

Cursus SALESFORCE

M2I - Formation - 2024

Donjon Audrey



Les fondamentaux du Web



Donjon Audrey
Formatrice Frontend

Contact :



www.linkedin.com/in/audrey-djn



[audrey_](#)



[audrey-donjon](#)

- ☐ Votre Prénom / Nom
- ☐ L'entreprise et le poste que vous occupez ou avez occupé dernièrement
- ☐ Votre parcours
- ☐ Vos attentes par rapport à la formation
- ☐ Êtes-vous familier avec le métier de développeur et le monde du numérique dans sa globalité ?
- ☐ Votre loisir préféré



Rappel du déroulé de la formation

- ☐ Début tous les jours à 9h00
- ☐ Pause le matin (~15min)
- ☐ Pause déjeuner de 12h30 à 13h30
- ☐ Pause l'après-midi (~15min)
- ☐ Fin de la journée de formation à 17h

Feuilles d'émargement à signer début matin et début après-midi

Objectifs :

- ☐ Comprendre les fondamentaux du web qui seront abordés durant ce module

Les fondamentaux du Web : sommaire général

1. Les métiers
2. Les concepts de base
3. Outils du développeur
4. Outils de développement
5. Utilisation des ressources disponibles

Les métiers autour du développement Web

Les métiers autour du développement Web / Logiciel

Cette formation vous permettra d'acquérir une base solide pour plusieurs parcours professionnels dans le développement web et la gestion de projets informatiques.

Développeur Web Front-End



Un développeur front-end crée la partie du site web que vous voyez et interagissez directement dans votre navigateur. Il utilise des langages comme HTML, CSS, et JavaScript, ainsi que des frameworks tels que Angular ou React, pour construire des interfaces utilisateur attrayantes et interactives.

Développeur Web Back-End



Le développeur back-end travaille sur le serveur, la base de données et les applications qui alimentent le site web. Il utilise des langages comme PHP, Java, ou Node.js, et s'occupe des bases de données SQL ou NoSQL, pour gérer la logique derrière les fonctionnalités d'un site.

Développeur Full Stack



Un développeur Full Stack combine les compétences du développement front-end et back-end. Il est capable de gérer à la fois la création d'interfaces utilisateur et la logique serveur, offrant une expertise complète en développement web.

Chef de projet Agile Scrum Master



Ce rôle implique la gestion de projets de développement logiciel en utilisant des méthodes agiles comme Scrum. Il se concentre sur la facilitation, la planification, et l'optimisation du processus de développement en collaboration étroite avec l'équipe de développement.

Les métiers autour du développement Web / Logiciel

Architecte Applicatif



L'architecte applicatif conçoit la structure technique d'applications et de systèmes informatiques. Il s'assure que l'architecture logicielle est robuste, évolutive et efficace.

Développeur Mobile



Spécialisé dans la création d'applications pour smartphones et tablettes. Utilise des langages et des plateformes spécifiques comme Java pour Android ou Swift pour iOS, ainsi que des technologies cross-platform comme Flutter ou React Native.

Spécialiste DevOps



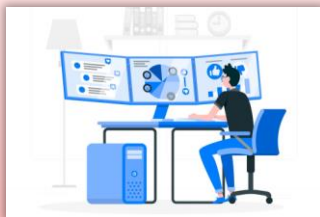
Le rôle DevOps implique de travailler à l'intersection du développement (Dev) et des opérations (Ops), se concentrant sur l'automatisation, l'intégration continue et l'amélioration de la collaboration entre les développeurs et les autres parties prenantes IT (Technologie de l'Information).

Ingénieur en Test Logiciel



Spécialisé dans le test des logiciels pour s'assurer qu'ils fonctionnent comme prévu. Cela inclut la réalisation de tests manuels et automatisés, la détection de bugs et la garantie de la qualité du logiciel.

Analyste de Données Spécialiste de Bases de Données



Un analyste de données travaille avec des données pour extraire des informations utiles, tandis qu'un spécialiste de bases de données gère et optimise les bases de données pour assurer leur efficacité et sécurité.

UX/UI Designer



Le UX (User Experience) Designer se concentre sur l'expérience globale de l'utilisateur avec une application, tandis que l'UI (User Interface) Designer se concentre sur la conception visuelle et l'interface de l'application.

Les métiers autour de la formation Java full stack

Cette formation Java full stack vous permettra de prétendre à une variété de métiers dans le domaine du développement web/logiciel comme :

- **Développeur Java junior** (*fondamentaux de Java/ concepts de la programmation orientée objet/ classes essentielles de Java SE / collections / BDD avec JDBC*)
- **Développeur Web Full Stack Junior** (*développement front-end + back-end*)
- **Développeur d'applications mobiles Android** (*après un apprentissage supplémentaire spécifique à Android*)
- **Développeur Front-End Junior** (*HTML, CSS, JavaScript, TypeScript, Angular*)
- **Développeur Back-End Junior** (*Java Jakarta EE, Spring, Spring Boot, Java Server Faces, JPA avec Hibernate, et création de services web REST.*)
- **Ingénieur DevOps Junior** (*concepts DevOps, utilisation de Docker, Jenkins, Kubernetes et Ansible*)
- **Développeur de bases de données Junior** (*SQL, JPA avec Hibernate*)
- **Développeur de tests logiciels Junior** (*test unitaire*)

Concepts de base en informatique

Concepts de Base en Informatique



- Composants d'un Ordinateur
- Systèmes d'Exploitation
- Bases de la Programmation
- Internet et Web
- Algorithme
- Bases de Données

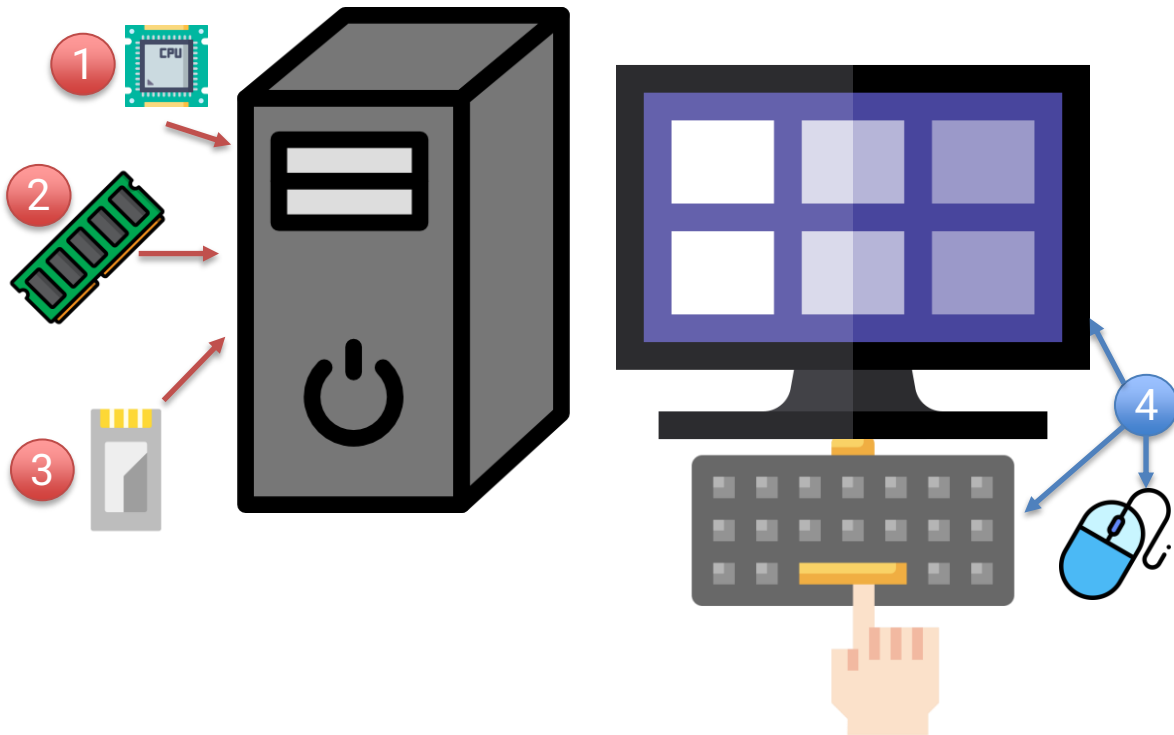
Composants d'un ordinateur

1 - **CPU (Unité central de traitement)** :
Cerveau de l'ordinateur, responsable du traitement des instructions.

2 - **Mémoire Vive (RAM)** :
Utilisée pour stocker temporairement les données pendant que le CPU les traite.

3 - **SSD (Stockage de données)** :
Pour le stockage à long terme des données.

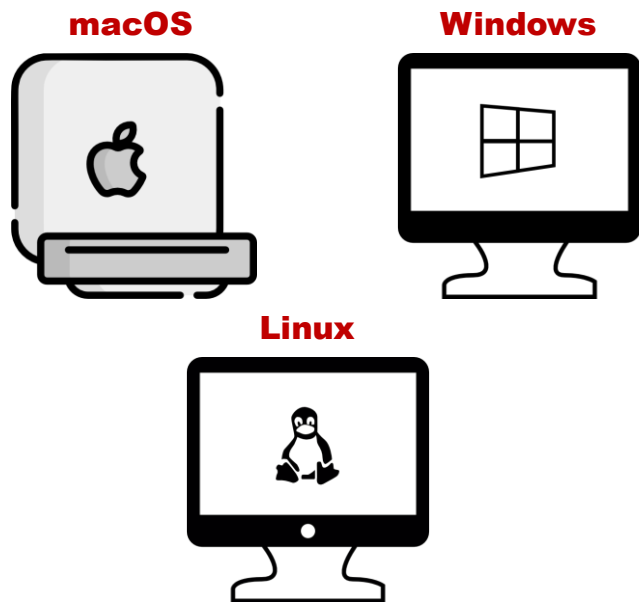
4 - **Périphériques d'Entrée/Sortie** :
Claviers, souris, écrans, imprimantes, etc.,
pour l'interaction avec l'utilisateur.



Système d'exploitation

Le **système d'exploitation** soit **os** : operating system sert d'interface entre l'utilisateur et le matériel de l'ordinateur, il permet de gérer les ressources de l'ordinateur tel que le CPU, la mémoire et les périphériques.

Exemple de système d'exploitation :



En résumé :

Windows est **populaire** et **largement utilisé**, avec un support étendu pour **divers outils de développement**. Des **IDE** comme **Visual Studio** offrent une **bonne expérience intégrée** pour le **développement web**.

MacOS offre une **bonne prise en charge** des **outils de développement web** et est souvent préféré pour son interface utilisateur **élégante** et sa **stabilité**, il s'agit d'un **écosystème fermé** et le **matériel Mac** est généralement **plus coûteux**.

Linux est très populaire parmi les développeurs pour sa **flexibilité**, sa **personnalisation** et **son accès ouvert** à une **large gamme d'outils de développement**. De nombreux **serveurs web fonctionnent sous linux**, ce qui rend le **test** et le **déploiement** des applications web **plus homogènes**, il peut tout de même nécessiter **plus de configurations initiales** et la **courbe d'apprentissage est plus abrupte** surtout pour ceux qui ne sont pas familiers avec les **interfaces de ligne de commande**.

Le **choix** de votre système d'exploitation dépendra surtout de vos **préférences personnelles**, de votre **familiarité avec le système d'exploitation** et des **besoins spécifiques de vos projets**. Ces trois systèmes sont capables de fournir un environnement de développement web robuste et efficace.

Bases de la programmation

Les **langages de programmation** sont des systèmes conçus pour **communiquer des instructions à une machine**. Pour pouvoir comprendre ces langages, il est important de se **familiariser avec leurs concepts de base**.

Synthèse des **différents concepts** qui seront abordés plus en profondeur pendant la formation :

❑ Variables et types de données :

Variables : Ce sont des espaces en mémoire où vous pouvez stocker des données. Elles peuvent être nommées et utilisées pour manipuler les données dans votre programme.

Types de Données : Les types de données définissent la nature des données que vous pouvez stocker dans une variable, tels que les nombres (entiers, flottants), les chaînes de caractères, ou les booléens (vrai/faux).

❑ Opérateurs :

Ce sont des **symboles** ou **mots-clés** qui indiquent au compilateur ou à l'interpréteur d'effectuer une **opération mathématique, logique, ou de comparaison**. Par exemple, + pour l'addition, > pour la comparaison, ou && pour un ET logique.

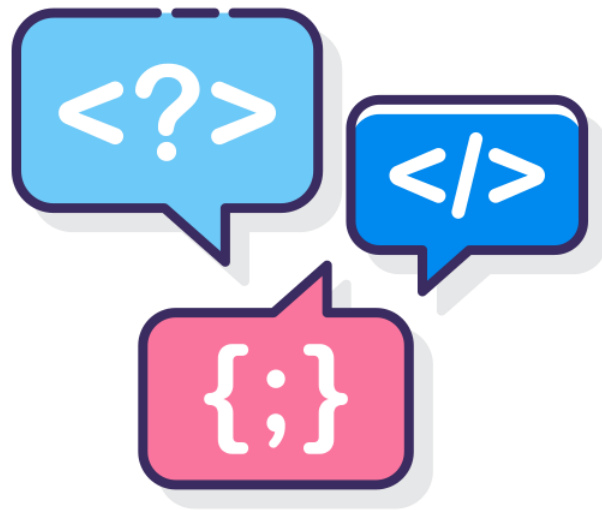
❑ Structures de contrôle :

Instructions Conditionnelles : if, else, et switch permettent d'exécuter des instructions en fonction de conditions spécifiques.

Boucles : for, while, et do-while sont utilisées pour répéter des blocs de code tant qu'une condition spécifiée est vraie.

❑ Fonctions :

Ce sont des **blocs de code qui effectuent une tâche spécifique**. Elles peuvent prendre des entrées (paramètres), exécuter des opérations, et retourner une sortie (valeur de retour).



Bases de la programmation

Synthèse des **différents concepts** qui seront abordés plus en profondeur pendant la formation :

❑ Structure de données

Les structures de données comme les tableaux, les listes etc. permettent de stocker et d'organiser les données de manière efficace.

❑ Programmation Orientée Objet (POO)

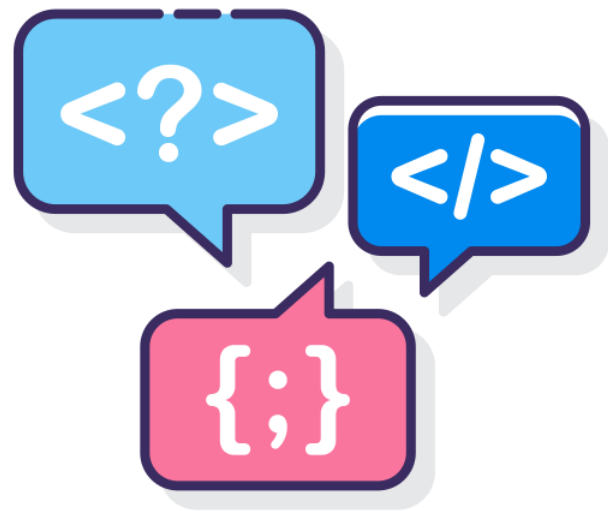
La POO est une approche de programmation basée sur des "objets", qui peuvent contenir des données (attributs) et des méthodes (fonctions). Elle utilise des concepts comme *l'encapsulation*, *l'héritage*, et le *polymorphisme*.

❑ Gestion des erreurs

Les langages de programmation offrent des mécanismes pour gérer les erreurs et les exceptions, permettant de traiter et de réagir aux problèmes qui surviennent pendant l'exécution.

❑ Bibliothèques et Frameworks

Il s'agit d'ensembles de code pré-écrit, sous forme de bibliothèques ou frameworks, sont disponibles pour simplifier le développement en offrant des fonctionnalités courantes sans avoir besoin de les réécrire.



Chaque **langage de programmation** a **sa propre syntaxe** et **ses particularités**, mais **ces concepts de base sont largement communs** et constituent **le fondement de la programmation**.

Bases de la programmation

L'un des concepts clé important également prévu : les API

❑ Qu'est-ce qu'une API ?

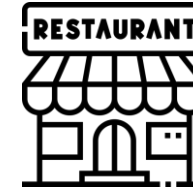
API signifie **Application Programming Interface**. C'est un **ensemble de règles** et de **mécanismes** qui permettent à différentes applications de **communiquer** entre elles. En d'autres termes, une **API** est comme un messenger qui **prend des requêtes, dit au système** ce que vous voulez faire, et vous **renvoie la réponse**.

❑ En quoi les API sont utiles ?

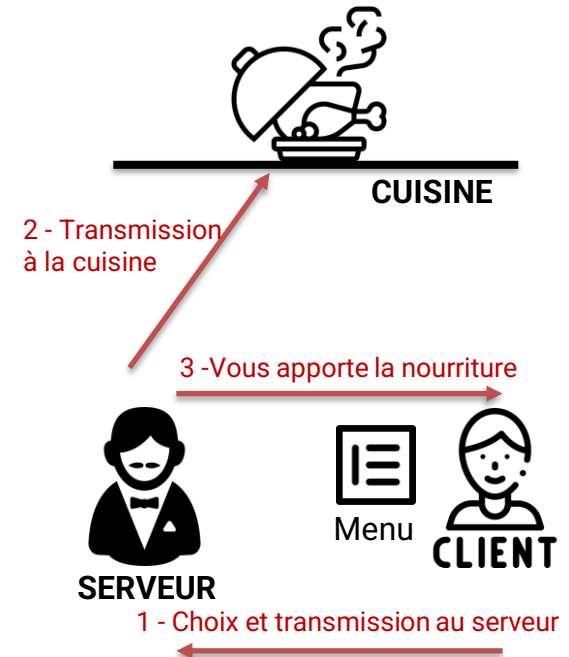
- ❑ **1 - Interconnexion** : Les API permettent aux différentes parties d'un logiciel, ou à différents logiciels, de se connecter et de travailler ensemble. Par exemple, une application de météo peut utiliser une API pour obtenir les prévisions météorologiques d'un service tiers
- ❑ **2 - Réutilisation** : Avec les API, vous n'avez pas à recréer la roue. Par exemple, si vous avez besoin d'un système de paiement, vous pouvez utiliser l'API d'un service comme PayPal au lieu de créer votre propre système de paiement.
- ❑ **3 - Évolutivité** : Les API permettent aux développeurs de créer des applications modulaires. Si un service change, vous n'avez besoin de mettre à jour que la partie qui communique avec ce service, pas toute votre application.

❑ Types d'API

- ❑ **1 - API Web** : Ce sont les plus courantes. Elles permettent aux applications de communiquer sur Internet en utilisant des protocoles comme HTTP. Par exemple, quand vous utilisez une application qui montre des cartes, elle utilise probablement une API comme Google Maps API.
- ❑ **2 - API de Système** : Utilisées pour permettre aux différentes parties d'un système d'exploitation ou d'un logiciel de communiquer entre elles.
- ❑ **3 - API de Bibliothèque ou de Framework** : Fournissent des fonctions et des méthodes que vous pouvez utiliser dans votre propre code. Par exemple, une bibliothèque graphique pour dessiner des éléments visuels dans votre application.



- **Nous** sommes le **client**
- **Le serveur** est l'**API**
- **La cuisine** est le **système** ou **service** auquel **on fait une requête**
- **Le menu** est la **documentation de l'API** qui nous montre ce qu'on peut demander



Compréhension du fonctionnement d'internet

Compréhension du Fonctionnement d'Internet

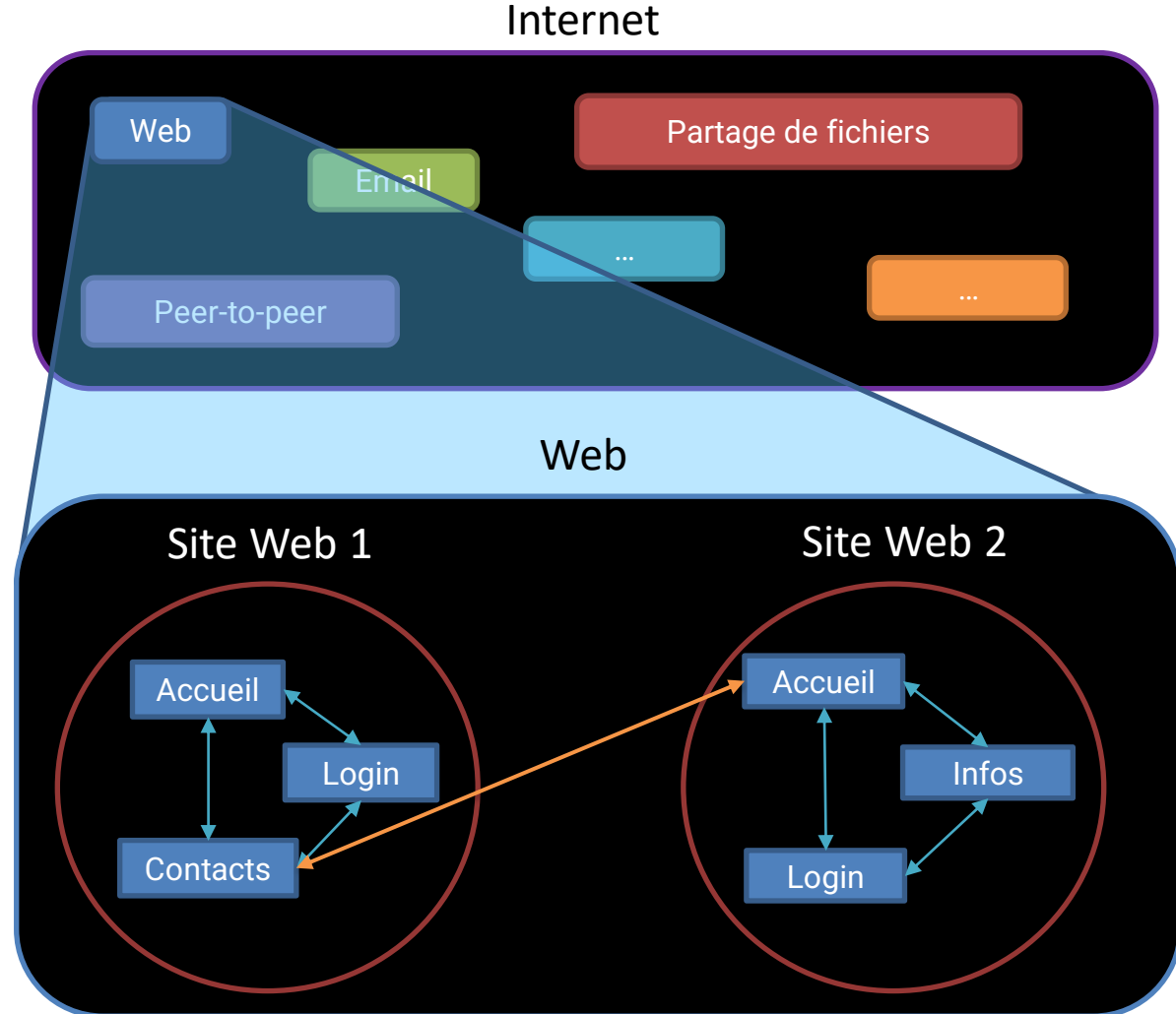
Internet ou web ?

Internet = Réseau mondial regroupant plusieurs systèmes de communication dont le **Web**.

Web = Réseau regroupant les **sites Web**. Il fait partie d'**Internet**. Aussi appelé World Wide Web.

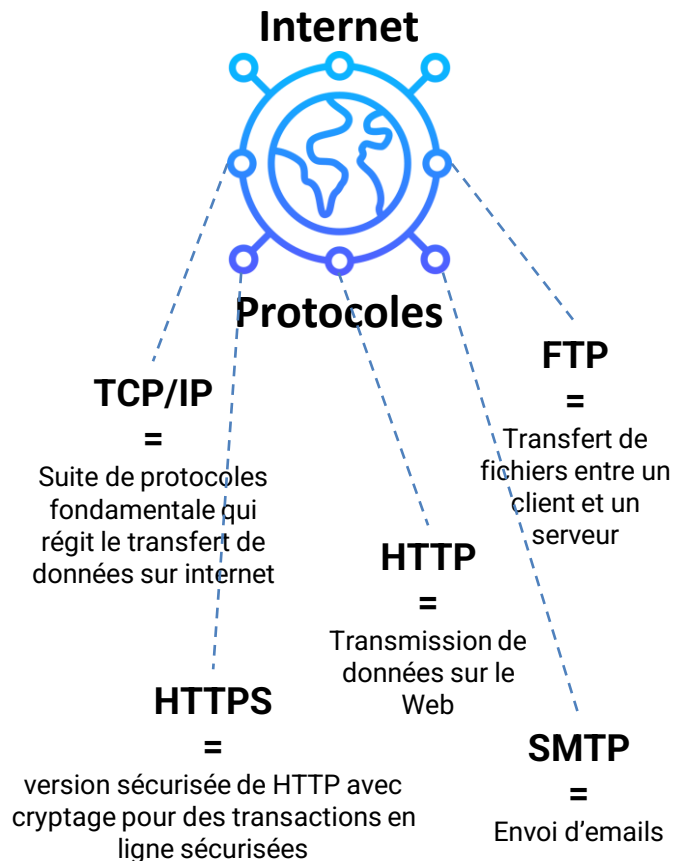
Site Web = Ensemble de pages reliées entre elles via des **hyperliens**, accessible sur le **Web**.

Hyperlien = Lien entre deux pages d'un même **site Web** (**lien interne**) ou entre deux pages de deux **sites Web** (**lien externe**).

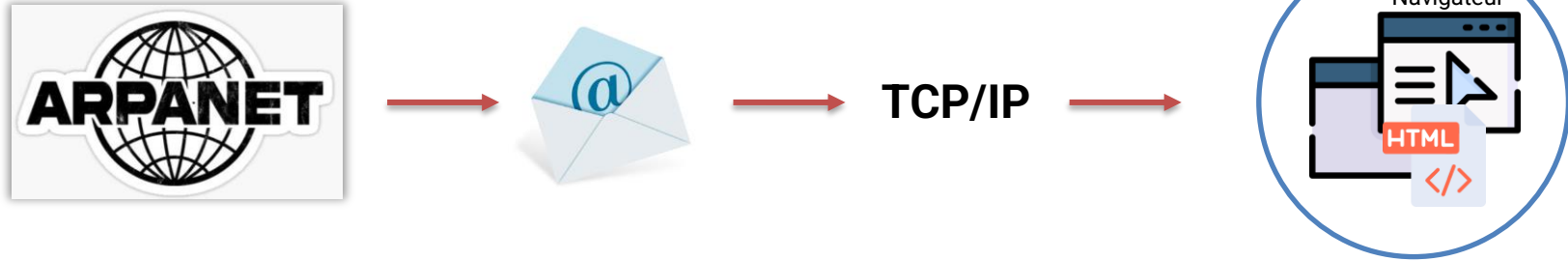


Compréhension du Fonctionnement d'Internet

- ❑ **Internet en tant que réseau global** : Internet est un réseau mondial de réseaux interconnectés qui utilise le protocole IP (Internet Protocol) pour la communication.
- ❑ **TCP/IP (Transmission Control Protocol/Internet Protocol)** : Le TCP/IP est la suite de protocoles fondamentale qui régit le transfert de données sur Internet. TCP gère la communication fiable entre les dispositifs, tandis qu'IP s'occupe de l'acheminement des paquets de données.
- ❑ **Protocole HTTP (Hypertext Transfer Protocol)** : HTTP est le protocole utilisé pour la transmission de données sur le Web, notamment pour charger des pages web dans un navigateur.
- ❑ **Protocole FTP (File Transfer Protocol)** : FTP est utilisé pour le transfert de fichiers entre un client et un serveur sur un réseau informatique.
- ❑ **Adresses IP et DNS (Domain Name System)** : Les adresses IP sont des identifiants uniques pour les dispositifs sur Internet. Le DNS traduit les noms de domaine faciles à retenir (comme `www.example.com`) en adresses IP numériques.
- ❑ **Autres protocoles importants** :
 - **SMTP (Simple Mail Transfer Protocol)** : Utilisé pour l'envoi d'emails.
 - **HTTPS (HTTP Secure)** : Une version sécurisée de HTTP avec cryptage pour des transactions en ligne sécurisées.
 - **Protocoles de Sécurité** : Comme SSL/TLS, utilisés pour sécuriser la communication sur Internet.
- ❑ **Infrastructure d'Internet** : Comprend des éléments comme les serveurs web, les réseaux de distribution de contenu (CDN), et les data centers.



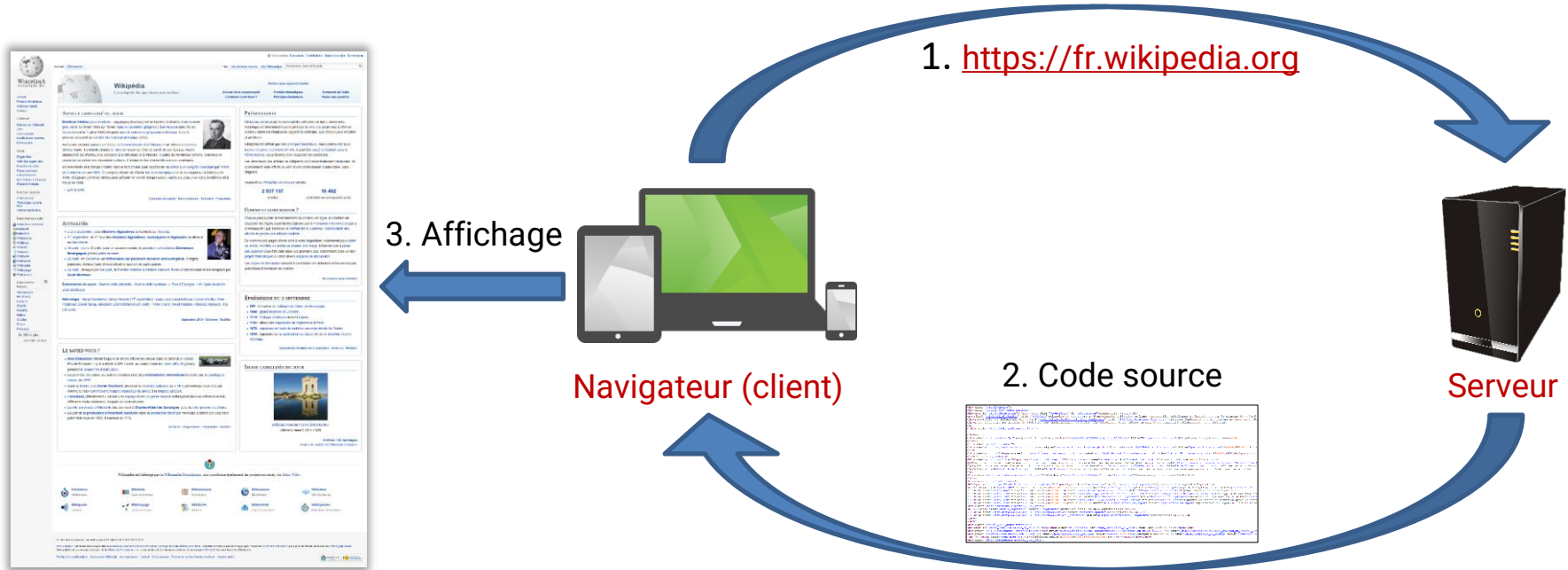
Un peu d'histoire...



- ❑ **1969** : L'ancêtre d'Internet, **Arpanet**, est créé par l'armée américaine pendant la guerre froide. C'est le premier **réseau décentralisé** en forme de « toile », pensé pour éviter qu'une attaque nucléaire ciblée puisse le détruire.
- ❑ **1972** : Création du système des **courriers électroniques**: la première application importante d'Arpanet.
- ❑ **1983** : **Arpanet** adopte la suite **TCP/IP** et devient **Internet**.
- ❑ **1990** : Création du **Web**, du premier **navigateur Web** et du langage **HTML** par **Tim Berners-Lee**, aussi appelé le « père du web ».

Processus de requête Client-Serveur

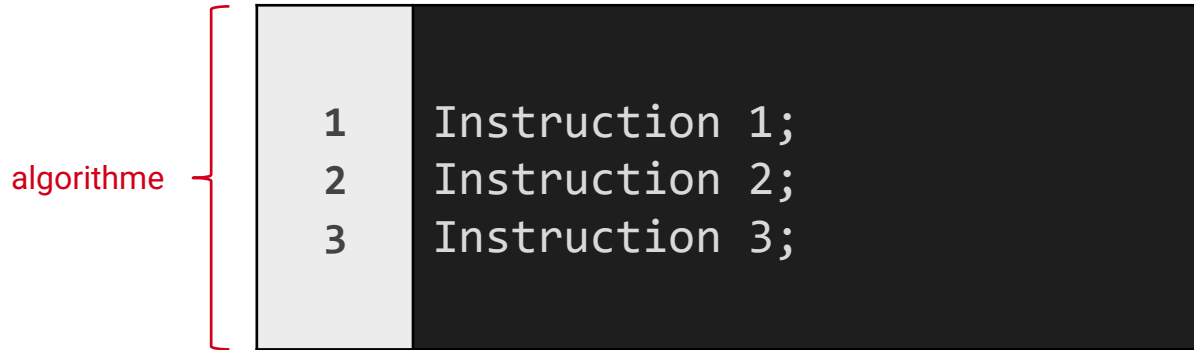
Processus de requête Client-Serveur



- ☐ Le **client** demande une page web via son URL (requête).
- ☐ Le **serveur** retourne au **client** le **code source** de la page demandée (réponse de requête).
- ☐ Le **navigateur** du client réceptionne le **code source** (HTML + CSS) et le **traduit visuellement à l'écran**.

Algorithmes

Comme vu précédemment un **langage de programmation** est un système conçu pour **communiquer des instructions à une machine**. Cet **ensemble d'instructions** est ce qu'on appelle un **algorithme**. Une **instruction** est un "**ordre**" donné via un langage de programmation pour **effectuer une action**. (exemple : Supprimer le carré rouge à l'écran).



Systèmes de Gestion de Base de données

Systèmes de Gestion de Base de Données

Un **SGBD** est un logiciel qui permet de **stocker**, **recupérer**, **mettre à jour** et **gérer** les **données dans une base de données**. Il fournit les moyens pour **manipuler** et **analyser** les **données** de manière structurée.

Types de SGBD :

- ❑ **SGBD Relationnels** : Comme MySQL, PostgreSQL, Oracle, et SQL Server. Ces systèmes utilisent le **langage SQL** (Structured Query Language) pour **interroger** et **manipuler** les **données stockées** dans des **tables relationnelles**.
- ❑ **SGBD NoSQL** : Comme MongoDB, Cassandra, et CouchDB. Ils sont conçus pour **gérer de grandes quantités de données non structurées** ou **semi-structurées**, et sont souvent utilisés pour les **applications nécessitant une grande évolutivité** et une **flexibilité** des **schémas de données**.

Stockage, récupération, mise à jour et gestion des données :

- ❑ Les **données sont stockées** dans des **formats structurés**, tels que des **tables dans les SGBD relationnels**, ou des **documents/objets dans les SGBD NoSQL**. Les **schémas** de base de données définissent la **structure**, les **types de données** et les **relations entre les différentes parties des données**.
- ❑ Les **SGBD** permettent d'extraire des données à l'aide de **requêtes**. Dans les SGBD **relationnels**, les **requêtes SQL** sont utilisées pour sélectionner, filtrer et trier les données.
- ❑ Les **SGBD** fournissent des **mécanismes** pour **mettre à jour les données existantes**, **ajouter** de nouvelles données et **supprimer** les données obsolètes.
- ❑ Les **SGBD** incluent des **fonctionnalités** pour la **gestion des transactions**, garantissant l'**intégrité** et la **cohérence** des **données**, en particulier dans les environnements multi-utilisateurs. Ils offrent également des outils pour la **sauvegarde**, la **récupération** et la **sécurité des données**.
- ❑ Les **SGBD** sont équipés d'outils pour **optimiser les requêtes** et gérer efficacement les **performances**, surtout lors de la manipulation de grandes quantités de données.
- ❑ Les **SGBD** modernes permettent l'**accès à distance** et facilitent la **collaboration** entre les utilisateurs et les applications sur différents sites ou plateformes.

Exercice 01 : Retrouve Le bon métier

Crée toi un compte sur Figma via ce lien :

<https://www.figma.com/fr-fr/>

Une fois le compte créé tu pourras avoir accès au lien que le formateur te transmettra pour réaliser l'exercice



Outils du développeur

Navigateurs Web



Mozilla Firefox



Google Chrome



Microsoft Edge



Internet Explorer



Safari



Opera



L'éditeur de texte pour écrire le code source.



Le navigateur pour visualiser.

Éditeurs de texte



Notepad++



Sublime Text



Visual Studio
Code



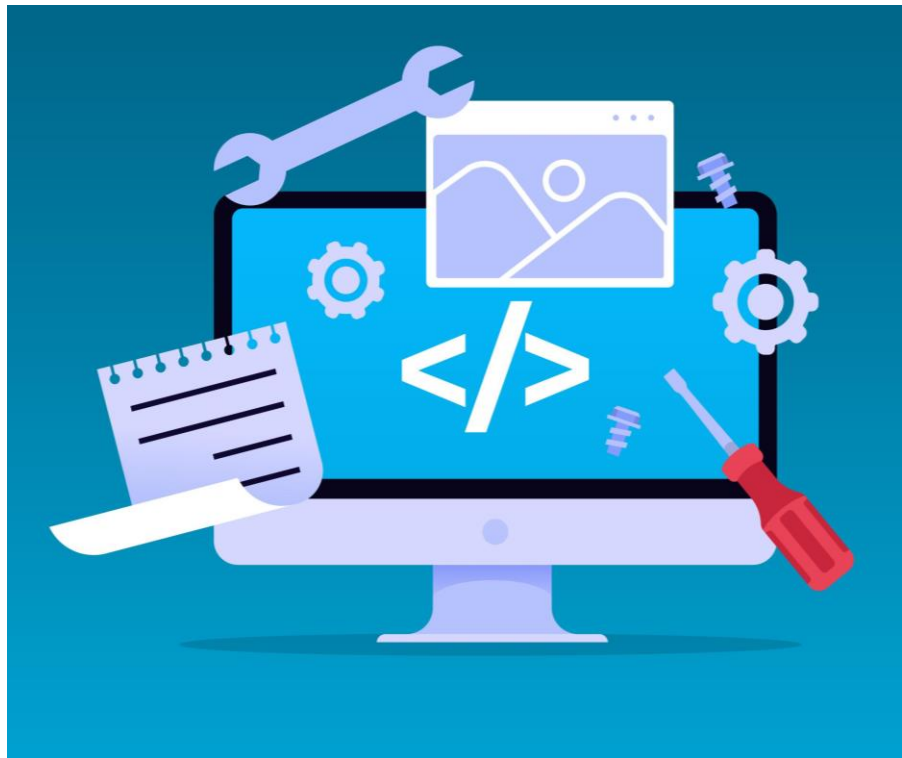
Brackets



Atom

Outils de développement

Outils de Développement



- Environnements de Développement Intégrés (IDE)
- Outils de contrôle de version
- Outils de développement Front-End
- Outils de développement Back-End
- Outils de test logiciel
- Outils de déploiement et de monitoring
- Outils de collaboration et de gestion de projet
- Outils d'Interface Graphique (GUI)

Découverte des outils de développement

De nombreux outils existent pour faciliter et optimiser le processus de développement de sites Web ou logiciels.

Environnements de Développement Intégrés (IDE) :

- Les IDE, comme **Visual Studio**, **Eclipse**, ou **IntelliJ IDEA**, offrent une plateforme unifiée pour la **codification**, le **débogage**, les **tests** et la **visualisation**.
- Ils intègrent généralement des **éditeurs de code**, des **outils de débogage**, et des **compilateurs**, et peuvent inclure des fonctionnalités pour la gestion de versions et la construction automatisée.

Outils de contrôle de version :

- Les systèmes de contrôle de version, tels que **Git**, permettent aux développeurs de **suivre et de gérer les changements** dans le code source.
- Des plateformes comme **GitHub**, **GitLab** ou **Bitbucket** facilitent la **collaboration**, la **révision de code** et la **gestion de projets logiciels**.

Outils de développement Front-End :

- Des **frameworks** et **bibliothèques** comme **React**, **Angular**, ou **Vue.js** aident à créer des **interfaces** utilisateur **interactives** et **efficaces**.
- Les **préprocesseurs CSS** (comme **Sass** ou **LESS**) et les **outils de construction** (comme **Webpack** ou **Gulp**) **automatisent** et **optimisent** le **développement front-end**.

Outils de développement Back-End :

- Des **frameworks** pour les langages de programmation backend, tels que **Node.js** pour **JavaScript**, **Django** pour **Python**, ou **Spring** pour **Java**, offrent des **structures** pour développer des **applications serveur efficaces**.
- Les systèmes de **gestion de bases de données** (comme **MySQL**, **PostgreSQL**, **MongoDB**) sont essentiels pour le **stockage** et la **gestion des données**.

Découverte des outils de développement

Outils de test logiciel :

- Des outils comme **JUnit** (pour Java), **PyTest** (pour Python), ou **Mocha** (pour JavaScript) sont utilisés pour **écrire et exécuter des tests automatisés**.
- Des outils d'**intégration continue** et de **livraison continue** (CI/CD), tels que **Jenkins**, **Travis CI** ou **CircleCI**, **automatisent le processus de test et de déploiement**.

Outils de déploiement et de monitoring :

- Des **plateformes cloud** comme **AWS**, **Azure** ou **Google Cloud** fournissent des **services pour héberger et déployer des applications**.
- Des **outils de monitoring** et de **performance**, tels que **New Relic** ou **Datadog**, aident à **surveiller les applications en production**.

Outils de collaboration et de gestion de projet :

- Des outils comme **Slack**, **Microsoft Teams**, ou **Trello** facilitent la communication et la collaboration au sein des équipes de développement.

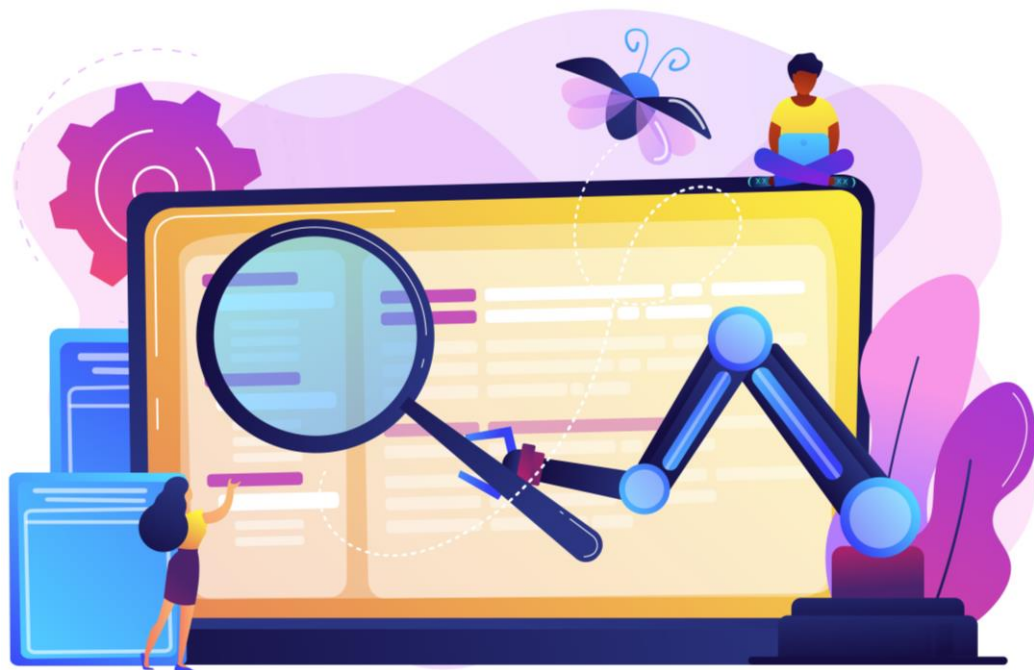
Outils d'Interface Graphique (GUI) :

- Des outils de **conception graphique** comme **Figma**, **Adobe XD** ou **Sketch** sont utilisés pour la création d'interfaces utilisateur et la maquette d'applications.

Les **outils de développement** jouent un rôle crucial dans **toutes les phases du cycle de vie** du développement logiciel, de la planification à la mise en production. Ils aident à améliorer l'efficacité, la qualité et la maintenance des logiciels.

Utilisation des ressources disponibles

Utilisation des Ressources en Ligne



- **Documentation** et tutoriels
- **Forums** et communautés en ligne
- **Références** et guides
- **Blogs** et articles
- **Ressources** Open Source
- **Outils** de collaboration en ligne
- **Podcasts** et Webinaires
- **Exercices** et challenges de codage

Bien exploiter les ressources disponibles

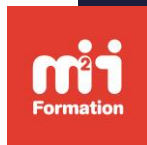
Bien exploiter les ressources disponibles pour apprendre, résoudre des problèmes, rester informé et collaborer.

Documentation et tutoriels	Sites web : MDN Web Docs , documentation officielle de langages et de Frameworks	Tuto en ligne : Codecademy, Coursera, Udemy ou freeCodeCamp
Forums et communautés en ligne	Forums : Stack Overflow, Reddit (r/programming, r/webdev)	Communautés en ligne : « Front-end Developer Network » sur LinkedIn ou « Javascript Developers » sur Facebook
Références et guides	W3Schools, GeeksforGeeks, ou Tutorialspoint	
Blogs et articles	TechCrunch, Ars Technica, ou Smashing Magazine	
Ressources Open Source	Plateformes : GitHub ou GitLab	Projets open source : sur Github (FreeCodeCamp , Bootstrap, Vue.js)
Outils de collaboration en ligne	Outils : GitHub, Bitbucket ou GitLab	Plateformes de communication : Slack ou Discord
Podcasts et Webinaires	Microsoft Events Catalog, Google Cloud Webinars	
Exercices et challenges de codage	LeetCode, HackerRank, Codewars, codingame, openclassrooms	

Exercice 02 : QCM

Lien vers QCM donné par le formateur





Fin du module



m2information.fr