des outils de ligne de commande simples et performants.
- **Scripts d'administration système** : Ruby peut être utilisé pour écrire des scripts d'administration système pour automatiser des tâches répétitives.

Conclusion :

Ruby est un langage de programmation puissant et polyvalent qui offre de nombreux avantages aux développeurs. Si vous recherchez un langage convivial, productif et flexible, Ruby est une excellente option à considérer.

2 – 17– HTML- CSS

Ce sont 2 langages complémentaires. Ils représentent le b.a.-ba du développement web. De son côté, HTML a pour rôle de structurer la page web. Il permet de constituer son squelette. C'est sur cette base que peuvent être intégrés des éléments graphiques et visuels à partir du langage CSS, puis les positionner les uns par rapport aux autres.

HTML : Structure et contenu

CSS : Mise en forme

Avantages

- HTML présente l'avantage d'être universel, et indépendant de tout système d'exploitation.
- Il fonctionne sur la base d'une structure distribuée, c'est-à-dire que les liens externes s'y intègrent parfaitement. Cela rend possible la modification d'un élément sans modifier la page où il se trouve.
- Enfin, HTML permet d'intégrer à la fois des éléments locaux et distants, via la connexion à un serveur.

Inconvénients

En tant que langage universel et simplifié, HTML peut s'avérer limité en termes de mise en forme d'une page.

Si vous souhaitez stocker vos données en local, il n'est également pas totalement adapté. Qui plus est, le stockage peut vite être alourdi, en fonction du nombre de pages de votre site web.

Conclusion :

HTML et CSS sont des langages essentiels pour quiconque souhaite créer des pages web. Ils sont relativement simples à apprendre et offrent une grande flexibilité pour créer des sites web attrayants et fonctionnel

2 – 18– F# - "Fsharp" - <https://Fsharp.org> - Un langage de programmation puissant et flexible

F# (F sharp) est un langage de programmation **open source** développé par **Microsoft Research** et intégré à **Visual Studio**. Il s'appuie sur la plateforme .NET et est dérivé du langage OCaml. F# est un langage hybride, combinant des aspects de programmation fonctionnelle, impérative et orientée objet.

Points forts de F# :

- **Concision** : F# permet d'écrire du code concis et expressif, ce qui améliore la lisibilité et la maintenabilité du code.
- **Efficacité** : F# est un langage performant, capable de rivaliser avec C et C++ en termes de vitesse d'exécution.
- **Sécurité du code** : F# offre un certain nombre de fonctionnalités qui contribuent à la sécurité du code, telles que le typage statique et les vérifications nulles.
- **Grandes bibliothèques et Framework** : F# possède un écosystème riche en bibliothèques et frameworks pour différents domaines d'application, tels que le réseau, la base de données, l'apprentissage automatique, etc.
- **Interopérabilité avec C#** : F# est entièrement interopérable avec C#, ce qui permet de mélanger du code F# et C# dans un même projet.

Utilisations de F# :

- **Applications web** : F# peut être utilisé pour développer des applications web back-end et front-end à l'aide de frameworks tels que ASP.NET Core et Fable.
- **Applications de bureau** : F# peut être utilisé pour développer des applications de bureau à l'aide de frameworks tels que WPF et Avalonia.

- **Jeux** : F# peut être utilisé pour développer des jeux à l'aide de frameworks tels que Unity et F#Game.
- **Outils de ligne de commande** : F# est un excellent choix pour développer des outils de ligne de commande simples et performants.
- **Bibliothèques et frameworks** : F# est utilisé pour développer des bibliothèques et frameworks pour différents langages et plateformes.

Conclusion :

F# est un langage de programmation puissant et polyvalent qui offre de nombreux avantages aux développeurs. Si vous recherchez un langage concis, efficace et flexible, F# est une excellente option à considérer.

a. – classement suivant le type d'utilisation

3– 1 – Langages pour un Web site statique

La réalisation d'un site web statique fait appel aux langages de programmation de base. HTML et CSS sont les plus répandus. JavaScript est, quant à lui, de plus en plus utilisé pour compléter le travail de programmation autour de la structure du site web

3– 1 – 1- HTML et CSS (§ 2 -17)

3– 1 – 2 - JAVASCRIPT. (§2 - 4)

3 – 1- 3 Générateur de sites statiques

Un générateur de site web statique (SSG) est un outil qui permet de créer un site web complet à partir de fichiers de données brutes et d'un ensemble de modèles. Les SSG génèrent des pages HTML pré-rendues qui peuvent ensuite être hébergées sur n'importe quel serveur web, sans avoir besoin d'un serveur web dynamique comme Apache ou Nginx.

Avantages des générateurs de sites web statiques :

- **Performance** : Les sites web statiques sont généralement beaucoup plus rapides que les sites web dynamiques, car les pages HTML sont déjà pré-rendues et n'ont pas besoin d'être générées à la volée.
- **Sécurité** : Les sites web statiques sont moins sensibles aux attaques que les sites web dynamiques, car il n'y a pas de code côté serveur à exploiter.
- **Simplicité** : Les SSG sont généralement plus simples à utiliser et à gérer que les systèmes de gestion de contenu (CMS) dynamiques.
- **Coût** : Les sites web statiques peuvent être moins chers à héberger que les sites web dynamiques, car ils ne nécessitent pas de serveur web puissant.

3 – 1 – 4 - Quelques générateurs des sites statiques

- **Jekyll** – www.jekyllrb.com
Jekyll est un générateur de site statique développé par Tom Preston-Werner, le fondateur de Github. C'est un logiciel libre écrit en langage Ruby. En 2017, il est considéré comme le générateur de site statique le plus populaire
- **Hugo** – <https://gohugo.io>
Hugo est un logiciel libre, générateur de site statique écrit en langage Go. C'est un concurrent du logiciel Jekyll écrit en langage Ruby .
En langage R, la bibliothèque logicielle Blogdown permet de développer des sites web statiques en utilisant Hugo .
- **Eleventy** – <https://www.11ty.dev>
Eleventy, également connu sous le nom de **11ty**, est un générateur de site statique². Il a été écrit en Javascript. Il a été lancé en 2017 par Zach Leatherman en tant

qu'alternative JavaScript à Jekyll³, l'un des premiers générateurs de sites statiques grand public, écrit en Ruby.

- **Hexo** – <https://hexo.io>
Hexo est extrêmement rapide et soutenu par le puissant moteur d'exécution de **Node.js**. D : il vous donne un ensemble 'outils centaines de fichiers ne prennent que quelques secondes à créer. Toutes les fonctionnalités de **GitHub Flavored Markdown** sont prises en charge. Le déploiement en une seule commande est l'une des meilleures fonctionnalités de Hexo.
- **Metalsmith** – <https://metalsmith.io>
Metalsmith.js est un générateur de site statique extrêmement simple
Et enfichage pour Node.JS
Metalsmith fonctionne plus comme une bibliothèque que comme un Framework :il vous doone un ensemble d'outils que l'on peut utiliser comme on veut
- **Pelican** – <https://getpelican.com>
Pelican a tout ce dont vous avez besoin pour créer votre site statique. Les constructions sont rapides et c'est bien agencé.
- **Wintersmith** – <http://wintersmith.io>
Wintersmith est un générateur de site statique simple mais flexible. Il prend du contenu (markdown, less, scripts, etc.), les transforme à l'aide de plugins et génère un site Web statique (html, css, images, etc.) que vous pouvez héberger n'importe où. Il est livré avec des plugins pour les modèles de [démarrage](#) et [de pug](#) (outils de templating),
- **Octopress** – <http://octopress.org>
Octopress est un framework de blogs statiques construit sur Jekyll. Il utilise des scripts pour créer des fichiers statiques à déployer sur un serveur
- **Middleman** – <https://middlemanapp.com>
Middleman est un générateur de site statique utilisant tous les raccourcis et outils du développement Web moderne. Middleman est distribué à l'aide du gestionnaire de packages RubyGems

3– 2 – Langage de programmation pour un site Web Dynamique

Le web dynamique désigne l'ensemble des technologies qui permettent de créer des pages web interactives et personnalisées en fonction de l'utilisateur. Contrairement aux pages web statiques qui sont pré-définies et ne changent pas, les pages web dynamiques sont générées à la volée par un serveur web en fonction de l'interaction de l'utilisateur avec le site web.

Fonctionnalités du web dynamique :

- **Interactivité** : Les pages web dynamiques peuvent réagir aux actions de l'utilisateur, comme les clics, les formulaires et les requêtes de recherche.
- **Personnalisation** : Les pages web dynamiques peuvent être personnalisées en fonction des préférences de l'utilisateur, comme la langue, la localisation et les produits précédemment consultés.
- **Mise à jour en temps réel** : Les pages web dynamiques peuvent être mises à jour en temps réel, sans avoir besoin de recharger la page entière.
- **Fonctionnalités avancées** : Le web dynamique permet de proposer des fonctionnalités avancées comme les paniers d'achat, les forums de discussion et les réseaux sociaux.

Technologies du web dynamique :

- **Langages de programmation côté serveur** : PHP, Python, Ruby, Java, etc.

- **Langages de programmation côté client** : JavaScript, TypeScript, etc.
- **Bases de données** : MySQL, PostgreSQL, MongoDB, etc.
- **Frameworks** : Laravel, Django, Rails, Spring Boot, etc.

Avantages du web dynamique :

- **Meilleure expérience utilisateur** : Les pages web dynamiques offrent une meilleure expérience utilisateur grâce à leur interactivité et leur personnalisation.
- **Meilleure engagement** : Les pages web dynamiques peuvent inciter les utilisateurs à rester plus longtemps sur le site web et à interagir davantage avec le contenu.
- **Meilleure conversion** : Les pages web dynamiques peuvent améliorer le taux de conversion des sites web e-commerce et des sites web de génération de leads.
-

Inconvénients du web dynamique :

- **Complexité accrue** : Le développement de sites web dynamiques est plus complexe que le développement de sites web statiques.
- **Coût accru** : Le développement et l'hébergement de sites web dynamiques peuvent être plus coûteux que le développement et l'hébergement de sites web statiques.
- **Sécurité** : Les sites web dynamiques peuvent être plus sensibles aux attaques que les sites web statiques.

Conclusion :

Le web dynamique offre de nombreux avantages pour les sites web qui ont besoin d'être interactifs, personnalisés et mis à jour en temps réel. Cependant, il est important de prendre en compte la complexité accrue, le coût et les implications de sécurité avant de choisir de développer un site web dynamique.

3 – 2 – 1 – les langages de programmation coté serveur

Un langage de programmation côté serveur est un langage utilisé pour créer des programmes qui s'exécutent sur un serveur web. Ces programmes génèrent ensuite du contenu HTML, JavaScript ou CSS qui est envoyé au navigateur web de l'utilisateur.

Exemples de langages de programmation côté serveur populaires :

- **PHP**: Un langage largement utilisé pour le développement web, notamment pour les sites web dynamiques et les systèmes de gestion de contenu (CMS). (§2-6)
- **Python** : Un langage polyvalent et puissant utilisé pour une variété d'applications, y compris le développement web, l'apprentissage automatique et l'analyse de données. (§2-1)
- **Java** : Un langage robuste et sécurisé utilisé pour une large gamme d'applications web, notamment les applications d'entreprise et les servelets. (§2-3)
- **Ruby** : Un langage expressif et convivial utilisé pour le développement web, notamment pour les frameworks Rails et Sinatra.
- **Node.js** : Un environnement d'exécution JavaScript permettant de créer des applications web asynchrones et performantes.

Choisir un langage de programmation côté serveur:

Le choix d'un langage de programmation côté serveur dépend de plusieurs facteurs, tels que :

- Le type d'application que vous souhaitez développer.
- Vos compétences et votre expérience en programmation.
- La popularité du langage et la communauté de développeurs qui l'entoure.
- Les performances et la sécurité du langage.

:
En conclusion,

Un site web dit « dynamique » est généralement un site qui gère des données. Celui-ci est alors connecté à un serveur, avec lequel il communique via un langage dédié. Tout comme pour les langages appelés « clients », il existe de nombreux langages de programmation dits « serveurs ». Ce type de développement est appelé *backend*.

3 -2 – 2 - Les langages de programmation côté client

Un langage de programmation côté client est un langage utilisé pour créer des programmes qui s'exécutent sur le navigateur web de l'utilisateur. Ces programmes permettent d'ajouter des fonctionnalités interactives et dynamiques aux pages web, comme des animations, des jeux, des formulaires de contact et des mises à jour en temps réel.

Le langage de programmation côté client le plus populaire est JavaScript. Il est utilisé pour une grande variété d'applications, y compris :

- **Ajouter des interactions aux pages web** : Animations, menus déroulants, formulaires de validation, etc.
- **Créer des jeux web** : Jeux simples et complexes, jeux 2D et 3D, etc.
- **Développer des applications web** : Applications web mono-page (SPA), applications web progressives (PWA), etc.
- **Communiquer avec le serveur** : Envoyer et recevoir des données du serveur, mettre à jour le contenu des pages web en temps réel, etc.

Autres langages de programmation côté client populaires :

- **TypeScript**: Un sur-ensemble de JavaScript qui offre des fonctionnalités supplémentaires telles que le typage statique.
- **CoffeeScript**: Un langage de programmation qui compile en JavaScript, avec une syntaxe plus concise et plus expressive.
- **Dart**: Un langage de programmation développé par Google, utilisé pour le développement web et mobile.

Choisir un langage de programmation côté client:

Le choix d'un langage de programmation côté client dépend de plusieurs facteurs, tels que :

- Le type d'application que vous souhaitez développer.
- Vos compétences et votre expérience en programmation.
- La popularité du langage et la communauté de développeurs qui l'entoure.
- Les performances et la compatibilité du langage avec les différents navigateurs web.

En conclusion,

Les langages de programmation côté client sont essentiels pour créer des pages web interactives et dynamiques. JavaScript est le langage de programmation côté client le plus populaire et il est utilisé pour une grande variété d'applications. D'autres langages comme TypeScript, CoffeeScript et Dart peuvent également être utilisés pour le développement web côté client. Le choix du langage le plus adapté à vos besoins dépend de plusieurs facteurs, tels que le type d'application que vous souhaitez développer, vos compétences en programmation et la communauté de développeurs qui entoure le langage.

3 – 2 – 3 – Langages complémentaires et utilitaire

3 -2- 3 – 1- MySQL- Langage de requête et SGBD

MySQL est à la fois un **langage de requête** et un **système de gestion** de bases de données (SGBD) relationnel open source. Il est largement utilisé pour stocker et gérer des données dans le cadre d'applications web et d'autres logiciels.

Le langage de requête de MySQL :

- **Langage SQL** : MySQL utilise le langage standard SQL (Structured Query Language) pour interroger et manipuler les données dans la base de données.
- **Fonctions et extensions** : MySQL propose également des fonctions et des extensions spécifiques qui ne font pas partie du standard SQL.

Le SGBD MySQL :

- **Stockage de données** : MySQL permet de stocker des données dans des tables, organisées en colonnes et en lignes.
- **Types de données** : MySQL supporte une variété de types de données, tels que les nombres entiers, les nombres décimaux, les chaînes de caractères, les dates et les heures.
- **Moteur de stockage** : MySQL utilise différents moteurs de stockage pour optimiser les performances pour différents types d'applications.
- **Connectivité** : MySQL peut être utilisé avec différents langages de programmation pour créer des applications web et d'autres logiciels.
-

Avantages de MySQL :

- : MySQL est gratuit et open source, ce qui le rend accessible à tous.
- **Facile à utiliser Open source**: Le langage SQL est relativement simple à apprendre et à utiliser.
- **Performant** : MySQL est un SGBD performant capable de gérer de grandes quantités de données.
- **Extensible** : MySQL offre de nombreuses fonctionnalités et extensions pour répondre aux besoins spécifiques des développeurs.

Inconvénients de MySQL :

- **Sécurité** : MySQL peut être vulnérable aux attaques si la configuration n'est pas sécurisée.
- **Complexité** : La gestion de bases de données complexes peut être difficile avec MySQL.
- **Manque de certaines fonctionnalités** : MySQL ne dispose pas de toutes les fonctionnalités des SGBD plus puissants.

Conclusion:

MySQL est un langage de requête et un SGBD open source largement utilisé pour stocker et gérer des données. Il est facile à apprendre et à utiliser, et offre de nombreuses fonctionnalités pour répondre aux besoins des développeurs. However, it is important to be aware of its limitations and take appropriate security measures.

3 – 2 – 3 – 2 - FTP- Le protocole FTP : transférer des fichiers en réseau

Le File Transfer Protocol (FTP), ou protocole de transfert de fichiers en français, est un ensemble de règles et de conventions qui permettent à des ordinateurs de communiquer entre eux pour échanger

des fichiers sur un réseau TCP/IP. Il s'agit d'un des protocoles les plus anciens et les plus répandus pour le transfert de fichiers, ayant vu le jour en 1971.

Fonctionnement du protocole FTP :

Deux acteurs principaux: Le protocole FTP implique deux acteurs principaux :

- **Client FTP:** Un logiciel installé sur votre ordinateur ou un appareil mobile et utilisé pour se connecter à un serveur FTP et gérer les transferts (téléchargement et envoi) de fichiers.
- **Serveur FTP:** Un logiciel exécuté sur un ordinateur distant qui stocke les fichiers et gère les connexions des clients FTP.

Connexion et communication:

Le client FTP établit une connexion contrôlée avec le serveur FTP sur le port 21 par défaut.

Une fois connecté, le client et le serveur échangent des commandes et des réponses en utilisant des codes spécifiques.

Le client envoie une commande pour ouvrir un canal de données séparé, généralement sur le port 20 par défaut.

Les données du fichier sont transférées sur le canal de données en utilisant un mode binaire pour garantir leur intégrité.

Une fois le transfert terminé, les deux canaux sont fermés et la connexion est interrompue.

Utilisations courantes du FTP :

- **Transfert de fichiers volumineux :** Idéal pour transférer des fichiers volumineux comme des vidéos, des logiciels ou des archives.
- **Mise à jour de sites web :** Utilisé par les développeurs web pour transférer des fichiers et des images vers le serveur d'hébergement d'un site web.
- **Sauvegarde de données :** Permet de sauvegarder des données importantes sur un serveur distant pour la sécurité et la récupération en cas de panne.

Limitations du FTP :

- **Sécurité :** Le protocole FTP n'est pas sécurisé par défaut. Les noms d'utilisateur, les mots de passe et les données transférées sont envoyés en clair sur le réseau, ce qui les rend vulnérables aux interceptions.
- **Manque de fonctionnalités avancées :** Le FTP est un protocole simple et ne dispose pas de fonctionnalités avancées comme le suivi de progression, la reprise automatique des transferts interrompus, ou la synchronisation de fichiers.

Alternatives sécurisées au FTP :

- **SFTP (SSH File Transfer Protocol) :** Utilise le protocole SSH pour crypter les données et garantir la sécurité des transferts.
- **FTPS (FTP over SSL/TLS) :** Une version sécurisée du FTP qui utilise les protocoles SSL ou TLS pour crypter les communications.
- **HTTPS :** Dans certains cas, le protocole HTTPS peut être utilisé pour transférer des fichiers en utilisant des requêtes HTTP POST et PUT, offrant un meilleur niveau de sécurité que le FTP.

Conclusion :

Le protocole FTP reste un outil important pour le transfert de fichiers, notamment en raison de sa simplicité et de sa large prise en charge. Cependant, il est essentiel d'être conscient de ses limitations de sécurité et d'envisager des alternatives plus sécurisées comme le SFTP ou le FTPS, en particulier pour les transferts de données sensibles.

3 – 2 – 3 – 3 - Perl : langage puissant et polyvalent

Perl est un langage de programmation puissant et polyvalent créé par Larry Wall en 1987. Il est largement utilisé pour une variété de tâches, notamment :

- **Traitement de texte** : Perl est particulièrement efficace pour manipuler des textes, grâce à ses expressions régulières puissantes et intégrées.
- **Administration système** : Les scripts Perl sont souvent utilisés pour automatiser des tâches d'administration système, comme la création de comptes utilisateurs, la configuration de serveurs et la gestion de fichiers.
- **Développement web** : Perl peut être utilisé pour créer des sites web dynamiques et des applications web, avec des frameworks comme Catalyst et Dancer.
- **Bioinformatique** : Perl est un langage populaire dans le domaine de la bioinformatique pour l'analyse de séquences d'ADN et d'autres données biologiques.
- **Programmation réseau** : Perl peut être utilisé pour créer des programmes réseau, comme des serveurs web et des clients FTP.

Avantages de Perl :

- **Puissant et flexible** : Perl est capable de gérer des tâches complexes et offre une grande liberté aux développeurs.
- **Facile à apprendre** : La syntaxe de Perl est relativement simple et claire, ce qui facilite son apprentissage.
- **Largement utilisé** : Perl dispose d'une communauté active et d'une grande quantité de ressources disponibles en ligne.
- **Multiplateforme** : Perl fonctionne sur une grande variété de systèmes d'exploitation, ce qui le rend portable.
- **Open source** : Perl est un langage libre et open source, ce qui le rend gratuit et accessible à tous.

Inconvénients de Perl :

- **Syntaxe complexe** : La syntaxe de Perl peut être perçue comme complexe et verbeuse par rapport à d'autres langages.
- **Manque de typage statique** : Perl est un langage typé dynamiquement, ce qui peut causer des erreurs difficiles à détecter à la compilation.
- **Problèmes de performance** : Perl peut être moins performant que d'autres langages pour certaines tâches.

Conclusion :

Perl est un langage de programmation puissant et polyvalent qui peut être utilisé pour une grande variété de tâches. Il est facile à apprendre et dispose d'une communauté active et de nombreuses ressources disponibles. Cependant, il est important de connaître ses inconvénients, comme sa

syntaxe complexe et ses problèmes de performance potentiels, avant de l'utiliser pour un projet import

3 – 2 – 3 – 4 -Delphi : langage de programmation orienté objet et EDI

Delphi est à la fois un langage de programmation orienté objet et un environnement de développement intégré (EDI) pour ce langage. Développé par Borland en 1995, il est principalement utilisé pour créer des applications Windows.

Caractéristiques de Delphi :

langage de programmation :

- **Basé sur Object Pascal**, un dérivé du langage Pascal avec des fonctionnalités orientées objet.
- **Syntaxe claire et lisible**, proche du langage C++.
- **Compilé en code natif** pour une exécution rapide.
- **Large éventail de bibliothèques** et de composants pour simplifier le développement.

Environnement de développement intégré (EDI):

- **Offre une interface** visuelle pour créer des applications Windows.
- **Outils de conception** de formulaires, d'édition de code, de débogage et de compilation.
- **Aide à la saisie** de code, IntelliSense et refactorisation.
- **Large éventail de plugins** et d'extensions disponibles.

Utilisations de Delphi :

- Développement d'applications Windows :
- Applications de bureautique et de gestion commerciale.
- Applications scientifiques et techniques.
- Jeux et logiciels multimédias.
- Applications mobiles avec Delphi Mobile (anciennement FireMonkey).

Développement web :

- Applications web avec RAD Studio et le framework WebBroker.
- Services web avec Delphi et REST.

Avantages de Delphi :

- **Rapidité de développement** : L'EDI Delphi permet de créer des applications rapidement grâce à ses outils visuels et à ses composants pré-définis.
- **Facilité d'apprentissage** : Le langage Delphi est relativement simple à apprendre, surtout pour les développeurs ayant déjà une expérience avec le langage Pascal ou C++.
- **Stabilité et performance** : Les applications Delphi sont généralement stables et performantes.
- **Large communauté** : Delphi dispose d'une communauté active de développeurs et de nombreuses ressources disponibles en ligne.

Inconvénients de Delphi :

- **Coût** : Delphi est un langage propriétaire et son utilisation nécessite une licence payante.
- **Prise en charge de Windows** : Delphi est principalement utilisé pour développer des applications Windows, ce qui peut limiter sa portabilité vers d'autres plateformes.

- **Manque de certaines fonctionnalités** : Delphi ne dispose pas de toutes les fonctionnalités des langages de programmation les plus modernes.

Alternatives à Delphi :

- **C++ Builder** : Un autre EDI pour le langage C++ avec des fonctionnalités similaires à Delphi.
- **Visual Studio** : L'EDI de Microsoft pour le développement d'applications Windows et web.
- **Lazarus** : Un EDI open source pour le langage Free Pascal, compatible avec Delphi.

Conclusion :

Delphi est un langage de programmation et un EDI puissant et efficace pour le développement d'applications Windows. Il est facile à apprendre et offre une grande rapidité de développement. Cependant, il est important de prendre en compte son coût, sa prise en charge limitée d'autres plateformes et ses fonctionnalités qui peuvent être limitées comparé to other modern languages

3 – 2 - 3 -5 - Pascal

Pascal est un langage de programmation impératif créé en 1970 par Niklaus Wirth. Il se caractérise par :

- **Une syntaxe claire**, rigoureuse et structurée, facilitant l'apprentissage et la création de programmes lisibles.
- **Un typage statique fort**, qui permet de détecter les erreurs de types dès la compilation.

Un ensemble de fonctionnalités riche, incluant des structures de données et des instructions de contrôle puissantes.

Utilisations

Pascal a été largement utilisé dans l'enseignement de la programmation en raison de sa simplicité et de sa rigueur. Il a également été utilisé pour le développement d'applications professionnelles, notamment dans les domaines de la science, de l'ingénierie et de la finance.

Avantages :

- Facilité d'apprentissage et d'utilisation
- Syntaxe claire et structurée
- Typage statique fort
- Large éventail de fonctionnalités

Inconvénients :

- Moins performant que certains langages plus modernes
- Manque de certaines fonctionnalités avancées présentes dans d'autres langages
- Relativement peu utilisé aujourd'hui

Conclusion

Pascal est un langage de programmation important qui a joué un rôle majeur dans l'histoire de l'informatique. Bien qu'il soit moins utilisé aujourd'hui qu'auparavant, il reste un langage pédagogique précieux et peut être utile pour le développement de certaines applications spécifiques.

3 – 3 - LOGICIEL de CRÉATION de SITES WEB

- **ZYRO** – <https://zyro.com>
Zyro est une plateforme de création de site qui a vu le jour en 2019. Pourtant, cet intervenant basé en Lituanie s’est rapidement illustré comme un des meilleurs dans sa catégorie. En plus d’offrir un service de qualité et une interface très conviviale, Zyro propose des tarifs compétitifs et une version gratuite
- **WIX** – <https://fr.wix.com>
Wix est un autre outil de conception de site internet très populaire. Cette plate-forme qui repose sur le glisser-déposer a tous les atouts d’un service de création de site web professionnel : elle est facile à utiliser, offre de jolis *templates* et vous permet de personnaliser presque tous les aspects du site.
 - Facile à utiliser avec un éditeur par glisser-déposer
 - Grand choix de modèles et de fonctionnalités
 - Forfait gratuit disponible
- **WORDPRESS.ORG** <https://fr.wordpress.org>
WordPress.org est le service de création de sites web gratuit le plus populaire au monde. Plus de 35% de tous les sites web sur Internet seraient gérés par WordPress. WordPress compte parmi les meilleurs services en raison de ses fonctionnalités puissantes, de son évolutivité et de sa facilité d’utilisation. WordPress est un outil de création de site internet gratuit et open source.
- **WORDPRESS.COM**
WordPress.com est la version entièrement hébergée de WordPress qui est gérée par la société Automattic. Il s’agit d’un service de création de site Web créé par Matt Mullenweg, le co-fondateur du logiciel open source WordPress, d’où le nom WordPress.com.
 - Plateforme open source très flexible
 - Large choix de plugins et de thèmes
 - Nécessite plus de connaissances techniques que Wix
- **WEBSSELF** <https://websself.net>
WebSelf est un éditeur en ligne qui vous permet de créer facilement un site Web. Aucune compétence en programmation n’est requise. WebSelf est un service facile à utiliser. Il suffit de glisser-déposer des blocs pour construire vos pages.
- **WEEBLY** – www.weebly.com
Weebly se démarque par des *templates* au design très actuel. Une très belle solution... à condition que vous n’ayez pas besoin d’options trop poussées. On apprécie l’usage du glisser-déposer, qui vous permet de configurer facilement vos pages exactement comme vous le souhaitez, et ses *templates* par défaut très bien pensés pour mobile
 - Bon rapport qualité-prix
 - Facile à utiliser avec un éditeur par glisser-déposer
 - Moins de fonctionnalités que Wix ou Squarespace
- **SHOPIFY** – <https://www.shopify.com>
Shopify est l’un des CMS spécialisés dans l’e-commerce les plus utilisés du marché. Sa solution complète offre la possibilité à n’importe quelle entreprise de créer sa boutique en ligne facilement grâce à un large choix de template
 - Plateforme de commerce électronique puissante
 - Facile à utiliser avec un large éventail de fonctionnalités
 - Forfaits payants requis
- **SQUARESPACE** – <https://fr.squarespace.com>

Squarespace une solution de création de site web clé en main qui offre la possibilité à n'importe qui de réaliser son propre site web. En plus des fonctionnalités de création de site de base, Squarespace vous donne également la possibilité de créer une boutique en ligne ou encore de mettre en place la réservation en ligne, fonctionnalité idéale pour les restaurants ou les salons de coiffure par exemple

- Conçu pour les sites web design
 - Modèles élégants et minimalistes
 - Facile à utiliser mais moins d'options que Wi
- **Agence de Créateur de site Web**
 - ✓ **IONOS** - <https://www.ionos.fr>
 - ✓ **SITE123** - <https://app.site123.com>
 - ✓ **HubSpot** - <https://www.hubspot.fr>
 - ✓ **JIMDO** - <https://www.jimdo.com>
 - ✓ **Webador** - <https://www.webador.fr>

3 - 4- Framework pour développement Web

En programmation informatique, un *framework* (appelé aussi **infrastructure logicielle, infrastructure de développement², environnement de développement**, est un ensemble cohérent de composants logiciels structurels qui sert à créer les fondations ainsi que les grandes lignes de tout ou partie d'un logiciel, c'est-à-dire une **architecture**.
il existe deux types de Framework : **les Framework back-end** et **les Framework front-end**.
Ainsi, le back-end concerne la partie cachée d'un site web ou une application tandis qu'une le front-end représente les premiers éléments visibles sur un site ou une application.

3 – 4 - 1- Framework Back-end (et mix front end)

- **Angular JS** – <https://angularjs.org>
Créé par Google en 2009, **AngularJS** est un framework **JavaScript** open-source qui prend en charge l'animation, la manipulation du DOM, le routage, l'injection de dépendance ou encore le data binding.
AngularJS est très populaire et dispose donc d'une communauté active
- **Reactjs** – <https://reactjs.org>
ReactJS est un framework développé par Facebook en 2013 et qui est aujourd'hui utilisé par Instagram, Netflix ou encore Yahoo.
Sa spécialité ? Le développement front-end, c'est-à-dire la création d'interfaces utilisateurs.
ReactJS utilise le langage **JSX** et intègre la notion de composants faciles à réutiliser entre différents projets.
- **VueJS** – <https://vuejs.org>
VueJS est un autre framework JavaScript basé sur l'utilisation de composants.
Si son utilisation est très proche de ReactJS, sa courbe d'apprentissage est toutefois jugée plus simple. Une autre différence est également à noter dans le langage utilisé puisqu'ici il s'agit du **JavaScript** et non du JSX...
- **Ruby on Rails** – <https://rubyonrails.org>
Le Framework **Ruby on Rails** est ce qu'on appelle un Model View Controller (**MVC**) et est idéal si vous débutez dans le développement web. En effet, il est facile à comprendre et à utiliser du fait de sa simplicité syntaxique.
- **Symfony** – <https://symfony.com>
Symfony est un Framework MVC open-source, écrit en PHP et 100% français.
Destiné aux projets plus complexes, voire haut de gamme, il permet de faciliter le

développement et offre une bonne flexibilité. Vous pouvez en effet le configurer comme vous le souhaitez et vous pouvez bénéficier de l'aide d'une grande communauté.

- **Django** – <https://www.djangoproject.com>
Django est un Framework en langage Python – et donc côté serveur – a été créé par des développeurs expérimentés.
Vous pouvez créer toute sorte de sites web, que ce soit des sites d'actualités, des systèmes de gestion de contenu ou encore des réseaux sociaux. Ce framework protège automatiquement votre site grâce à un moyen sécurisé de gérer les comptes utilisateurs et les mots de passe.
- **Laravel** – <https://laravel.com>
Laravel est un Framework PHP composé de plusieurs bibliothèques issues d'autres frameworks. Vous pouvez gérer les systèmes de cache, les envois d'emails, les sessions utilisateurs, la pagination de votre site ou encore créer des requêtes SQL. Concernant la documentation, celle-ci est très complète et de qualité, ce qui est idéal si vous êtes débutant.
- **ASP.MVC** - <https://dotnet.microsoft.com/en-us/apps/aspnet>
Développé par Microsoft et supporté par Windows, **ASP.NET** est lié au framework .NET.
Avec ASP.NET vous pouvez développer des sites web complexes et dynamiques, notamment grâce au Model View Controller.
- **Meteor** – <https://www.meteor.com>
Meteor est un Framework hautement évolutif basé sur le **JavaScript** et qui intègre **NodeJS** et **MongoDB**.
Rapide à utiliser, vous pouvez gérer à la fois le front-end et le back-end de votre site. Meteor est idéal pour créer des sites web collaboratifs ou encore des messageries où il est nécessaire d'afficher des données en temps réel.
- **Spring** - <https://spring.io>
Spring est l'un des plus anciens frameworks Java mais certainement l'un des meilleurs. Il offre de nombreux outils pour développer, configurer et sécuriser des sites web. Malgré sa complexité, la documentation est très fournie et la communauté

3 – 4 – 2- - Framework Front end

- **Bootstrap** – <https://getbootstrap.com>
Bootstrap est sans doute la plus grande référence des frameworks front-end. Développé par les équipes à l'origine du réseau social Twitter, il utilise les langages HTML, CSS et JavaScript. Il a été créé pour développer des sites qui s'adaptent à tout type d'écran.
- **Semantic UI** – <https://semantic-ui.com>
Semantic UI est un autre framework de grande qualité qui brille notamment par sa simplicité.
À la différence de nombreux frameworks, celui-ci garde une cohérence linguistique dans son code. Il fournit un large jeu de couleurs et est reconnu pour son esthétisme de qualité.
- **Skeleton** – <https://getskeleton.com>
Skeleton est un vrai poids plume avec seulement 400 lignes de code. Il va droit à l'essentiel puisqu'il ne contient que les éléments fondamentaux pour créer un site web. Facile à prendre en main, vous pourrez rapidement développer un site internet qui s'affichera sur toutes tailles d'écrans avec Skeleton.

- **KickStart** - <http://www.99lime.com/elements/>
Léger et fonctionnel, **KickStart** inclut tout le nécessaire pour créer un site internet en un temps record.
Il est souvent utilisé pour concevoir des pages web et est apprécié des utilisateurs pour son côté pratique et fonctionnel. Il intègre beaucoup d'éléments préconçus qui vous feront gagner du temps dans la création de votre site.
- **Foundation** – [https:// get.foundation](https://get.foundation)
Ce framework très avancé permet de créer des sites de grande ampleur et complètement responsive, mais n'est pas le plus accessible.
Développé par ZURB, **Foundation** s'adresse davantage à des professionnels, et aux développeurs qui souhaitent apporter leurs contributions à la plateforme.
- **Kube** – <https://github.com/imperavi/kubeframework>
Kube (« kyu: b ») est un Framework Web pour les développeurs et les concepteurs. Kube est construit et conçu pour fournir le framework CSS et JS le plus flexible et le plus puissant à la communauté. Contrairement à de nombreux autres frameworks Web, Kube est idéal à la fois pour l'amorçage ultra-rapide et pour un développement Web de pointe sérieux et à l'épreuve du temps.
- **Ulkit** – <https://getulkit.com>
Ce Framework très modulable s'est taillé une vraie réputation, notamment pour créer des interfaces utilisateur avancées.
Ulkit a été créé par YOOtheme, expert dans la création de **thèmes pour WordPress**, ainsi que dans la conception de templates pour **Joomla!**
- **PureCSS** – <https://purecss.io>
Créé par les développeurs de Yahoo!, **PureCSS** est un framework léger idéal pour créer des sites parfaitement responsives.
Il propose une alternative plus minimaliste et flexible que Bootstrap, tout en offrant toutes les fonctionnalités pour la conception d'un site web standard.
- **KnaCSS** – <https://groundwork.> – <https://www.knacss.com>
Framework français, **KNACSS** a tout ce qu'il faut pour lancer la création de **toute sorte de site internet**.
Il peut être considéré comme un mini framework. Pour l'exploiter au mieux, il faut des bases solides en CSS, et ne pas débiter dans le codage informatique.
- **GroundworkCSS** – <https://groundworkcee.github.com>
GroundworkCSS est un framework **front-end** flexible doté d'un système de grilles particulièrement abouti.
Un framework idéal pour créer une application web accessible sur ordinateur comme sur smartphone ou tablette. Il s'adresse lui aussi aux initiés.
- **Cascade. Framework** - <https://jslegers.github.io/cascadeframework/>
Cascade Framework se distingue de Bootstrap en donnant plus de contrôle au développeur.
Il s'adapte aussi bien aux navigateurs récents ou aux plus anciens. Il sera donc possible au besoin de rétrograder votre conception afin qu'elle
- **Jeet** – <https://jeet.gs>
Jeet propose un système de grille très avancé qui vous aide à créer vos sites plus vite avec moins de code. Il permet un langage simplifié dans le code mais demande une bonne prise en main.
- **Milligram** - <https://milligram.io>

Milligram est lui aussi un **framework CSS** très léger qui se distingue notamment par un système de grille singulier.

C'est l'un des mini framework les plus simples d'utilisation sur le marché. Il est facile à comprendre et il s'intègre rapidement sur tous les sites.

- **MontageJS** - <https://montagejs.org>

MontageJS se spécialise dans la construction d'applications sur une seule page, garantissant une **expérience utilisateur maximale**.

Il permet de simplifier le développement des applications HTML5, pour une utilisation plus fluide sur des appareils aux ressources limitées.

- **Susy** – <https://www.oddbird.net/susy/>

Susy n'est pas un Framework à proprement parler, mais plutôt un générateur de grilles "à la demande" qui peut être extrêmement utile.

Vous pourrez construire n'importe quel design, mais attention, vous devez obligatoirement utiliser le préprocesseur Sass.

- **Topcoat** – <http://topcoat.io>

Créé par Adobe, **Topcoat** met la priorité sur la performance tout en étant assez customisable.

C'est une excellente bibliothèque **CSS open source**, qui aide les développeurs à créer des applications web toujours plus rapides. Pour une meilleure expérience des utilisateurs.

- **Ink** - <https://ink.sapo.pt>

Ink est un framework front-end pensé pour développer plus vite et plus simplement les interfaces utilisateur.

Une boîte à outils simple d'utilisation qui vous permettra de gagner du temps sur vos projets de création de sites web, sans avoir à recoder les basiques.

- **Backbone.js** - <https://backbonejs.org>

Ce framework s'utilise également en JavaScript. Plutôt simple d'utilisation et avec un ensemble de fonctionnalités plus que correct, **BackboneJS** devrait intéresser plus d'un développeur.

Il est principalement utilisé pour développer des applications web one-page

- **CodeIgniter** – <https://codeigniter.com>

CodeIgniter existe depuis 2006 et il a rapidement conquis de grandes entreprises grâce à sa rapidité et sa simplicité.

Sa prise en main est facile et plus vous l'utiliserez, plus vous parviendrez à maîtriser toutes ses fonctionnalités. Petit bonus, ce framework permet d'accélérer toutes vos applications web.

- **YiiFramework** – <https://www.yiiframework.com>

Yii Framework est l'un des frameworks proposant le plus de fonctionnalités de sécurité !

Il permet de coder en quelques minutes grâce à une drôle de fonctionnalité permettant de créer un modèle à adapter sur d'autres projets. Très pratique et véritable gain de temps pour les développeurs travaillant sur plusieurs projets à la fois.

- **Mithril** - <https://mithril.js.org>

Mithril.js est un framework JavaScript moderne côté client pour la création d'applications à page unique. Il est petit (< 10 Ko gzip), rapide et fournit des utilitaires de routage et XHR prêts à l'emploi.

- **Fat-Free Framework** – <https://fatfreeframework.com>

Fat-Free Framework est également un micro-framework PHP. Sa rapidité d'exécution est donc naturelle et toutes les fonctionnalités dont vous aurez besoin pour développer sont disponibles.

De plus, vous pouvez débiter dans le codage informatique et réussir parfaitement à utiliser Fat-Free Framework. Bonne nouvelle pour les apprentis développeurs.

➤ **Svelte** - <https://svelte.dev>

Svelte est un framework et compilateur JavaScript performant. Il permet de créer des applications web rapides et réactives. Une des meilleures solutions du moment.

3 – 4 – 3 - - Autres Framework

Flask - KOA pour Node JS - Express - Rails . Zend Framework - Codelgniter -
CakePHP - .Proton Native - NW.JS - -Polymer Project - Slim -. FuelPHP - Phalcon- -Material UI -
Materialize - Knaccs -- jQuery-Pyramid -. Bottle -. TurboGears-CherryPy - Aurelia --Next.JS -
Electron -Drupal -Web2Py -Sinatra

3 – 6 - Langages pour une application mobile

Les mobinautes devenant de plus en plus nombreux, les applications mobiles se doivent de répondre à des fonctionnalités toujours plus avancées pour répondre à leurs attentes. Si vous avez pour projet de développer une application mobile, le choix du langage de programmation va dépendre des systèmes d'exploitation

▪ **JAVA Android (§ -1-2)**

Le langage Java est adapté au développement d'une application mobile native sur Android. Il dispose d'une documentation importante, étant partagé par une grande communauté de développeurs dans le monde

▪ **SWIFT IOS (§- 1 – 9)**

Swift se base sur le langage Objective-C. Les éléments Swift peuvent s'y intégrer directement

▪ **C#,C ouC++ (§-1-4,1-6)**

▪ **OBJECTIVE- C (§-1 – 8)**

▪ **KOTLIN Android (§- 1 -14)**

▪ **FLUTTER**

Flutter est un cadre de développement d'applications mobiles gratuit et open source créé par **Google**. Il est utilisé pour développer des applications pour Android et iOS, écrites en **Dart**, un langage de programmation développé par Google. Avec Flutter, vous pouvez créer rapidement et facilement de belles applications natives.

▪ **REACT NATIVE – <https://reactnative.dev>**

React Native est né comme projet lors d'un hackathon interne chez Facebook, et s'est avéré si bon qu'il est devenu l'un des langages de codage les plus populaires, et qui peut donc créer des applications qui fonctionnent sur n'importe quelle plateforme.

▪ **CORDOVA – <https://cordova.apache.org>**

Apache Cordova ou plus anciennement Apache Callback ou PhoneGap, est un **framework** open-source développé par la Fondation Apache. Il permet de créer des applications pour différentes plateformes (Android, Firefox OS, iOS, Ubuntu, Windows 8...) en HTML, CSS et JavaScript.

OUTILS de CRÉATION D'APPLICATION MOBILE

Solutions clé en main pour générer des applications mobiles AppsBuilder très simplement et en quelques minutes, sans avoir besoin de connaître un langage particulier

- **GoodBarberS** – <https://fr.goodbarber.com>
GoodBarber est une vraie référence dans le domaine des générateurs d'applications mobile. Cet outil français s'est fait un nom grâce à ses templates très esthétiques, la simplicité de son éditeur et ses prix accessibles. GoodBarber est aussi à la pointe en matière de Progressive Web Apps.
- **Appligo** – <https://appligo.biz>
AppliGo est un générateur d'applications mobile simple d'utilisation, bénéficiant d'un support réactif et disponible en français.
- **Swiftic** – <https://www.swiftic.com>
Swiftic est l'un des générateurs d'applications leaders du marché. Son éditeur performant et ses nombreuses fonctionnalités en font un outil de choix. Disponible en français.
- **AppsBuilder**- [https://www.appypre.com/app\(builder/appmaker](https://www.appypre.com/app(builder/appmaker)
Avec son outil de création d'application en drag and drop, **Appsbuilder** est une des plateformes les plus intuitives et les plus user-friendly pour le développement d'une application en ligne.
- **AppMachine** – <https://www.appmachine.com>
Avec son éditeur original sous forme de blocs Légo, **AppMachine** est une solution très puissante pour créer votre application sans développeurs ni aucune connaissance.
- **Business Apps** – <https://www.businessapps.com>
La grande force de **Business Apps** est le nombre d'outils externes que l'on peut incorporer dans son application : de Magento à Instagram en passant par MailChimp. Il permet d'envoyer facilement des notifications push ou encore de mettre en place un programme de fidélité.
- **Shoutem** – <https://shouyen.com>
Shoutem est un outil de création d'applications bien pensé et proposant des designs très léchés, mais il est probablement moins accessible.
- **Mobile Roadie** - <https://mobileroadie.com>
Mobile Roadie est plutôt destiné à la création d'application pour les grandes organisations. Et, pour cause, il propose des fonctionnalités très avancées comme le ciblage géolocalisé du contenu de l'application, le marketing automation, des fonctions communautaires et sociales...
- **Mobincube** – <https://mobilecube.com/fr/>
Mobincube est un générateur d'applications mobile qui se veut accessible à tous, autant financièrement que techniquement
- **Siberian**- <https://www.siberiancms.com>
Siberian est un CMS open source permettant de créer une application gratuite.
- **Appy Pie** – <https://www.appypie.com/app-builder/appmaker>
Basé en Inde, la promesse d'**Appy Pie** est celle d'une création d'app mobile simple comme bonjour. Une fois que votre application est prête, vous pouvez la publier sur Google Play et App Store.
- **PandaSuite** – <https://pandasuite.com/fr/>
PandaSuite vous propose de créer votre application sans coder grâce à une interface drag and drop très intuitive. Profitez de sa librairie de composants pour ajouter des animations à votre application.
L'application peut également être connectée à un fournisseur de données interne ou externe.
- **BuildFire** – <https://buildfire.com>

BuildFire vous permet de construire facilement votre appli iOS ou Android avec un haut degré de personnalisation.

- **ModAppCreator** – <https://www.modappcreator.com>
MobAppCreator est un très bon outil pour développer votre propre application native avec simplicité.
- **Appsheet** – <https://about.appsheet.com>
Appsheet, c'est le générateur d'application mobile pour tous : instantané et abordable, il ne nécessite lui non plus aucune connaissance du code.
- **Appsmoment** – <https://appsmoment.com>
AppsMoment est un outil complet pour créer son appli, aux tarifs flexibles (d'une version gratuite limitée à presque 200 \$ / an).
- **Appinstitut** – <https://appinstitut.com>
AppInstitute est un générateur d'application mobile pratique doté d'un éditeur drag and drop, qui peut s'adapter à toutes sortes d'activités.
- **Glide** – <https://www.glideapps.com> Glide est un nouvel outil qui vous permet de créer des applications sans apprendre à coder. Cela peut être très utile si vous avez une idée d'application, mais que vous n'avez pas le temps d'apprendre à la concrétiser

3 – 4 – Langage de programmation pour IA

L'intelligence artificielle s'intègre de plus en plus dans le web et ses usages. Avec elle, de nombreux langages de programmation se développent. Les systèmes informatiques alimentés par l'IA

effectuent des tâches telles que :

- **la prise de décision complexe,**
- **la traduction linguistique,**
- **les transactions financières,**
- **la reconnaissance visuelle et l'autocorrection.**

3 – 4 – 1 – langages classiques

Java et Python sont les 2 codes les plus utilisés en programmation d'intelligence artificielle

- Python – voir § 3-1
- Java – voir § 3-2
- R – voir § 3 - 7

3 – 4 – 1- Le nouveau : Mojo

Mojo est un langage de programmation développé par [Modular IA](#), une entreprise d'infrastructure IA fondée par Chris Lattner. Il se présente comme une réponse aux besoins spécifiques des développeurs en matière d'intelligence artificielle, d'automatisation et de systèmes embarqués.

Grâce à sa syntaxe simple et intuitive, Mojo permet aux programmeurs d'écrire du code avec rapidité et efficacité. Il assure en retour d'excellentes performances. Ces deux avantages font de ce langage un outil réellement compétitif pour le [développement web](#).

En termes de vitesse d'exécution, **Mojo surpasse Python**. Ses créateurs affirment qu'il est 35 000 fois plus rapide. Cela s'explique par le fait qu'il associe la simplicité de ce langage à la performance de C/C+. De ce fait, Mojo est très complet et promet donc d'excellents résultats à l'utilisation. Il vise :

- Une compatibilité totale avec l'écosystème Python ;
- Un contrôle de bas niveau pour assurer des performances prévisibles ;
- L'absence de fragmentation de l'écosystème ;
- La possibilité de déployer des sous-ensembles de code sur des accélérateurs IA.

➤ **Les caractéristiques de Mojo**

Les inconditionnels du langage Python se laisseront facilement tenter par Mojo, en raison de ces caractéristiques impressionnantes. Découvrons trois d'entre elles.

- **Une syntaxe combinant Python et la portabilité de C**

Mojo possède une particularité : sa syntaxe est épurée, comparable à Python. En fait, le langage s'applique à rendre le code informatique lisible. Il permet ainsi aux développeurs de comprendre et de modifier facilement leurs lignes de code en utilisant des structures logiques. De plus, le langage propose un système de vérification et d'inférence. Celui-ci permet de traquer les erreurs de compile-time et donc de réduire les risques d'erreurs runtime. Mojo prend également en charge l'interopérabilité C/C++ et facilite de ce fait la vie aux programmeurs.

- **La présence du MILR (Multi-Level Intermediate Representation)**

Le langage Mojo prend en charge MILR. En d'autres termes, il permet aux développeurs d'accéder à une gamme complète de nouvelles fonctionnalités avancées telles que les unités matérielles AI, les threads et les vecteurs.

Le MILR améliore les performances du langage et lui permet de fonctionner avec un traitement parallèle sur plusieurs cœurs. À côté, Python ne prend en charge qu'un seul thread. C'est pour cela qu'il est beaucoup moins rapide que Mojo.

-**La compilation statique pour une exécution plus rapide**

L'entreprise Modular AI a développé le langage Mojo sur des technologies de compilateur de nouvelle génération. Celles-ci permettent de gagner significativement en performances lorsque vous ajoutez des types à vos programmes.

Elles permettent aussi de définir des abstractions sans coût et de profiter d'une sécurité de mémoire de type Rust. Cette approche de compilation statique garantit une vitesse d'exécution sans égale.

3 – 4 – 2 – récent

Julia et Scala ? Ces 2 langages de programmation sont très récents et connaissent une croissance notable ces dernières années.

3 – 4 -2 – 1 Julia ,

Julia est un langage proche de Python du point de vue de la syntaxe. Il est dy 4 -2-1 namique, facile à prendre en main, stable et libre à la fois. Sa première version a été lancée en 2009. Julia a notamment été repris par la [Climate Modeling Alliance](https://climate-modeling-alliance.org) pour la construction de leur futur modèle climatique. Julia aest un projet open source sur licence du MIT – <https://julia-lang.org>

Les écarts les plus significatifs de Julia par rapport aux langages dynamiques typiques sont:

- Le langage de base impose très peu; Julia Base et la bibliothèque standard sont écrites en Julia elle-même, y compris les opérations primitives comme l'arithmétique des entiers

- Un langage riche de types pour construire et décrire des objets, qui peut également être utilisé en option pour effectuer des déclarations de type
- La possibilité de définir le comportement des fonctions sur de nombreuses combinaisons de types d'arguments via [plusieurs distributions](#)
- Génération automatique de code efficace et spécialisé pour différents types d'arguments
- de bonnes performances, se rapprochant de celles des langages compilés statiquement comme C

3- 4 – 2 – 2 -Scala – <https://scalalang.org>

Scala est un beau **croisement des paradigmes de programmation orientée objet et fonctionnelle**. Il est rapide, robuste, et très populaire auprès de nombreux professionnels du Big Data. Les frameworks Apache **Spark** et Apache **Kafka** par exemple ont été construits sur Scala, ce qui témoigne de sa puissance.

Scala fonctionne sur la JVM, ce qui signifie que les codes écrits en Scala peuvent être facilement utilisés dans un écosystème Big Data basé sur Java. **Scala est beaucoup moins verbeuse par rapport à Java**. Vous pouvez écrire des centaines de lignes de code Java en moins de 15 lignes dans Scala. Scala présente cependant un aspect négatif : sa **courbe d'apprentissage abrupte**. Par rapport à des langages comme Go et Python, Scala pourrait décourager les débutants.

2 – 4 – 2 – 3 -Prolog

Prolog est un langage utilisé dans les domaines de l'Intelligence Artificielle et la Programmation Logique avec Contraintes. Sa syntaxe et son principe de fonctionnement sont radicalement différents de langages impératifs tels que C ou Java. Le raisonnement se rapproche plus de langages fonctionnels tels que Caml ou Lisp. Pourtant, Prolog n'est pas un langage fonctionnel. Prolog est le premier langage de programmation logique. Il existe différents outils pour vous permettre de programmer en Prolog.

SWI Prolog

Possède un débogueur graphique ainsi que plusieurs solveurs de contraintes

www.swi-prolog.org

GNU Prolog

Propose un solveur de contraintes sur domaine fini

<http://gnu-prolog.inria.fr>

Sicstus Prolog (payant)

Possède des extensions en plus, dont plusieurs solveurs de contraintes

www.sics.se/sicstu

3 – 4 – 2 – 4 - LISP

Lisp est développé à l'origine pour la notation mathématique dans les logiciels informatiques, Lisp a été **l'un des premiers langages de programmation d'IA**. Malgré son ancienneté, on se rend compte que de nombreux composants critiques des systèmes d'exploitation de robots sont encore codés en Lisp. En tant que tel, Lisp est également **un langage avant-gardiste** dans le domaine de l'intelligence artificielle. En fait, c'est l'inventeur de LISP, John McCarthy, qui a inventé le terme « intelligence artificielle » **La gestion automatique du stockage des données, la saisie dynamique et les fonctionnalités d'ordre supérieur** ne sont que quelques-uns de ses multiples avantages. Lisp (ou LISP, pour LISt Processing, « traitement de listes ») est une famille de langages de programmation créée en 1958 par John McCarthy, pionnier de l'intelligence artificielle.

Points forts de Lisp :

- **Simplicité syntaxique:** Lisp utilise une syntaxe basée sur des listes entre parenthèses, considérée comme élégante et facile à apprendre.
- **Programmation fonctionnelle:** Lisp met l'accent sur la programmation fonctionnelle, où les fonctions sont des citoyens de première classe et peuvent être manipulées comme des données.
- **Flexibilité et puissance:** Lisp est un langage très flexible et puissant, capable de s'adapter à une grande variété de tâches.
- **Large adoption en IA:** Lisp a été l'un des langages de choix pour la recherche en intelligence artificielle pendant plusieurs décennies.

Utilisations actuelles :

- Intelligence artificielle: Lisp est toujours utilisé dans certains domaines de l'IA, comme le traitement du langage naturel.
- **Développement de logiciels:** Lisp est utilisé pour développer des logiciels dans divers domaines, comme la finance, la biotechnologie et l'aéronautique.
- Recherche et éducation: Lisp est un langage populaire pour la recherche en informatique et l'enseignement de la programmation.

Avantages :

- Syntaxe simple et élégante
- Puissance et flexibilité
- Large adoption en IA
- Grand nombre de bibliothèques et d'outils

Inconvénients :

- Syntaxe atypique pouvant rebuter certains développeurs
- Moins répandu que d'autres langages
- Communauté moins active

En conclusion :

Lisp est un langage de programmation puissant et flexible avec une longue histoire et une large adoption dans le domaine de l'intelligence artificielle. Il offre une syntaxe simple et élégante et une grande variété de fonctionnalités.

Si vous êtes intéressé par la programmation fonctionnelle, l'intelligence artificielle ou si vous recherchez un langage puissant et flexible, **Lisp est un excellent choix.**

2 – 4 – 2 – 5 -Hasket

Haskell est un autre langage attrayant pour les programmeurs d'IA, offrant des avantages tels que :

une gestion efficace de la mémoire,
des systèmes de type,
et des capacités de réutilisation du code.

Haskell, c'est un langage de programmation fonctionnel qui met l'accent sur **l'exactitude du code.**

Haskell est l'un des langages de programmation d'IA **les plus sécurisé**, de sorte que les développeurs utilisant Haskell bénéficient de **la plus grande flexibilité et des capacités**

de gestion des erreurs. Par-dessus tout, Haskell procure l'efficacité de penser plutôt que de coder.

3 – 4 – 3 - Framework pour IA

Le terme « **Framework pour IA** » désigne un ensemble d'outils et de bibliothèques destinés à simplifier le développement d'applications d'intelligence artificielle. Ces frameworks fournissent un environnement structuré avec des fonctionnalités prédéfinies, permettant aux développeurs de se concentrer sur la logique et les algorithmes spécifiques de leur projet d'IA au lieu de tout construire à partir de zéro.

➤ **TensorFlow**

Open source issue de Google, **TensorFlow** est un Framework IA complet. Comportant plusieurs bibliothèques et outils de technologies prédictives, il a été créé dans le but de simplifier les processus de création des modèles d'apprentissage machine et autres réseaux neuronaux.

Grâce à des modèles graphiques et des organigrammes bien détaillés, ce dispositif permet une meilleure exploitation des données, sans oublier la rationalisation du développement de solutions informatiques très élaborées.

➤ **Amazon SageMaker Neo**

Amazon SageMaker Neo est un framework IA très récent. Après étude de son code source, récemment rendu public, il s'est avéré qu'il est principalement dédié aux développeurs d'IA souhaitant créer des modèles d'apprentissage machine et les exécuter sur le Cloud.

De par sa sophistication, Amazon SageMaker Neo est plus adapté pour fonctionner sur des appareils informatiques de pointe ou des capteurs de l'Internet des objets (IoT), vu leur nécessité à faire des prévisions rapides et à faible latence.

➤ **Microsoft CNTK -Cognitive ToolKit**

Microsoft CNTK est un framework IA open source basé sur des réseaux neuronaux. Supportant plusieurs langages de programmation, il est souvent mobilisé comme un outil d'apprentissage machine.

Grâce à sa rapidité et sa polyvalence, cet outil est utilisé par de grandes sociétés comme Skype, Bing ou Cortana, qui ont besoin d'une plateforme d'apprentissage évolutive et hautement optimisée pour gérer leurs ensembles de données.

Microsoft CNTK a été développé de sorte à être en parfaite adéquation avec les dernières avancées technologiques de l'intelligence artificielle.

➤ **Theano**

Basé sur Python, **Theano** est un framework IA qui dote les systèmes informatiques d'une très grande puissance de calcul. À la base, le développement de cette solution résulte d'un besoin criant de dispositifs IA pouvant assurer l'implémentation et la compilation des expressions mathématiques de manière efficace. Theano c'est aussi un framework qui permet d'exécuter le code source en utilisant à la fois le CPU et le GPU. Ce point bien précis constitue le principal avantage de cet outil.

➤ **KERAS**

Compatible avec TensorFlow, Theano et Microsoft CNTK, Keras fait partie des frameworks IA les plus puissants. Facile d'utilisation, il offre aux développeurs la capacité de prototyper rapidement les solutions qu'ils développent. Les capacités

prédictives de cette solution sont telles que plusieurs grandes entreprises l'utilisent pour la prévision et l'analyse de leur satisfaction client.

➤ **PyTorch**

PyTorch est un *Framework IA open source*, destiné à soutenir des opérations de calculs à grande échelle. Utilisé dans les laboratoires IA de Facebook et Twitter, il favorise la réalisation de tâches telles que la création, l'indexation, le découpage ou encore le transport de tableaux numériques multidimensionnels. En plus de s'intégrer facilement avec les systèmes d'exploitation mobiles iOS et Android, il présente plusieurs qualités qui justifient sa présence dans notre liste. En voici quelques-unes :

- Facilité d'utilisation grâce à des modèles préexistants ;
- Vitesse d'exécution ;
- Compatibilité avec un large panel de technologies de programmation.

➤ **Caffe**

Caffe est un réseau d'apprentissage en profondeur qui vient avec l'ensemble préchargé de réseaux de neurones entraînés. Cela devrait être votre premier choix si votre date limite est proche.

Connu pour ses capacités de traitement d'image, ce framework a également une prise en charge étendue de MATLAB.

Les caractéristiques

- Tous ses modèles sont écrits dans des schémas en clair
- Il offre une vitesse massive et un travail très efficace car il est déjà préchargé.
- Une communauté open source active pour la discussion et le code collaboratif

➤ **Apache Mahout**

Apache Mahout est un projet de la fondation Apache visant à créer des implémentations d'algorithmes d'apprentissage automatique distribués.

D'abord développé au-dessus de la plate-forme Hadoop, Mahout a ensuite utilisé Apache Spark. Mahout est encore en cours de développement ; le nombre d'algorithmes implémentés a rapidement augmenté, mais certains manquent encore

3 – 5 – langages de programmation graphique / programmation visuelle

Un **langage de programmation graphique** ou **visuel** est un langage de programmation dans lequel les programmes sont écrits par assemblage d'éléments graphiques. Sa syntaxe concrète est composée de symboles graphiques et de textes, qui sont disposés spatialement pour former des programmes. De nombreux langages visuels se basent sur les notions « de boîtes et de flèches » : les boîtes (ou d'autres d'objets) sont traitées comme des entités, reliées par des flèches ou des lignes qui représentent des relations.

Avantages

- Moins coûteux
- Base de connaissance accessible
- Développement plus rapide des logiciels
- Courbe d'apprentissage

Inconvénient

- Evolutivité limitée de la plateforme
- Potentiel limité

- Fastidieux et encombrant
- Domaine d'application particulier

Langages graphiques

3 – 5 -1 -LabVIEW- : Un langage graphique pour l'instrumentation et le contrôle

LabVIEW (contraction de Laboratory Virtual Instrument Engineering Workbench) est un langage de programmation graphique développé par **National Instruments**. Il est utilisé pour l'instrumentation et le contrôle d'instruments, l'acquisition et l'analyse de données, et le développement de systèmes embarqués.

Fonctionnement :

LabVIEW utilise un langage graphique appelé **G** pour construire des programmes.

Les programmes sont créés en assemblant des blocs fonctionnels prédéfinis, appelés fonctions, sur un diagramme.

Les fonctions sont reliées entre elles par des fils qui transportent les données.

Caractéristiques principales :

- **Environnement de développement graphique** : facilite la création et la compréhension des programmes.
- **Large bibliothèque de fonctions**: offre des outils pour de nombreux domaines d'application.
- **Connectivité matérielle** : permet de communiquer avec une grande variété d'instruments et de périphériques.
- **Compilateur intégré** : optimise les programmes pour une exécution rapide et efficace.

Utilisations de LabVIEW:

- **Acquisition de données** : permet de collecter des données à partir de capteurs et d'instruments.
- **Contrôle d'instruments** : permet de piloter des instruments et des machines.
- **Analyse de données** : offre des outils pour traiter et analyser les données collectées.
- **Développement de systèmes embarqués** : permet de créer des applications pour des systèmes autonomes.

Avantages de LabVIEW :

- **Rapidité de développement** : permet de créer des prototypes et de tester des concepts rapidement.
- **Réduction des erreurs** : la structure visuelle minimise les erreurs de syntaxe courantes.
- **Facilité d'apprentissage** : le langage graphique est intuitif et accessible aux débutants.
- **Flexibilité** : permet de créer des programmes complexes et personnalisés.

Site officiel de LabVIEW : <https://www.ni.com/en/shop/labview.html>

Tutoriels en ligne : <https://learn.ni.com/learn/article/labview-tutorial>

3– 5 – 2 -SCRATCH

Le langage graphique le plus connu et le plus répandu est **Scratch**. Il a été publié pour la toute première fois en 2007 avec pour ambition d'apprendre la programmation **de façon simple** aux enfants. Bouture de Scratch, **ScratchJr** est disponible sous forme d'application gratuite et s'adresse aux enfants de 5 à 7 ans. D'autres langages graphiques ont été développés à partir de Scratch, comme BYOB 4.0/Snap

3 – 6 – 3 - Blockly

Blockly est une bibliothèque logicielle. Elle offre un **éditeur de programmation graphique** qui est ajouté aux applications Android, iOS et aux applications web. Blockly utilise également des blocs graphiques imbriqués. Les algorithmes programmés peuvent par exemple être **exportés** sous forme de code JavaScript ou Python

3 – 5 -4 - NEPO.

NEPO est inspiré de Scratch et utilise la bibliothèque de Blockly en accès libre qui a été **complétée par d'autres fonctionnalités**. Les interfaces ouvertes pouvant également commander d'autres systèmes matériels ou robots constituent l'un des avantages de NEPO.

3– 5- 5App Inventor : langage de programmation pour applications mobiles

App Inventor est un langage de programmation graphique développé par le Massachusetts Institute of Technology (MIT). Il permet de créer des applications mobiles pour Android sans connaissances en programmation Java.

Fonctionnement :

App Inventor utilise un langage visuel par blocs pour construire des programmes.

Les blocs sont assemblés sur une interface graphique intuitive pour créer l'interface utilisateur et la logique de l'application.

- Il est possible d'intégrer des fonctionnalités avancées comme la géolocalisation, l'appareil photo et **les notifications**.

Caractéristiques principales:

- **Gratuit et open-source:** accessible à tous et modifiable librement.
- **Interface intuitive:** facile à prendre en main pour les débutants.
- **Large bibliothèque de composants:** offre des éléments pour créer des interfaces utilisateur riches.
- **Connectivité avec les services Google:** permet d'intégrer des services comme Google Maps et Google Drive.

Utilisations d'App Inventor:

- **Création de jeux simples :** jeux d'arcade, jeux de mémoire, etc.
- **Applications éducatives :** simulations, quiz, etc.
- **Outils de productivité:** agendas, listes de tâches, etc.
- Applications de domotique : contrôle d'appareils domestiques intelligents.

Avantages d'App Inventor :

- **Rapidité de développement:** permet de créer des prototypes et de tester des concepts rapidement.
- **Réduction des erreurs:** la structure visuelle minimise les erreurs de syntaxe courantes.
- **Facilité d'apprentissage:** le langage graphique est intuitif et accessible aux débutants.
- **Motivation pour l'apprentissage:** permet aux étudiants de découvrir la programmation de manière ludique.

App Inventor est un outil pédagogique précieux pour l'initiation à la programmation et au développement d'applications mobiles. Il permet aux étudiants de créer des applications fonctionnelles et interactives en s'amusant.

3- 5 – 6 - Lego Mindstorms

Une série de produits du fabricant de jouets Lego dont la pièce maîtresse est une **brique de lego programmable** : elle permet de construire et de programmer des robots et d'autres systèmes interactifs avec des moteurs électriques, des capteurs et des pièces de Lego Technic.

3 – 6 – Langage de script

Un **langage de script** est un langage de programmation **interprété** qui permet d'automatiser des tâches et d'ajouter des fonctionnalités à des logiciels ou des systèmes informatiques. Contrairement aux langages compilés, les scripts ne sont pas convertis en code machine avant l'exécution. Ils sont plutôt interprétés ligne par ligne par un interpréteur, ce qui les rend plus rapides à développer et à tester.

Utilisations des langages de script :

- **Automatisation** : Les langages de script peuvent être utilisés pour automatiser des tâches répétitives, ce qui permet de gagner du temps et d'améliorer la productivité.
- **Ajout de fonctionnalités**: Les langages de script peuvent être utilisés pour ajouter des fonctionnalités à des logiciels ou des systèmes existants.
- **Développement d'applications** : Les langages de script peuvent être utilisés pour développer des applications web et des applications autonomes.
- **Administration système** : Les langages de script peuvent être utilisés pour automatiser des tâches d'administration système.

Avantages des langages de script:

- **Rapidité de développement** : Les langages de script sont généralement plus rapides à développer que les langages compilés.
- **Facilité d'apprentissage** : Les langages de script sont généralement plus faciles à apprendre que les langages compilés.
- **Portabilité** : Les langages de script sont généralement portables entre différents systèmes d'exploitation.

Inconvénients des langages de script :

- **Performances** : Les langages de script peuvent être moins performants que les langages compilés.
- **Sécurité** : Les langages de script peuvent être plus vulnérables aux attaques que les langages compilés.

- **Conclusion**

Les langages de script sont des outils puissants qui peuvent être utilisés pour automatiser des tâches, ajouter des fonctionnalités à des logiciels et développer des applications. Ils sont généralement plus rapides à développer et à apprendre que les langages compilés, ce qui les rend populaires auprès des développeurs et des administrateurs système.

3 – 6 -1 – assimilable à langages de programmation de haut niveau

Python, PHP, Perl, Rubis ; Matlab, Julia, JavaScript, R ,Coffee,

3 – 6 -2 – nouveaux langages de programmation de haut niveau

3 – 6 -2 – 1 - LUA – <https://www.lua.org>

Lua est un langage de script rapide et léger. Le mot « lua » signifie « lune » en portugais, car la langue est développée et maintenue par l'Université catholique pontificale de Rio de Janeiro au Brésil. Lua soutient les paradigmes de programmation procédurale, orientée objet et fonctionnelle.

Comme l'interpréteur de Lua est écrit en C, il peut être facilement intégré dans des applications grâce à son API C. Cela étant dit, vous pouvez utiliser Lua pour étendre des applications existantes écrites dans des langages basés sur le C tels que C, C++, C#, Java, Perl, Ruby et d'autres. Lua est fréquemment utilisé pour développer des jeux vidéo, tels que Angry Birds, World of Warcraft et Grim Fandango. Comme il est facile à intégrer, il est également un choix fréquent pour les dispositifs intégrés tels que les décodeurs, les tableaux de bord de voitures (par exemple Volvo), les caméras IP (par exemple Cisco), et autres.

Il peut également être utilisé dans le développement web, car les serveurs Apache et Nginx disposent tous deux d'un module Lua

3– 6 – 2 – 2 -Groovy – <https://groovy-lang.org>

Apache Groovy est un langage **puissant, typé et dynamique**, avec des capacités de typage statique et de compilation statique, pour la plate-forme Java visant à améliorer la productivité des développeurs grâce à un **syntaxe**. Il s'intègre en douceur à n'importe quel programme Java et fournit immédiatement à votre application fonctionnalités puissantes, y compris les capacités de script, la création de langage spécifique au domaine, Méta-programmation et programmation fonctionnelle au moment de l'exécution et de la compilation. Cependant, il existe également de nombreuses implémentations de Groovy qui permettent de l'utiliser comme langage de script. Les plus populaires sont le framework web **Grails** (anciennement connu sous le nom de Groovy on Grails) et l'outil d'automatisation de la construction **Gradle**, mais il en existe aussi **plusieurs autres**.

3– 6 – 2 – 3 -GML

L'acronyme GML signifie **GameMaker Language**. C'est un bon exemple de langage de script spécifique à un domaine utilisé dans le développement de jeux. GML est un langage de script propriétaire appartenant à GameMaker Studio 2, un moteur de jeu multi-plateformes et une plateforme de développement appartenant à YoYo Games, qui en assure la maintenance. Même si GML est principalement utilisé pour contrôler les objets du jeu, ce n'est pas un langage orienté objet mais un langage procédural. Il vous permet d'appeler des scripts personnalisés à partir de n'importe quel objet de jeu.

Outre le langage de script GML, GameMaker Studio 2 dispose également d'un outil de script visuel appelé Drag and Drop (DnD). En raison de la nature flexible de GameMaker Studio 2, vous pouvez également mélanger DnD avec vos scripts GML ; ne pas confondre ce langage avec GML (Geography Markup Language) pour encoder et échanger des données géographique et GML (Generalized Markup Language), langage de balisage proposé par IBM en 1980N, ancêtre du XML

3 -6 – 3 - script shell

Le shell (ou interface système en français) est un programme qui reçoit des commandes informatiques données par un utilisateur à partir de son clavier pour les envoyer au système d'exploitation qui se chargera de les exécuter.

Aux débuts de l'informatique, le shell était la seule interface utilisateur disponible sur un système de type **Unix** tel que **Linux**. De nos jours et avec l'arrivée de la souris, tous les ordinateurs utilisent des interfaces utilisateur graphiques (**GUI**, pour « **graphical user interface** ») comme macOS ou Windows , en plus des interfaces de ligne de commande (CLI, pour « command line interface »), comme le shell.

Le mot « shell » est employé ici comme une métaphore. Shell signifie enveloppe ou coque en français : à l'inverse du noyau d'un ordinateur, le shell désigne la couche la plus haute de toutes les interfaces des systèmes Unix (Linux, macOS).

➤ **Bash**

Bash est l'un des outils de scripting les plus populaires disponibles sous Unix. C'est l'acronyme de Bourne Again Shell. C'est un outil puissant pour tout utilisateur de Linux ou administrateur système. Les scripts Bash sont un élément extrêmement puissant et utile pour le développement. Il permet de raccourcir les tâches répétitives en un seul appel de fonction. De nombreuses commandes longues peuvent être consolidées en un seul code exécutable.

Bash est facilement disponible sur presque toutes les versions de Linux et ne nécessite aucune installation séparée. La liste des shells disponibles peut être vérifiée en tapant la commande suivante : `cat etc/shell` . les variantes de bash sont :

- ✓ Unix a 2 grandes catégories de shells.
 - Bourne-type shell
 - C shell

Bourne shell est également classé comme :

- Korn shell (ksh)
- Bourne shell (sh)
- POSIX shell (sh)
- Bourne Again shell (bash)

C shell est également classé comme :

- C shell (csh)
- TENEX (TOPS) C shell (tcsh)

➤ **Powershell**

À l'origine, **PowerShell** était un shell en ligne de commande (successeur de `command` et un langage de script uniquement pour le système d'exploitation Windows. Depuis lors, Microsoft l'a développé en open source et l'a fait passer du [.NET Framework](#), qui ne peut créer que des applications Windows, au [.NET Core](#), qui peut créer des applications pour Windows, Linux et MacOS. Cela signifie que PowerShell est désormais multi-plateforme.

Il a également été renommé de Windows PowerShell en PowerShell Core, correspondant au framework sous-jacent. Contrairement à la plupart des shells en ligne de commande, PowerShell accepte et renvoie des objets .NET au lieu de texte en clair, ce qui ouvre de nouvelles possibilités d'automatisation des tâches

➤ **VBA**

VBA signifie Visual Basic for Applications et est une implémentation du langage de programmation **Visual Basic 6** (non développé activement depuis 2008). Il a été créé pour les applications Microsoft Office afin de permettre aux développeurs d'automatiser les tâches répétitives, d'ajouter de nouvelles fonctionnalités et d'interagir avec les utilisateurs finaux des documents.

Tout comme Visual Basic, VBA suit le paradigme de la programmation pilotée par les événements qui place des événements tels que les actions de l'utilisateur au centre du déroulement du programme.

Comme les applications Microsoft Office ont une interface utilisateur graphique, vous pouvez attacher des scripts VBA aux boutons de menu, aux raccourcis clavier, aux macros (patterns programmables) et aux événements **OLE** (Object Linking and Embedding qui vous permet de contrôler une application à partir d'une autre; c'est une technologie propriétaire de Microsoft). Visual Basic 6 est remplacé par VisualBasic.NET sur la plateforme .NET de Microsoft

➤ VBScript

VBScript (diminutif de Microsoft Visual Basic Scripting Edition, créé par Microsoft) est un sous-ensemble de Visual Basic utilisé en tant que [langage de script](#) d'usage général. Il est souvent comparé au JScript. Comme tout langage de script, VBScript est un langage interprété. Il ne nécessite pas de compilation avant d'être exécuté. En revanche, il nécessite que la machine destinée à les exécuter possède un interpréteur, un programme capable de comprendre l'ensemble des instructions présentes dans le programme. Selon l'utilisation les différents "interpréteurs" sont :

- [ASP](#) (*asp.dll*) dans un environnement Web
- wscript.exe dans un environnement [Windows](#)
- cscript.exe dans un environnement à ligne de commande
- mshta.exe pour les [applications HTML](#).

Les fichiers VBScript pour Windows Scripting Host ont généralement l'extension de fichier .vbs

4 – Classement par l'indice TIOBE (Pays-Bas- <https://www.tiobe.com>)

L'indice TIOBE Programming community est un indicateur de la popularité de la programmation. L'index est mis à jour une fois par mois. Les notes sont basées sur le nombre d'ingénieurs qualifiés dans le monde, les cours et les fournisseurs tiers. Les moteurs de recherche populaires tels que Google, Bing, Yahoo, Wikipédia, Amazon, YouTube et Baidu sont utilisés pour calculer les notes ; Il est important de noter que l'indice TIOBE ne concerne pas le **meilleur** langage de programmation ou *le langage dans lequel la plupart des lignes de code ont été écrites*.

4 – 1 indice TIOBE de janvier 2024

Titre de janvier : C++ est le langage de programmation de l'année 2024 de TIOBE !

C++ est le langage de programmation de TIOBE de l'année 2022. Il a remporté ce titre parce que C++ a gagné en popularité (+4,62 %) en 2022. Les finalistes sont C (+3,82 %) et Python (+2,78 %).

L'indice peut être utilisé pour vérifier si vos compétences en programmation sont toujours à jour ou pour prendre une décision stratégique sur le langage de programmation à adopter lors du démarrage de la construction d'un nouveau système logiciel. La définition de l'indice TIOBE se trouve [ici](#) :

Feb 2024	Feb 2023	Change	Programming Language		Ratings	Change
1	1			Python	15.16%	-0.32%
2	2			C	10.97%	-4.41%
3	3			C++	10.53%	-3.40%
4	4			Java	8.88%	-4.33%
5	5			C#	7.53%	+1.15%
6	7	^		JavaScript	3.17%	+0.64%
7	8	^		SQL	1.82%	-0.30%
8	11	^		Go	1.73%	+0.61%
9	6	v		Visual Basic	1.52%	-2.62%
10	10			PHP	1.51%	+0.21%
11	24	^		Fortran	1.40%	+0.82%
12	14	^		Delphi/Object Pascal	1.40%	+0.45%
13	13			MATLAB	1.26%	+0.27%
14	9	v		Assembly language	1.19%	-0.19%
15	18	^		Scratch	1.18%	+0.42%
16	15	v		Swift	1.16%	+0.23%
17	33	^		Kotlin	1.07%	+0.76%
18	20	^		Rust	1.05%	+0.35%
19	30	^		COBOL	1.01%	+0.60%
20	16	v		Ruby	0.99%	+0.17%

a. – **2 - Le top 50 complet** des langages de programmation est répertorié ci-après

.

Position	Programming Language				
21	(Visual) FoxPro	31	COBOL	41	PowerShell
22	SAS	32	Scala	42	Awk
23	Objective-C	33	F#	43	Prolog
24	Lua	34	Logo	44	CFML
25	Kotlin	35	TypeScript	45	Haskell
26	Ada	36	Groovy	46	D
27	Fortran	37	Bash	47	LabVIEW
28	Lisp	38	Dart	48	Scheme
29	Julia	39	RPG	49	ABAP
30	Transact-SQL	40	PL/SQL	50	OCaml

4 – 3 – liste des 10 langages non nommés issus du classement Tiobe

4 – 3 – 1 – SAS -rang 22- <https://www.sas.com>

Le langage de commande de **SAS**, pour *Statistical Analysis System*, est un langage propriétaire de programmation de quatrième génération (L4G) édité par SAS Institute depuis 1976.

Depuis 2004, SAS en est à la version 9, ce qui correspond à une évolution majeure dans le logiciel car il intègre une nouvelle brique conceptuelle destinée à s'implanter dans le monde des logiciels d'informatique décisionnelle. Le logiciel SAS traditionnel se compose d'un ensemble de modules permettant de répondre aux besoins suivants par la programmation :

- Création et gestion de bases de données ;
- Traitements analytiques des bases de données ;
- Création et diffusion de rapports de synthèse et de listing.

4 – 3 – 2 – Ada – rang 26 - <https://www.ada-France.org>

Ada est un langage de programmation qui a été normalisé en 1983 après avoir été conçu en réponse à un cahier des charges conçu par le DOD (département de la Défense) des États-Unis. En 1995, une nouvelle version de ce langage a été définie afin de renforcer la sécurité et les capacités concernant la programmation orientée objet. Une nouvelle version Ada2005 renforce les capacités du langage concernant la programmation orientée objet, la programmation concurrente et les bibliothèques prédéfinies.

Il doit son nom à Ada Lovelace. Ada est un langage à typage statique, modulaire et offrant une syntaxe claire inspirée de **Pascal**. Il est souvent utilisé dans des systèmes temps-réel nécessitant un haut niveau de fiabilité

4 - 3 – 3 – Fortran -rang 27 – <https://fortran-lang.org> – pour le calcul scientifique

Fortran (FORmula TRANslating system) est un langage de programmation généraliste dont le domaine de prédilection est le calcul scientifique et le calcul numérique.

Historique:

Créé en 1954 par John Backus chez IBM.

Premier langage de programmation de haut niveau largement utilisé.

A connu plusieurs versions : Fortran II, Fortran IV, Fortran 77, Fortran 90, Fortran 95, Fortran 2003, Fortran 2008, Fortran 2018 et Fortran 2023.

Caractéristiques :

- Typé statique fort: Permet de détecter les erreurs de programmation et d'optimiser le code.
- **Syntaxe concise et lisible**: Facilite l'apprentissage et la maintenance du code.
- Particulièrement performant pour les calculs scientifiques: Offre un large éventail de bibliothèques dédiées.
- Largement utilisé dans la communauté scientifique: Bénéficie d'un support important et de nombreux outils.

Domaines d'application :

- **Simulateurs numériques**: Modélisation de phénomènes physiques et chimiques.
- **Analyse de données**: Traitement et interprétation de données scientifiques.
- **Calcul haute performance**: Applications nécessitant une puissance de calcul importante.
- **Intelligence artificielle**: Développement d'algorithmes d'apprentissage automatique.

Avantages :

- **Performance** : Code optimisé pour les calculs scientifiques.
- **Fiabilité** : Détection et correction des erreurs de programmation.
- **Portabilité** : Code compatible avec différents systèmes d'exploitation.
- **Facilité d'apprentissage** : Syntaxe simple et intuitive.

Inconvénients :

- **Syntaxe verbeuse**: Comparé aux langages plus modernes.
- **Manque de fonctionnalités**: Pour le développement d'applications générales.
- **Image vieillissante**: Moins populaire auprès des nouveaux développeurs.

Conclusion:

Fortran est un langage de programmation puissant et fiable pour le calcul scientifique. Il offre une performance optimale et un large éventail de bibliothèques dédiées. Bien qu'il puisse paraître vieillot, il reste un choix pertinent pour les développeurs qui ont besoin de puissance et de précision dans leurs applications scientifiques.

4 – 3 – 4 - Cobol -rang 31

Le **COBOL**, ou "Common Business Oriented Language", [existe depuis 1959](#) et est largement reconnu pour avoir contribué à la création du secteur des logiciels informatiques tel que nous le connaissons aujourd'hui. Il a dominé les systèmes informatiques des entreprises et des administrations au cours des décennies suivantes, et les grandes organisations s'appuient toujours sur ce langage pour faire fonctionner leurs ordinateurs centraux et leurs systèmes commerciaux de base, comme les systèmes de paie et de comptabilité. Cobol survit aujourd'hui dans quelques incarnations. IBM maintient activement ses propres mises en oeuvre Cobol et gère de nombreuses applications qui les exploitent. Micro Focus Cobol est une édition commerciale qui s'exécute sur Windows de Microsoft, compile les applications Cobol vers Java et .Net et se déploie même dans des environnements clouds comme Azure. On trouve aussi des mises en oeuvre open source de Cobol, telles que GnuCobol, qui est disponible gratuitement et compile vers le code machine source. Toutefois, elles peuvent ne pas disposer des fonctionnalités de déploiement ou de débogage les plus avancées des Cobol commerciaux

Avantages :

- **Facilité d'apprentissage** : Syntaxe simple et intuitive.

- **Lisibilité**: Code facile à comprendre et à maintenir.
- **Portabilité** : Code compatible avec différents systèmes d'exploitation.
- **Fiabilité** : Détection et correction des erreurs de programmation.

Inconvénients :

- **Verbosité** : Code plus long que les langages modernes.
- **Manque de fonctionnalités** : Pour le développement d'applications web et mobiles.
- **Image vieillissante** : Moins populaire auprès des nouveaux

4 - 3 – 5 - Dart-rang 38- <https://dart.dev>

Dart est un langage de programmation optimisé pour les applications sur plusieurs plateformes. Il est développé par Google et est utilisé pour créer des applications mobiles, de bureau, de serveur et web.

Dart est un langage orienté objet à ramasse-miettes (récupérateur de mémoire)avec une syntaxe de type C++¹. Dart peut se compiler en code natif ou en JavaScript. Il prend en charge les interfaces, les mixins, les classes abstraites, les génériques réifiés et l'inférence de type. **Flutter est** un kit de développement logiciel (SDK) d'interface utilisateur open-source créé par Google. Il **est** utilisé pour développer des applications pour Android, iOS, Linux, Mac, Windows, Google Fuchsia et le web à partir d'une seule base de code. **Flutter est un framework** et non un langage de programmation.

4 – 3 – 6 - RGP -rang 39

RPG est l'acronyme de **R**eport **P**rogram **G**enerator, francisé en GAP pour **G**énérateur **A**utomatique de **P**rogrammes.

RPG est avant tout un langage destiné à la gestion (au même titre que [Cobol](#)) et à la manipulation de bases de données. Il est équipé d'instructions spécialisées pour manipuler les fichiers de données de façon la plus efficace et la plus concise possible.

Le langage est né en 1961 avec une syntaxe très rigide, en *colonnes* héritées des cartes perforées, où la position des opérandes et des opérateurs était imposée. Ses incarnations plus récentes, en particulier RPG/IV apparu en 1995, permettent de disposer beaucoup plus librement le code source.

RPG est indissolublement lié aux systèmes IBM, en particulier l'AS/400 : la très grande majorité des applications développées sur ces systèmes le sont en RPG.

4 – 3 – 7 -Awk -rang 42 -

awk — dont le nom vient des trois créateurs, Alfred Aho, Peter Weinberger et Brian Kernighan — est un langage de traitement de lignes, disponible sur la plupart des systèmes Unix et sous Windows avec Cygwin ou Gawk. Il est principalement utilisé pour la manipulation de fichiers textuels pour des opérations de recherches, de remplacement et de transformations complexes.

Awk est le plus souvent utilisé pour la production de fichiers plats aux spécifications particulières (échanges entre différents systèmes d'informations hétérogènes). Il est aussi utilisé comme "parser" de fichiers XML ou de fichiers textes pour générer des commandes SQL à partir des données extraites. Il peut être utilisé aussi pour des opérations de calculs complexes et mise en forme de données brutes pour faire des tableaux statistiques.

4 – 3 – 8 - CFML-rang 44

ColdFusion Markup Language , plus communément appelé **CFML** , est un langage de script pour le développement Web qui s'exécute sur la JVM , le framework .NET et Google App Engine . Plusieurs implémentations commerciales et open source de moteurs CFML sont

disponibles, notamment **Adobe ColdFusion** , **Lucee** , **New Atlanta BlueDragon** (qui proposent à la fois une version basée sur Java et une version basée sur .NET), **Railo** et **Open BlueDragon** ainsi que d' autres moteurs de serveur CFML . Dans sa forme la plus simple, Web, CFML complète les fichiers HTML standard avec des commandes de base de données, des opérateurs conditionnels , des fonctions de formatage de haut niveau et d'autres éléments pour produire des applications Web . CFML comprend également de nombreuses autres constructions, notamment les composants ColdFusion (CFC), la version des objets de CFML , qui permettent de séparer la logique métier de la présentation.

4 – 3 – 9 -Scheme-rang 48 – <https://www.scheme.com>

Scheme (prononciation : /skim/) est un langage de programmation dérivé du langage fonctionnel Lisp, créé dans les années 1970 au Massachusetts Institute of Technology (MIT) par Gerald Jay Sussman et Guy L. Steele.

Le but des créateurs du langage était d'épurer le Lisp en conservant les aspects essentiels, la flexibilité et la puissance expressive. Scheme a donc une syntaxe extrêmement simple, avec un nombre très limité de mots-clés. Comme en Lisp, la notation préfixée permet de s'affranchir d'une précédence des opérateurs. De plus, la puissance des macros de Scheme lui permet de s'adapter à n'importe quel problème, notamment de le rendre orienté objet et donc multi-paradigme.

4 – 3 - 10 -Abap-rang 50

ABAP est un langage de programmation propriétaire, faisant partie de l'ensemble logiciel SAP. Il s'agit actuellement du langage utilisé dans la programmation des Web Application Server faisant partie de la plateforme NetWeaver pour la réalisation de progiciels. Sa version *ABAP/4* est objet, le chiffre 4 faisant de plus référence à son appartenance à la classe des langages de quatrième génération. ABAP (Advanced Business Application Programming)est l'un des successeurs du Cobol et est apparu dans les années 1980 dans la vague des langages de quatrième génération (L4G). Il s'agit d'un dérivé du langage permettant de réaliser des sorties de données (appelées *rappports*) de l'application SAP R/2, sur lequel de nombreuses multinationales avaient bâti leur architecture d'application professionnelle. ABAP a par la suite été maintenu comme langage de référence pour les applications SAP R/3 qui sont apparus en 1992.

5 – PYPL Popularité du Langage de programmation – <https://pypl.github.io>

Le PYPL PopularitY of Programming Language Index est créé en analysant la fréquence à laquelle les didacticiels de langage sont recherchés sur Google : plus un didacticiel de langage est recherché, plus le langage est supposé être populaire. C'est un indicateur avancé. Les données brutes proviennent de Google Trends.Si vous croyez en la sagesse collective, l'indice de popularité du langage de programmation PYPL peut vous aider à décider quel langage étudier ou lequel utiliser dans un nouveau projet logiciel.

Rang		Langage	part	tendance
1		Python	27.93 %	-0.9 %
2		Java	16.78 %	-1.3 %
3		JavaScript	9.63 %	+0.5 %
4		C#	6.99 %	-0.3 %
5		C/C++	6.9 %	-0.5 %
6		PHP	5.29 %	-0.8 %

7		R	4.03 %	-	0.2 %
8	↑↑↑	TypeScript	2.79 %		+1.0 %
9		Swift	2.23 %		+0.3 %
1	↓↓	Objective	-2.2 %		-0.1 %
11	↑↑	Go	1.94 %		+0.7 %
12	↑↑↑	Rust	1.9 %		+0.9 %
13	↓	Kotlin	1.81 %		+0.1 %
14	↓↓↓↓	Matlab	1.63 %		-0.1 %
15	↑	Ruby	1.13 %		+0.3 %
16	↓↓	VBA	1.03 %		-0.0 %
17		Ada	0.89 %		+0.2 %
18	↑↑↑	Dart	0.86 %		+0.5 %
19		Scala	0.62 %		+0.0 %
20	↓↓	Visual Basic	0.56 %		-0.1 %
21	↑↑↑	Lua	0.55 %		+0.2 %
22	↓↓	Abap	0.5 %		+0.1 %
23	↑↑	Haskell	0.35 %		+0.1 %
24	↑↑↑↑	Julia	0.34 %		+0.1 %
25	↓↓↓	Groovy	0.34 %		-0.0 %
26		Cobol	0.33 %		+0.1 %
27	↓↓↓↓	Perl	0.33 %		+0.0 %
28	↓	Delphi/Pascal	0.12 %		-0.1 %

En France , les part de marché en 2023 des langages sont :

:

	Change	Language	Share	Trend
1		Python	24.22 %	-0.7 %
2		Java	11.19 %	-0.7 %
3	↑↑↑↑↑	JavaScript	6.95 %	+2.8 %
4	↑↑↑↑↑	C/C++	5.71 %	+1.8 %
5	↓↓	C#	4.32 %	-3.1 %
6	↑↑↑↑	R	3.83 %	+0.2 %

6 – Environnement développement d'application

En programmation informatique, un **environnement de développement** est un ensemble d'outils qui permet d'augmenter la productivité des programmeurs qui développent des logiciels. Il comporte un éditeur de texte destiné à la programmation, des fonctions qui permettent, par pression sur un bouton, de démarrer le compilateur ou l'éditeur de liens ainsi qu'un débogueur en ligne, qui permet d'exécuter ligne par ligne le programme en cours de construction. Certains environnements sont dédiés à un langage de programmation en particulier.

6 – 1 – différents modèles

En pratique, on a différents modèles qui constituent tout ou en partie un environnement de développement :

6 – 1 – 1 – Framework

Un Framework en informatique est un ensemble d'outils et de composants logiciels qui aident les développeurs à créer des applications plus rapidement et plus facilement. Ils fournissent une structure et une base de code commune, ce qui permet aux développeurs de se concentrer sur la logique métier de leur application plutôt que de réinventer la roue à chaque fois.

Types de Framework

Il existe de nombreux types de Framework, chacun ayant ses propres forces et faiblesses. Parmi les types les plus courants, on trouve :

- **Web frameworks:** Ils sont utilisés pour créer des applications web dynamiques. Parmi les exemples populaires, on trouve Django (Python), Ruby on Rails (Ruby), Spring Boot (Java) et Laravel (PHP).
- **Mobile frameworks:** Ils sont utilisés pour créer des applications mobiles natives. Parmi les exemples populaires, on trouve React Native (JavaScript), Flutter (Dart) et Kotlin Multiplatform Mobile (Kotlin).
- **Desktop frameworks:** Ils sont utilisés pour créer des applications de bureau. Parmi les exemples populaires, on trouve Electron (JavaScript), Qt (C++) et wxWidgets (C++).
- **Micro-frameworks:** Ils sont plus légers et plus simples que les frameworks traditionnels. Ils sont souvent utilisés pour créer des applications web simples et des API. Parmi les exemples populaires, on trouve Flask (Python), Express (JavaScript) et Sinatra (Ruby).

Avantages des Framework

Les Framework offrent de nombreux avantages aux développeurs, notamment :

- **Productivité accrue :** Ils permettent aux développeurs de créer des applications plus rapidement et plus facilement.
- **Code plus propre et plus maintenable :** Ils fournissent une structure et une organisation au code, ce qui le rend plus facile à comprendre et à maintenir.
- **Meilleure qualité du code :** Ils incluent souvent des fonctionnalités de test et de débogage qui permettent d'améliorer la qualité du code.
- **Réduction des coûts:** Ils peuvent réduire les coûts de développement en permettant aux développeurs de réutiliser du code et des composants existants.

Inconvénients des Framework

Les Framework peuvent également avoir quelques inconvénients, notamment :

- **Courbe d'apprentissage:** Il peut être nécessaire d'investir du temps pour apprendre à utiliser un nouveau framework.
- **Manque de flexibilité:** Les frameworks peuvent être rigides et ne pas toujours s'adapter parfaitement aux besoins spécifiques d'une application.
- **Dépendance au framework:** Le choix d'un framework peut créer une dépendance à ce Framework, ce qui peut rendre difficile le changement de framework ultérieurement.

Conclusion

Les frameworks peuvent être un outil précieux pour les développeurs. Ils peuvent aider à améliorer la productivité, la qualité du code et la maintenabilité des applications. Cependant, il est important de choisir le bon framework pour les besoins spécifiques d'une application et de prendre en compte les inconvénients potentiels.

6 – 1 – 2 – Atelier de génie logiciels

Un atelier de génie logiciel (AGL), aussi appelé atelier CASE (Computer Aided Software Engineering), est un ensemble d'outils logiciels qui assistent au développement de logiciels. Il s'agit d'un environnement de développement intégré qui propose des fonctionnalités pour toutes les phases du cycle de vie d'un logiciel, de la conception à la maintenance.

Fonctions principales d'un AGL:

- **Modélisation:** Création de diagrammes et de modèles pour représenter les différents aspects du logiciel, tels que l'architecture, les exigences, les processus métier et les données.
 - **Gestion de code :** Stockage, organisation et modification du code source.
 - **Compilation et génération de code:** Traduction du code source en langage machine et création des exécutables.
 - **Test et débogage :** Identification et correction des erreurs dans le logiciel.
 - **Gestion de configuration :** Suivi des modifications apportées au logiciel et contrôle des versions.
 - **Documentation :** Génération de la documentation technique et des guides d'utilisation.

Avantages d'utiliser un AGL:

- **Amélioration de la productivité:** Automatisation des tâches répétitives et centralisation des informations du projet.
- **Meilleure qualité du logiciel:** Détection et correction des erreurs plus facile grâce aux outils de test et de débogage.
- **Meilleure communication et collaboration:** Facilitation du travail en équipe et du partage d'informations entre les différents intervenants du projet.
- **Meilleure maintenance du logiciel:** Facilite la mise à jour et l'évolution du logiciel grâce à la gestion de configuration.

Le choix d'un AGL dépend de plusieurs facteurs, tels que la taille et la complexité du projet, les méthodologies de développement utilisées, les compétences des développeurs et le budget disponible. Il est important de bien comparer les différents outils disponibles avant de faire un choix.

Quelques AGF (Wikipédia ‘ Atelier de génie logiciel) :

AD-Cycle d'IBM, AGL Merise/2, AGL-X (Prisme), ArgoUM, ARIS d'IDS-Scheer, ASA & GEODE, Advantage Plex, Adélia (Hardis), Application Builder d'Enablon, BOUML, Case*dictionary d'Oracle, Clarion, CodeFluent Entities de SoftFluent, Corporate Modeler de Casewise, C++ Builder, DB-MAIN, Dalyo, DoMIS, Enterprise Architect de Sparx Systems, Envision SART, Excelerator, IntelliJ IDEA, KDevelop, Mega, Method/1, Modelio, NATSTAR de Nat System, Net Express de Micro Focus, NetBeans, NSDK de Nat System, Objectteering, Optima PLC atelier d'automatisme sur PC (norme IEC 61131-3), Pacbase, PACKflow de Mica-Systems, Perceptory, PowerAMC de Sybase, Rational Rose d'IBM, (en) Synon, UNIFACE, Visual Paradigm, Visual Studio, WEBDEV, WINDEV, WINDEV Mobile

6 - 1 - 3 – Environnement intégré (EDI/IDE)

Dans un environnement de développement « intégré » (abrégé **EDI** en français ou **IDE** en anglais, pour ***integrated development environment***), les outils sont prévus pour être utilisés **ensemble** (le produit d'un outil peut servir de matière première pour un autre outil)¹. Les outils peuvent être intégrés dès le départ, c'est-à-dire qu'ils sont construits dans le but d'être utilisés ensemble. Il peut aussi s'agir d'un ensemble d'outils développés sans lien entre eux et intégrés a posteriori⁴.

Un **AGL** couvre toutes les phases d'un projet (de la conception à la validation, en passant par la gestion de projet) alors qu'un **IDE** ne couvre que la phase de réalisation. On attribue à tort l'étiquette 'AGL' à n'importe quel IDE, de nos jours..

6 – 2 – EDI par langage

6 – 2 – 1 – Développement JAVA

Intellij IDEA, NetBeans, Eclipse, VScode, BlueJ, JDeveloper, DrJava, JCreator, JGRASP, Greenfoot, Xcode, MyEclipse, red Hat OpenShift Dev Spaces

6 – 2 – 2 - IDE PHP

Eclipse IDE, PHPStorm, VScode, NetBeans, Komodo, Eclipse Che, CodeLite, Zend Studio, codelobster ; Cloud9

6 – 2 – 3 – IDE Python

Idle, eric, PyDev, PyCharm, Thonny, VScode, Atom, JupyterLab, Spider, CodeVisuel

6 – 2 – 4 - IDE C , C++

Code ::blocks, Netbeans, Codelite, Turbo++, C++Builder, Anjuta, Clion, Visual Studio, QT Creator, Studio Gnat

6 – 2 – 5 – IDE Javascript/Node/ TypeScript

Visual Studio Code, Web, Storm, Vim, Sublime Text, VScode, Atom, Brackets, Emabs, codepen, JS Bin, AWS Cloud9, JSFiddle

6 – 2 – 5 – IDE HTML/CSS

VScode, Atom

6 -2 - 6 – IDE pour SDL

Adminer, DBeaver, Microsoft SQL Server, RazorSDL, Squirrel, MySQL, HeidiSQL, Oracle SQL, Valentina Studio, Toad For SQL

6 – 3 – EDI par application

6 – 3 – 1 - Développement des sites Web

Eclipse , NetBeans, Scilab, Visual Studio, Visual Studio Express Edition, Delphi, Code::Blocks, MonoDevelop, SharpDevelop, KDevelop, Access, C++ Builder, MATLAB, , LabVIEW, Dev-C++, Qt Creator, XCode, Lazarus ,WinDev, 4D, Dreamweaver, Zend Studio

6 – 3 – 2 - Application mobile (Android)

Android Studio (intellij IDEA), Eclipse, Visual studio (avec Xamarin), intellij IDEA, NetBeans, Komodo , Cordova, PhoneGap, Accelerator Titanium, App inventor

6 – 3 – 3 – IDE pour les "Data scientits" (Big Data)

Grâce à sa flexibilité et la simplicité de sa syntaxe, Python figure parmi les langages dont le traitement est le plus rapide. Cela le rend parfait pour le Big Data en raison de la volumétrie des informations à traiter en un temps réduit.

Spider, JupyterLabs, PyCharm, VS code, Atom, Tonny DataSpell, JetBrains

6 – 3 – 4- plateforme IA et ML(Machine Learning)

Amazon AI services, TensorFlow, Google AI Services, H2O, Petuum, Polyaxon, DataRobot, NeuralDesigner, predictionI

6 – 4 – Classement suivant l'indice PYPL

6 –4 –1 - TOP EDI index

1	Visual Studio	12	Code ::Block	23	JCreator
2	Visual Studio Code	13	Vim	24	DrJava
3	Eclipse	14	PhpStorm	25	Code 2
4	PYCharm	15	Komodo	26	Monkey Studio
5	Android Studio	16	Xamarin	27	RubyMine
6	Intellij	17	QtCreator	28	MonoDevelop
7	NetBeans	18	Emacs	29	Aptane
8	Sublim Text	19	Geany	30	Eric Python
9	RStudio	20	Jdeveloper	31	Julia Studio
10	Atom	21	Light table	32	SharpDevelop
11	Xcode	22	Rad Studio	33	Zend Studio

6 – 3 – 2- TOP EDI en Ligne : ODI

L'index TOP ODE est créé en analysant la fréquence à laquelle les noms des IDE en ligne sont recherchés sur Google.

Plus un IDE en ligne est recherché, plus l'IDE en ligne est supposée être populaire. Les données brutes proviennent de **Google Trends**.

Si vous croyez en la sagesse collective, l'index TOP ODE peut vous aider à décider quel IDE en ligne utiliser pour votre projet de développement logiciel.

1	JSFidle	11	EDICloud9
2	Codage	12	Violon Python
3	Codio	13	EDI PHP en ligne
4	PythonAnywhere	14	Codiade
5	Jeu de fléchettes	15	Ice Coder
6	Idée	16	CodePique-nique

7	Repl.it	17	MajModifier
8	AWScloud9	18	Nitrous.io
9	Goorm	19	Envie de code
10	Coden'importe ou	20	EDI Orion

Annexe 1 : Ressources

- 1 – **Yeeply** - <https://fr.yeeply.com/blog/langages-programmation-type-developpement/>
- 2 – **indice TIOBE** : <https://www.tiobe.com/tiobe-index>
- 3 – **indice PYPL** – <https://pypl.github.io/PYPL.html>
- 4 - **cleveroad** - <https://www.cleveroad.com/blog/programming-langages-ranking/>
- 5 – **Digitiz** - <https://digitiz.fr/blog/langages-de-programmation/>
- 6 – **Warketingdigital** - <https://www.warketingdigital.net/langages-de-programmation-2022/>
- 7 – **Webitech** - <https://webitechparis.com/blog/langages-de-programmation-en-2022/>
- 8 – **Kinska** – <https://kinsta.com/fr/blog/langages-de-script/>
- 9 – **Codeur** – <https://www.codeur.com/blog/top-langages-de-programmation>
- 10 – **Mojo**- <https://blog.lesjeudis.com/mojo-langage-programmation-ia-plus-rapide-que-python>

Annexe 2 : position des langages

Langage zone page			Langage zone page			Langage zone page		
Abap	4-3-10	63	HTML/CSS	2-17	30	RGP	4-3-6	63
Ada	4-3-2	61	JAVA	2-3	20	RuBy	2-16	30
AppInventor	3-6-5	55	JavaScript	2-4	21	Rust	2-12	26
Awk	4-3-7	63	Julia	3-4-2-1	49	SAS	4-3-1	61
Blockly	3-5-3	55	KotLin	2-15	29	Scala	3-4-2-2	50
C#	2-5	22	labView	3-5-1	54	Scheme	4-3-9	63
C++	2-7	23	Latex	1-8-2	7	Scratch	3-5-2	54
CC	2-2	19	Legomindstorms	3-5-6	56	SGML/HTML	1-8-5	8
CFMI	4-3-8	63	Lisp	3-4-2-4	50	SPARQL	1-10-6	15
Cobol	4_3_4	62	LUA	3-6-2-1	57	SQL	1-10-7	16
CSS	1-9-1	10	Matlab	2-11	26	Swift	2-10	25
Dart	4-3-5	63	MDX	1-10-3	14	TEI	1-8-6	9
Datalog	1-10-1	13	Mojo	3-4-1	48	TypeScript	2-13	27
Delphi	3-2-3-3	39	MySQL	3-2-3-1	35	XML	1-8-3	8
DMX	1-10-2	13	Nepo	3-5-4	55	Xpath	1-10-9	17
DSSSL	1-9-3	11	ObjectifC	2-9	25	Xquery	1-10-8	16
F#	2-18	31	OOL	1-10-4	14	XSL	1-9-2	10
Fortran	4-3-3	61	Pascal	3-2-3-4	40			
FTP	3-2-3-2	36	Perl	3-2-3-3	38			
gencode	1-8-1	7	PHP	2-6	23			
GML	1-8-4	8	ProLog	1-10-5	15			
GML	3-6-2-3	57	Prolog	3-4-2-3	50			
GO	2-14	28	Python	2-1	18			
Groovy	3-6-2-2	57	QBE	1-10-10	18			
Hasket	3-4-2-5	51	R	2-8	24			