



# introduction aux langages informatiques

**Note importante** : Ce document est conçu comme un guide de référence, mais n'est pas exhaustif. La capacité à rechercher des informations complémentaires et à résoudre des problèmes de manière autonome est une compétence essentielle pour tout développeur. Si vous rencontrez des difficultés ou si certaines étapes ne fonctionnent pas comme prévu, n'hésitez pas à consulter d'autres ressources (documentation officielle, tutoriels YouTube, forums spécialisés comme Stack Overflow). Apprendre à trouver l'information manquante et à déboguer par vous-même est probablement la compétence la plus précieuse que vous développerez dans votre parcours professionnel.

## Table des matières

[Table des matières](#)

[Les trois principaux types de langages informatiques](#)

[Les langages de balisage](#)

[Les langages de style](#)

[Les langages de programmation](#)

[Historique de HTML et CSS](#)

[Ressources complémentaires](#)

## ▼ Les trois principaux types de langages informatiques

- Le langage de balisage (Markup Language)

- Les langages de programmation
- Les langages de style

```
<!-- HTML, un langage de balisage  
<button id="demo">Demo</button>
```

```
/** CSS, un langage de style */  
#demo {  
  background: blue;  
  color: white;  
}
```

```
//Javascript, un langage de programmation  
element.addEventListener("click",  
function myFunction() {  
  const buttonDemo = document.  
  buttonDemo.innerHTML = "Hello"  
})
```

## ▼ Les langages de balisage

### Introduction



Les langages de balisage sont utilisés pour structurer et décrire des documents numériques, en définissant la sémantique des éléments.

## WEB

### HTML (HyperText Markup Language)



HTML est le langage de balisage standard pour créer des pages web.

### ▼ Structure de base d'un document HTML5

```
<!DOCTYPE html>  
<html lang="fr">  
<head>
```

```

<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale
=1.0">
<title>Titre de la page</title>
<link rel="stylesheet" href="styles.css">
</head>
<body>
  <header>
    <h1>Titre principal</h1>
    <nav>
      <ul>
        <li><a href="#section1">Section 1</a></li>
        <li><a href="#section2">Section 2</a></li>
      </ul>
    </nav>
  </header>

  <main>
    <section id="section1">
      <h2>Section 1</h2>
      <p>Contenu de la section 1.</p>
    </section>

    <section id="section2">
      <h2>Section 2</h2>
      <p>Contenu de la section 2.</p>
      
    </section>
  </main>

  <footer>
    <p>&copy; 2025 - Tous droits réservés</p>
  </footer>

  <script src="script.js"></script>
</body>
</html>

```

## ▼ Éléments HTML5 sémantiques

HTML5 a introduit de nombreux éléments sémantiques qui décrivent mieux la structure et le contenu:

- `<header>` : En-tête de page ou de section
- `<nav>` : Navigation principale
- `<main>` : Contenu principal
- `<section>` : Section thématique
- `<article>` : Contenu autonome
- `<aside>` : Contenu connexe
- `<footer>` : Pied de page
- `<figure>` et `<figcaption>` : Contenu illustré avec légende

## Outils (Github, Notion, Obsidian...)

### Markdown



Markdown est un langage de balisage léger conçu pour être facile à lire et à écrire.

#### ▼ Syntaxe de base Markdown

# Titre de niveau 1

## Titre de niveau 2

Paragraphe avec du **texte en gras** et du *texte en italique*.

[Lien vers un site](https://example.com)

![Texte alternatif de l'image](chemin/vers/image.jpg)

- Élément de liste 1
- Élément de liste 2
  - Sous-élément de liste

1. Élément numéroté 1
2. Élément numéroté 2

> Citation de texte

```code en ligne```

```
```javascript
// Bloc de code
function hello() {
  console.log("Hello, world!");
}
```
```

---

|           |           |
|-----------|-----------|
| Colonne 1 | Colonne 2 |
| -----     | -----     |
| Cellule 1 | Cellule 2 |

## Logiciels (OpenOffice, Word)

### XML (eXtensible Markup Language)



XML est un langage de balisage extensible utilisé pour stocker et transporter des données.

#### ▼ Exemple de documents XML

```
<?xml version="1.0" encoding="UTF-8"?>
<livres>
  <livre id="1">
    <titre>Le Petit Prince</titre>
    <auteur>Antoine de Saint-Exupéry</auteur>
    <annee>1943</annee>
  </livre>
```

```
<livre id="2">  
  <titre>1984</titre>  
  <auteur>George Orwell</auteur>  
  <annee>1949</annee>  
</livre>  
</livres>
```

## ▼ OOXML (Office Open XML)



Format utilisé par Microsoft Office (Word, Excel, PowerPoint) depuis 2007.

# ▼ Les langages de style

## Introduction



Les langages de style sont utilisés pour définir la présentation visuelle des documents et des pages web. Ils séparent le contenu (structure) de sa présentation.

## WEB

## CSS (Cascading Style Sheets)



CSS est le langage de style standard pour le web, permettant de contrôler l'apparence des éléments HTML.

## ▼ Structure de base du CSS

```
/* Sélecteur { propriété: valeur; } */  
body {
```

```

font-family: 'Arial', sans-serif;
background-color: #f5f5f5;
margin: 0;
padding: 0;
}

.container {
  max-width: 1200px;
  margin: 0 auto;
}

#header {
  background-color: #333;
  color: white;
  padding: 1rem;
}

/* Sélecteurs avancés */
article p:first-child {
  font-weight: bold;
}

button:hover {
  background-color: #0066cc;
  color: white;
}

```

## ▼ Concepts clés du CSS

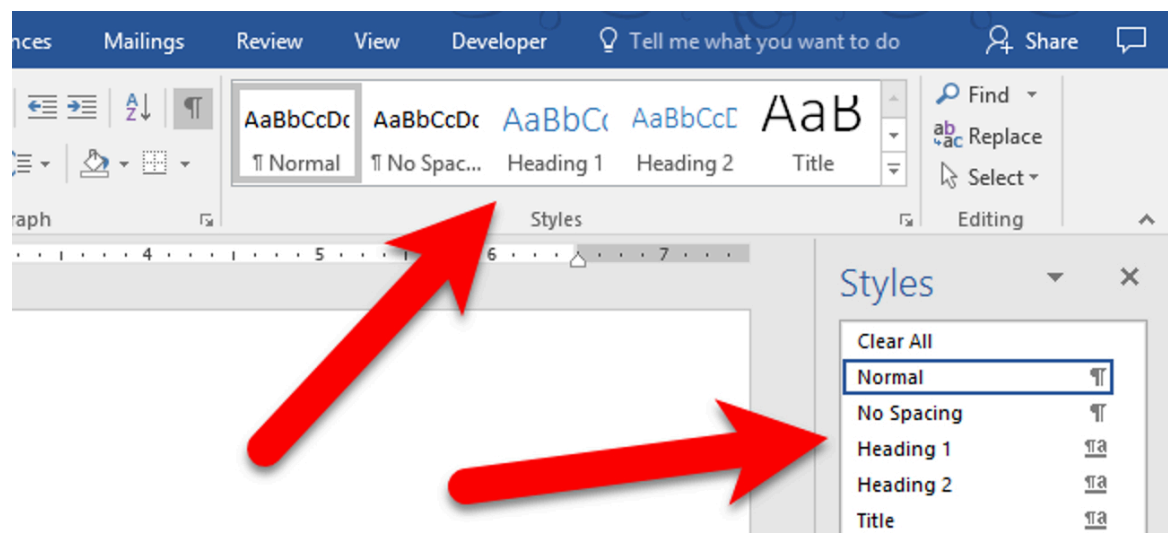
1. **La cascade:** Ordre d'application des styles (importance, spécificité, ordre source)
2. **L'héritage:** Transmission des propriétés des éléments parents aux enfants
3. **La spécificité:** Mécanisme déterminant quel style s'applique en cas de conflit
4. **Le box model:** Modèle de boîte définissant comment les éléments sont dimensionnés

## ▼ Tendances modernes en CSS

- **CSS Grid & Flexbox**: Systèmes de mise en page modernes
- **Variables CSS (custom properties)**: Variables natives en CSS
- **CSS Animations & Transitions**: Animation sans JavaScript
- **Media queries**: Design responsive
- <https://sass-lang.com/>

## Logiciels (Word)

### Outils de style



## ▼ Les langages de programmation

### Introduction



Les langages de programmation sont des systèmes de notation destinés à écrire des logiciels et des programmes informatiques. Ils permettent de spécifier des algorithmes, manipuler des données et contrôler le fonctionnement des ordinateurs.

### Classification par domaine d'application



# Développement Web (applications accessibles via un navigateur Web)

Les applications web sont divisées en deux parties principales:

## 1. Frontend (Client)

- Interface permettant d'accéder à l'application et d'interagir avec les données
- **Langages principaux:** JavaScript, TypeScript
- **Frameworks populaires:** React, Vue.js, Angular, Svelte
- **Rôle:** Gestion de l'interface utilisateur, interactions, animations, validation de formulaires

## 2. Backend (Serveur)

- Hébergement de l'application, traitement et récupération des données
- **Langages principaux:** PHP, Java, JavaScript (Node.js), Python, Ruby, Go
- **Frameworks populaires:** Express.js, Laravel, Spring, Django, Ruby on Rails
- **Rôle:** Gestion des bases de données, authentification, logique métier, API

**Exemples d'applications web:** Blogs, Chats, Forums, Sites E-Commerce, Applications SaaS

# Développement Logiciel

Le développement logiciel concerne la création de programmes exécutables directement sur un ordinateur.

- **Langages bas niveau:** Assembleur, C
- **Langages haut niveau:** C++, C#, Java, Python, Rust
- **Frameworks pour les jeux:** Unity (C#), Unreal Engine (C++)
- **Rôle:** Applications de bureau, Jeux vidéo, Systèmes d'exploitation, Logiciels métier

# Classification par paradigme

Les langages de programmation peuvent être classés selon différents paradigmes:

1. **Programmation impérative:** Instructions séquentielles modifiant l'état du programme (C, Pascal)
2. **Programmation orientée objet:** Organisation du code en objets encapsulant des données et des méthodes (Java, C++, Python)
3. **Programmation fonctionnelle:** Traitement basé sur l'évaluation de fonctions sans effets secondaires (Haskell, Lisp, JavaScript moderne)
4. **Programmation déclarative:** Description du résultat attendu plutôt que des étapes pour y parvenir (SQL, HTML/CSS)

## Structure d'un programme JavaScript moderne

```
// Import de modules (ES Modules)
import { useState } from 'react';

// Déclaration de fonction (avec approche fonctionnelle)
const calculateTotal = (items) => {
  return items.reduce((total, item) => total + item.price, 0);
};

// Classe (programmation orientée objet)
class ShoppingCart {
  constructor() {
    this.items = [];
  }

  addItem(item) {
    this.items.push(item);
  }

  getTotal() {
    return calculateTotal(this.items);
  }
}

// Utilisation de l'API moderne (Promises, async/await)
```

```
async function fetchProducts() {
  try {
    const response = await fetch('https://api.example.com/products');
    const data = await response.json();
    return data;
  } catch (error) {
    console.error('Erreur lors de la récupération des produits:', error);
    return [];
  }
}
```

## ▼ Historique de HTML et CSS

### ▼ Évolution de HTML

#### Les débuts (1990-1997)

- **1990**: Tim Berners-Lee crée HTML comme langage de balisage pour le World Wide Web
- **1991-1993**: HTML informel, sans standard officiel
- **1995**: HTML 2.0, première spécification
- **1997**: HTML 3.2, puis HTML 4.0 rapidement après

#### La transition XHTML (1999-2004)

- **2000**: XHTML 1.0, reformulation de HTML 4 en XML
- **2001**: XHTML 1.1, modularisation de XHTML
- Le but: plus de rigueur, favoriser l'interopérabilité, préparer le Web sémantique

#### Renaissance de HTML (2004-présent)

- **2004**: Formation du WHATWG (Web Hypertext Application Technology Working Group) par des employés d'Apple, Mozilla et Opera, frustration face à la lenteur du W3C
- **2007**: Le W3C reconnaît les travaux du WHATWG, coopération
- **2008**: Premières ébauches de HTML5
- **2014**: HTML5 devient la recommandation officielle du W3C

- **2019:** Le WHATWG obtient le contrôle officiel de la spécification HTML, approche "Living Standard" (standard évolutif)

## ▼ Évolution de CSS

### Première génération (1996-2000)

- **1996:** CSS 1 créé par Håkon Wium Lie et Bert Bos
- **1998:** CSS 2 avec positionnement, z-index, media types

### Deuxième génération (2000-2010)

- **2001-2010:** Développement progressif de CSS 2.1 (correction de bugs)
- Adoption lente et inégale par les navigateurs (période difficile)

### CSS3 et au-delà (2010-présent)

- **CSS3:** Approche modulaire (plus de 50 modules séparés)
- Modules majeurs: Sélecteurs, Box Model, Backgrounds & Borders, Text Effects, 2D/3D Transformations, Animations, Flexbox, Grid
- **Custom Properties** (variables CSS)
- **Media Queries** pour le responsive design
- **CSS Houdini:** API de bas niveau pour étendre CSS

## ▼ Organismes de standardisation

### W3C (World Wide Web Consortium)

- Fondé en 1994 par Tim Berners-Lee
- Processus formel avec plusieurs étapes (Working Draft, Candidate Recommendation, Proposed Recommendation, W3C Recommendation)
- Approche par "snapshots" figés

### WHATWG (Web Hypertext Application Technology Working Group)

- Fondé en 2004

- Approche "Living Standard": spécifications constamment mises à jour
- Plus agile, reflète mieux la réalité de l'implémentation dans les navigateurs

## ▼ Ressources complémentaires

### Documentation officielle

- [MDN Web Docs](#)
- <https://developer.mozilla.org/fr/>
- <https://www.w3.org/>
- <https://html.spec.whatwg.org/>
- ⚡ [CSS Specifications](#)
- <https://tc39.es/ecma262/>

### Tutoriels et cours en ligne

- <https://www.w3schools.com/>
- <https://css-tricks.com/>
- <https://javascript.info/>
- <https://grafikart.fr/>

### Outils

- <https://validator.w3.org/>
- <https://caniuse.com/>
- <https://codepen.io/>
- <https://jsfiddle.net/>
- <https://developer.chrome.com/docs/lighthouse/overview?hl=fr>

**Protection intellectuelle :** Ce cours est intégralement protégé par le droit d'auteur conformément au Code de la propriété intellectuelle français. Toute reproduction, diffusion ou utilisation non autorisée est strictement interdite.

L'intelligence artificielle a été utilisée uniquement comme outil d'assistance pour la correction orthographique et la reformulation de certaines phrases, sous ma supervision et validation. Le contenu pédagogique et intellectuel reste entièrement ma création originale et personnelle.