



# Les pages web en 2024

Le format HTML  
La mise en forme avec CSS3 et  
L'interaction avec Javascript

# Organisation du cours / TD

- **Session de 30 minutes pour maîtriser un concept puis exercice de prise en main**
- **Objectif: Etre en mesure de réaliser une mise en forme au format html.**
- **Projet directeur: Créer son cv ou ses savoir faire en html**
- **Projet en TP: Creation d'un quizz qui evalue les reponses**

# Pre-requis

- **Disposer d'un PC avec un editeur de code (notepad++, visual studio, ...) et un navigateur à jour (Chrome de preference)**
- **Connaître le fonctionnement d'un PC ce qu'est un fichier texte, un logiciel.**
- **Aucun pre-requis sur le html css ou javascript**
- **Les ressources utiles :**
  - W3Schools: <https://www.w3schools.com/>: <https://www.w3schools.com/>
  - Mozilla Developer Network: <https://developer.mozilla.org/en-US/>:  
<https://developer.mozilla.org/en-US/>
  - Codecademy: <https://www.codecademy.com/>: <https://www.codecademy.com/>
  - Bootstrap: <https://getbootstrap.com/>: <https://getbootstrap.com/>
  - HTMx: <https://htmx.org/>: <https://htmx.org/>

# Sur 4 jours

## Jour 1

### HTML5 (CM: 2h)

Introduction au HTML et le protocole HTTP

Les balises/ Attributs / Elements / Tableaux /Formulaires HTML

### CSS3 (CM: 2h)

Introduction au CSS

Les sélecteurs / propriétés / positionnement / bordure / police / Arrière plan CSS

### HTML/CSS (TP: 3h)

Mise en page d'un CV ou d'un service proposé en html

Mise en page d'un questionnaire pour évaluer si vous voulez donner des informations personnel (email, photo,...) à la personne qui vous visite

La combinatoire des reponses = le nom d'une page web à afficher.

## Jour 2

### JavaScript (CM: 4h)

Introduction au JavaScript

Les variables / opérateurs / conditions / boucles / fonctions / événements / animations

### JS (TP: 3h)

Modeliser le questionnaire en format JSON

Afficher chaque question une par une et calculer un score en fonction des réponses créer une redirection vers une page contenant les bonne reponses dans son nom.html. Dereferencé la page des moteurs de recherche

## Jour 3

### Bootstrap V5 (CM:2h)

Introduction à Bootstrap V5

Les composants / grilles / formulaires / personnalisation Bootstrap

### HTMX (CM:2h)

Introduction à htmx

Les composants / interactions / ajax htmx

### Bootstrap/htmx (TP:3h)

Refaire la page du CV en créant un menu déroulant avec un menu contact qui ouvre le questionnaire.

Afficher un formulaire de contact à la suite du questionnaire

Charger le questionnaire dispo <https://>

Créer une animation parallax et un carroussel d'image sur la page principale

## Jour 4

Projet (TP:4h) Finaliser la page du CV / service avec bootstrap V5 (en créant une charte graphique code couleur, font).

Reprendre le questionnaire et afficher question par question en adaptant la presentation de chaque question :

Affichage avec une video ou une image, une question et une liste de reponses possible à cocher

Recuperer un questionnaire sur un autre serveur et calculer un score de bonne reponse et créer une page de resultat qui affiche ou pas le formulaire de contact.

### Evaluation:

Une note basée sur les reponses au quiz sur 10 et sur la qualité du projet sur 10 pour faire une note individuelle (50% quizz 50% projet collectif)

# **1. HTML 5**

# Introduction

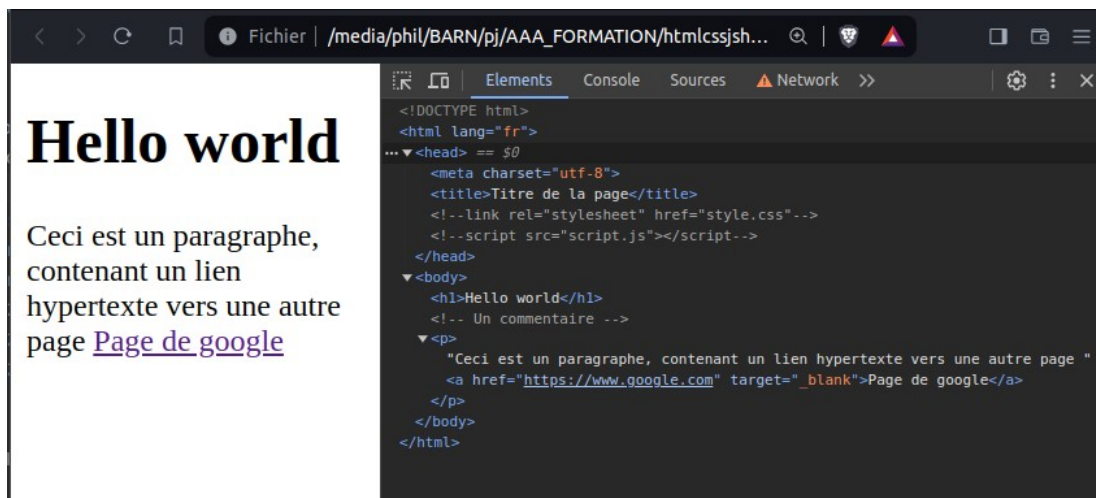
## HTML HyperText Markup Language

- **HTML n'est pas :**

- Un langage de programmation.
- Un langage de mise en forme de document

- **HTML est :**

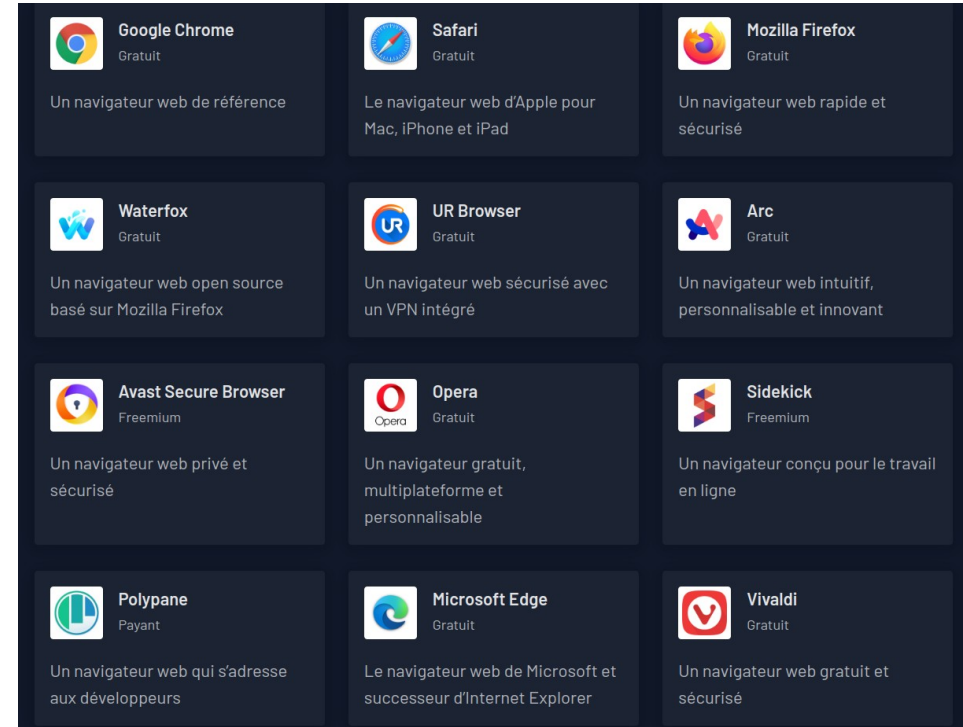
- Une organisation de contenu texte dans une page web
- Une indexation d'un ensemble de pages web
- Une déclinaison du XML qui est une représentation de données plus complexe



# Introduction

## HTML HyperText Markup Language

- **HTML est un fichier texte.**
  - Il a un encodage
  - Il a une langue
  - Il est composé de balises caractérisées par des attributs qui permettent à des logiciels appelés navigateur d'interpréter son contenu et de le mettre en forme.



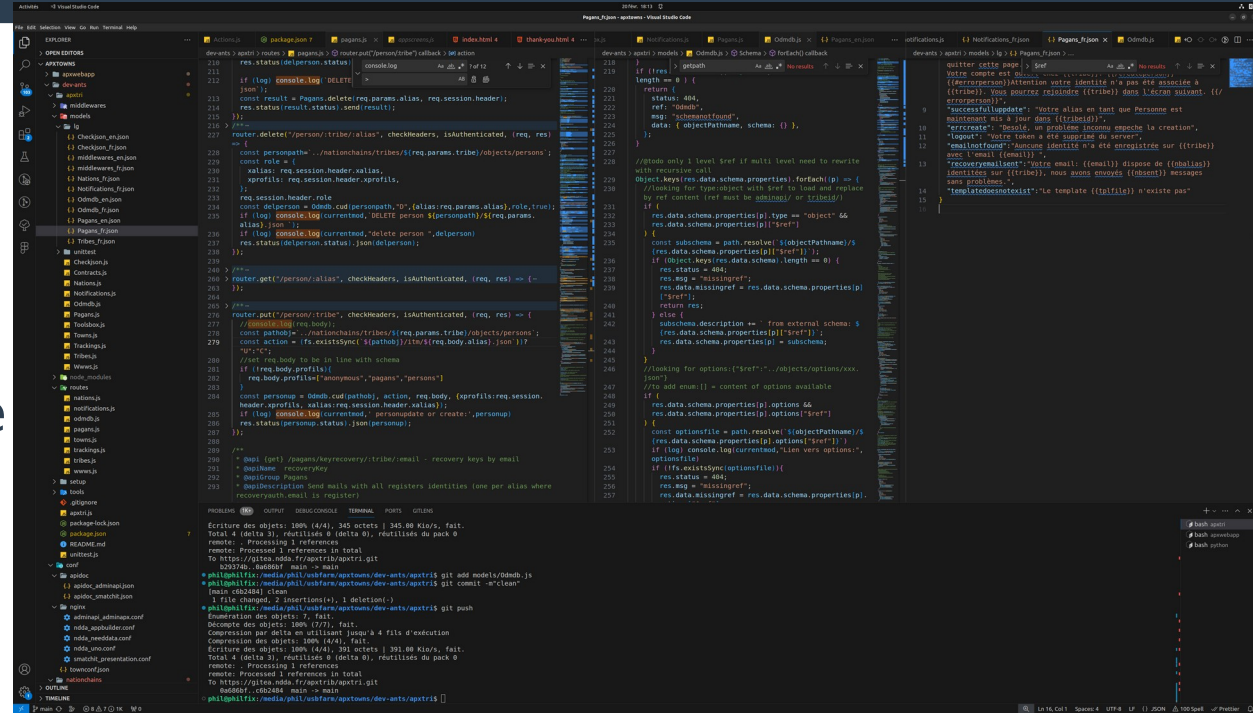
# Introduction

## HTML HyperText Markup Language

- **Un editeur de texte/code**

- Affiche le texte sans modification
- Permet d'exécuter du code
- Permet de prévisualiser le rendu html

- **Notepad++ / visual studio / bracket / ...**





# Un site ou une app web

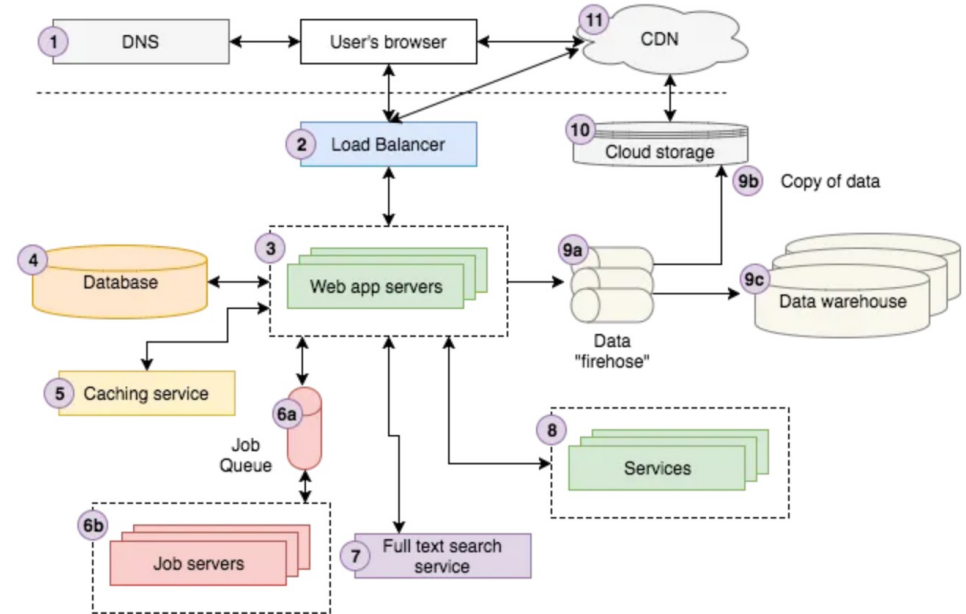
## Définition & objectifs

- **Définition d'une architecture web**

- L'architecture web est l'organisation des composants et des interactions qui constituent une application web.
- Elle définit la structure, le comportement et les interactions entre les différents éléments d'une application web.

- **Objectifs d'une architecture web**

- Fournir une base solide pour le développement et le déploiement d'applications web
- Garantir la cohérence, la fiabilité et la performance des applications web
- Faciliter l'évolutivité et l'extensibilité des applications web



Modern web application architecture overview

# Le réseau

## Les noms de domaines et les Domain Name Server

- **Sur internet IPv4 ou v6 sont difficiles à retenir et surtout peuvent être dynamiques.**

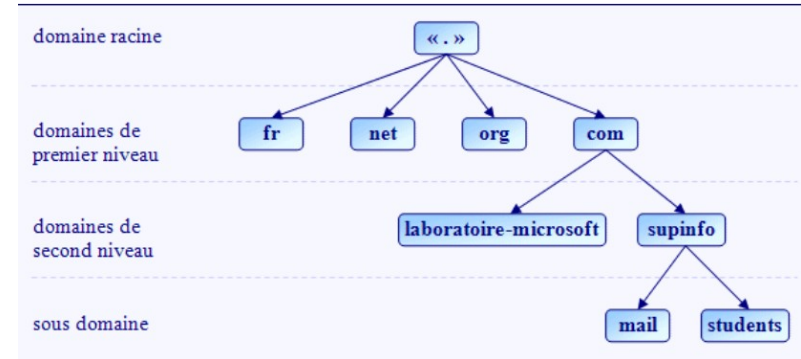
- Un annuaire hiérarchique permet de traduire un nom de domaine en adresse IPv4 ou IPv6
- Il existe 7 annuaires DNS physiques qui se synchronisent

- **Pour charger une page web 4 serveurs DNS sont impliqués**

- Recurseur DNS ([www.google.com](http://www.google.com) va être découpé en sous requête vers les 3 autres serveurs)
- Serveur de noms racine (.fr .com ...)
- Serveur de noms TLD (top level domain)
- Serveur de noms de référence (sous-domaine)

- **Des registrars gèrent les domaines racine et les TLD.**

- On passe par ovh, ionos, 1&1, ... ont des accès à ces registrars et revendent des TLD
- Ils disposent d'interface de configuration type



Accueil > Gérer > Liste de vos noms de domaine



### LISTE DE VOS NOMS DE DOMAINES

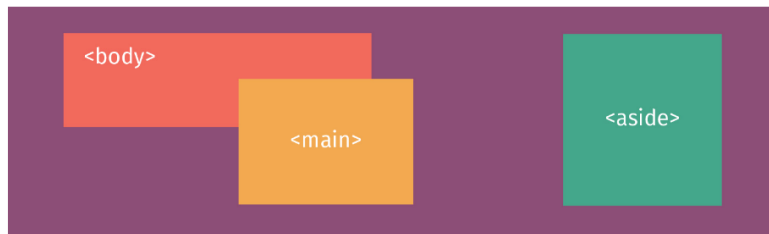
Nom de domaine	Contacts	Services	Expire le	Sélection
20 par page				
<a href="#">mesure3d.fr</a>	RATF	dnsperso	24 janvier 2023	<input type="checkbox"/>
<a href="#">mecoiffe.fr</a>	RATF	dnsperso	28 novembre 2023	<input type="checkbox"/>
<a href="#">mdb-batiment.fr</a>	RATF	dnsperso	25 février 2026	<input type="checkbox"/>
<a href="#">maison-prajault.fr</a>	RATF	dnsperso	23 février 2023	<input type="checkbox"/>
<a href="#">maison-boucault.fr</a>	RATF	dnsperso	12 juin 2023	<input type="checkbox"/>
<a href="#">maison-boucault.com</a>	RATF	dnsperso	14 avril 2023	<input type="checkbox"/>
<a href="#">maildigit.fr</a>	RATF	dnsperso	04 février 2023	<input type="checkbox"/>
<a href="#">maildigit.com</a>	RATF	dnsperso	04 février 2023	<input type="checkbox"/>
<a href="#">lucasmartin.fr</a>	RATF	dnsperso	07 novembre 2023	<input type="checkbox"/>

# Les balises

## HTML HyperText Markup Language

- Les balises permettent d'isoler des blocs de textes
- Ces balises permettent aux navigateurs d'y associer des comportements particuliers
- <https://jaetheme.com/balises-html5/>

### Liste des balises HTML5



Les balises avec des hachures sont obsolètes

<!-- -->		!DOCTYPE				
<b>a</b>						
a	abbr	acronym	address	applet	area	article
aside	audio					
<b>b</b>						
b	base	basefont	bdo	bdi	big	blockquote
body	br	button				
<b>c</b>						
canvas	caption	center	cite	code	col	colgroup
command						
<b>d</b>						
datalist	dd	del	details	dfn	dir	div
dl	dt					
<b>e</b>						
em	embed					
<b>f</b>						
fieldset	figcaption	figure	font	footer	form	frame

# Les balises

## HTML HyperText Markup Language

- **Règle générale : une balise s'ouvre et se ferme pour encapsuler un contenu**

- `<h1>titre <a></a></h1>`

- `<h1> titre <a></h1></a>`

- **La racine de base**

- `<html>`

- **Les metadata**

- `<base> <head> <link> <meta> <style> <title>`

- **La racine de section**

- `<body>`

- **Contenu**

- `<article> <aside> <footer> <header> <hx> <main> <nav> <section>`

- **Contenu textuel en bloc**

- `<div> <hr> <ul> <li> <menu> <ol> <p> <pre>`

- **Contenu texte en ligne**

- `<a> <b> <i> <s> <small> <span> <time> <br> <data> <code> ...`

- **Contenu images et medias**

- `<img> <audio> <video> <area> <map> <track>`

- **Contenu embarqué**

- `<embed> <objects> <portal> <iframe>`

- **Contenu svg et mathML**

- `<svg> <matlm>`

- **Scripts**

- `<canvas> <noscript> <script>`

- **Contenu tabulaire**

- `<table> <th> <tr> <td> <caption> <col> <tfoot>`

- **Formulaire**

- `<form> <input> <button> <select> <option> <textarea>`

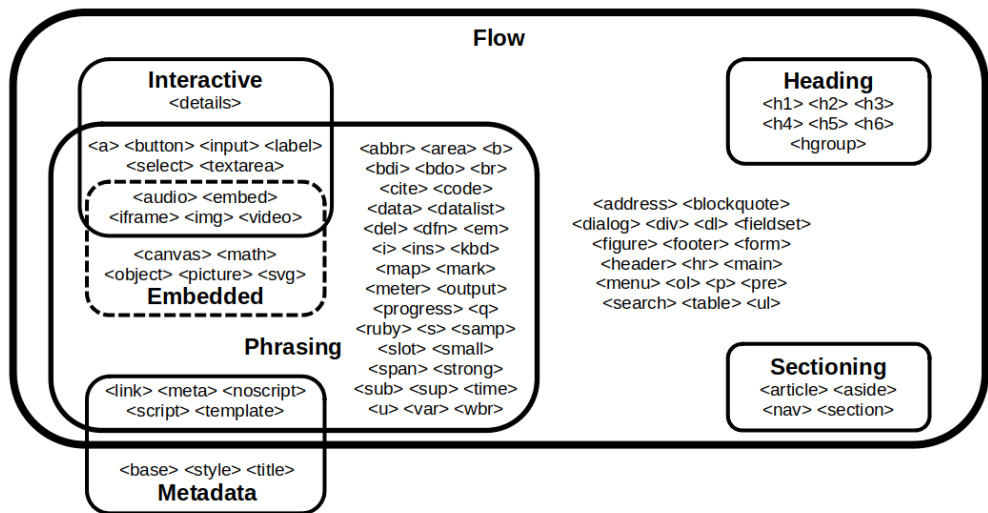
- **Elements interactifs**

- `<details> <summary> <dialog>`

- **Composant web (web components)**

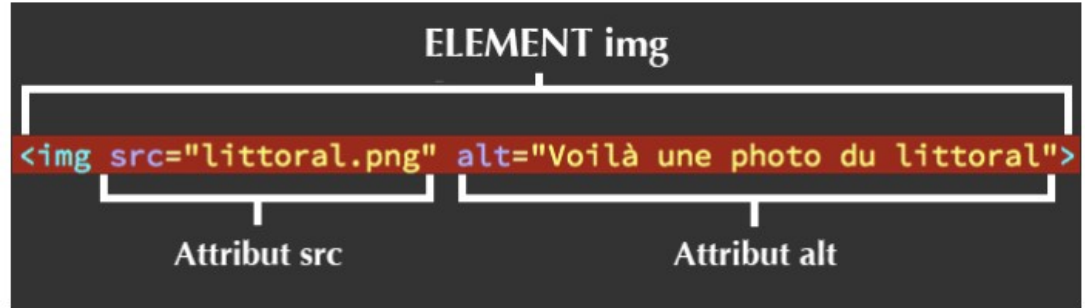
- `<slot> <template>`

### HTML element content categories



# Les attributs

HTML HyperText Markup Language



- Pour qualifier des balises, une balise a une liste d'attributs réservés :
- `<tag attribute="valeur"></tag>`
- Les plus utilisés :
  - id : identifiant unique sur la page
  - class : pour typer un élément via une class css
  - src : indique la source de l'élément
  - data-xx : personnalisable utile dans des programme
  - alt, placeholder : utile pour remplacer un élément inaccessible ou en modèle
  - width, height, size, rows, color, mise en forme de balise

# Les Elements

HTML

## Hello world

Ceci est un paragraphe,  
contenant un lien  
hypertexte vers une autre  
page [Page de google](https://www.google.com)

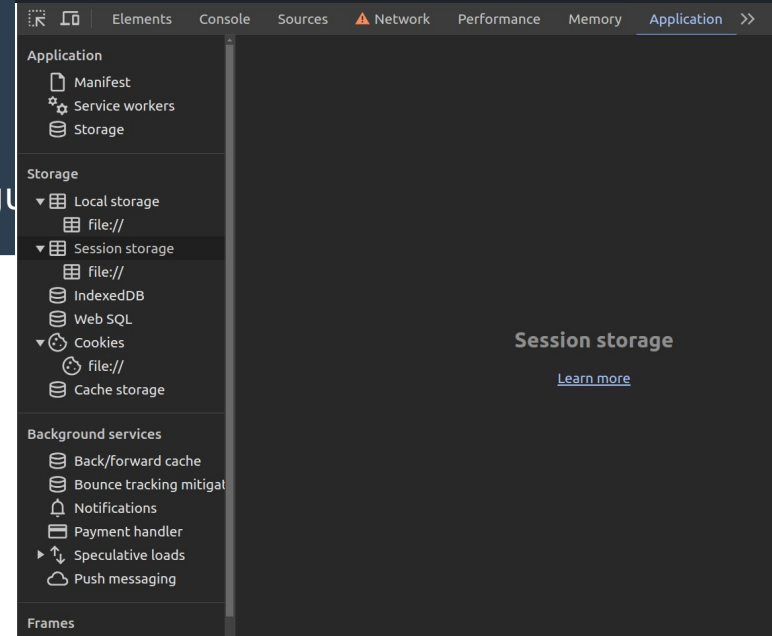
```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="utf-8">
    <title>Titre de la page</title>
    <!-- link rel="stylesheet" href="style.css" -->
    <!-- script src="script.js" -->
  </head>
  <body>
    <h1>Hello world</h1>
    <!-- Un commentaire -->
    <p>
      "Ceci est un paragraphe, contenant un lien hypertexte vers une autre page "
      <a href="https://www.google.com" target="_blank">Page de google</a>
    </p>
  </body>
</html>
```

- Une page = ensemble d'éléments :
- Ouvrez dans votre navigateur <https://opinion.ndda.fr/pagebase.html>
- Enregistrez en local le fichier pagebase.html
  - Affichez le dans votre explorateur de fichier
  - ouvrez le dans votre editeur de texte
  - puis ouvrez le dans votre navigateur
- Modifier Hello world par hello {votre prenom}
  - Class : pour typer un element via une class css
- Passer en mode dev votre navigateur F12 ou click droit et inspecter

# Les navigateurs

HTML HyperText Markup Language

- **Le mode développeur est l'outil de debugage**
  - Éléments : visualise le code html
  - Console : permet de voir les erreurs et les traces `console.log('hello')` dans un programme javascript
  - Sources : visualise tous les fichiers en ligne
  - Network : liste et trace les requetes http
  - Application : stockage de données en local



# Les bases

## HTML HyperText Markup Language

- Les titres `<h1>` `<h6>` `<h1> titre</h1>`
- Les paragraphes `<p>blabla</p>`
- Les blocs `<p>Texte<span class='rouge'>texte en rouge</span></p>`
- Les retours à la ligne : `<p> texte sur ligne1<br> texte sur ligne2</p>`
- Les barres horizontale `<hr>`
- Mise en forme de base : `<b>` gras, `<s>` barré `<i>` italic
- Les espaces ne sont pas affichés sauf `&nbsp;` ( existe double ou quadruple espace)

- **Exercice :**

- Rédiger une présentation de qui vous êtes professionnellement (ou voulez être) avec un titre, et un ou 2 paragraphes avec mise en forme.



# Les tableaux

HTML HyperText Markup Language

- **Les tag clés : table, caption, tbody, tr (ow) , th (head) td (cellule)**

**<table>**

```
<caption>Titre</caption>
```

```
<tr><th>Titre col1</th><th>Titre  
col2</th></tr>
```

```
<tr><td>Cellule col1 ligne 1</td>
```

```
    <td> Cellule col2 ligne1</td>
```

```
</tr>
```

**</table>**

- **Exercice :**

- Créer un tableau qui liste vos compétences avec 2 colonnes. La première avec le nom de la compétence et dans la deuxième votre niveau (débutant, maîtrise, expert)

# Les formulaires

## HTML HyperText Markup Language

- Les tag clés : form, input, textarea, select/options, button

```
<form action="/ma-page-de-traitement" method="post">
<ul>
  <li>
    <label for="name">Nom&nbsp;&nbsp;&nbsp;</label>
    <input type="text" id="name" name="user_name" />
  </li>
  <li>
    <label for="mail">E-mail&nbsp;&nbsp;&nbsp;</label>
    <input type="email" id="mail" name="user_mail" />
  </li>
  <li>
    <label for="msg">Message&nbsp;&nbsp;&nbsp;</label>
    <textarea id="msg" name="user_message"></textarea>
  </li>
</ul>
<div class="button">
  <button type="submit">Envoyer le message</button>
</div>
</form>
```

- **Exercice :**

- Créer un formulaire de contact, nom , email, message et un select avec 2 options (personnel ou professionnel)
- Modifier le select pour le rendre ‘multiselectionnable’

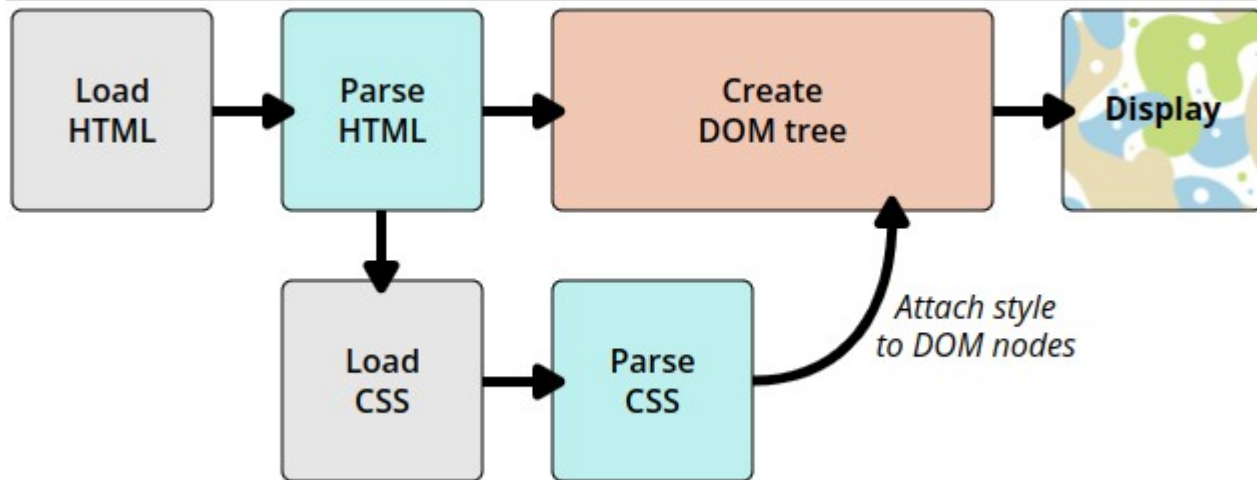
## **2. CSS 3**

# Le CSS

## Cascading Style Sheets

### Mise en forme d'élément HTML (ou XML)

- 2 possibilités pour ajouter du style
- `<style></style>` dans le `<head>` ou un lien `<link rel= 'stylesheet' href='style.css'>`



```
<p>
  Let's use:
  <span>Cascading</span>
  <span>Style</span>
  <span>Sheets</span>
</p>
```

```
p
├ "Let's use:"
├ SPAN
│   └ "Cascading"
├ SPAN
│   └ "Style"
└ SPAN
    └ "Sheets"

span {
  border: 1px solid black;
  background-color: lime;
}
```

Let's use: **Cascading Style Sheets**

### Style sheet (interne ou externe)

```
Selecteur {  
  directive : valeur;  
}
```

### ou Inline

```
<h1 style="color:blue;background-  
color:yellow;border:1px solid  
green ;"> Texte mis en forme</h1>
```

- **Exercice :**

- Modifier vos titres et vos paragraphes (utiliser span pour modifier des morceaux de texte)

# Le css / les selecteurs

HTML HyperText Markup Language

## Selecteur {directive : valeur;}

**#identifiant** => <div id='identifiant'></div>

**.reponse** => <div class='reponse'></div>

**p** => <p></p>

**a[title]** => <a title= »qq chose »></a>

**input[name='email']** => <input name='email'>

**p , h1** => <h1></h1> et <p></p>

**a:hover** => <a></a> au passage du curseur

**p::fisrt-line** => <p>blabla

lala</p> selectionne la 1ere ligne

**article > p** => <article><p></p></article> uniquement le 1<sup>er</sup> p à l'interieur de article.

**section p** => tous les p de la section

Sélectionne un tag et applique les directives à l'ensemble des éléments enfants.

Voir la table de reference des selecteurs

[https://developer.mozilla.org/fr/docs/Learn/CSS/Building\\_blocks/Selectors](https://developer.mozilla.org/fr/docs/Learn/CSS/Building_blocks/Selectors)

## • Exercice :

- Ajouter des identifiants et des class pour appliquer des styles identiques sur plusieurs sélecteurs et évaluer l'impact des sélecteurs

# Le css / les directives

HTML HyperText Markup Language

Selecteur **{directive : valeur;}**

- Mise en forme de texte

[https://developer.mozilla.org/fr/docs/Learn/CSS/Styling\\_text](https://developer.mozilla.org/fr/docs/Learn/CSS/Styling_text)

- Directive display / position / float le positionnement

[https://developer.mozilla.org/fr/docs/Learn/CSS/CSS\\_layout](https://developer.mozilla.org/fr/docs/Learn/CSS/CSS_layout)

**Préparation d'une charte graphique :**

- Code couleur <https://paletton.com>
- Font <https://fonts.google.com>

- **Exercice :**

- Préparer une mise en forme de page web avec des lignes et des colonnes

# Le css / les directives

HTML HyperText Markup Language

Selecteur {directive : valeur;}

- Texte css: text-align, line-height, letter-spacing, color, margin, padding, text-shadow
- Animations (si changement alors délai)  
#elt {transition : margin-left 5s;}
- Box-sizing (taille de bloc <> height width  
\* { box-sizing : border-box;}
- Positionnement  
De base : utiliser weight, height, max-width avec margin et padding  
Complexe : Utiliser principalement display :flex (bloc) ou display:inline (forcer la ligne)
- Utile avec js :  
{ display : none ; visibility:hidden;}
- Responsive design :  
html : <meta name="viewport" content="width=device-width, initial-scale=1">  
css : @media all and (min-width: 760px) and (max-width: 959px) {}
- Fonts  
<html><head> : <link href="http://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet" type="text/css" />  
CSS : html { font-size:10px ; font-family : "Open Sans", sans-serif ; }

## • Exercice :

- Installer plusieurs fonts sur un site
- Choisir une police pour les <h> et une autre pour le <p> <a> <li> <ul>





**Javascript**

# Les bases du js : variables

Atelier : Les variables basejs.js

- **Une variable permet de stocker une valeur**

- **Declaration**

**let** mavariable='texte' ; variable dont la portée est celle du bloc courant (lecture/écriture)

**const** mavariable='texte' ; constante dont la portée est celle du bloc (lecture uniquement)

**var** mavariable='texte' ; variable globale

Avant ES6 var

Après ES6 let const (variable scoppées moins de risque d'erreurs)

- Const objet = {} n'est pas modifiable mais ses propriétés le sont ex : objet.key=val

Variable	Explication	Exemple
<a href="#">Chaîne de caractères</a>	Une suite de caractères connue sous le nom de chaîne. Pour indiquer que la valeur est une chaîne, il faut la placer entre guillemets.	<pre>let myVariable = 'Bob';</pre>
<a href="#">Nombre</a>	Un nombre. Les nombres ne sont pas entre guillemets.	<pre>let myVariable = 10;</pre>
<a href="#">Booléen</a>	Une valeur qui signifie vrai ou faux. true / false sont des mots-clés spéciaux en JS, ils n'ont pas besoin de guillemets.	<pre>let myVariable = true;</pre>
<a href="#">Tableau</a>	Une structure qui permet de stocker plusieurs valeurs dans une seule variable.	<pre>let myVariable = [1, 'Bob', 'Étienne', 10]; Référez-vous à chaque élément du tableau ainsi : myVariable[0], myVariable[1], etc.</pre>
<a href="#">Objet</a>	À la base de toute chose. Tout est un objet en JavaScript et peut être stocké dans une variable. Gardez cela en mémoire tout au long de ce cours.	<pre>let myVariable = document.querySelector('h1'); tous les exemples au dessus sont aussi des objets.</pre>

# Les bases du js : Objets standards

[https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global\\_Objects](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global_Objects)

## Objets globaux standards (par catégorie)

### Propriétés - valeurs

Les propriétés globales renvoient une valeur simple, elles ne possèdent aucune propriété méthode :

- [Infinity](#)
- [NaN](#)
- [undefined](#)
- le littéral [null](#)
- [globalThis](#)

### Propriétés - fonctions

Les fonctions globales, appelées globalement (et non par rapport à un objet), renvoient directement leur résultat à l'objet appelant.

- [eval\(\)](#)
- [uneval\(\)](#) ▲
- [isFinite\(\)](#)
- [isNaN\(\)](#)
- [parseFloat\(\)](#)
- [parseInt\(\)](#)
- [decodeURI\(\)](#)
- [decodeURIComponent\(\)](#)
- [encodeURIComponent\(\)](#)
- [encodeURIComponent\(\)](#)
- [escape\(\)](#) 🗑
- [unescape\(\)](#) 🗑

# Les bases du js : Objets standards

[https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global\\_Objects](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global_Objects)

## Objets fondamentaux

Ces objets sont les objets fondamentaux de JavaScript. Parmi ces objets, on retrouve les objets génériques, les fonctions et les erreurs.

- [Object](#)
- [Function](#)
- [Boolean](#)
- [Symbol](#)
- [Error](#)
- [EvalError](#)
- [InternalError](#) ⚠
- [RangeError](#)
- [ReferenceError](#)
- [StopIteration](#)
- [SyntaxError](#)
- [TypeError](#)
- [URIError](#)

## Données structurées

Ces objets permettent de représenter et de manipuler des tampons de données (*buffers*) et des données utilisant la notation JSON (JavaScript Object Notation).

- [ArrayBuffer](#)
- [SharedArrayBuffer](#) ⚠
- [Atomics](#) ⚠
- [DataView](#)
- [JSON](#)

## Objets de contrôle d'abstraction

- [Promise](#)
- [Generator](#)
- [GeneratorFunction](#)
- [AsyncFunction](#) ⚠

## Nombres et dates

Ces objets permettent de manipuler les nombres, dates et calculs mathématiques.

- [Number](#)
- [BigInt](#)
- [Math](#)
- [Date](#)

## Manipulation de textes

Ces objets permettent de manipuler des chaînes de caractères.

- [String](#)
- [RegExp](#)

Voir le site <https://regex101.com>

Et d'autres :  
Collection indexées  
Collection avec clés  
Introspection  
Internationalisation  
webassembly

# Les bases du js : Expressions & opérateurs

Atelier : basejs.js

<https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Operators>

- Expressions primaires
  - Expression « vers la gauche »
- Incrementation et décrémentation
  - Opérateurs unitaires
- Opérateurs arithmétiques
  - Opérateurs relationnels
  - Opérateurs d'égalité
- Opérateurs de décalage binaires
  - Opérateurs binaires booléens
  - Opérateurs logiques
- Opérateur conditionnel ternaire
  - Opérateur d'affectation

# Les bases du js : Instructions et déclarations

Atelier : Les variables basejs.js

<https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Statements>

Contrôle de flux (bloc, break, if..then, try...catch)

Déclaration de variable (vu au début)

Fonctions et classes

Itération

Autres (export, import)

# Les bases du js : Functions , Class, Error

Atelier : basejs.js

<https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Functions>

<https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Classes>

<https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Errors>

# Exercice js

- **Exercice :**

- Créer 2 input pour saisir des entiers
- Créer 3 boutons + - \* au click appliquer l'opération et afficher le resultat dans un paragraphe.

- **Exercice :**

- Créer une variable questionnaire =  

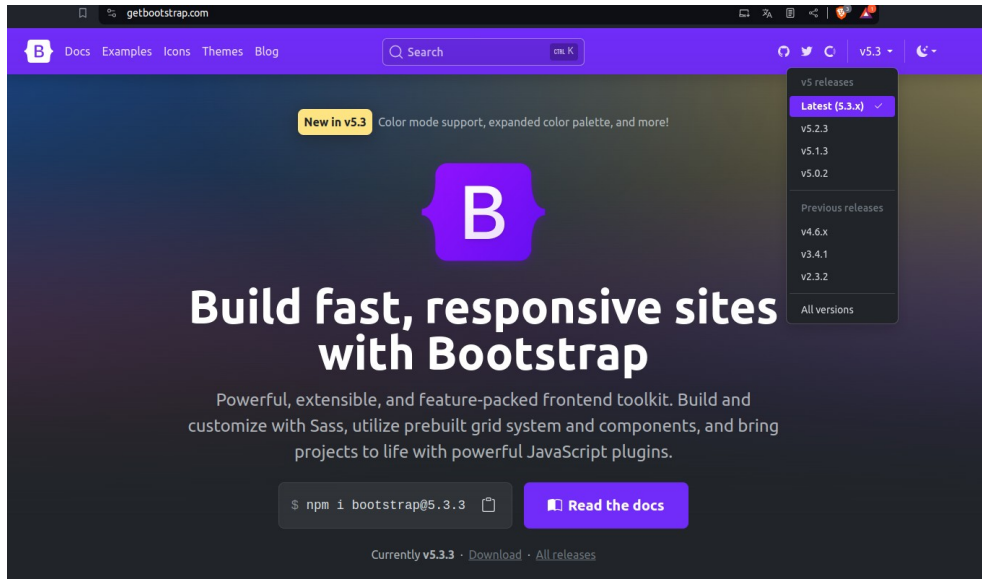
```
[{  
  question : 'question1',  
  reponses:[{id:1,reponse : 'Vrai'},  
             {id:2,reponse : 'faux'}]  
}]
```
- Afficher une par une la question avec les boutons
- Au click alimenter une variable reponses=[] avec le code de la reponse clickée puis passer à la question suivante.





**Bootstrap V5**

# Bootstrap : Une usine à web



The screenshot shows the Bootstrap website homepage. The navigation bar includes links for Docs, Examples, Icons, Themes, and Blog, along with a search bar and a version selector set to v5.3. The main heading reads "Build fast, responsive sites with Bootstrap". Below this, a subheading states: "Powerful, extensible, and feature-packed frontend toolkit. Build and customize with Sass, utilize prebuilt grid system and components, and bring projects to life with powerful JavaScript plugins." A code snippet shows the command to install Bootstrap via npm: `$ npm i bootstrap@5.3.3`. A button labeled "Read the docs" is also present. A sidebar on the right lists "v5 releases" with "Latest (5.3.x)" selected, and "Previous releases" including v4.6.x, v3.4.1, and v2.3.2. A footer note indicates the current version is v5.3.3 with links to download and all releases.

getbootstrap.com

Docs Examples Icons Themes Blog

Search

New in v5.3 Color mode support, expanded color palette, and more!

**B**

## Build fast, responsive sites with Bootstrap

Powerful, extensible, and feature-packed frontend toolkit. Build and customize with Sass, utilize prebuilt grid system and components, and bring projects to life with powerful JavaScript plugins.

```
$ npm i bootstrap@5.3.3
```

[Read the docs](#)

Currently **v5.3.3** · [Download](#) · [All releases](#)

v5 releases

- Latest (5.3.x) ✓
- v5.2.3
- v5.1.3
- v5.0.2

Previous releases

- v4.6.x
- v3.4.1
- v2.3.2

All versions



### Install via package manager

Install Bootstrap's source Sass and JavaScript files via npm, RubyGems, Composer, or Meteor. Package managed installs don't include documentation or our full build scripts. You can also [use any demo from our Examples repo](#) to quickly jumpstart Bootstrap projects.

```
$ npm install bootstrap@5.3.3
```

```
$ gem install bootstrap -v 5.3.3
```

[Read our installation docs](#) for more info and additional package managers.



### Include via CDN

When you only need to include Bootstrap's compiled CSS or JS, you can use [jsDelivr](#). See it in action with our simple [quick start](#), or [browse the examples](#) to jumpstart your next project. You can also choose to include Popper and our JS [separately](#).

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/"
```

```
<script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dis"
```

# Sass / Less : Preprocesseur scss

## Include all of Bootstrap's Sass

Import one stylesheet and you're off to the races with every feature of our CSS.

```
// Variable overrides first
$primary: #900;
$enable-shadows: true;
$prefix: "mo-";

// Then import Bootstrap
@import "../node_modules/bootstrap/scss/bootstrap";
```

Learn more about our [global Sass options](#).

## Include what you need

The easiest way to customize Bootstrap—include only the CSS you need.

```
// Functions first
@import "../node_modules/bootstrap/scss/functions";

// Variable overrides second
$primary: #900;
$enable-shadows: true;
$prefix: "mo-";

// Required Bootstrap imports
@import "../node_modules/bootstrap/scss/variables";
@import "../node_modules/bootstrap/scss/variables-dark";
@import "../node_modules/bootstrap/scss/maps";
@import "../node_modules/bootstrap/scss/mixins";
@import "../node_modules/bootstrap/scss/root";

// Optional components
@import "../node_modules/bootstrap/scss/utilities";
@import "../node_modules/bootstrap/scss/reboot";
@import "../node_modules/bootstrap/scss/containers";
@import "../node_modules/bootstrap/scss/grid";
@import "../node_modules/bootstrap/scss/helpers";
@import "../node_modules/bootstrap/scss/utilities/api";
```

- **Une video pour en savoir plus sur les pre-processeurs css (7mn) :**

– <https://www.youtube.com/watch?v=DPVGByyOGqA>

# Css : Gestion de variables css --



## Build and extend in real-time with CSS variables

Bootstrap 5 is evolving with each release to better utilize CSS variables for global theme styles, individual components, and even utilities. We provide dozens of variables for colors, font styles, and more at a `:root` level for use anywhere. On components and utilities, CSS variables are scoped to the relevant class and can easily be modified.

[Learn more about CSS variables](#) →

### Using CSS variables

Use any of our [global :root variables](#) to write new styles. CSS variables use the `var(--bs-variablename)` syntax and can be inherited by children elements.

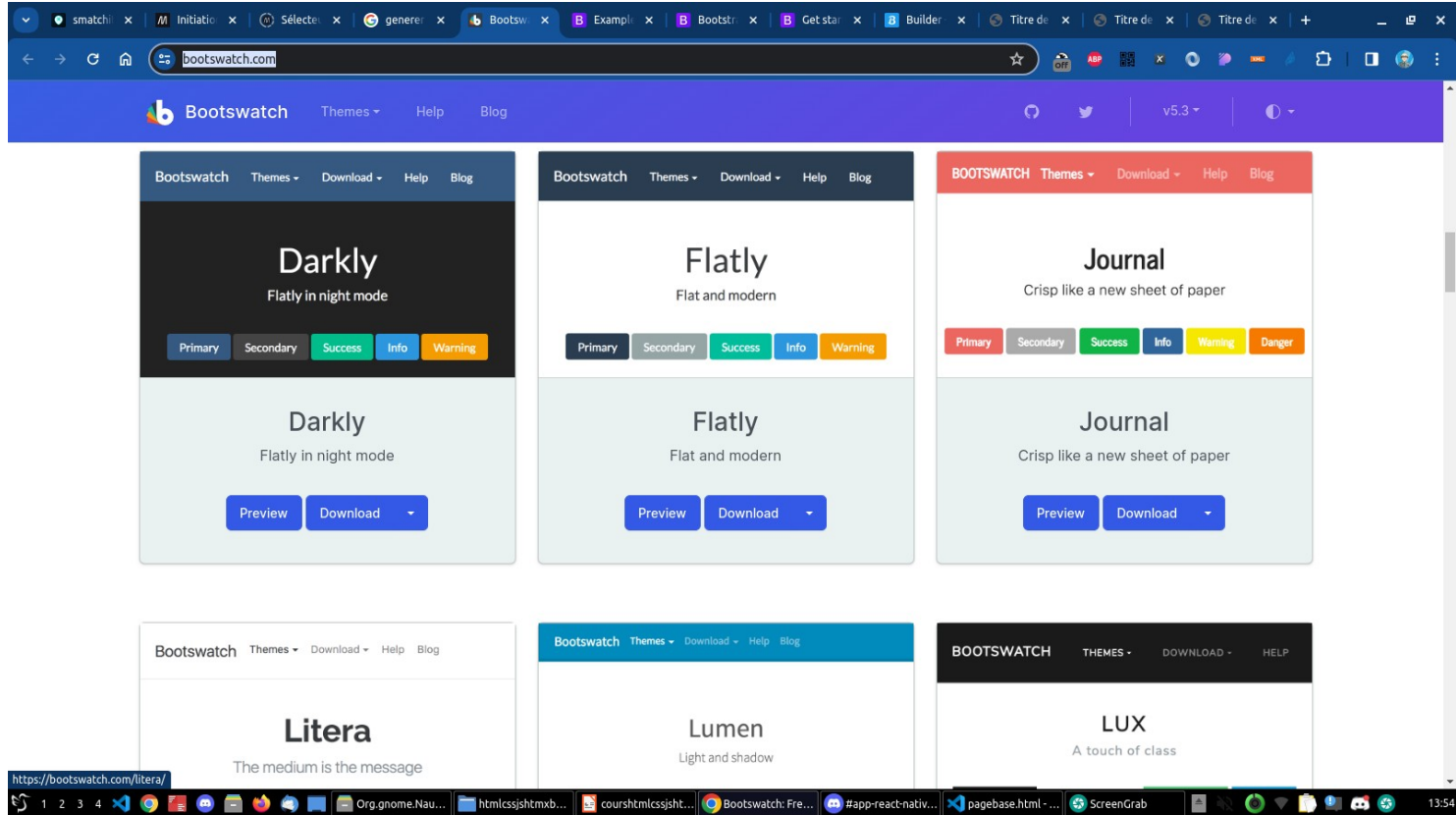
```
.component {  
  color: var(--bs-gray-800);  
  background-color: var(--bs-gray-100);  
  border: 1px solid var(--bs-gray-200);  
  border-radius: .25rem;  
}  
  
.component-header {  
  color: var(--bs-purple);  
}
```

### Customizing via CSS variables

Override global, component, or utility class variables to customize Bootstrap just how you like. No need to redeclare each rule, just a new variable value.

```
body {  
  --bs-body-font-family: var(--bs-font-monospace);  
  --bs-body-line-height: 1.4;  
  --bs-body-bg: var(--bs-gray-100);  
}  
  
.table {  
  --bs-table-color: var(--bs-gray-600);  
  --bs-table-bg: var(--bs-gray-100);  
  --bs-table-border-color: transparent;  
}
```

# Prendre un theme tout fait



# Installer et personnaliser bootstrap

## 1. Installation de Bootstrap v5:

- Téléchargez la dernière version de Bootstrap depuis <https://getbootstrap.com/docs/5.0/getting-started/download/>.
- Décompressez le fichier téléchargé.
- Copiez les fichiers CSS et JavaScript de Bootstrap dans votre projet Web.
- Ajoutez les liens vers les fichiers CSS et JavaScript dans votre page HTML.

## 2. Personnalisation des variables CSS:

- Créez un fichier CSS personnalisé dans votre projet Web.
- Importez les variables Bootstrap depuis le fichier \_variables.scss.
- Définissez les valeurs personnalisées pour les variables que vous souhaitez modifier.
- Importez votre fichier CSS personnalisé dans votre fichier CSS principal.

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Exemple Bootstrap v5</title>

  <link rel="stylesheet" href="bootstrap/css/bootstrap.min.css">

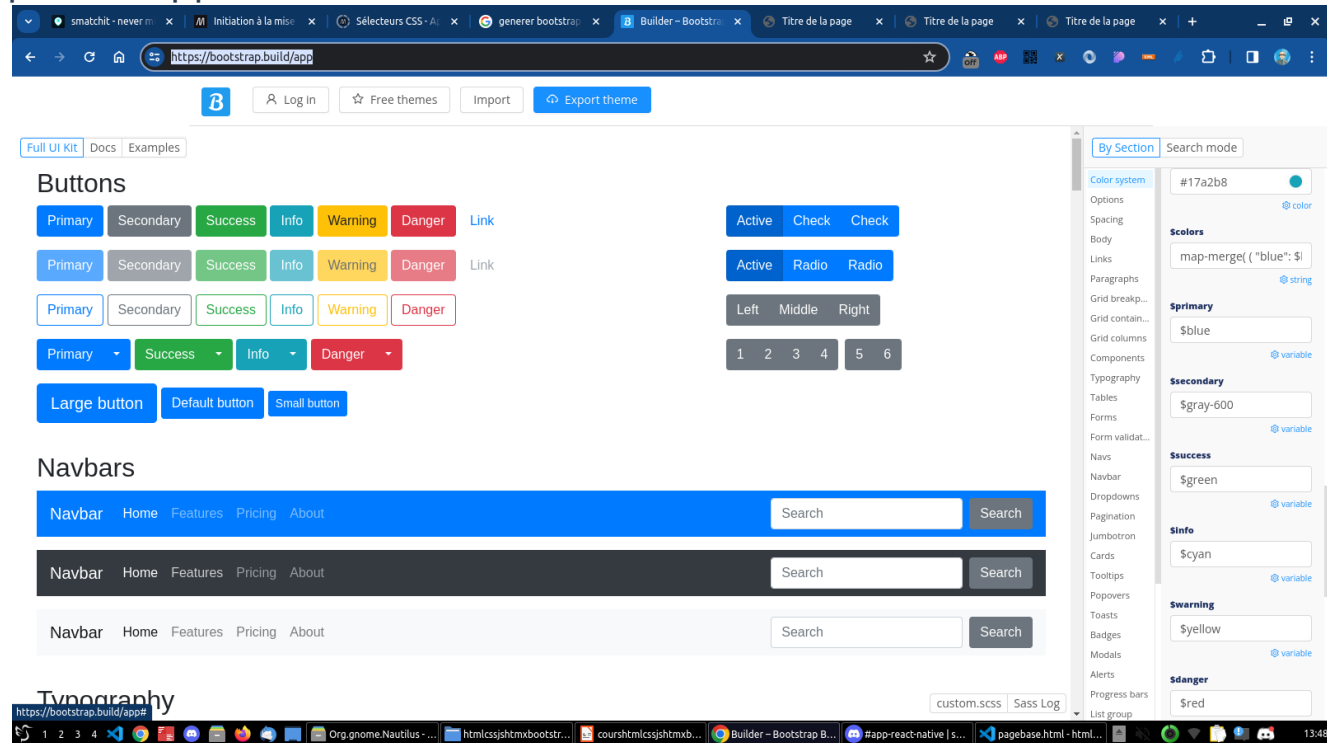
  <link rel="stylesheet" href="style.css">
</head>
<body>
  <h1>Exemple d'utilisation de Bootstrap v5</h1>

  <script src="bootstrap/js/bootstrap.min.js"></script>
</body>
</html>
```

# Installer et personnaliser bootstrap

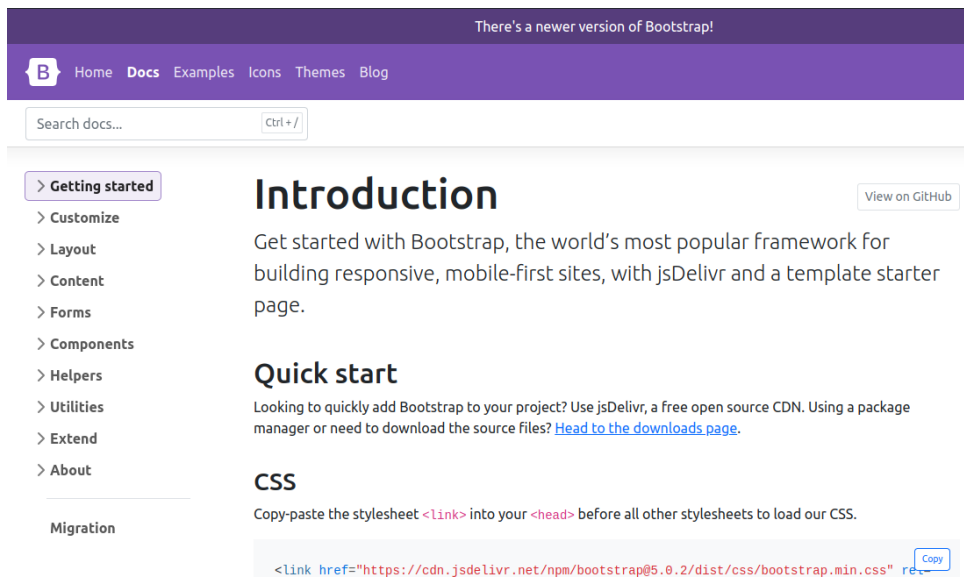
## Un generateur de thème en ligne :

- <https://bootstrap.build/app>



# Tester les outils bootstrap

Tester les outils bootstrap :  
<https://getbootstrap.com/docs/5.0/getting-started/introduction/>







**TP**

# TP livrable

- **Envoyer à [mvai.phc@ndda.fr](mailto:mvai.phc@ndda.fr) un zip nommé nom\_prenom.zip contenant :**
  - /index.html (votre CV)
  - /css/bootstrap.css
  - /css/custom.css
  - /js/main.js votre code javascript
  - /bootstrap.bundle.js
  - / {pagecontact}.html
- **Critères d'évaluation :**
  - 40 % si le site est opérationnel
  - 10 % sur le bouton brute force
  - 50 % sur la qualité du code (lisibilité et gestion des erreurs)

# TP : Personnalisation de bootstrap

- **Objectif:**

- Générer 2 fichiers style.css et bootstrap.min.js autonome de bootstrap et personnalisé à des codes couleurs

- **Process :**

- Installer node.js npm et nvm
- Installer bootstrap et sass:  
npm init  
npm install [bootstrap@5](#)  
npm install -g sass
- Créer un repertoire /css et /scss

# TP : Création d'une page web

```
$primary: #007bff;  
$secondary: #6c757d;  
$success: #28a745;  
$info: #17a2b8;  
$warning: #ffc107;  
$danger: #dc3545;  
$light: #f8f9fa;  
$dark: #212529;  
$font-family-base: 'Roboto', sans-serif;  
$font-family-google: 'Montserrat', sans-serif;  
  
@import  
"../node_modules/bootstrap/scss/bootstrap.scss"
```

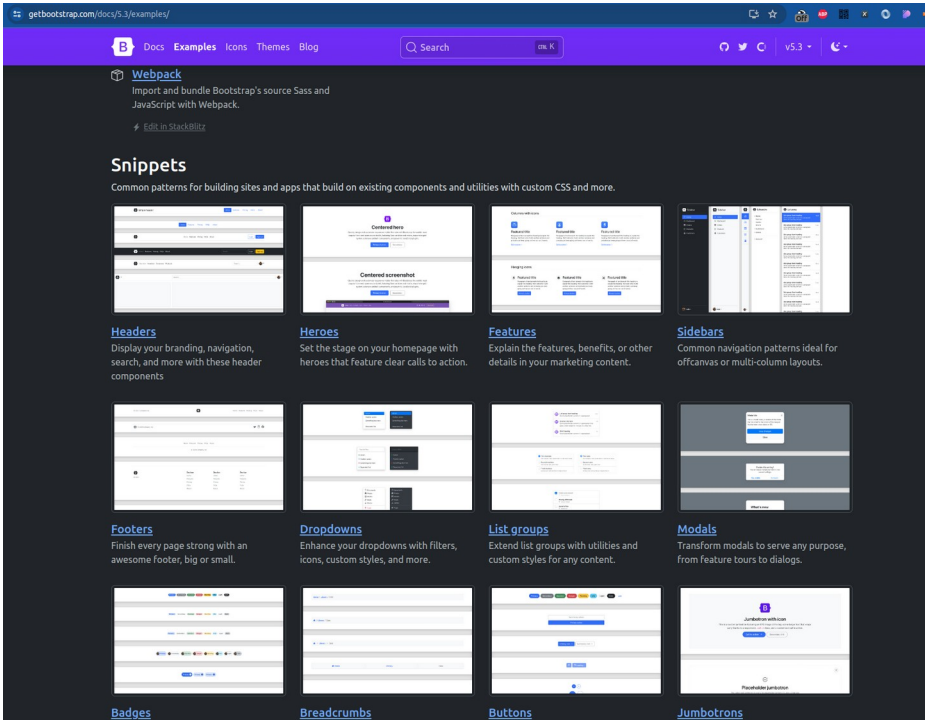
- **Choix :**

- Choisir une font et l'ajouter dans la page html avec  
`<link href="https://fonts.googleapis.com/css2?family=Montserrat:wght@300;400;500;700&display=swap" rel="stylesheet">`
- Créer scss/perso.scss
- `sass scss/perso.scss css/bootstrap.css`

# TP : Installer & tester bootstrap

## Modifier la page html

- Lien css, js, fonts
- Copier coller des composants bootstrap dans votre page pour tester leur rendu
- Créer votre page personnalisée (CV ou portfolio ou passion ...)



# TP : Rendre interactif votre page avec js

- **Reprendre les exercices du cours :**

- Créer une section avec la class cours
- Implementer les exercices avec bootstrap

- **Exercice :**

- Créer 2 input pour saisir des entiers
- Créer 3 boutons + - \* au click appliquer l'opération et afficher le resultat dans un paragraphe.

- **Exercice :**

- Créer une variable questionnaire =  

```
[{  
  question : 'question1',  
  reponses:[{id:1,reponse : 'Vrai'},{id:2,reponse : 'faux'}]  
}]
```
- Afficher une par une la question avec les boutons
- Au click alimenter une variable reponses=[] avec le code de la reponse clickée puis passer à la question suivante.

# Travaux pratique : Page

- **Créer une page web avec son CV ou une passion** sans afficher de données personnelles (pas de noms) en utilisant les composants de bootstrap (accordion, image, slideshow, ...)
- **Ajouter un bouton « Me contacter »** avec un texte qui explique que pour vous contacter vous devez répondre de façon correcte à une liste de questions.
- **Créer un fichier js** contenant une constante questionnaire du type [{qlabel : 'label de la question',qid:1,reponses:[{rlabel:'label de la reponse1',rid:1},{rlabel:'label de la reponse2',rid:2}]]]. Il n'y a qu'une bonne réponse à choisir parmi les réponses
- **Ecrire un code qui va boucler sur la liste des questions** pour poser la question (avec un template string) et injecter le code html de la question avec la liste des réponses sous forme de boutons clickables dans votre page
- **Ecrire une fonction evenement au « clique »** sur un bouton réponse va concaténer chaque réponse dans une variable reponses=A{qid}\_{rid}... et qui affiche la question suivante
- **Créer une autre page web** au même endroit que la page principale avec un nom qui correspond aux bonnes reponses, chaque groupe de question commence par A{qid}\_{rid}... .html Toujours avec bootstrap créer un formulaire de contact (message, email, nom, prenom) qui va utiliser la fonction mailto pour ouvrir la messagerie client et pre-renseigner le message pour envoi à votre adresse email.
- **Revenir sur la page precedente ajouter à la fin du questionnaire le teste de l'existence ou pas de la page {reponses}.html** si elle existe rediriger le navigateur vers cette page, si elle n'existe pas afficher un message que « suite à vos réponses vous ne souhaitez pas être contacter »
- **Créer un bouton brute force** qui déclenche une fonction dans votre page CV pour hacker la bonne réponse et rediriger vers la page contact (brute force)

# TP : Ecrire du css

**La notion de  
!default !important**

- **Exercice :**
  - Avec une ligne de css transformer vos boutons avec des coins carrés
  - Ajouter un bouton pour changer dynamiquement la presentation



# TP : Quelques solution

## Tester si une url existe ou pas

```
var request;  
if(window.XMLHttpRequest)  
    request = new XMLHttpRequest();  
else  
    request = new ActiveXObject("Microsoft.XMLHTTP");  
request.open('GET', 'http://www.mozilla.org', false);  
request.send(); // there will be a 'pause' here until the  
response to come.  
// the object request will be actually modified  
if (request.status === 404) {  
    alert("The page you are trying to reach is not  
available.");  
}
```