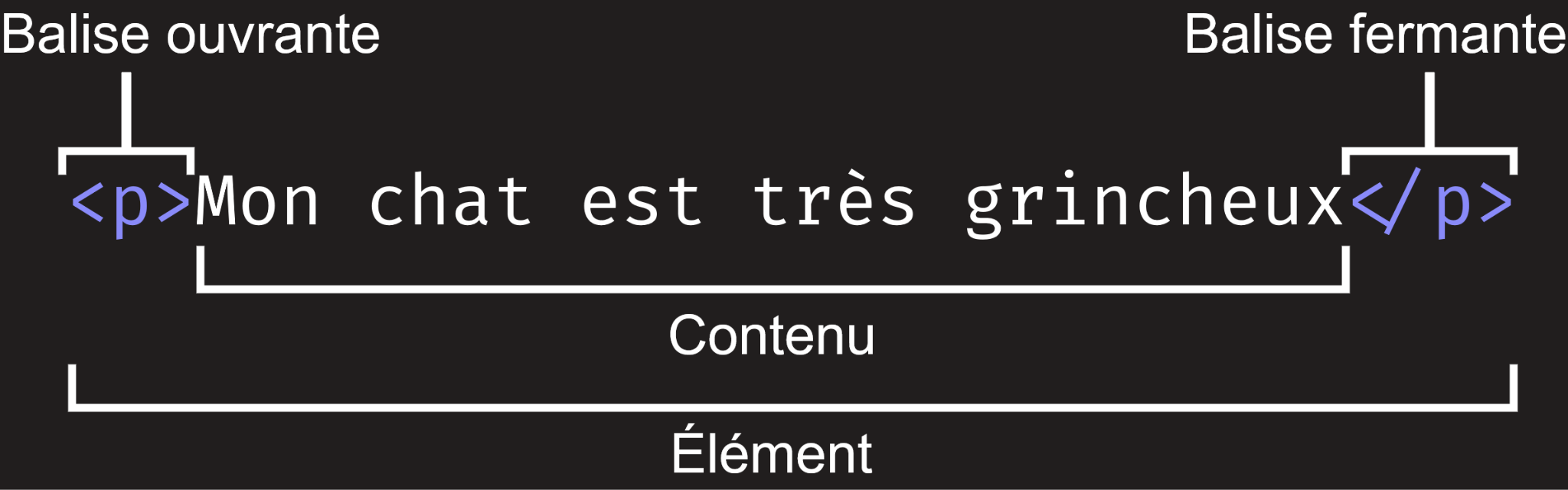
“ICT Culture”

***HTML***

*Le HTML, acronyme de « HyperText Markup Language » (langage de balisage hypertexte), sert à organiser le contenu et la structure des pages web. On peut l'utiliser pour diviser le contenu en paragraphes, créer des listes, insérer des images et des tableaux de données. HTML fonctionne en encadrant le contenu avec des* ***balises****, ce qui définit sa structure. Chaque document HTML est constitué d'une série* ***d'éléments*** *qui entourent différentes parties du contenu, permettant ainsi de les afficher ou de les formater de manière spécifique. En utilisant des* ***balises****, on peut par exemple transformer du texte en lien vers une autre page, définir des titres, etc.*



Les parties principales d'un élément sont :

- La balise ouvrante : composée du nom de l'élément entre chevrons (< et >), marquant le début de l'élément.

- La balise fermante : incluant une barre oblique avant le nom de l'élément, indiquant la fin de celui-ci.

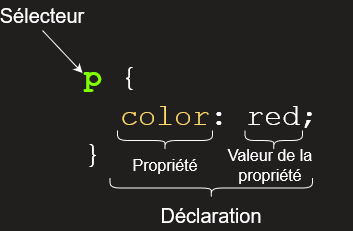
- Le contenu : ce qui se trouve à l'intérieur de l'élément, tel que du texte dans cet exemple.

- L'élément : la combinaison de la balise ouvrante, de la balise fermante et du contenu forme l'élément.

* + Link:
  + <https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/HTML_basics>
  + <https://chat.openai.com>

***CSS***

* + Les **CSS** (Cascading Style Sheets en anglais, soit « feuilles de style en cascade ») sont utilisées pour la mise en forme des pages web.
  + Tout comme HTML, CSS n'est pas vraiment un langage de programmation. C'est un langage de feuille de style qui permet d'appliquer des **styles** à différents éléments sélectionnés dans un document HTML. Par exemple, il est possible de sélectionner tous les paragraphes d'une page HTML et de définir leur couleur de texte en rouge à l'aide du code CSS suivant :



* + Link:
  + <https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/CSS_basics>
  + <https://chat.openai.com>

1. ***Javascript***

JavaScript est un langage de programmation de haut niveau, interprété et orienté objet, principalement utilisé pour créer des pages web interactives et dynamiques. Il est souvent intégré dans le code HTML des pages web pour ajouter des fonctionnalités telles que des animations, des effets visuels, des validations de formulaire, des calculs dynamiques et des interactions avec l'utilisateur. En plus de son utilisation côté client dans les navigateurs web, JavaScript peut également être utilisé côté serveur grâce à des environnements comme Node.js. Il est connu pour sa polyvalence, sa facilité d'apprentissage et sa compatibilité avec de nombreuses technologies web.

let myHeading = document.querySelector("h1");

myHeading.textContent = "Bonjour, monde !";

Le texte du titre a été changé en «Bonjour, monde ! » en utilisant JavaScript. Pour cela, on a utilisé une fonction appelée [querySelector()](https://developer.mozilla.org/fr/docs/Web/API/Document/querySelector) pour obtenir une référence sur l'en‑tête et la stocker dans une variable appelée myHeading. C'est assez proche de ce qu'on a fait avec les sélecteurs CSS. Lorsqu'on souhaite manipuler un élément, il faut d'abord le sélectionner.

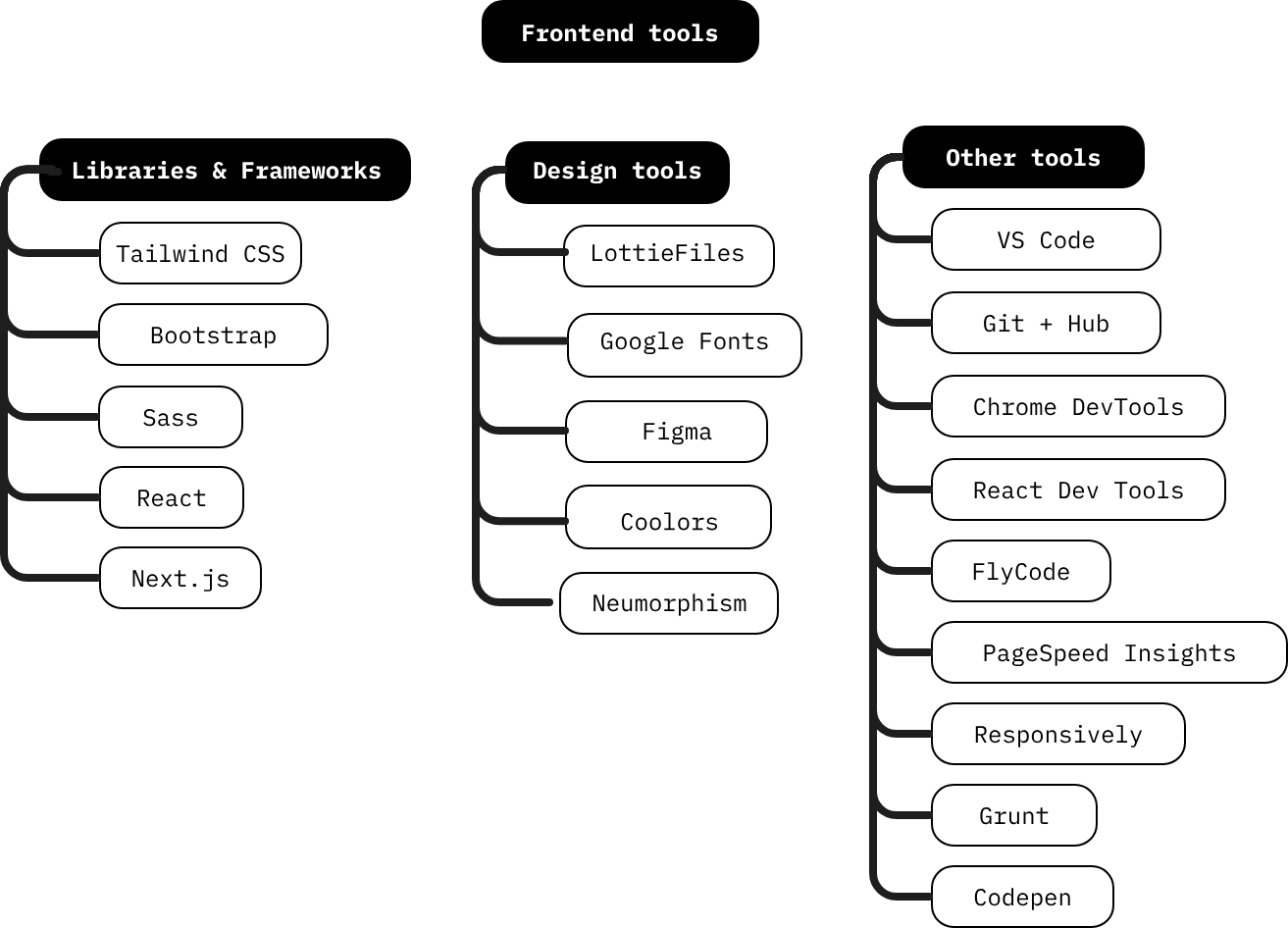
Document Object Model (DOM), qui vous permet de manipuler les documents.



* + Link: <https://developer.mozilla.org/fr/docs/Learn/Getting_started_with_the_web/JavaScript_basics>
  + <https://chat.openai.com>

***Front-end***

Le **front-end** désigne la partie d'une application ou d'un site web avec laquelle **l'utilisateur interagit** directement dans son navigateur web ou sur son appareil. Cela inclut l'interface utilisateur, les éléments visuels et **interactifs** tels que les menus, les boutons, les formulaires, les animations… Le front-end est généralement développé en utilisant des langages de programmation tels que HTML, CSS et JavaScript, qui permettent de structurer le contenu, de le styliser et de le rendre dynamique et interactif respectivement. Les développeurs front-end sont chargés de concevoir et de développer cette partie de l'application afin de fournir une expérience utilisateur intuitive et agréable.



* + Link:

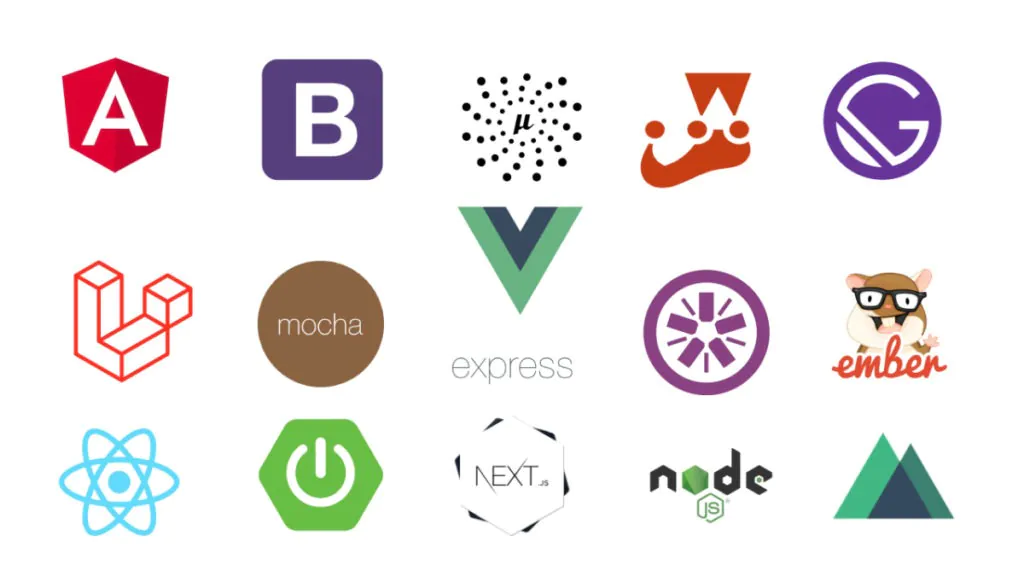
<https://www.wearedevelopers.com/magazine/best-tools-for-front-end-development>

[https://chat.openai.com](https://chat.openai.com/c/c7320c6b-051c-4d29-b4f7-1b675aa23c5f)

***Framework***

Un **framework** est un ensemble **structuré** d'outils, de bibliothèques, de conventions et de bonnes pratiques de développement logiciel, conçu pour **faciliter la création**, la maintenance et l'évolutivité des applications informatiques. Il fournit un ensemble de fonctionnalités **prêtes à l'emploi** et un cadre de travail cohérent pour les développeurs, leur permettant de se concentrer sur la logique métier spécifique de leur application plutôt que sur les détails techniques courants. Les frameworks peuvent être utilisés dans divers domaines, tels que le développement web, le développement d'applications mobiles, le développement de jeux, etc. Ils visent généralement à accélérer le processus de développement, à améliorer la qualité du code et à favoriser la réutilisation des composants.

Exemples de frameworks célèbres : AngularJS, ReactJS, Django, Ruby on Rails, Spring…



Les **inconvénients** des frameworks incluent une **complexité initiale élevée**, une **surcharge de fonctionnalités inutiles**, une **rigidité** potentielle dans la conception, une **dépendance à une technologie** spécifique, une possible **baisse de performance** et une **augmentation de la taille du code**.

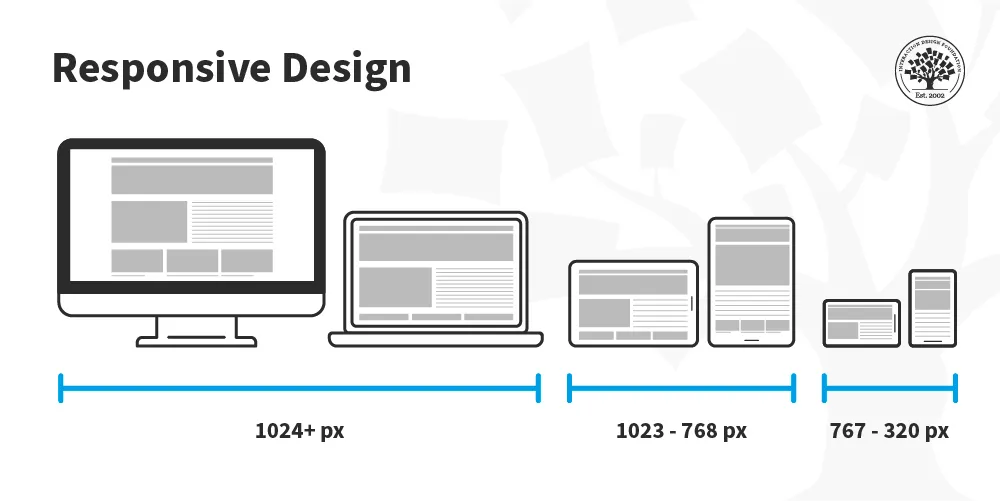
* + Link:

<https://developer.mozilla.org/en-US/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/Main_features>

[https://chat.openai.com](https://chat.openai.com/c/c7320c6b-051c-4d29-b4f7-1b675aa23c5f)

***Responsive***

Le **responsive** design est une approche de conception web qui vise à créer des sites internet qui s'adaptent automatiquement à différents appareils et **tailles d'écran**. Cela garantit une expérience utilisateur optimale, quelle que soit la plateforme utilisée, que ce soit un ordinateur de bureau, une tablette ou un smartphone. Le responsive design utilise des techniques de mise **en page flexibles,** des grilles proportionnelles et des médias queries pour ajuster dynamiquement le contenu et les éléments de navigation en fonction de la résolution de l'écran.



* + Link:
  + <https://public-images.interaction-design.org/literature/articles/materials/flTR2AEh0Xm8QMKevDdSR4x08XS9oDCs2FrMxpjj.jpg>
  + [https://chat.openai.com](https://chat.openai.com/c/c7320c6b-051c-4d29-b4f7-1b675aa23c5f)

***Full-stack***

Le terme "**full-stack**" désigne un développeur ou une équipe de développement capable de travailler à la fois sur la partie **front-end** (côté client) et sur la partie **back-end** (côté serveur) d'une application informatique. Un développeur full-stack est compétent dans plusieurs technologies de développement, tant pour la création de l'interface utilisateur que pour la gestion des données et la logique métier côté serveur. Ils sont capables de gérer l'ensemble du processus de développement d'une application, de la conception à la mise en production.

Avantages et Inconvénients :

1. Polyvalence : Capables de gérer tous les aspects d'une application.

2. Compréhension globale : Vue d'ensemble pour des décisions éclairées.

Connaissance moins approfondie : Risque de moins de spécialisation.

3. Efficacité : Réduction des dépendances et des temps d'attente.

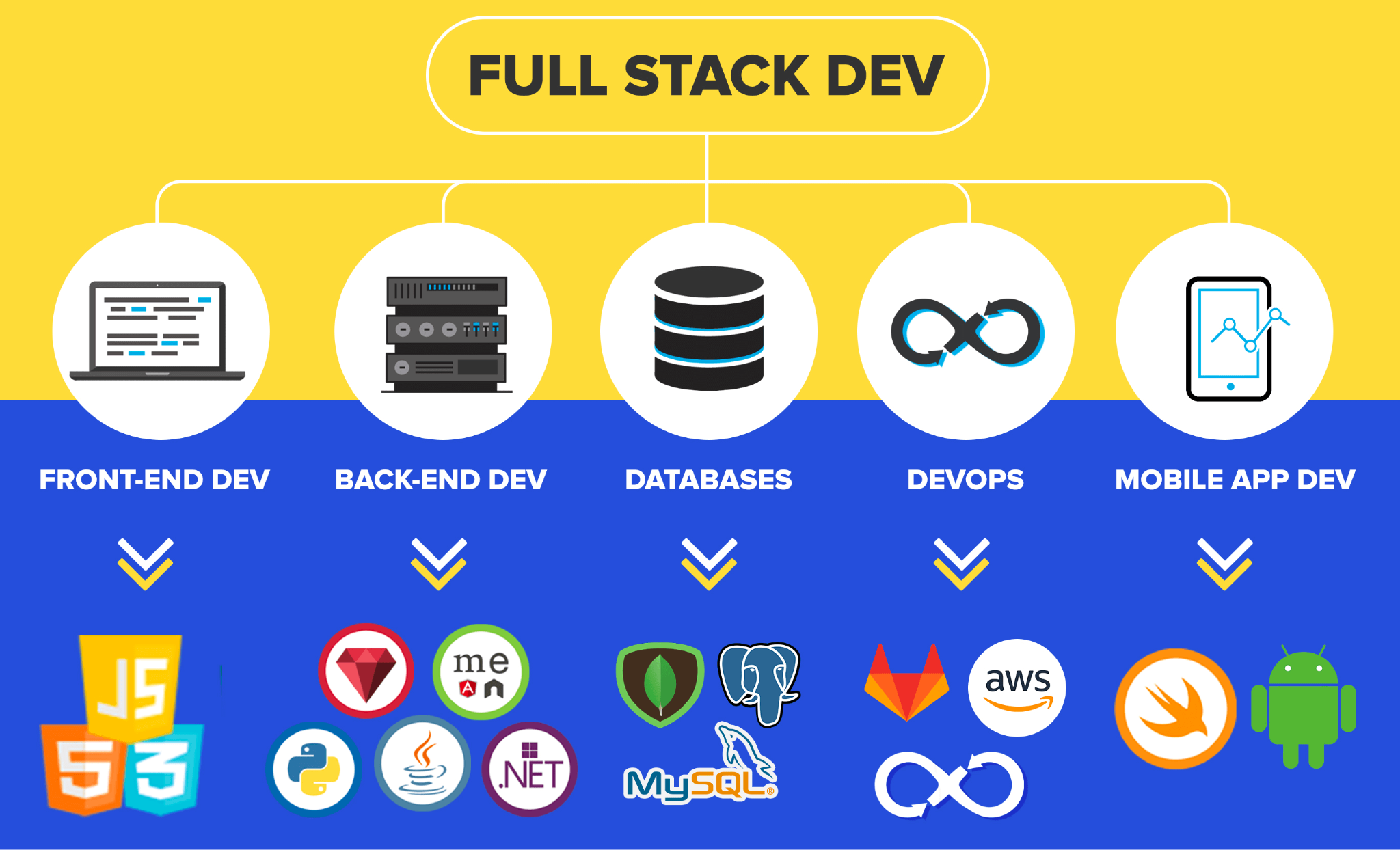
4. Résolution de problèmes : Capacité à identifier et à résoudre les problèmes à différentes couches.

1. Connaissance moins approfondie : Risque de moins de spécialisation.

2. Surcharge de travail : Gérer plusieurs tâches peut être intense.

3. Évolution rapide des technologies : Nécessite une mise à jour constante.

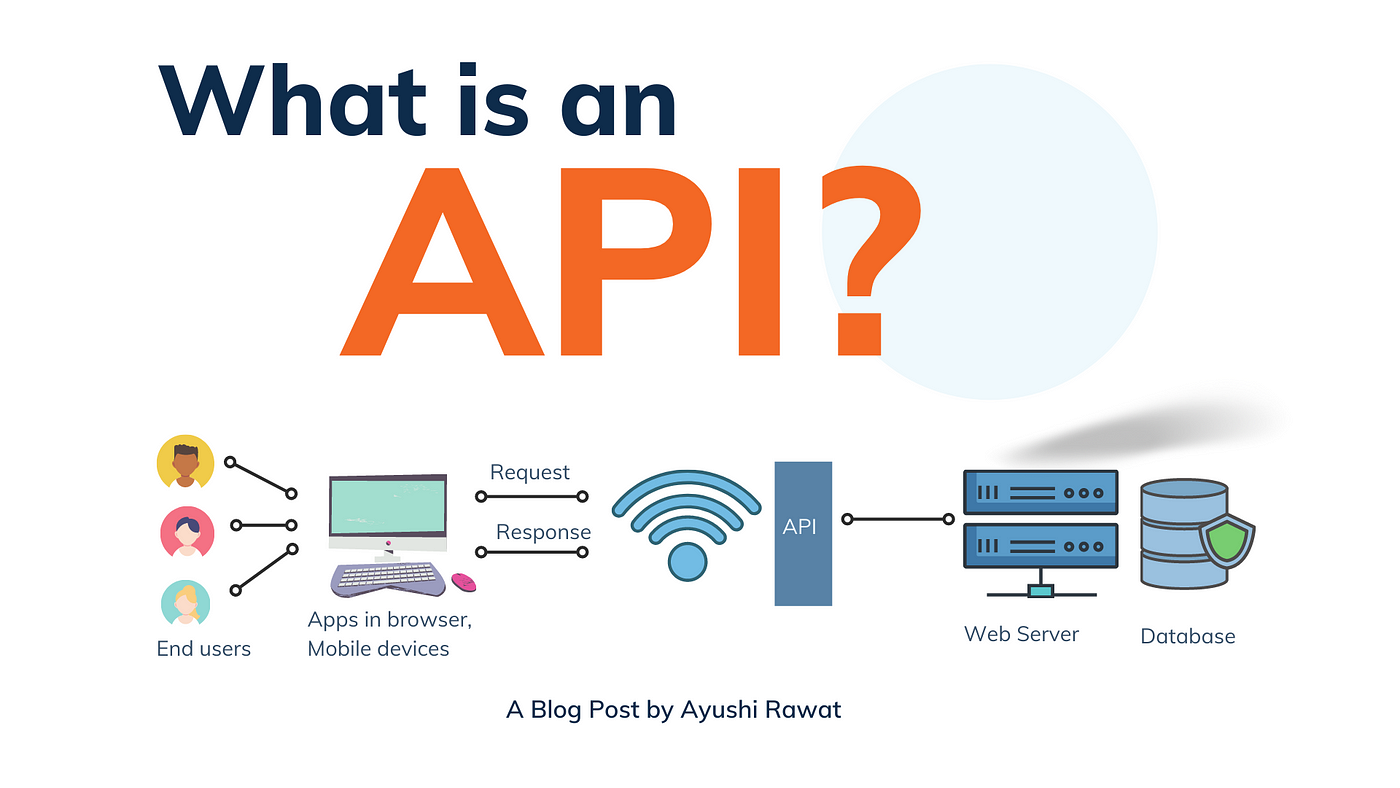
4. Difficulté à se spécialiser : Peut être difficile de se spécialiser dans un domaine spécifique.



* + Link:
  + <https://www.musteducation.tn/wp-content/uploads/2022/12/full-stack-devlopment-min.png>
  + [https://chat.openai.com](https://chat.openai.com/c/c7320c6b-051c-4d29-b4f7-1b675aa23c5f)

2. ***API***

**API** **permet à un logiciel ou à une application d'accéder aux fonctionnalités ou aux données d'un autre logiciel,** d'un service en ligne ou d'un système informatique de manière standardisée et sécurisée. Par exemple, une API peut être utilisée par un site web pour **récupérer des données** d'une base de données externe, envoyer des notifications à des appareils mobiles, ou intégrer des fonctionnalités d'un service tiers comme le partage sur les réseaux sociaux. En résumé, les API facilitent l'intégration et l'interopérabilité entre différents logiciels et systèmes, ce qui permet de créer des applications plus riches et plus fonctionnelles.



* + Link:

[https://miro.medium.com/v2/resize:fit:1400/1\*2GMSybCim32TTec9afVGxw.png](https://miro.medium.com/v2/resize:fit:1400/1*2GMSybCim32TTec9afVGxw.png)

[https://chat.openai.com](https://chat.openai.com/c/c7320c6b-051c-4d29-b4f7-1b675aa23c5f)

***Algorithm***

Un **algorithme** est une série d'instructions ou de règles permettant de **résoudre un problème ou d'accomplir une tâche spécifique**. Ces instructions décrivent une séquence logique d'opérations à exécuter dans un ordre précis pour atteindre un objectif donné. Les algorithmes sont largement utilisés en informatique pour la conception de programmes et d'applications, mais ils sont également utilisés dans de nombreux autres domaines, tels que les mathématiques, les sciences naturelles, l'intelligence artificielle, la cryptographie, etc. Un algorithme peut être exprimé de différentes manières, notamment sous forme de texte, de pseudo-code, de diagrammes ou de représentations mathématiques.

if (pokemonIa.hp >= vieInitialIA / 2) {

ctx.fillStyle = "green";

ctx.fillRect(130, 110, 130 \* (pokemonIa.hp / vieInitialIA), 15); //450,380

} else if (

pokemonIa.hp < vieInitialIA / 2 &&

pokemonIa.hp > vieInitialIA / 5

) {

ctx.fillStyle = "orange";

ctx.fillRect(130, 110, 130 \* (pokemonIa.hp / vieInitialIA), 15);

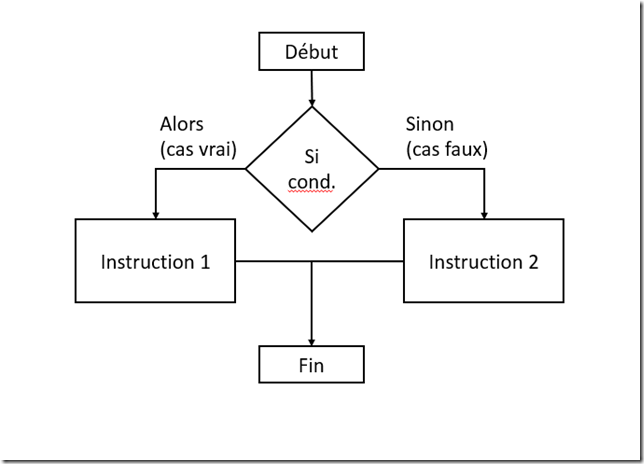
} else {

ctx.fillStyle = "red";

ctx.fillRect(130, 110, 130 \* (pokemonIa.hp / vieInitialIA), 15);

}

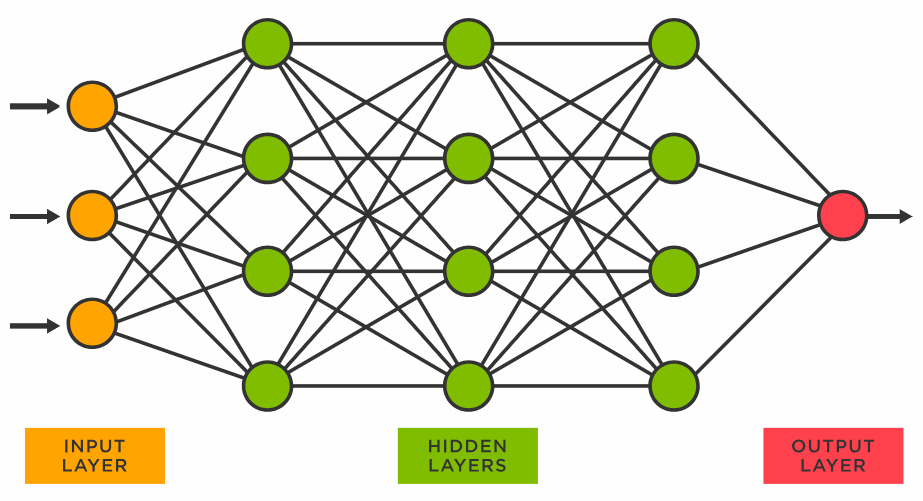
}

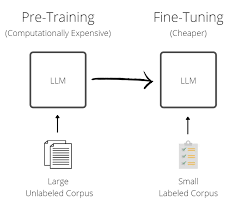


* + Link: [https://chat.openai.com](https://chat.openai.com/c/c7320c6b-051c-4d29-b4f7-1b675aa23c5f)
  + <https://pro-du-code.com/wp-content/uploads/2018/10/IfThenElse_1.png>

***Large Language Model***

Un **Large Language Model** (LLM) est un type de modèle de langage qui est **capable de générer du texte ou de comprendre le langage naturel** avec une grande complexité et une grande richesse. Ces modèles sont souvent construits à l'aide de réseaux de neurones profonds, **Un réseau de neurones est constitué d'un ensemble de nœuds interconnectés appelés neurones.** Chaque neurone reçoit des signaux d'entrée, les traite et produit une sortie. Ces signaux sont propagés à travers les différentes couches du réseau, où chaque couche peut effectuer des opérations de traitement sur les données.Ils sont particulièrement efficaces pour modéliser des données complexes et non linéaires, et sont souvent entraînés à l'aide d'algorithmes d'apprentissage **supervisé ou non supervisé** pour ajuster leurs poids et leurs paramètres afin de réaliser des tâches spécifiques. Les LLM sont généralement entraînés sur de vastes ensembles de données textuelles pour apprendre la structure et la sémantique du langage. Ils peuvent être utilisés pour diverses tâches liées au langage naturel, telles que la génération de texte, la traduction automatique, le résumé automatique, la réponse aux questions, la reconnaissance de l'intention et bien d'autres. Un exemple bien connu de LLM est GPT





* + Link [https://chat.openai.com](https://chat.openai.com/c/c7320c6b-051c-4d29-b4f7-1b675aa23c5f)
  + [https://miro.medium.com/v2/resize:fit:923/1\*X4qGpTGe3A6BsHPeUhn4PA.png](https://miro.medium.com/v2/resize:fit:923/1*X4qGpTGe3A6BsHPeUhn4PA.png)