

Fondamentaux du Web

Achref El Mouelhi

Docteur de l'université d'Aix-Marseille
Chercheur en programmation par contrainte (IA)
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`

- 1 Introduction
- 2 Internet
- 3 TCP
- 4 IP
 - Protocole IP
 - Adresse IP
 - Adresse MAC
 - IPv4
 - IPv6

Plan

- 5 HTTP
- 6 Navigateur
- 7 URL, URN et URI
 - URL
 - URN
 - URI
- 8 Site web vs Application web
- 9 Langages
 - Paradigme
 - Compilation et interprétation
 - Analyse du code et linkage
 - Classification par usage
 - Comparaison
 - Frontend et Backend
 - Critères de choix d'une stack web

Plan

- 10 Base de données
- 11 Serveur
- 12 Framework
- 13 Métiers dans le domaine du développement
- 14 Cloud
- 15 Logiciels de gestion
- 16 Architectures des applications Web
- 17 IDE

Fondamentaux du Web

Développement Web

```
graph TD; A([Développement Web]) --> B[Développement]; A --> C[Web];
```

Développement

- De manière générale : processus de croissance, d'évolution ou de création progressive de quelque chose.
- Dans le contexte informatique : concevoir, créer, tester et maintenir des logiciels ou des applications.

Web

- Un service **internet**
- Appelé officiellement **World Wide Web** (www)
- Ensemble de pages (ou d'informations) accessibles via **Internet**

Fondamentaux du Web

Développement Web

- est le processus de création de sites web ou d'applications accessibles via **Internet**.
- englobe diverses activités, notamment :
 - La conception de l'architecture et de l'interface utilisateur.
 - La programmation des fonctionnalités (front-end et back-end).
 - La gestion des bases de données.
 - Le déploiement et la maintenance des sites et applications.
- utilise des **langages de programmation** et des technologies web pour construire et maintenir de simples pages web ou d'applications complexes.

Fondamentaux du Web

Remarque

Le développement ne se limite pas uniquement au web.

Fondamentaux du Web

Exemples d'autres types d'applications que l'on peut développer

- Logiciels de bureau
 - conçus pour être exécutés sur des ordinateurs de bureau ou portables.
 - langages de programmation couramment utilisés : **C++**, **Java** et **C#**.
- Applications mobiles
 - conçues pour fonctionner sur des appareils mobiles tels que les smartphones et les tablettes.
 - utilise des langages de programmation spécifiques tels que **Swift** (pour **iOS**) et **Kotlin** (pour **Android**).
- Jeux vidéo
 - implique la création de jeux pour diverses plateformes : ordinateurs, consoles et appareils mobiles.
 - langages souvent utilisés : **C++** et **C#**.
- Logiciels embarqués
 - conçus pour contrôler des appareils électroniques spécifiques, tels que les systèmes de contrôle automobile, les appareils médicaux et les appareils électroménagers.
 - nécessite souvent une connaissance approfondie des langages de bas niveau tels que **C** et **C++**.

Fondamentaux du Web

Internet

- Réseau mondial d'ordinateurs inter-connectés
- Utilisant le protocole **TCP/IP** pour la communication
- Fonctionnant grâce à un ensemble complexe de routeurs, de serveurs et de câbles (fibre optique, câbles sous-marins...)
- Permettant l'échange de données (textes, images, vidéos...)
- Chaque ordinateur de ce réseau a une adresse **IP**.
Par exemple : 192.168.22.3
- Certains ordinateurs ou serveurs sont identifiés par un nom de domaine, plus facile à retenir.
Par exemple : `www.google.fr`

Fondamentaux du Web

DNS : Domain Name System

- un système qui traduit les noms de domaine (par exemple, `google.com`) en adresses **IP** (par exemple, `172.217.168.238`) : mapping entre adresses **IP** en noms de domaine
- agit comme un annuaire téléphonique pour l'Internet, facilitant l'accès aux sites web et autres ressources en ligne.

© Achref EL MOULI

Fondamentaux du Web

DNS : Domain Name System

- un système qui traduit les noms de domaine (par exemple, `google.com`) en adresses **IP** (par exemple, `172.217.168.238`) : mapping entre adresses **IP** en noms de domaine
- agit comme un annuaire téléphonique pour l'Internet, facilitant l'accès aux sites web et autres ressources en ligne.

DNS : fonctionnement

- En saisissant une adresse web dans le navigateur, l'ordinateur envoie une requête à un serveur **DNS**.
- Le serveur **DNS** recherche l'adresse **IP** correspondante au nom de domaine demandé, puis renvoie cette information à l'ordinateur.
- Le navigateur peut alors utiliser l'adresse **IP** pour se connecter au serveur web où se trouve le site demandé.

Fondamentaux du Web

Dans **Internet**, on peut utiliser plusieurs protocoles

- **HTTP** : **H**yper**T**ext **T**ransfer **P**rotocol et son homologue sécurisé **HTTPS** (**H**yper**T**ext **T**ransfer **P**rotocol **S**ecure)
- **FTP** : **F**ile **T**ransfer **P**rotocol
- **SMTP** : **S**imple **M**ail **T**ransfer **P**rotocol
- ...

Fondamentaux du Web

Dans **Internet**, on peut utiliser plusieurs protocoles

- **HTTP** : **H**yper**T**ext **T**ransfer **P**rotocol et son homologue sécurisé **HTTPS** (**H**yper**T**ext **T**ransfer **P**rotocol **S**ecure)
- **FTP** : **F**ile **T**ransfer **P**rotocol
- **SMTP** : **S**imple **M**ail **T**ransfer **P**rotocol
- ...

Web

Un des services utilisés sur **Internet** utilisant le protocole **HTTP**

Fondamentaux du Web

TCP : Transmission Control Protocol

- chargé de transporter les données entre deux ordinateurs connectés à Internet.
- garantit que les données sont livrées à destination sans erreur, perte ou duplication.
- utilise des mécanismes de contrôle (accusés de réception, retransmission) pour détecter et corriger les erreurs de transmission.

Fondamentaux du Web

TCP : fonctionnement

- Il établit une connexion entre l'émetteur et le récepteur avant d'envoyer les données.
- Il découpe les données en paquets, les numérote, et vérifie qu'ils sont tous reçus correctement.
- En cas de perte, il redemande les paquets manquants.
- Les données sont livrées dans l'ordre où elles ont été envoyées, même si les paquets peuvent arriver dans le désordre.

Fondamentaux du Web

IP : Internet Protocol

- Famille de protocoles de communication de réseaux informatiques
- Conçu par le chercheur américain **Vint Cerf** pour être utilisé sur **internet**
- Permettant
 - la gestion d'adresse identifiant les machines se connectant sur **internet**,
 - e routage des paquets de données.

Fondamentaux du Web

TCP vs IP

- **IP** gère l'adressage et le routage.
- **TCP** gère la fiabilité et l'ordre des données.

Fondamentaux du Web

Adresse IP

- Identifiant attribué à toute machine utilisant le protocole **IP**
- Deux versions possibles : **IPv4** et **IPv6**

© Achref EL MOULI

Fondamentaux du Web

Adresse IP

- Identifiant attribué à toute machine utilisant le protocole **IP**
- Deux versions possibles : **IPv4** et **IPv6**

À ne pas confondre avec adresse **MAC**

- Adresse **MAC (Media Access Control)** : adresse physique unique pour chaque matériel défini par le constructeur
- Adresse **IP** : adresse logique fourni par un fournisseur d'accès internet.

Fondamentaux du Web

Adresse **MAC**

- Codée sur 6 octets (48 bits)
- Représentée généralement sous forme hexadécimale
- ":" ou "-" pour séparer les différentes octets
- exemple : 54-13-79-6B-45-D0

© Achref EL

Fondamentaux du Web

Adresse **MAC**

- Codée sur 6 octets (48 bits)
- Représentée généralement sous forme hexadécimale
- ":" ou "-" pour séparer les différentes octets
- exemple : 54-13-79-6B-45-D0

Pour récupérer l'adresse **MAC**

- Lancer un terminal (**Invite de commandes, Power Shell...**)
- Exécuter la commande `ipconfig/all`
- Aller chercher l'Adresse physique

Fondamentaux du Web

Adresse IPv4

- Codée sur 4 octets (32 bits)
- Représentée généralement sous forme décimale
- Chaque octet contient le codage d'un nombre compris entre 0 et 255
- Les différents octets sont séparés par "."
- Théoriquement, 2^{32} adresses possibles (soit 4 294 967 296)
- Exemple : 192.168.1.13

Fondamentaux du Web

Adresse IPv4

- Codée sur 4 octets (32 bits)
- Représentée généralement sous forme décimale
- Chaque octet contient le codage d'un nombre compris entre 0 et 255
- Les différents octets sont séparés par "."
- Théoriquement, 2^{32} adresses possibles (soit 4 294 967 296)
- Exemple : 192.168.1.13

2^{32} adresses possibles \Rightarrow énorme mais insuffisant en considérant tous les périphériques connectés à Internet : ordinateurs, ordinateurs portables, les tablettes, les smartphones...

Fondamentaux du Web

Adresse IPv6

- Codée sur 16 octets (128 bits) : 8 groupes de deux octets
- Représentée généralement sous forme hexadécimale
- Les différents groupes sont séparées par ":"
- Exemple : 2a01:e35:2f72:e740:dd1f:ca5e:5201:e895

© Achre

Fondamentaux du Web

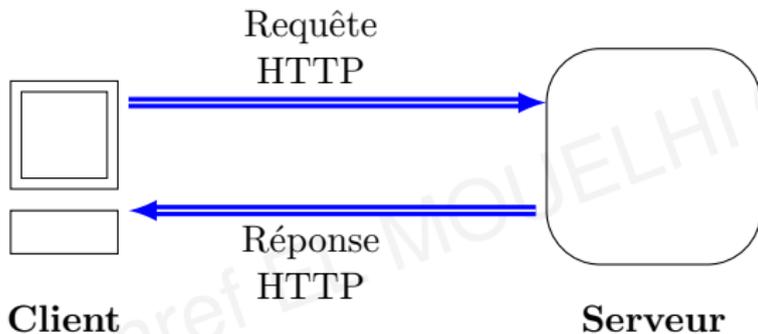
Adresse IPv6

- Codée sur 16 octets (128 bits) : 8 groupes de deux octets
- Représentée généralement sous forme hexadécimale
- Les différents groupes sont séparées par ":"
- Exemple : 2a01:e35:2f72:e740:dd1f:ca5e:5201:e895

IPv4 et IPv6 sont incompatibles

Une machine ne disposant que d'une adresse IPv4 ne peut communiquer avec une machine ne disposant que d'une adresse ipv6

Fondamentaux du Web



Architecture client/serveur et protocole HTTP

Fondamentaux du Web

Protocole HTTP

- Créé par une équipe composée principalement de **Tim Berners Lee** et **Roy Fielding**
- Utilisant le port 80
- La requête et la réponse sont composées de deux parties : entête (headers) et corps (body)
- Entête de la requête **HTTP** : méthode, cookies, encodage...
- Corps de la requête **HTTP** : paramètres de formulaire, données **JSON** ou **XML**...
- Entête de la réponse **HTTP** : code d'état, type du contenu...
- Corps de la réponse **HTTP** : page **Web**, texte brut, données **JSON** ou **XML**,...

Fondamentaux du Web

Méthode (ou verbe) HTTP

- **GET** pour la récupération (ne prend pas de corps)
- **POST** pour ajouter une nouvelle ressource
- **DELETE** pour supprimer une ressource
- **PUT** et **PATCH** pour modifier une ressource
- ...

Code d'état (**Status Code**) : code composé de 3 chiffres

- 2xx (Success - Succès)
 - 200 OK
 - 201 Created
 - 202 Accepted
- 3xx (Redirection)
- 4xx (Client Error - Erreur du client)
 - 400 Bad Request
 - 401 Unauthorized
 - 403 Forbidden
 - 404 Not Found
 - 405 Method Not Allowed
- 5xx (Server Error - Erreur du serveur)

Fondamentaux du Web

Exemple de données au format JSON

```
{  
  "nom": "John Doe",  
  "age": 30,  
  "ville": "Marseille"  
}
```

Exemple de données au format XML

```
<personne>  
  <nom>John Doe</nom>  
  <age>30</age>  
  <ville>Marseille</ville>  
</personne>
```

Fondamentaux du Web

Exemple de page HTML

```
<!DOCTYPE html>
<html>
<head>
  <title>Exemple de page Web</title>
</head>
<body>
  <h1>Bienvenue sur notre site Web!</h1>
  <p>Ceci est une page de démonstration.</p>
</body>
</html>
```

Scénario avec DNS, TCP, IP et HTTP

On ouvre le navigateur et on saisit `www.exemple.com` dans la barre d'adresse, puis on appuie sur .

- L'ordinateur envoie une requête à un serveur **DNS** (souvent celui du fournisseur d'accès ou un public comme `8.8.8.8` de **Google**).
- Le serveur **DNS** répond avec l'adresse **IP** associée, par exemple `93.184.216.34`.
- L'ordinateur utilise l'adresse **IP** (`93.184.216.34`) comme destination et son propre **IP** comme source.
- **IP** encapsule les données dans des paquets et les envoie via le réseau (routeurs, switches...) jusqu'au serveur cible.
- **TCP** ouvre une connexion avec le serveur et prépare le transport fiable.
- Le navigateur envoie une requête **HTTP**.
- Le serveur reçoit cette requête via **TCP**, interprète qu'il doit renvoyer la page d'accueil.
- **TCP** transporte cette réponse jusqu'à notre ordinateur.

Fondamentaux du Web

Navigateur

- Application logicielle
- Permettant aux utilisateurs d'accéder, de visualiser et d'interagir avec des pages web (corps d'une réponse **HTTP**)
- Interprétant le code **HTML**, **CSS** et **JavaScript** retournés par le serveur

Fondamentaux du Web

Exemple de **Navigateur**

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
- Safari
- Opéra
- ...

Fondamentaux du Web

URL : Uniform Resource Locator

- Moyen de localiser une ressource de manière unique
- Contenant généralement
 - le protocole utilisé (comme `http` ou `https`)
 - le nom de domaine (ou l'adresse **IP**)
 - le chemin d'accès vers la ressource spécifique sur le serveur
 - les paramètres

Fondamentaux du Web

Protocol

Port

Parameters

Hostname

Path

`http://localhost:8080/routeName?param1=value1¶m2=value2¶m3=value3`

URL

Fondamentaux du Web

URN : Uniform Resource Name

- Nom unique attribué à une ressource
- Ne donne pas d'information sur la localisation
- Exemple : `urn:isbn:0451450523`

Fondamentaux du Web

URI : Uniform Resource Identifier

Terme global englobant **URL** et **URN**

© Achref EL MOUËL

Fondamentaux du Web

URI : Uniform Resource Identifier

Terme global englobant **URL** et **URN**

Remarque

En pratique, les termes **URL** et **URI** sont souvent utilisés de manière interchangeable dans de nombreuses situations, car les **URL** sont les **URI** les plus courantes et familières.

Fondamentaux du Web

Page Web

- Corps d'une réponse **HTTP**
- Contenant du code **HTML**, **CSS** et **JavaScript**
- Appartenant à un site Web ou à une application Web

Fondamentaux du Web

Site Web

- Composé de plusieurs pages **Web** liées entre elles
- Accessible via une adresse **URL** commune
- Pouvant être statique ou dynamique (lié à une base de données)
- Pouvant contenir des informations statiques ou semi-dynamique, des images, des vidéos...
- Souvent très peu interactif avec l'utilisateur

Fondamentaux du Web

Exemple de sites Web

- Site de news
- Site de météo
- Un blog
- Wikipédia
- ...

Fondamentaux du Web

Application **Web**

- Souvent plus avancée en termes de fonctionnalités interactives
- Permettant aux utilisateurs d'effectuer des actions complexes, de manipuler des données en temps réel...
- Très souvent dynamique

Fondamentaux du Web

Exemple d'applications **Web**

- Gmail
- Facebook
- Site de l'URSSAF
- Site de France Travail
- ...

Fondamentaux du Web

Caractéristiques d'une application **Web**

- Fort niveau d'interaction avec l'utilisateur
- Utilisation fréquente de **JavaScript** (parfois frameworks comme **React**, **Angular...**)
- Authentification, stockage de données, logique métier
- Mise à jour dynamique des contenus sans recharger la page (**AJAX**, **API REST**)
- ...

Fondamentaux du Web

Langages (de programmation)

- est un ensemble de règles **syntaxiques** et **sémantiques**.
- permet d'écrire des instructions compréhensibles par une machine.
- traduit en langage machine via un *compilateur* ou un *interpréteur*.

Fondamentaux du Web

Caractéristiques Principales

- **Syntaxe** Règles d'écriture du code (comme `print("Hello")` en **Python**).
- **Sémantique** Signification des instructions exécutées.
- **Niveau d'abstraction**
 - Bas niveau : proche du matériel (comme **Assembleur**).
 - Haut niveau : abstrait et lisible (comme **Python, Java...**).

Fondamentaux du Web

Plusieurs paradigmes de programmation

- Déclaratif
- Impératif
- Procédural
- Orienté objet
- Fonctionnel

Fondamentaux du Web

Paradigme Impératif

Le programme est une séquence d'instructions exécutées étape par étape, modifiant l'état du système.

© Achref EL MOUËZ

Fondamentaux du Web

Paradigme Impératif

Le programme est une séquence d'instructions exécutées étape par étape, modifiant l'état du système.

Exemple : C

```
int x = 0;  
x = x + 1;  
printf("%d", x);  
(Incrémente une variable et l'affiche)
```

Fondamentaux du Web

Paradigme Procédural

Un sous-ensemble de la programmation impérative où le programme est structuré en procédures (ou fonctions).

© Achref EL MOUËL

Fondamentaux du Web

Paradigme Procédural

Un sous-ensemble de la programmation impérative où le programme est structuré en procédures (ou fonctions).

Exemple : JavaScript

```
function somme(a, b) {  
    return a + b  
}  
  
console.log(somme(2, 3))
```

Fondamentaux du Web

Paradigme Objet

Le programme est organisé autour d'objets (entités combinant données et comportements).

© Achref EL MOULI

Fondamentaux du Web

Paradigme Objet

Le programme est organisé autour d'objets (entités combinant données et comportements).

Exemple : Java

```
class Chien { void aboyer() {  
System.out.println("Wouf"); } }
```

(Définit une classe avec une méthode)

Fondamentaux du Web

Paradigme Fonctionnel

Le programme repose sur l'évaluation de fonctions pures, sans effets secondaires.

© Achref EL MOU

Fondamentaux du Web

Paradigme Fonctionnel

Le programme repose sur l'évaluation de fonctions pures, sans effets secondaires.

Exemple : Haskell

```
double x = x * 2
```

(Fonction pure qui double une valeur)

Fondamentaux du Web

Paradigme Déclaratif

Le programme décrit "quoi" obtenir sans préciser "comment" le faire.

© Achref EL MOULI

Fondamentaux du Web

Paradigme Déclaratif

Le programme décrit "quoi" obtenir sans préciser "comment" le faire.

Exemple : SQL

```
SELECT nom FROM utilisateurs WHERE age > 18;
```

(Récupère des données sans détailler l'algorithme)

Fondamentaux du Web

L'exécution peut se dérouler de deux manières différentes

- Pour les langages **compilés** : le compilateur lit le programme en entier et le transforme en binaire (code machine) ou en **bytecode**.
 - **Code machine** (binaire) : code exécutable directement par le processeur.
 - **Bytecode** : code intermédiaire compilé, exécuté par une machine virtuelle (**VM**).
- Pour les langages **interprétés** : l'interpréteur opère ligne par ligne : lit une ligne de programme, puis exécute immédiatement les instructions machines correspondantes.

Fondamentaux du Web

Compilation en bytecode vs compilation en code binaire

Type de compilation	Explication	Exemples de langages
Compilation en code binaire	Transforme directement le code source en code machine spécifique au processeur.	C, C++, Assembleur, Rust
Compilation en bytecode	Transforme le code source en bytecode intermédiaire, nécessitant une VM.	Java (JVM), C# (CLR), Python (PyCode)

Compilation : avantage

- La traduction se fait une seule fois, avant l'exécution.
- Généralement plus rapide en exécution, car le code est déjà traduit.
- Erreurs détectées à la compilation (avant l'exécution).

Compilation : inconvénient

- Le code compilé est souvent spécifique à un **OS** et une architecture matérielle.
- Développement plus lent (nécessite une compilation à chaque modification).
- Traduction Longue (compilation complète nécessaire avant exécution).

© Achref EL M...

Compilation : avantage

- La traduction se fait une seule fois, avant l'exécution.
- Généralement plus rapide en exécution, car le code est déjà traduit.
- Erreurs détectées à la compilation (avant l'exécution).

Compilation : inconvénient

- Le code compilé est souvent spécifique à un **OS** et une architecture matérielle.
- Développement plus lent (nécessite une compilation à chaque modification).
- Traduction Longue (compilation complète nécessaire avant exécution).

Interprétation : avantage

- Développement plus rapide (pas de compilation à chaque modification).
- Plus portable (l'interpréteur doit juste être présent).

Interprétation : inconvénient

- La traduction se fait à chaque exécution.
- Généralement plus lente, car la traduction se fait en temps réel.

Fondamentaux du Web

Qu'est-ce que le linkage ?

- Étape qui assemble les morceaux compilés
- Résout les références (fonctions, bibliothèques)
- Produit le fichier exécutable final

© Achref EL

Fondamentaux du Web

Qu'est-ce que le linkage ?

- Étape qui assemble les morceaux compilés
- Résout les références (fonctions, bibliothèques)
- Produit le fichier exécutable final

Remarque

Pas de linkage dans l'interprétation classique, mais parfois des liens dynamiques à l'exécution.

Fondamentaux du Web

Les étapes d'analyse du code

1 Lexicale (Lexing)

- Transforme le texte brut en unités de base appelées tokens.
- Dans : $x = 5 + y;$
- Tokens : $x, =, 5, +, y, ;$

2 Syntaxique (Parsing)

- Vérifie la grammaire du langage
- Exemple : $5 = x + ; \Rightarrow$ erreur de syntaxe !

3 Sémantique

- Est-ce que les variables sont déclarées ?
- Est-ce qu'on utilise les bons types ?
- Est-ce qu'une opération est logique ?

Fondamentaux du Web

Étapes quand un programme est compilé ou interprété

- 1 Analyse lexicale
- 2 Analyse syntaxique
- 3 Analyse sémantique
- 4 Compilation
- 5 Linkage

Fondamentaux du Web

Classification des langages par usage

- **Bas niveau / Systèmes : C, C++, Rust, Assembly**
- **Applications d'entreprise : Java, C#, Delphi, COBOL**
- **Data Science et IA : Python, R, Julia, Lisp, Prolog**
- **Automatisation et scripting : Bash, Perl, Python, PowerShell**
- **Jeux vidéo : C++, C#, Lua, GDScript**
- **Applications de bureau : Java, C#, Python**
- **Applications mobiles : Kotlin (pour Android), Swift (pour iOS)**
- ...

Fondamentaux du Web

Langages pour compilation et analyse syntaxique

- **Lex / Flex** : analyse lexicale
- **Yacc / Bison** : analyse syntaxique
- **OCaml** : conception de compilateurs, preuve de programme
- **Utilisation** : développement de compilateurs et interpréteurs

Fondamentaux du Web

Langages pour l'IA et la logique

- **Python** : Machine Learning, Deep Learning
- **R** : analyse statistique et modélisation
- **Julia** : calcul scientifique haute performance
- **Lisp** : traitement du langage naturel
- **Prolog** : systèmes experts et logique déclarative

Fondamentaux du Web

Langages pour la finance et les statistiques

- **R** : analyse de données, finance, bio-informatique
- **Julia** : calcul scientifique et haute performance
- **COBOL** : logiciels bancaires et systèmes financiers

Fondamentaux du Web

Langages spécialisés pour entreprise et RAD

- **Pascal** : enseignement, applications bureautiques
- **Delphi (Object Pascal)** : développement rapide d'applications (Windows, mobile)
- **Windev** : développement rapide d'applications
 - **ERP (Enterprise Resource Planning)** : logiciel de gestion intégré qui permet aux entreprises de gérer toutes leurs activités depuis un seul système (gestion comptable et financière, gestion des stocks et des achats, gestion des ressources humaines...)
 - **CRM (Customer Relationship Management)** : logiciel de gestion de la relation client (historique des achats, préférences, newsletters, campagnes email, devis, contrats, factures...)
- **WebDev** : développement web **SaaS (Software as a Service)** en mode **RAD (Rapid Application Development)**

Fondamentaux du Web

Langages pour les Jeux Vidéo

- **C++** : moteurs de jeu
- **C#** : jeux **Unity**
- **GDScript** : moteur Godot
- **Lua** : scripting dans les jeux

Fondamentaux du Web

Langage	Année de création	Créateur	Utilisation principale
C	1972	Dennis Ritchie (Bell Labs)	Systèmes d'exploitation, embarqué
C++	1983	Bjarne Stroustrup (Bell Labs)	Logiciels performants, jeux vidéo
C#	2000	Microsoft (Anders Hejlsberg)	Web, desktop, jeux Unity
Java	1995	Sun Microsystems (James Gosling)	Web, mobile, desktop
PHP	1995	Rasmus Lerdorf	Web
Python	1991	Guido van Rossum	IA, Data Science, Web, desktop

Fondamentaux du Web

Langage	Compilation/Interprétation	Portable	Typage	Paradigme	Typage
C	Compilé	Oui	Statique	Impératif	Faible
C++	Compilé	Oui	Statique	Impératif, Objet, Fonctionnel	Fort
C#	Compilé	Oui	Statique	Objet, Fonctionnel, Procédural	Fort
Java	Compilé	Oui	Statique	Objet, Fonctionnel	Fort
PHP	Interprété	Oui	Dynamique	Objet, Procédural	Faible
Python	Interprété	Oui	Dynamique	Objet, Procédural, Fonctionnel	Fort

Fondamentaux du Web

Qu'est-ce qu'un langage web ?

- Langages conçus pour le développement d'applications ou contenus pour le web.
- Permettent de créer des pages web interactives, de gérer des données, et d'interagir avec les utilisateurs.
- Peuvent être classés en deux grandes catégories
 - Langages client (frontend)
 - Langages serveur (backend)

Fondamentaux du Web

Langages clients

- Langages de programmation exécutés du côté du navigateur de l'utilisateur.
- Responsables de l'interface utilisateur et de l'interaction avec l'utilisateur dans le navigateur
 - **Structure** : HTML (contenu).
 - **Présentation** : CSS (apparence).
 - **Comportement** : **JavaScript** (interactivité, données).

Fondamentaux du Web

Langages serveurs

- Langages serveurs sont des langages de programmation exécutés du côté du serveur web.
- Utilisés pour
 - traiter les requêtes des utilisateurs
 - accéder aux bases de données
 - effectuer des opérations côté serveur
 - générer dynamiquement les pages **Web** à renvoyer au navigateur

Fondamentaux du Web

Exemples de langage serveur

- **Java** avec des frameworks comme **Spring**
- **C#** avec des frameworks comme **ASP.NET Core**
- **PHP** avec des frameworks comme **Symfony** ou **Laravel**
- **Python** avec des frameworks comme **Django** ou **Flask**
- ...

Fondamentaux du Web

Critères de choix d'une stack web

- Taille et complexité du projet
- Budget et ressources disponibles
- Architecture souhaitée (microservices, monolithique...)
- Compétences de l'équipe
- Performances et scalabilité attendues
- Communauté et écosystème

Fondamentaux du Web

Comparatif backend : vue d'ensemble

Critère	Java	ASP.NET	Python	PHP
Performance	++	++	+	+
Projet cible	Moyen à gros	Moyen à gros	Petit à moyen	Petit à moyen
Apprentissage	Difficile	Moyen	Facile	Facile
Communauté	Très large	Large	Très large	Très large
Hébergement	Coûteux	Moyen	Peu coûteux	Très économique
Typage	Fort	Fort	Dynamique	Dynamique

Fondamentaux du Web

Résumé Backend

- **Java (Spring)** : pour des SI critiques et des architectures robustes
- **ASP.NET Core** : moderne, performant, idéal avec Azure
- **PHP (Symfony)** : efficace pour le web traditionnel (sites, e-commerce)

Fondamentaux du Web

Comparatif frontend : vue d'ensemble

Critère	Angular	React	Vue.js
Complexité	Élevée	Moyenne	Faible
Structure	Très structuré	Flexible	Léger
Projet cible	Moyen à gros	Tout type	Tout type
Communauté	Grande	Énorme	Grande
Apprentissage	Raide	Moyenne	Douce
Performance	Excellente	Excellente	Très bonne

Fondamentaux du Web

Résumé Frontend

- **Angular** : pour les gros projets structurés
- **React** : polyvalent, populaire, idéal avec des APIs **REST/GraphQL**
- **Vue.js** : intuitif, très productif pour les petits/moyens projets

Fondamentaux du Web

Pour conclure

- Il n'existe pas une seule bonne stack.
- Choisir selon les besoins, les contraintes et les compétences de l'équipe.
- La cohérence entre backend et frontend est aussi un facteur clé.

Fondamentaux du Web

Base de données

Ensemble structuré de données organisées de manière à faciliter le stockage, la recherche, la récupération et la gestion de l'information

© Achref EL MOU

Fondamentaux du Web

Base de données

Ensemble structuré de données organisées de manière à faciliter le stockage, la recherche, la récupération et la gestion de l'information

Système de Gestion de Base de Données (SGBD)

Logiciel permettant de créer, de manipuler et de gérer une base de données

Fondamentaux du Web

Serveur

- Un ordinateur ou un système informatique
- Fournissant des services, des ressources ou des fonctionnalités à d'autres ordinateurs, appelés clients

Plusieurs types de serveur

- **Serveur Web** :
 - Connu aussi sous le nom de serveur **HTTP**
 - Fournit des pages web aux navigateurs des utilisateurs
 - Situé dans un centre de données (data center), dans le **cloud** ou localement
 - Connecté à **Internet** via une connexion à large bande à haute vitesse, assurant une connectivité rapide et fiable pour répondre aux demandes des utilisateurs.
- **Serveur de fichiers** : Stocke et gère les fichiers à d'autres ordinateurs du réseau.
Exemple : **Samba**, **FileZilla**...
- **Serveur de messagerie** : Gère l'envoi, la réception et le stockage des e-mails.
- **Serveur de base de données** : Stocke et gère les bases de données, permettant aux applications d'accéder et de manipuler des données de manière structurée.
- **Serveur DNS (Domain Name System)** : Traduit les noms de domaine en adresses **IP**, facilitant ainsi la navigation sur Internet.
- **Serveur DHCP (Dynamic Host Configuration Protocol)** : Distribue automatiquement les adresses **IP** aux clients du réseau.
- ...

Fondamentaux du Web

Exemples de serveurs Web par langage/platforme

- **Java** : Apache Tomcat, WebLogic Server, GlassFish, Jetty, WildFly (anciennement JBoss)...
- **C#** : Microsoft Internet Information Services (IIS)...
- **PHP** : Apache HTTP Server, Nginx, IIS...
- **Python** : Gunicorn, Waitress...
- ...

Fondamentaux du Web

Exemples de serveurs données

- **Oracle Corporation : Oracle, MySQL**
- **Microsoft : Microsoft SQL Server**
- Quelques serveurs open-sources : **MariaDB , PostgreSQL ...**
- ...

Fondamentaux du Web

Framework

- Plusieurs traductions
 - cadriciel
 - environnement de développement
 - cadre d'applications
 - ...
- Ensemble de composants logiciels
- Facilitant la réalisation d'une (partie de l') application
- Imposant une certaine structure, logique, syntaxe...

Fondamentaux du Web

Plusieurs types de Framework

- Frameworks applicatifs pour le développement d'applications web :
 - **Vue.js** pour **JavaScript**,
 - **Spring** pour **Java**,
 - **Symfony** pour **PHP**
 - ...
- Frameworks de présentation de contenu web :
 - **Bootstrap** pour **CSS**
 - ...
- Frameworks de persistance
- Frameworks de logging
- ...

Fondamentaux du Web

Framework : avantages

- Gain de temps
- Meilleure organisation de projet
- Respect de bonnes pratiques
- Meilleure efficacité
- Faciliter le travail d'équipe
- ...

Fondamentaux du Web

Framework : inconvénients

- Complexité
- Apprentissage
- ...

Fondamentaux du Web

Quelques frameworks pour

- **Java** : Apache Struts, Spring
- **C#** : ASP.NET MVC
- **PHP** : Symfony, Laravel
- **Python** : Django, Flask
- **JavaScript** : Angular, Express.js, Vue.js, Nest.js

Fondamentaux du Web

Métiers dans le domaine du développement

- Développeur front-end
- Développeur back-end
- Développeur full-stack : maîtrise à la fois le front-end et le back-end, ce qui lui permet de travailler sur l'ensemble d'un projet web.
- Développeur web mobile : spécialisé dans la création d'applications web pour les appareils mobiles.
- Intégrateur web : transforme les maquettes graphiques en code **HTML**, **CSS** et **JavaScript** pour créer des pages web.
- Webmaster : responsable de la gestion technique et de la maintenance d'un site web.
- ...

Fondamentaux du Web

Évolutions de carrière (avec de l'expérience)

- Chef de projet informatique
- Lead développeur
- Architecte logiciel
- Expert en développement web ou mobile
- ...

Fondamentaux du Web

Ces métiers peuvent s'exercer dans divers environnements

- Entreprises de services numériques (**ESN**)
- Agences web
- Start-ups
- Grandes entreprises
- En freelance
- ...

Fondamentaux du Web

Ces métiers peuvent s'exercer dans divers environnements

- Entreprises de services numériques (**ESN**)
- Agences web
- Start-ups
- Grandes entreprises
- En freelance
- ...

Remarque

ESN : un terme qui a remplacé **SSII** (Société de Services en Ingénierie Informatique).

Fondamentaux du Web

Cloud

- Ensemble des ressources (services) informatiques partagées via Internet
 - **Stockage de données** : pour sauvegarder et accéder à des fichiers sans nécessiter de support physique local.
 - **Hébergement d'applications** : pour déployer des applications sur des serveurs distants sans avoir à gérer l'infrastructure physique.
 - **Logiciels et applications** : accessibles en ligne, souvent sous forme d'abonnements.
 - **Puissance de calcul** : pour des tâches gourmandes en ressources, comme l'analyse de données ou l'intelligence artificielle.
- Remplaçant les ressources physiques comme les serveurs et les logiciels installés localement sur un ordinateur ou dans une entreprise

Fondamentaux du Web

Le cloud et le développement

- **Développement simplifié** : créer, tester et déployer des applications plus rapidement et à moindre coût.
- **Scalabilité** : s'adapter facilement à une augmentation du nombre d'utilisateurs (montée en charge).
- **DevOps et CI/CD** : avec des outils comme **GitHub Actions**, **GitLab CI/CD**, **AWS CodePipeline** pour faciliter l'automatisation du déploiement et des tests.
- **IA et ML** : des services cloud comme **Google AI**, **AWS SageMaker** offrent des modèles **IA** pré-entraînés accessibles via **API**.
- **Bases de données managées** : des solutions comme **Amazon RDS**, **Firebase Firestore** ou **MongoDB Atlas** permettent d'héberger des bases de données sans s'occuper de leur administration.

Fondamentaux du Web

XaaS (Anything as a Service)

- Ce qui se traduit en : tout en tant que service
- Des modèles de services informatiques proposés via le cloud

Fondamentaux du Web

Modèles de services cloud (XaaS)

Nom	Signification	Définition	Exemples
SaaS	Software as a Service	Logiciels accessibles via Internet, sans installation locale.	Google Drive, Salesforce, Dropbox
PaaS	Platform as a Service	Environnements de développement et de déploiement dans le cloud.	Heroku, Google App Engine, Microsoft Azure App Services
IaaS	Infrastructure as a Service	Services cloud fournissant des serveurs, du stockage et du réseau.	AWS EC2, Google Compute Engine, Azure Virtual Machines
FaaS	Function as a Service	Exécution de fonctions sans gestion de serveur (serverless computing).	AWS Lambda, Google Cloud Functions, Azure Functions

Fondamentaux du Web

Fournisseur	Lancement	Types de services	Description
Amazon Web Services (AWS)	2006	IaaS, PaaS, SaaS	Leader du marché, offre plus de 200 services, dont le calcul, le stockage (S3), et l'IA.
Microsoft Azure	2010	IaaS, PaaS, SaaS	Plateforme polyvalente avec intégration forte aux outils Microsoft et support hybride.
Google Cloud Platform (GCP)	2008	IaaS, PaaS, SaaS	Réputé pour ses capacités en analyse de données et apprentissage automatique.
Alibaba Cloud	2009	IaaS, PaaS, SaaS	Dominant en Asie, propose des solutions compétitives pour le commerce électronique.
Oracle Cloud	2016	IaaS, PaaS, SaaS	Spécialisé dans les bases de données et les solutions d'entreprise hybrides.
IBM Cloud	2011	IaaS, PaaS, SaaS	Axé sur les environnements hybrides et les services d'IA (Watson).

Fondamentaux du Web

Logiciels de gestion

- des solutions technologiques qui visent à améliorer l'efficacité, la productivité et la prise de décision dans une entreprise
- s'adaptent aux besoins spécifiques de chaque organisation

Fondamentaux du Web

Logiciels de gestion

Nom	Signification	Définition	Exemples
CRM	Customer Relationship Management	Gestion de la relation client et optimisation des ventes.	Salesforce, HubSpot, Zoho CRM
ERP	Enterprise Resource Planning	Gestion intégrée des ressources et processus d'une entreprise.	SAP, Odoo, Microsoft Dynamics
SCM	Supply Chain Management	Gestion de la chaîne logistique et des fournisseurs.	SAP SCM, Oracle SCM, Blue Yonder
BPM	Business Process Management	Automatisation et optimisation des processus métier.	Camunda, IBM BPM, BonitaSoft
PLM	Product Lifecycle Management	Gestion du cycle de vie des produits, de la conception à la fabrication.	Dassault ENOVIA, Siemens Teamcenter, PTC Windchill

Fondamentaux du Web

Contexte

- Plusieurs architectures possibles pour construire une application Web.
- Le choix dépend du type d'application, des besoins techniques et des contraintes de l'équipe.
- Focus sur 3 architectures majeures : Monolithique, SPA, Microservices.

Fondamentaux du Web

Monolithique

- Application centralisée : frontend, backend, base de données dans un même projet.
- Génération de pages côté serveur (Server-side rendering).
- Exemple : **PHP (Laravel)**, **Java (Spring)**, **Python (Django)**.

Fondamentaux du Web

Avantages

- Simplicité de développement et déploiement.
- Bon **SEO** par défaut.
- Moins de technologies à maîtriser.

© Achref EL

Fondamentaux du Web

Avantages

- Simplicité de développement et déploiement.
- Bon **SEO** par défaut.
- Moins de technologies à maîtriser.

Inconvénients

- Moins flexible pour les interfaces riches.
- Difficulté à scaler certaines parties de manière indépendante.

Fondamentaux du Web

SPA

- Application chargée une seule fois, tout le reste est dynamique via **JavaScript**.
- Utilise une **API REST** ou **GraphQL** pour communiquer avec le backend.
- Frameworks : **React**, **Angular**, **Vue.js**.

Fondamentaux du Web

Avantages

- **UX** fluide et rapide.
- Séparation claire frontend / backend.
- Bon pour les applications riches et dynamiques.

© Achref EL M...

Fondamentaux du Web

Avantages

- **UX** fluide et rapide.
- Séparation claire frontend / backend.
- Bon pour les applications riches et dynamiques.

Inconvénients

- **SEO** plus difficile sans **SSR**.
- Dépend fortement de **JavaScript**.
- Setup plus complexe.

Fondamentaux du Web

Microservices

- Application décomposée en petits services indépendants.
- Chaque service a son propre domaine métier, base de données et **API**.
- Communication via **HTTP** ou message broker (**Kafka**, **RabbitMQ**...).

Fondamentaux du Web

Avantages

- Scalabilité et résilience.
- Équipes autonomes sur chaque service.
- Déploiements indépendants.

© Achref EL M...

Fondamentaux du Web

Avantages

- Scalabilité et résilience.
- Équipes autonomes sur chaque service.
- Déploiements indépendants.

Inconvénients

- Complexité accrue (orchestration, monitoring...).
- Risque de duplication de logique.
- Nécessite une bonne culture **DevOps**.

Fondamentaux du Web

Comparaison des Architectures

Critère	Monolithique	SPA	Microservices
Complexité	Faible	Moyenne	Élevée
Scalabilité	Limitée	Bonne	Excellente
SEO	Très bon	Faible (sans SSR)	Variable
Maintenance	Facile au début	Moyenne	Complexe
Réactivité UI	Faible	Excellente	Dépend du front
Déploiement	Simple	Moyen	Complexe

Fondamentaux du Web

Pour conclure

- Pas d'architecture parfaite, tout dépend du contexte.
- Pour un projet simple : **Monolithique**.
- Pour une UI riche et dynamique : **SPA**.
- Pour des projets à grande échelle : **Microservices**.

Fondamentaux du Web

IDE

- **I**ntegrated **D**evelopment **E**nvironment (ou Environnement de Développement Intégré en français)
- est un logiciel qui regroupe un ensemble d'outils conçus pour faciliter le développement de programmes informatiques.
- inclut généralement :
 - Un éditeur de texte avancé avec coloration syntaxique et autocomplétion.
 - Un débogueur pour identifier et corriger les erreurs dans le code.
 - Un compilateur ou un interpréteur pour exécuter le code.
 - Des outils de gestion de projet (comme la gestion des fichiers ou des dépendances).
 - Parfois, des fonctionnalités supplémentaires comme l'intégration avec des systèmes de contrôle de version (Git, par exemple).

Fondamentaux du Web

IDE Gratuits

- **Visual Studio Code** (Microsoft) : *Langages : JS, TS, Python, C++, Java, PHP...*
- **Eclipse** (Eclipse Foundation) : *Langages : Java, C++, Python, PHP, JavaScript*
- **PyCharm Community** (JetBrains) : *Langages : Python*
- **IntelliJ IDEA Community** (JetBrains) - *Langages : Java, Kotlin, Scala*
- **NetBeans** (Apache) : *Langages : Java, C/C++, PHP, JavaScript*
- **Android Studio** (Google) : *Langages : Java, Kotlin*
- **Xcode** (Apple) : *Langages : Swift, Objective-C, C, C++*

Fondamentaux du Web

IDE Payants

- **Visual Studio Pro/Enterprise** (Microsoft) : *Langages : C#, .NET, C++, Python, JavaScript*
- **PyCharm Professional** (JetBrains) : *Langages : Python, Django, Flask, JS*
- **IntelliJ IDEA Ultimate** (JetBrains) : *Langages : Java, Kotlin, Scala, JS, TS*
- **WebStorm** (JetBrains) : *Langages : JavaScript, TypeScript, HTML, CSS, Node.js*
- **PhpStorm** (JetBrains) : *Langages : PHP, JavaScript, HTML, CSS*
- **CLion** (JetBrains) : *Langages : C, C++, Rust*
- **Rider** (JetBrains) : *Langages : C#, .NET*