

# LIFWEB - CM1

Oumar Adam ADAM

Semestre printemps 2023-2024 UCBL

## 1 Introduction

Introduction au Web et à JavaScript.

## 2 Organisation de l'UE

- Informations générales
- Liens emploi du temps
- MCC
- Programme et supports

Site web : <http://lifweb.pages.univ-lyon1.fr/>

## 3 Doctrine

- Documentation de référence
- Pré-requis variés
- Use the platform : les standards d'abord
  - <http://vanilla-js.com/>
  - <https://youmightnotneedjquery.com/>
  - <http://youmightnotneedjs.com/>
- Toute la chaîne de la conception au déploiement
- Focus sur la qualité, l'automatisation et l'efficacité

## 4 Faire

- Les exercices 0 sont à préparer avant.
- Les TPs sont à finir en intégralité hors BONUS.
  - Si vous avez le temps en séance.
  - Sinon pour approfondir avant les épreuves.
- Les assistants LLMs sont autorisés une fois qu'on sait évaluer la qualité de leurs réponses.

## 5 Programmation Web

Mousquetaires du Web. HTML5, CSS3, JavaScript et HTTP!

## 6 L'écosystème du web

Les différents métiers du Web.

- <https://roadmap.sh/frontend>
- <https://roadmap.sh/backend>
- <https://roadmap.sh/devops>
- <https://roadmap.sh/full-stack>

Beaucoup de concepts, de technique et d'évolutions.

## 7 HTTP

HTTP status code 413.

## 8 Le protocole du Web

Pile réseau du web.

## 9 Basic aspects of HTTP

- HTTP is simple
- HTTP is extensible (via headers)
- HTTP is stateless, but not sessionless

## 10 Méthodes des requêtes HTTP

Appelées aussi verbes.

- GET : get a specific resource
- POST: create a new resource
- PUT: update an existing resource (or create)
- DELETE: delete the specified resource
- HEAD: get the metadata information
- TRACE, OPTIONS, CONNECT, PATCH: avancées

## 11 Démonstrations

- Devtools (Firefox, Chrome)
- <https://httpie.io/> et <https://curl.se/>
- Analyseur de protocole <https://www.wireshark.org/>

## 12 HTML5 et CSS3

## 13 Conseils généraux

- Travaillez systématiquement en navigation privée
- Utilisez des sources de référence
- Validez vos documents HTML5/CSS3

## 14 JavaScript : Introduction

JavaScript est un langage de programmation de haut niveau, principalement utilisé pour développer des applications web interactives et dynamiques. Voici quelques caractéristiques clés de JavaScript :

- **Côté client** : JavaScript est principalement exécuté côté client dans les navigateurs web, ce qui permet d'ajouter des fonctionnalités interactives aux pages web.
- **Polyvalent** : En plus du développement web, JavaScript peut également être utilisé pour développer des applications côté client et côté serveur.

- **Interprété** : JavaScript est un langage interprété, ce qui signifie que le code est exécuté ligne par ligne par un moteur JavaScript intégré au navigateur.
- **Syntaxe inspirée de Java** : Bien que le nom puisse suggérer une relation avec Java, JavaScript est un langage indépendant avec une syntaxe légèrement inspirée de Java.
- **Programmation orientée objet et fonctionnelle** : JavaScript prend en charge les paradigmes de programmation orientée objet et fonctionnelle, offrant une grande flexibilité dans la façon dont le code est structuré.
- **Large écosystème** : JavaScript dispose d'un vaste écosystème de bibliothèques, de frameworks et d'outils qui facilitent le développement web moderne.

## 14.1 Types de données

- Cette section traite des différents types de données disponibles en JavaScript, tels que les types primitifs (`null`, `undefined`, `boolean`, `number`, `string`, `symbol`) et les objets.

## 14.2 Déclaration de variables

- Cette section couvre les différentes façons de déclarer des variables en JavaScript, notamment l'utilisation de `var`, `let`, et `const`, ainsi que la portée des variables associées à chaque méthode de déclaration. Par exemple :

```
var x = 10;
let y = 20;
const z = 30;
```

## 14.3 Boucles

- Cette section présente les différentes boucles disponibles en JavaScript, y compris `for`, `while`, et `do...while`, ainsi que les boucles spécifiques comme `for...in` et `for...of`. Par exemple :

```
for (let i = 0; i < 5; i++) {
    console.log(i);
}
```

## 14.4 Affectation par décomposition

- Cette section explique le concept d'affectation par décomposition en JavaScript, une fonctionnalité qui permet d'extraire des valeurs d'objets ou de tableaux et de les affecter à des variables distinctes. Par exemple :

```
const person = { name: 'John', age: 30 };
const { name, age } = person;
console.log(name, age);
```

## 14.5 JS est impératif / structuré

- Cette section met en évidence les caractéristiques impératives et structurées de JavaScript, telles que l'affectation de variables, les séquences d'instructions, les boucles et les conditionnelles. Par exemple :

```
let x = 10;
if (x > 5) {
  console.log('x is greater than 5');
}
```

## 14.6 JS est orienté objet

- Cette section aborde la nature orientée objet de JavaScript, en mettant en avant la création d'objets par prototypage et le rôle des fonctions en tant qu'objets de première classe. Par exemple :

```
function Person(name) {
  this.name = name;
}
const person = new Person('John');
console.log(person.name);
```

## 14.7 JS est fonctionnel

- Cette section explore le côté fonctionnel de JavaScript, en mettant l'accent sur les fonctions en tant que valeurs de première classe, les fonctions fléchées, les fermetures lexicales et l'utilisation des fonctions pour les opérations de transformation de données. Par exemple :

```
const add = (a, b) => a + b;
console.log(add(2, 3));
```

## 14.8 API fonctionnelle des tableaux

- Cette section présente l'API fonctionnelle des tableaux en JavaScript, en montrant comment utiliser les méthodes telles que `map`, `filter`, `reduce`, et `every` pour manipuler les tableaux de manière fonctionnelle. Par exemple :

```
const numbers = [1, 2, 3, 4, 5];
const doubled = numbers.map(num => num * 2);
console.log(doubled);
```

## **14.9 JS est événementiel**

- Cette section explique le caractère événementiel de JavaScript, en mettant en évidence la suspension de l'exécution des fonctions, l'adaptation aux longues actions imprévisibles et le modèle de concurrence.

## **15 Node.js**

- Cette section présente Node.js, un environnement d'exécution JavaScript côté serveur, en soulignant ses fonctionnalités et son utilisation dans le développement web.