


# Technologies et Programmation Web

*Cours sur Google Classroom*




*Scan me!*

Code : **2bbmr6ao**

Enseignante : **Dorra DHAOU**

Email : [dorra.dhaou@fst.utm.tn](mailto:dorra.dhaou@fst.utm.tn) / [dorradhaou.isimm@gmail.com](mailto:dorradhaou.isimm@gmail.com)


AU. 2025 - 2026



1

## Contenu du cours

1. Introduction au développement web
2. HTML – Structure des pages web
3. CSS – Mise en forme et design
4. JavaScript – Interactivité côté client
5. PHP – Bases du côté serveur
6. Formulaires et interaction client-serveur
7. Connexion entre PHP et MySQL



2

## Objectifs du cours et Prérequis

3



### Objectifs :

- Comprendre les **enjeux** et les **domaines d'application** du développement web.
- Identifier les **composants fondamentaux** d'une application web.
- Distinguer les **rôles** des technologies **front-end** et **back-end**.
- Préparer son **environnement** de développement.



### Prérequis

- Compétences en logique et **algorithmique**
- Maîtrise de l'**environnement** informatique
- Culture générale du **Web**
- Notions de base en base de données
- Connaissance du langage **XML**

3

## C'est quoi le développement Web ?

- Le **développement Web** désigne l'ensemble des techniques et des outils permettant de créer des sites et des applications accessibles via Internet.
- C'est un domaine en constante évolution, au cœur de notre quotidien numérique : réseaux sociaux, plateformes de streaming, sites e-commerce, blogs, services en ligne...
- Il combine **design**, **programmation**, et **architecture réseau** pour offrir des expériences interactives aux utilisateurs.

4

# C'est quoi le développement Web ?

## Trois piliers essentiels :

- **Le front-end** : ce que voit et utilise l'utilisateur (interface, design, interactivité).
- **Le back-end** : ce qui se passe derrière le rideau (serveur, base de données, logique métier).
- **La communication entre les deux** : via des protocoles et des APIs.

Pilier	Rôle	Exemple
Front-end	Interface visible par l'utilisateur	Boutons, menus, animations
Back-end	Traitement des données et logique serveur	Connexion, base de données
Communication	Échange entre client et serveur	Requêtes HTTP, APIs

5



## Chapitre 1

# Introduction à la programmation web

**Dorra DHAOU**



AU. 2025 - 2026

6

# PLAN du cours

7

1. Qu'est-ce que le Web ?
2. Panorama des technologies web
3. Front-end vs Back-end
4. Cycle de vie d'une requête web
5. Outils du développeur web
6. Le système multitâche dans Android
7. Premier projet : Hello Web

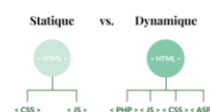


7

## Qu'est-ce que le Web ?



- Le **World Wide Web (WWW)** est un système hypertexte accessible via Internet, permettant de consulter des pages web reliées entre elles par des liens. Il repose sur des technologies comme **HTML**, **URL** et le protocole **HTTP**.
- **1990s** : Web statique → pages HTML simples, sans interaction.
- **2000s** : Web dynamique → contenu généré côté serveur (PHP, ASP).
- **2010s** : Web interactif → JavaScript, AJAX, APIs, SPAs (React, Angular).
- **Aujourd'hui** : Web intelligent → PWA, IA, Web 3.0, responsive design.



8

# Qu'est-ce que le Web ?



## Qu'est-ce qu'un URI ?

- Un **URI** (*Uniform Resource Identifier*) est une chaîne de caractères qui permet **d'identifier une ressource** sur le Web, de manière unique et permanente.
- Un **URL** (*Uniform Resource Locator*) est un **type spécifique d'URI**, utilisé pour **localiser** une ressource (ex. : une page web, une image, un fichier).
- *Tous les URL sont des URI, mais tous les URI ne sont pas forcément des URL.*

9

# Qu'est-ce que le Web ?



## Structure d'un URI

Partie	Rôle	Exemple
<b>Protocole</b> (comment ?)	Indique le mode d'accès à la ressource	http:// ou https://
<b>Adresse</b> (où ?)	Spécifie le serveur ou le domaine	www.example.com/
<b>Document</b> (quoi ?)	Désigne le fichier ou la ressource ciblée	index.html, image.jpg

### Exemple:

<https://www.w3.org/index.php>

→ **https://** = protocole

→ **www.w3.org/** = adresse

→ **index.php** = document

10

# Qu'est-ce que le Web ?

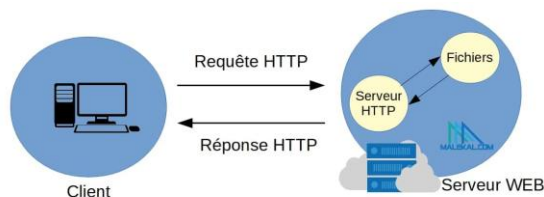


## Architecture client-serveur

- **Client** : navigateur (Chrome, Firefox...) → envoie des requêtes.
- **Serveur** : héberge les fichiers web → répond avec du contenu.
- Communication via le protocole **HTTP** (c'est comme le langage que le client et le serveur utilisent pour se comprendre).

### Cycle simplifié :

Utilisateur → Navigateur → Requête HTTP → Serveur → Réponse HTML → Affichage



11

# Qu'est-ce que le Web ?



## Protocoles fondamentaux : HTTP & HTTPS

- **HTTP (HyperText Transfer Protocol)** : C'est le protocole de communication utilisé pour transférer des pages web entre un **client** (navigateur) et un **serveur**. Il définit comment les requêtes et les réponses sont structurées et échangées.
- **HTTPS (HyperText Transfer Protocol Secure)** : c'est la version sécurisée → avec chiffrement SSL/TLS pour protéger les données.

Protocole	Fonction	Sécurité
HTTP	Transfert de données entre client et serveur	Non sécurisé
HTTPS	HTTP + chiffrement SSL/TLS	Sécurisé (🔒)

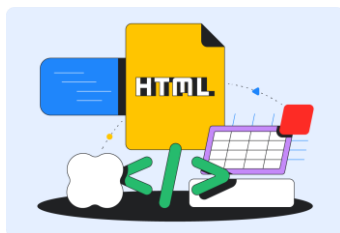
→ **HTTPS** est indispensable pour les sites modernes, surtout ceux qui traitent des données sensibles.

12

## Panorama des technologies web

### HTML – Structure du contenu

- Langage de balisage utilisé pour créer la structure des pages web.
- Organise les éléments : titres, paragraphes, images, liens, formulaires...
- **Exemple** : `<h1>Bienvenue</h1>`, `<p>Voici mon site</p>`



13

## Panorama des technologies web

### CSS – Mise en forme et design

- Langage de style qui contrôle l'apparence des éléments HTML.
- Permet de définir les couleurs, polices, marges, animations...
- **Exemple** : `p { color: blue; font-size: 16px; }`



14

## Panorama des technologies web

### ⚙️ JavaScript – Interactivité côté client

- Langage de programmation exécuté dans le navigateur.
- Permet de rendre les pages dynamiques : menus déroulants, validations de formulaires, animations...

#### □ Exemple :

```
document.getElementById("btn").onclick = function() { alert("Bonjour !"); }
```



15

## Panorama des technologies web

### 🖨️ Serveurs web et langages côté serveur

Serveur	Fonction
Apache	Serveur HTTP open-source, très utilisé
Nginx	Serveur performant, adapté aux fortes charges

Langage	Fonction
PHP	Langage classique pour le back-end, facile à apprendre
Node.js	Permet d'utiliser JavaScript côté serveur, moderne et rapide



### Bases de données

Base	Type	Usage
MySQL	Relationnelle	Stockage structuré, idéal pour les sites classiques
MongoDB	NoSQL	Stockage flexible, adapté aux applications modernes

16

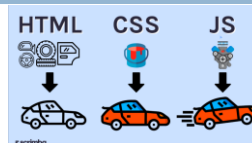


## Front-end vs Back-end



### Front-end : ce que voit l'utilisateur

- Interface graphique visible dans le navigateur.
- Construit avec **HTML, CSS, JavaScript**.
- Gère l'affichage, l'interaction, l'expérience utilisateur.



**Exemples** : boutons, menus, formulaires, animations.



### Back-end : ce qui se passe derrière le rideau



- Partie invisible qui gère la logique, les données et les traitements.
- Fonctionne sur le **serveur** avec des langages comme **PHP, Node.js, Python**.
- Accède aux **bases de données**, vérifie les informations, génère des réponses.

**Exemples** : vérification de mot de passe, enregistrement d'un message, calculs.

17

## Front-end vs Back-end



### Communication entre les deux via API

- Le front-end envoie des **requêtes** (souvent en JavaScript) pour demander ou envoyer des données.
- Le back-end répond avec des **données** (souvent en format **JSON**).
- Cette communication se fait via des **API** (Application Programming Interface).

**Exemple** : GET /produits → le serveur renvoie la liste des produits



#### Exemple

- L'utili
- Le fr
- Le b
- base de
- Le se
- Le fr



18

## Outils du développement web

### Éditeurs de code

- **Visual Studio Code** : puissant, gratuit, très utilisé dans l'industrie. 
- **Sublime Text** : rapide et léger, idéal pour les débutants. 
- Ces outils permettent d'écrire, organiser et colorer le code HTML, CSS, JS.

### Navigateurs web + outils de développement

- **Google Chrome et Mozilla Firefox** :  
navigateurs modernes avec outils intégrés. 
- **DevTools** : inspecter le DOM, modifier le CSS en direct, analyser les requêtes réseau. 
- Utiles pour comprendre comment le navigateur interprète le code et détecter les erreurs.

19

## Outils du développement web

### Serveurs locaux

- **XAMPP / WAMP** : simulent un serveur web sur l'ordinateur, utiles pour PHP et MySQL.
- **Live Server (extension VSCode)** : recharge automatique du navigateur à chaque modification.
- Permettent de tester les pages web localement avant de les mettre en ligne.

20

## Organisation des fichiers d'un projet web

/mon-projet

— index.html	→ page principale
— style.css	→ styles de la page
— script.js	→ logique JavaScript
— /images	→ dossier pour les visuels
— /assets	→ polices, icônes, etc.

→ Une bonne organisation facilite la maintenance et la collaboration.