

# **Cours Applications web**

**IENAC 21**



P. CIVEYRAC – SAFRAN - philippe.civeyrac@safrangroup.com

J. PIN – SAFRAN - joackim.pin@safrangroup.com

K. PUECHMOREL - ENAC Z134 - karine.puechmorel@enac.fr

# TABLE DES MATIERES

<b>PLAN DU COURS</b>		<b>SLIDES</b>
Objectifs .....		2
Planning des cours .....		3
<b>1. Le www .....</b>		<b>5</b>
1.1. Définition et historique .....		7
1.2. Les concepts du web .....		8
1.3. Les navigateurs .....		11
<b>2. HTML5 / CSS3 .....</b>		<b>13</b>
<b>2.1. Les langage HTML5 .....</b>		<b>14</b>
2.1.1. Principe du langage .....		14
2.1.2. Les balises .....		20
2.1.3. Les médias .....		27
2.1.4. Les icônes .....		31
2.1.5. Les formulaires .....		34
<b>2.2. 2. Les feuilles de style .....</b>		<b>44</b>
2.2.1. Définition et principe du langage .....		44
2.2.2. Classe et format .....		55
2.2.3. Cascades et Héritage .....		56
2.2.4. Bootstrap .....		58
2.2.5. Préprocesseurs en CSS .....		69
<b>3. Infrastructure web .....</b>		<b>75</b>
3.1. Le serveur web Flask .....		76
3.2. Principe du routage .....		85
3.3. Passage de paramètres .....		91
3.4. Moteur de template JINJA .....		95
3.5. Templating .....		100
<b>4. Traitement des données .....</b>		<b>124</b>
4.1. Données enregistrées en BDD .....		126
4.2. Sessions et cookies .....		128
4.3. Données enregistrés dans un fichier .....		129
<b>5. Gestion de l'interface (sur E-campus) .....</b>		



## **ANNEXES**

Annexe 1 : Installation de la plate-forme d'hébergement

Annexe 2 : Feuilles de styles

Annexe 2 : Récapitulatifs requêtes sql dans bdd.py : select, insert, update, delete





Mars 2022

...

# APPLICATIONS WEB

## IENAC 21

Présenté par:

Joackim PIN – joackim.pin@safrangroup.com  
Philippe CIVEYRAC -- philippe.civeyrac@safrangroup.com  
Karine PUECHMOREL – karine.puechmorel@enac.fr

1



## Objectifs



### Les concepts du web

Définition, historique et environnement



### Conception et réalisation

Savoir réaliser un site web pour présenter l'information



Travail en équipe



### Gestion des données distantes

Savoir traiter et explorer des données distantes (SGBD) à travers une application web.



### Mécanismes client / serveurs

Comprendre les relations clients/serveurs ainsi que les spécificités des langages.

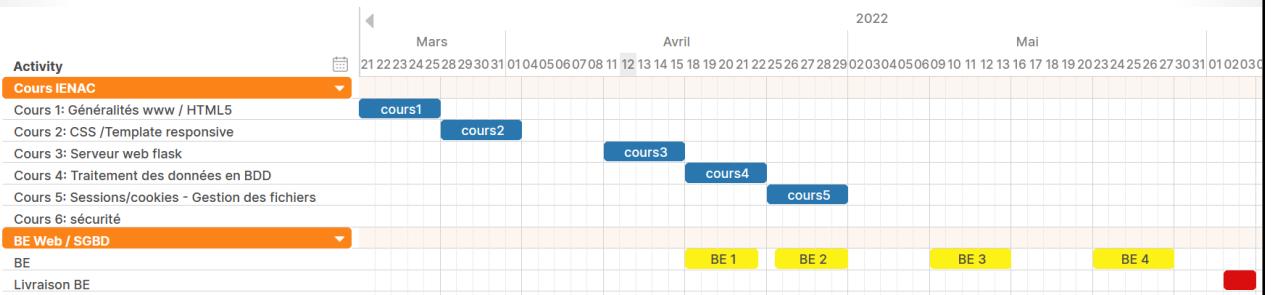
2

2

1



# Planning



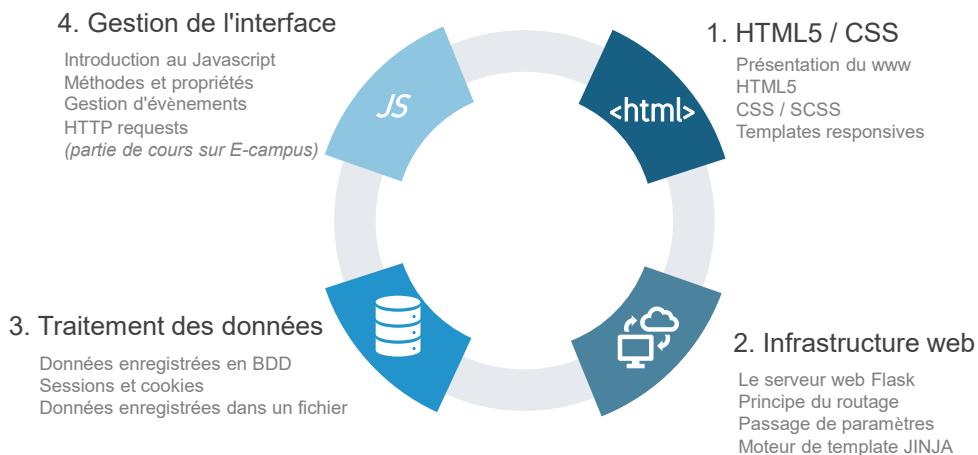
Planning prévisionnel – peut être modifié selon l'état du parc informatique ENAC.

3

3



# Plan du cours



4

4



## HTML5 / CSS



I. Présentation du www

II. HTML5

III. CSS / SCSS

IV. Templates responsives

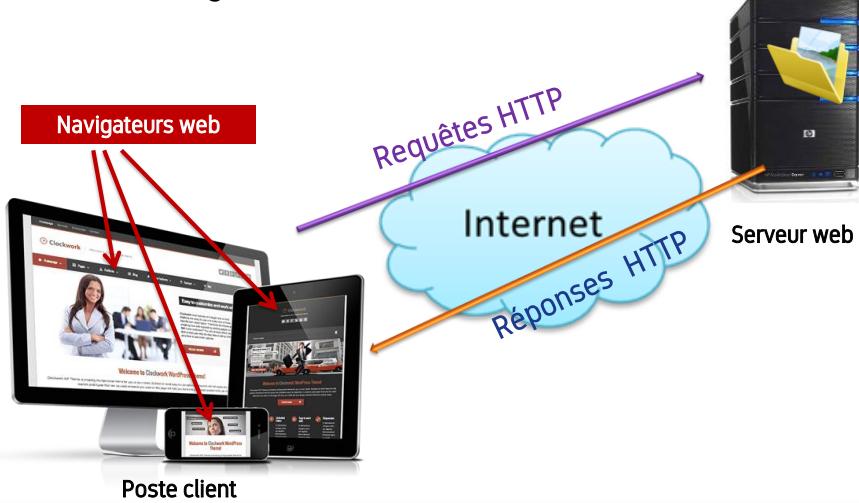
5

5



## Qu'est-ce qu'une application web ?

Une **application web** désigne un logiciel applicatif hébergé sur un serveur et accessible via un navigateur web





## Le web: définition et historique

Le WWW - *World Wide Web* - littéralement la « toile mondiale », est un système hypertexte réparti sur Internet et qui permet de consulter, avec un navigateur, des pages Web mises en ligne.

- 1990-1993 : le Centre Européen de Recherche Nucléaire et le National Center for Superconducting Application créé le www.
- 1993 : premier navigateur (gratuit) : MOSAIC.
- 1994 : création du W3C
- 2000 : 10 millions de sites - 368 millions d'internautes.
- 2008 : En France: 31,77 millions d'internautes.: 155 millions de sites web
- 2015: 3,025 milliards d'internautes  
2,060 milliards d'inscrits sur les réseaux sociaux, soit 68% des internautes.
- En 2022:
  - 5 milliards d'internautes (monde)
  - 60,9 millions d'internautes (France ~ 93% de la population)

*Statistiques intéressantes:* <https://fr.wizcase.com/blog/statistiques-incroyables-sur-internet-et-les-reseaux-sociaux/>



## Les concepts du web

- ✓ un moyen d'accéder facilement à n'importe quelle ressource sur internet : **Uniform Ressource Locator**
- ✓ Un langage de mise en forme simple des documents présents sur internet : **HyperText Markup Language**
- ✓ un protocole de communication entre le serveur web et les clients : **HyperText Tranfert Protocol**



Organisme de gestion: **le W3C**



## Le protocole HTTP

Le protocole HTTP permet de demander à un serveur d'effectuer une action via une requête. Dans le cadre d'une application web c'est le navigateur qui se charge d'envoyer les requêtes au serveur et de traiter la réponse.

- Le type de requête est défini via une méthode particulière.
- L'action concerne généralement une ressource identifiée par une URL.
- Il existe différentes méthodes dans le protocole  
HTTP : GET, HEAD, POST, OPTIONS, CONNECT, TRACE, PUT, PATCH, DELETE.

*Exemple:* Requête GET envoyée pour récupérer la page d'accueil de google

*Dans le cadre de ce cours, deux méthodes seront utilisées :*

**GET** : les données à envoyer au serveur sont écrites directement dans l'URL

**POST** : écrit les paramètres dans la requête HTTP

9

1 > HTML5 / CSS > Présentation du www

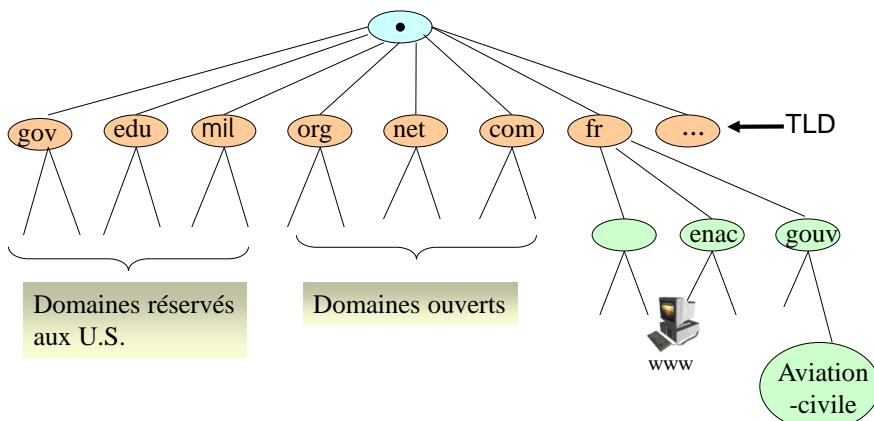
9



## Les espaces de noms de domaine

Un nom de domaine est une chaîne de caractères (ex : prodomaines) associée à une extension (ex : .com, .fr...). Il constitue ainsi un nom familier associé à une adresse IP.

domaine racine (=root)



*Exemple de nom de domaine:* www.aviation-civile.gouv.fr

1 > HTML5 / CSS > Présentation du www

10

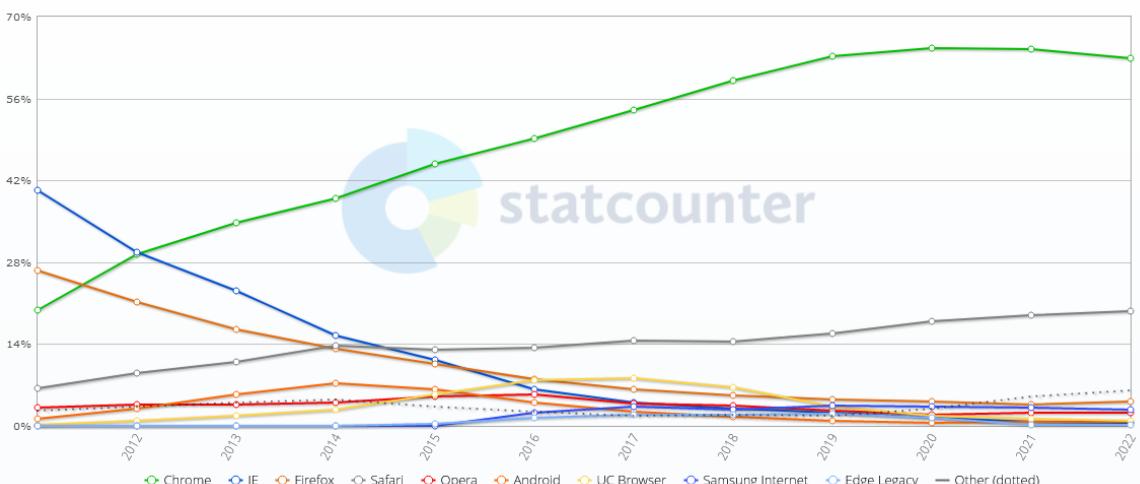
10



## Les navigateurs

Browser Market Share Worldwide  
2011 - 2022

[Edit Chart Data](#)



1 > HTML5 / CSS

> Présentation du www.statcounter.com/browser-market-share#yearly-2011-2022

11

11



## Quelques commandes bien pratiques

Visual studio Code comprend un certain nombre de commandes utiles pour l'écriture du code.

Pour voir et modifier la liste des commandes, Cliquer sur >> Raccourcis Clavier

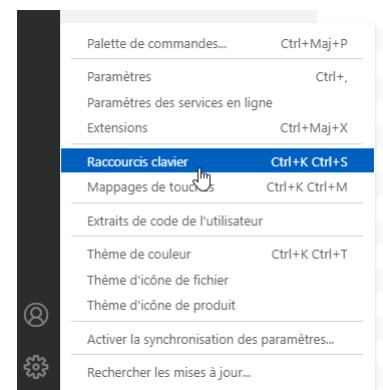
Sélectionner tout le texte: **CTRL+A**

Rechercher dans le texte: **CTRL+F**

Remplacer dans le texte: **CTRL+H**

ajoute / supprime un commentaire: **CTRL+K CTRL+C**

Mettre en forme le document (Indentation du texte):  
**(souris) clic droit** => mettre en forme le document



1 > HTML5 / CSS

> Présentation du www

12

12



# HTML5 / CSS



## I. Présentation du www

## II. HTML5

- Principales balises
- Médias
- icônes
- Formulaires

## III. CSS / SCSS

## IV. Templates responsives

13

13



# HTML

Le langage HTML est un langage de représentation de données composé de balises.

**balise (ou tag):** instruction interprétée par le navigateur et définissant l'affichage d'un élément particulier. Une balise est composée de deux parties :

- une partie dite « **ouvrante** » : *début de l'instruction*  
`<instruction attribut1="valeur" attribut2="valeur" >`
- une partie dite « **fermante** » : *fin de l'instruction*  
`</instruction>`



## Balises structurantes

### Le doctype <!DOCTYPE html>

- définit le type du document: règles de la syntaxe
- première ligne du fichier
- Indispensable



### L'en-tête

- balise <head></head>
- Indispensable
- Spécifie: les métadonnées, les feuilles de styles, les fonctions JavaScript, l'encodage de caractères

### La zone racine

- balise <html></html>
- Indispensable
- indique le début et la fin du document
- la langue du document <html lang="fr">

### Le corps du document

- balise <body></body>
- Indispensable
- contient tous les éléments à afficher dans la page



## L'en-tête du document <head></head>

### ■ Encodage des caractères

<meta charset="UTF-8" />

### ■ Titre de l'onglet

<title>ma page web</title>

### ■ Favicon

<link rel="shortcut icon" type="image/png" href="favicon.png" />

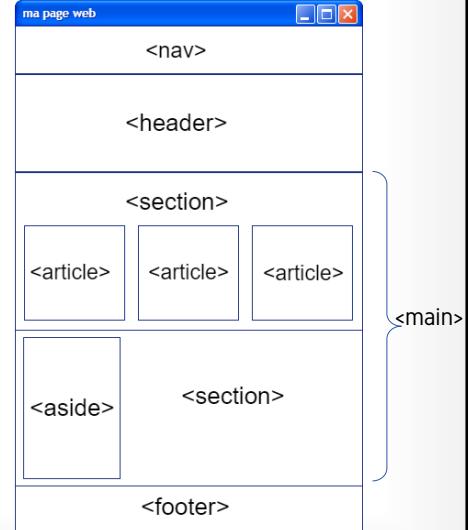
```

index.html ...
1  <!DOCTYPE html>
2  <html lang="fr">
3
4    <head>
5      <meta charset="UTF-8" />
6      <title>ma page web</title>
7      <link rel="shortcut icon" type="image/png" href="favicon.png"/>
8    </head>
9
10   <body>
11     Contenu du site
12   </body>
13
14 </html>
  
```



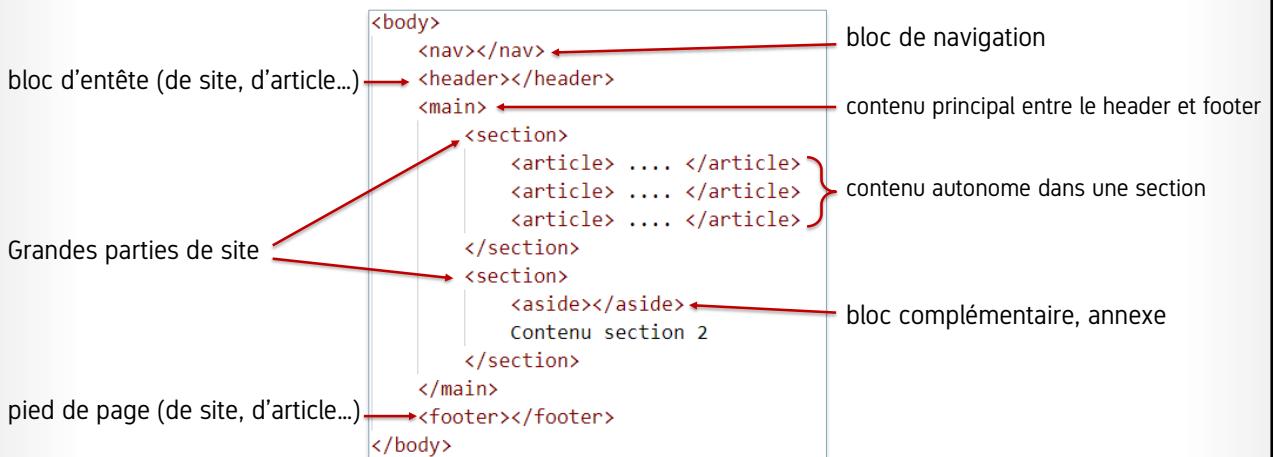
## Le corps du document <body></body>

- <nav> définit un bloc de navigation
- <header> définit un bloc d'entête (de site, d'article...)
- <main> entre le header et footer, définit le contenu principal
- <section> définit une partie de site
- <article> représente un contenu autonome dans un document
- <aside> définit un bloc d'informations complémentaires, annexes
- <footer> définit un pied de page (de site, d'article...)



## Le corps du document <body></body>

Certaines balises définissent la structure de la page et possède une sémantique particulière.





## Règles d'écriture

- Chaque balise nécessite une fermeture.

<p>texte 1

<p> texte 1</p>

N.B: Pour certaines balises, il suffit de rajouter un " / " à la fin. Ex: <img />, <br />

- Imbriquer correctement les éléments.

<p>Paragraphe en <i> italique </p></i>

<p>Paragraphe en <i> italique </i></p>

4  
Règles

- Chaque valeur d'attribut doit être entre guillemets

<div id=AbCde></div>

<div id="AbCde"></div>

- Les balises et leurs attributs s'écrivent en minuscule.

<TEXTAREA ID="AbCde"></TEXTAREA>

<textarea id="AbCde"></textarea>



## Les caractères spéciaux

Quand les caractères accentués ne s'affichent pas correctement, ils peuvent être remplacés par un codage particulier. La liste complète de ces codes est disponible sur:

<https://www.commentcamarche.net/contents/489-caracteres-speciaux-html>

è - &egrave;  
é - &acute;  
ê - &ecirc;  
ë - &euml;

à - &agrave;  
â - &acirc;  
ä - &auml;  
æ - &aelig;

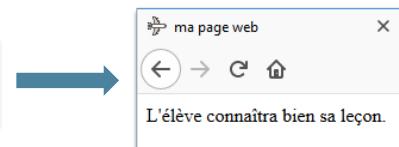
ô - &ocirc;  
ö - &ouml;  
œ - &oelig;  
ç - &ccedil;

î - &icirc;  
ï - &iuml;  
û - &ucirc;  
Ù - &ugrave;

" - &quot;  
& - &amp;  
< - &lt;  
> - &gt;

Exemple:

```
<body>
  L'&eacute;l&egrave;ve conna&icirc;tra bien sa le&ccedil;on.
</body>
```



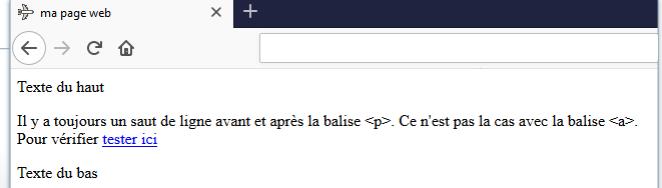


## Les balises HTML

Il existe deux catégories de balises :

- Les balises **block** créent un bloc autonome précédé d'un saut de ligne.  
*Exemple:* <p>, <div>, <ul>, <li>, <h1>...<h6>, <table>....
- Les balises **inline** ne créent pas de bloc autonome. Elles s'inscrivent à la suite d'un contenu. Il n'y a donc pas de saut de ligne,  
*Exemple:* <a>, <span>, <strong>, <em>, <img>, <sub>, <sup>....

```
Texte du haut
<p> Il y a toujours un saut de ligne avant et après la balise &lt;p&gt;.
Ce n'est pas le cas avec la balise &lt;a&gt;.<br />
Pour vérifier <a href="/index.html">tester ici</a>
</p>Texte du bas
```



1 > HTML5 / CSS > Langage HTML5

21

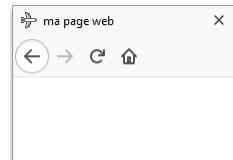


## Balises

*Commentaires:* <!-- -->

*Exemple:*

```
<!-- Je suis un commentaire.
Je n'apparais pas dans le navigateur. -->
```

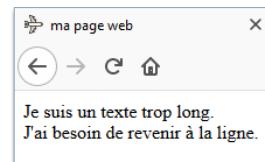


N.B: Commande visual studio code pour commenter / décommenter: CTRL+K CTRL+C

*Retour à la ligne:* <br /> (break)

*Exemple:*

```
Je suis un texte trop long. <br /> J'ai besoin
de revenir à la ligne.
```



1 > HTML5 / CSS > Langage HTML5

22

22

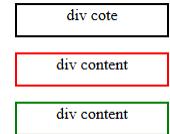


## Balises block

*Conteneur:*      `<div> </div>`

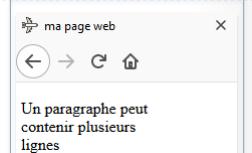
- Structure la page
- N'a pas de sémantique particulière

```
<section>
  <div id="cote"></div>
  <div class="content"></div>
  <div class="content"></div>
</section>
```



*Paragraphe:*      `<p> </p>`

`<p>` Un paragraphe peut contenir plusieurs lignes`</p>`



*Les titres:*

- `<h1></h1>`
- `<h2></h2>`
- `<h3></h3>`
- `<h4></h4>`
- `<h5></h5>`
- `<h6></h6>`

```
<h1>Titre 1</h1>
<h2>Titre 2</h2>
<h3>Titre 3</h3>
<h4>Titre 4</h4>
<h5>Titre 5</h5>
<h6>Titre 6</h6>
```

**Titre 1**

**Titre 2**

**Titre 3**

**Titre 4**

**Titre 5**

**Titre 6**



## Balises block

*Les listes à puces*

- `<ul></ul>`: liste non ordonnée  
`<li></li>`: item dans la liste

*Types possibles:*  
square, disc, circle, none

```
<ul type="disc">
  <li>Item 1</li>
  <li>Item 2</li>
  <li>Item 3</li>
  <li>Item 4</li>
</ul>
```

- Item 1
- Item 2
- Item 3
- Item 4

- `<ol></ol>`: liste ordonnée  
`<li></li>`: item dans la liste

*Types possibles:*

- Numériques: 1,2...
- Lettres: A, a
- Chiffres romains: i, I

```
<ol type="I">
  <li>Romain 1</li>
  <li>Romain 2</li>
  <li>Romain 3</li>
  <li>Romain 4</li>
</ol>
```

- I. Romain 1
- II. Romain 2
- III. Romain 3
- IV. Romain 4



## Balises block

### Les tableaux

- **<table></table>**: début / fin du tableau
- **<thead></thead>**: en-tête du tableau
- **<tbody></tbody>**: corps du tableau
- **<tr></tr>**: une ligne
- **<th></th>**: une colonne dans l'en-tête
- **<td></td>**: une colonne dans le corps
  
- **rowspan**: fusion de 2 lignes
- **colspan**: fusion de 2 colonnes

#	Colonne 1	Colonne 2	Colonne 3
1	L1,C1	L1,C2	L1,C3
2	L2,C2		
3	L3,C2		L3,C3

1 &gt; HTML5 / CSS

&gt; Langage HTML5

25

```

<table>
  <thead>
    <tr>
      <th>#</th>
      <th>Colonne 1</th>
      <th>Colonne 2</th>
      <th>Colonne 3</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>L1,C1</td>
      <td>L1,C2</td>
      <td>L1,C3</td>
    </tr>
    <tr>
      <td>2</td>
      <td rowspan="2">L2,C1</td>
      <td colspan="2" rowspan="2">L2,C2</td>
    </tr>
    <tr>
      <td>3</td>
      <td>L3,C2</td>
      <td>L3,C3</td>
    </tr>
  </tbody>
</table>

```

25



## Balises inline

### Lien hypertexte: **<a></a>**

Un lien hypertexte est un mot ou ensemble de mots qui, par un simple clic, permet d'accéder à une autre partie de la page, à une autre page ou à un autre site.



Exemple:

```

<a href="index.html">Accueil</a>
<a href="https://www.ecologie.gouv.fr/direction-generale-laviation-civile-dgac" target="_blank">
  DGAC
</a>
<a href="https://www.ecologie.gouv.fr/direction-generale-laviation-civile-dgac#scroll-nav_5">
  Antilles-Guyane
</a>

```

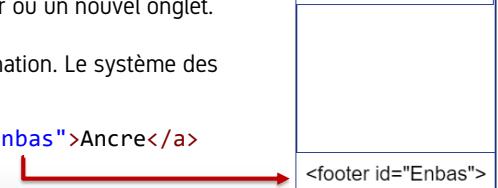
- **target = " \_blank "**: Ouverture de la page dans un nouveau navigateur ou un nouvel onglet.
- Par défaut, les liens hypertextes renvoient au début du site de destination. Le système des ancrées permet d'être redirigé plus bas dans la page demandée.

```
<a href="index.html#Enbas">Ancre</a>
```

1 &gt; HTML5 / CSS

&gt; Langage HTML5

26



26



# HTML5 / CSS



## I. Présentation du www

## II. HTML5

- Principales balises
- Médias
- icônes
- Formulaires

## III. CSS / SCSS

## IV. Templates responsives



# Les images

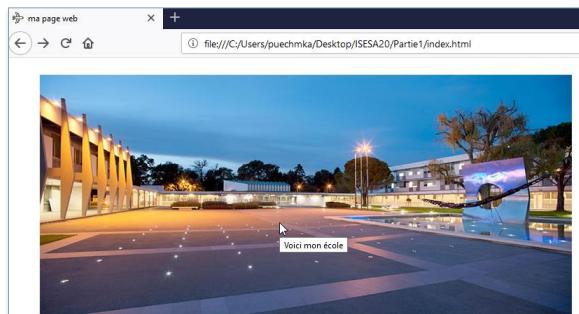
*balise: <img />*

```

```

*Les attributs:*

- **src**: définit l'adresse de l'image à afficher (absolue ou relative)
- **alt** : désigne le texte alternatif de l'image. Ne se voit que lorsque l'image ne peut pas s'afficher
- **title**: texte apparaissant lorsque la souris passe sur l'image.

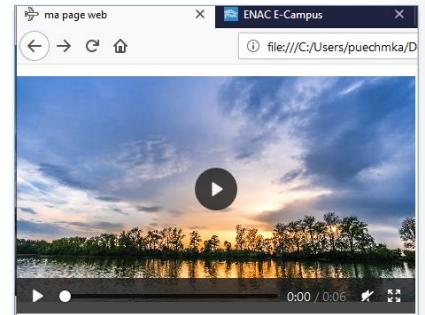




## Vidéo & audio

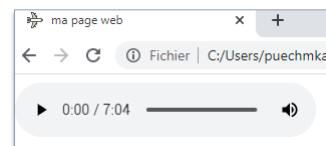
Balise: `<video>`

```
<video width="320" height="240" controls="controls">
  <source src="movie.mp4" type="video/mp4" />
  <source src="movie.ogg" type="video/ogg" />
  Votre navigateur ne supporte pas la balise vidéo
</video>
```



Balise: `<audio>`

```
<audio controls="controls">
  <source src="song.ogg" type="audio/ogg" />
  <source src="song.mp3" type="audio/mp3" />
  Votre navigateur ne supporte pas la balise audio
</audio>
```



Attributs pour `<audio>` et `<video>`:

Autoplay, controls, loop, preload, src, height, width, poster, muted



## Vidéo Youtube & carte Google Map

Vidéos youtube:

- Sur Youtube, aller sur la page de la vidéo
- Sélectionner "partager"



- Sélectionner "intégrer"

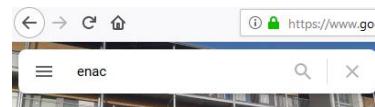


- Inclure dans votre fichier html, le code qui vous est proposé



Carte Google Map

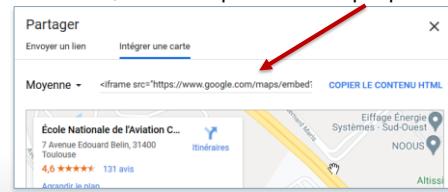
- Sur Google Map, aller sur la localisation à visualiser



- Un fois le zoom réglé, Sélectionner "partager"



- Sélectionner "intégrer une carte", puis, inclure dans votre fichier html, le code qui vous est proposé





# HTML5 / CSS



## I. Présentation du www

## II. HTML5

- Principales balises
- Médias
- **icônes**
- Formulaires

## III. CSS / SCSS

## IV. Templates responsives



# Les icônes

Afin d'implémenter des icônes, bootstrap suggère l'utilisation de frameworks alternatifs tels que fontawesome

### *Installer fontawesome:*

1. Se rendre sur le site officiel du framework : <https://fontawesome.com/>
2. Télécharger et décompresser fontawesome (version free for web) dans le répertoire **vendor**
3. Rajouter dans le <head> le lien vers le fichier all.min.css

```
<link rel="stylesheet" href=" vendor/fontawesome-free-X.X.X-web/css/all.min.css" />
```

4. Rajouter juste avant le </body> le lien vers le fichier all.min.js

```
<script src="vendor/fontawesome-free-X.X.X-web/js/all.min.js"></script>
```

**Autre solution:** par lien CDN -- <https://cdnjs.com/libraries/font-awesome>

Tous les icônes disponibles sont sur: <https://fontawesome.com/>  
Cliquer sur le menu "icon" puis sélectionner le filtre "free" à gauche



## Les icônes

Pour personnaliser la mise en forme des icônes, voir la documentation officielle sur le site <https://fontawesome.com/>

### Exemples d'icônes



```
<i class="fas fa-home"></i>
```



```
<i class="fas fa-user-times"></i>
```



```
<i class="fas fa-plane"></i>
```

#### *Modification de la taille*

```
<i class="fas fa-trash-alt fa-xs"></i>
<i class="fas fa-trash-alt fa-sm"></i>
<i class="fas fa-trash-alt fa-lg"></i>
<i class="fas fa-trash-alt fa-2x"></i>
<i class="fas fa-trash-alt fa-3x"></i>
<i class="fas fa-trash-alt fa-5x"></i>
<i class="fas fa-trash-alt fa-7x"></i>
<i class="fas fa-trash-alt fa-10x"></i>
```



#### *Modification de la couleur (idem texte normal)*

```
<i class="fas fa-plus-square text-success"></i>
```



33

33



## HTML5 / CSS

### I. Présentation du www



### II. HTML5

- Principales balises
- Médias
- icônes
- Formulaires

### III. CSS / SCSS

### IV. Templates responsives

34

34



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

Un **formulaire HTML** délimite une zone dans laquelle la page Web est capable de dialoguer avec l'internaute à travers des champs de saisie et boutons. Les formulaires sont traités côté serveur.

```
<form action="/nomRoute" method="POST"></form>
```

*Les attributs:*

- **action** : page ou route qui reçoit et traite le formulaire
- **method** : Type de méthode HTTP utilisé pour l'envoi.
  - **GET**: les données sont envoyées dans la barre d'adresse du navigateur.
  - **POST**: les données sont envoyées cachées (en métadonnées)

The screenshot shows a window titled "ma page web" containing a form structure. The form includes:
 

- A text input field labeled "Votre login" with a placeholder "Votre login".
- A password input field with a placeholder "Votre login".
- A section for "choix1" and "choix2" with two sets of checkboxes and radio buttons.
- A dropdown menu labeled "liste déroulante".
- A text area labeled "vos commentaires".
- A blue "Envoyer" button at the bottom right.



## Champs de formulaires

*Type Text:*

```
<label for="login">Champ input text</label>
<input type="text" name="login" id="login" placeholder="Votre login" />
```

The screenshot shows a text input field with the label "Champ input text" above it and the placeholder "Votre login" inside the field.

*Type password:*

```
<label for="motPasse">Champ Mot de passe</label>
<input type="password" name="motPasse" id="motPasse" placeholder="Votre mot de passe" />
```

The screenshot shows a password input field with the label "Champ input password" above it and the placeholder "Votre mot de passe" inside the field.

*Type email:*

```
<label for="sonMail">Champ Email</label>
<input type="email" name="sonMail" id="sonMail" placeholder="Votre email" />
```

The screenshot shows an email input field with the label "Champ input email" above it and the placeholder "Votre email" inside the field.



## Champs de formulaires

Type number:

```
<label for="age">Champ input number</label>
<input type="number" name="age" id="age" placeholder="Votre âge" />
```

Champ input number

Type file:

```
<label for="fichier">Champ input file</label>
<input type="file" name="fichier" id="fichier" />
```

Champ input file

 Aucun fichier sélectionné.

Type Range:

```
<label for="customRange">Champ input range</label>
<input type="range" name="customRange" id="customRange">
```

Champ input range

## Les champs dates

Type date

```
<label for="maDate"></label>
<input type="date" name="maDate" id="maDate"/>
```

jj / mm / aaaa

juin 2021

lun.	mar.	mer.	jeu.	ven.	sam.	dim.
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

Type time

```
<label for="monHeure">Champ input time</label>
<input type="time" name="monHeure" id="monHeure"/>
```

08:10

1 > HTML5 / CSS > Langage HTML5

38

38

19



## Les champs dates



les champs suivants ne marchent pas sur tous les navigateurs.



juin 2021							↑	↓
lu	ma	me	je	ve	sa	di	17	44
31	1	2	3	4	5	6	18	45
7	8	9	10	11	12	13	19	46
14	15	16	17	18	19	20	20	47
21	22	23	24	25	26	27	21	48
28	29	30	1	2	3	4	22	49
5	6	7	8	9	10	11	23	50
Aujourd'hui								

type datetime-local

2021						
janv.	févr.	mars	avr.	mai	juin	juil.
sept.	oct.	nov.	déc.	Ce mois		
---	---	---	---	---	---	---

type month

juin 2021							↑	↓
Semaine	lu	ma	me	je	ve	sa	di	
22	31	1	2	3	4	5	6	
23	7	8	9	10	11	12	13	
24	14	15	16	17	18	19	20	
25	21	22	23	24	25	26	27	
26	28	29	30	1	2	3	4	
27	5	6	7	8	9	10	11	
							Cette semaine	

type week



## Les champs checkbox et radio

### Type checkbox:

Réponse à choix **multiple**

Choix 1    Choix 2    Choix 3

```
<input type="checkbox"
       name="choix[]"
       id="C1" value="1" />
<label for="C1">Choix 1</label>

<input type="checkbox"
       name="choix[]"
       id="C2" value="2" />
<label for="C2">Choix 2</label>

<input type="checkbox"
       name="choix[]"
       id="C3" value="3" />
<label for="C3">Choix 3</label>
```

### Type radio

Réponse à choix **unique**

Choix 1    Choix 2    Choix 3

```
<input type="radio"
       name="rad[]"
       id="R1" value="1" />
<label for="R1">Choix 1</label>

<input type="radio"
       name="rad[]"
       id="R2" value="2" />
<label for="R2">Choix 2</label>

<input type="radio"
       name="rad[]"
       id="R3" value="3" />
<label for="R3">Choix 3</label>
```



## Listes déroulantes & zone de texte large

*Liste déroulante: <select> </option>*

```
<select name="monSelect" aria-label="monSelect">
  <option selected disabled>
    Faites votre choix
  </option>
  <option value="2">1+1</option>
  <option value="4">2*2</option>
  <option value="6">12:2</option>
</select>
```

*Zone de texte large: <textarea>*

```
<label for="comment">
  Zone de texte large
</label>

<textarea name="comment" id="comment">
  Je suis bavard. J'aime beaucoup écrire.
</textarea>
```



## Boutons

*Type submit:* bouton de soumission

```
<button type="submit"> Envoyer </button>
```

*Type button:*

bouton d'agrément (n'envoie pas le formulaire)

```
<button type="button">
  Cliquez sur moi
</button>
```

### Validation des formulaires à la volée

L'attribut **required** permet de rendre obligatoire le remplissage d'un champ et bloquer la validation du formulaire si l'un des champs n'a pas été renseigné.



# HTML5 / CSS



I. Présentation du www

II. HTML5

## III. CSS / SCSS

- Définition et principes de base
- Classes et formats
- Cascades et héritage
- Bootstrap
- Préprocesseurs en CSS

IV. Templates responsives

43

43



# Les feuilles de styles

## *CSS: Cascading Style Sheets*

Elles permettent (entre autres) de modifier l'aspect de tout un site en ne touchant qu'à un seul fichier (la feuille de style), et donc d'alléger grandement la gestion d'un site.

1996: CSS1

1998: CSS2

2011: CSS3

La version CSS4 a commencé en parallèle de CSS3 mais n'est toujours pas devenue une norme.



# Principes CSS

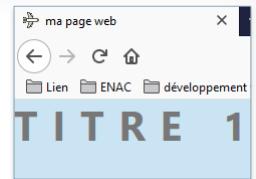
Fichier index.html

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8" />
    <title>ma page web</title>
  </head>
  <body>
    Ce style s'applique à la balise body
    <h1>Titre 1</h1> Ce style s'applique à la balise h1
  </body>
</html>
```

Fichier style.css

```
body {
  background: rgb(202, 228, 243);
  font-size: 16px;
}

h1 {
  text-transform: uppercase;
  font-weight: 700;
  color: #777;
  font-size: 2.5em;
  letter-spacing: 15px;
}
```



## Liens: html ↔ CSS

- Solution 1: Tous les css dans le <head> de index.html



```
index.html
<head>
  <link rel="stylesheet" href="css/style1.css" />
  <link rel="stylesheet" href="css/style2.css" />
  <link rel="stylesheet" href="css/style3.css" />
</head>
```

- Solution 2: dans le <head> de index.html puis dans all.css

```
index.html
<head>
  <link rel="stylesheet" href="css/all.css" />
</head>
```

```
all.css
@import url("style1.css");
@import url("style2.css");
@import url("style3.css");
```



## Liens: html ↔ CSS



**Méthode 2:** dans <head><style>

```
<html lang="fr">
  <head>
    <meta charset="UTF-8" />
    <title>ma page web</title>
    <style>
      body {
        background: #fff;
        font-weight: 300;
        color: #777;
        line-height: 1.65em;
        font-size: 16px;
      }
    </style>
  </head>

  <body>
    <h1>Titre 1</h1>
  </body>
</html>
```



**Méthode 3:** dans une balise

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta charset="UTF-8" />
    <title>ma page web</title>
  </head>

  <body>
    <h1 style="text-transform: uppercase;
font-weight: 700;color: #ed786a;font-size:2.5em;
letter-spacing: 15px;">
      Titre 1
    </h1>
  </body>
</html>
```



## Quelques éléments CSS

**background-color:** Couleur de fond d'écran.

**background-image:** image de fond.

**color :** couleur du texte.

**font-family :** police de caractère (*Arial, Times, Tahoma ...*)

**font-size:** taille de la police.

**font-weight:** poids de police (*gras, fin ...*)

**font-style:** inclinaison de la police (*italique, oblique ..*)

**text-align:** aligner le texte (*gauche, droit, centré, justifié*)

**width:** longueur d'un élément

**height:** hauteur d'un élément.

**text-decoration:** décoration du texte (*souligné, surligné, barré, clignotant..*)

**List-style:** définition de la puce introduisant une liste

*Voir Annexe 2*



## Unités de mesure

pt	Point. 1 point correspond à 1/72 pouce	font-size:12pt; line-height:14pt;
pc	Pica. 1 pica correspond à 12 points.	font-size:1pc; line-height:1.2pc;
in	1 pouce correspond à 2.54 centimètres	border-width:0.1in; margin-left:in;
mm	Millimètre. 1/1000 du mètre étalon.	margin-bottom:10mm; width:70mm;
cm	Centimètre. 1/100 du mètre étalon.	font-size:1cm; top:2.54cm;
px	Pixel. Relative d'un mode de sortie à l'autre, absolue par rapport à un seul et unique mode de sortie.	border-width:3px; margin-right:60px;
em	Embed. Dimension par rapport à la taille de police de l'élément.	font-size:1.2em; line-height:1.5em;
%	pourcentage. Relatif à la taille de l'élément proprement-dit, ou à la taille de l'élément parent ou à un contexte plus général.	font-size:10pt; line-height:120%;



## Unités de couleur

Les couleurs peuvent être déclarées :

- par leur nom (en anglais)
- par leur son code hexadécimal:  
**#RRVVBB**
- par le RGB- [R],[V],[B]
- par le RGBA:  
**rgba( 156, 25, 26, 0.8);**  
*0,8 est la transparence*

Transparent	White	Golden Yellow	Signal Yellow	Yellow
Light Yellow	Brimstone Yellow	Purple Red	Burgundy	Dark Red
Red	Light Red	Orange Red	Orange	Light Orange
Pastel Orange	Purple	Violet	Lavender	Lilac
Pink	Soft Pink	Deep Sea Blue	Steel Blue	Dark Blue
Cobalt Blue	King Blue	Brilliant Blue	Blue	Traffic Blue
Gentian Blue	Gentian	Azure Blue	Sky Blue	Light Blue
Ice Blue	Turquoise Blue	Turquoise	Mint	Dark Green
Forest Green	Green	Grass Green	Light Green	Yellow Green
Lime-Tree Green	Brown	Nut Brown	Light Brown	Beige
Cream	Black	Dark Grey	Grey	Telegrey
Middle Grey	Light Grey	Silver Grey (met)	Gold (met)	Copper (met)



# Polices particulières

## HELLO CSS 3 WORLD!

De nombreuses polices de caractères sont téléchargeables sur internet:  
Google Fonts, fontsquirrel....

*Exemple: utilisation de google fonts*

1. Choisir une fonte sur les site officiel de google fonts.
2. Sélectionner le lien + [Select this style](#) en face de la fonte choisie
3. Dans le cadre qui vient d'apparaître, cliquer sur l'option
4. Dans style.css, rajouter en haut du fichier la ligne commençant par: @import url('https://fonts.googleapis.com/css?family=FamilyName'); (enlever <style> et </style>)
5. Dans style.css, rajouter dans "h1", la ligne décrite dans la partie "css rules to specify families" et commençant par: font-family: ....

```

 <link>  @import

<style>
@import url('https://fonts.googleapis.com/css?family=Dancing+Script&display=swap');
</style>

CSS rules to specify families

font-family: 'Dancing Script', cursive;

```



# Espacement et marges

## Principe d'utilisation

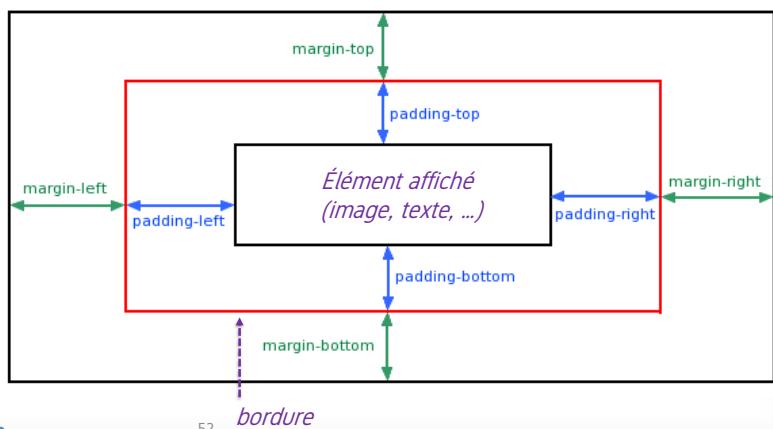
margin-top:10px;  
margin-right:20px;  
margin-bottom:30px;  
margin-left:40px;

margin: 10px 20px 30px 40px;  
top right bottom left

margin: 20px 30px;  
[top bottom] [right left]

margin: 50px;  
top right bottom left

- padding: espacement à l'intérieur de la bordure
- margin: espacement à l'extérieur de la bordure





# HTML5 / CSS



## I. Présentation du www

## II. HTML5

## III. CSS / SCSS

- Définition et principes de base
- Classes et formats
- Cascades et héritage
- Bootstrap
- Préprocesseurs en CSS

## IV. Templates responsives

53

53



# Attributs id et class

**Les attributs id et class permettent de nommer et de différentier des balises HTML.**  
Il peuvent être utilisés pour gérer les styles CSS.

L'attribut **id** signifie identifiant. Il permet de nommer précisément une balise.

Un id doit être **unique**. Il ne peut y avoir qu'un seul id identique par page.

```
<section id="aboutUs">
    .....
</section>
```

*style.css*

```
div#aboutUs
{
    color:#ed786a;
    margin: 50px;
    border-radius: 5px;
}
```

L'attribut **class** permet de donner un même nom à un ensemble de balises

*style.css*

```
<div class="fondGris">
    .....
</div>
<section>
    <article class="fondGris">....</article>
    <article class="fondGris">....</article>
    <article class="fondGris">....</article>
</section>
```

class ⇔ • (dans le css)  
 id ⇔ # (dans le css)

```
.fondGris
{
    background-color: #777;
    padding: 10px 0;
    border: 2px solid gainsboro;
}
```



## Classes et formats

```
style.css

p , div{color:red; text-align:center;}

.lien    {font-style:italic;}
.p.lien   {background-color:yellow; }

#lien    {font-weight:bold;}
div#lien {background-color:green; }

div h3
{
  color: #777;
  border:3px dashed blueviolet;
}
```

*index.html*

```
<p class= "lien">Texte1</p>
<div id= "lien">Texte2</div>
<div>
  <h3>Texte 3</h3>
</p>
```

Résultat

Texte1

Texte2

Texte 3

Une pseudo-classe est une manière de cibler un élément sans ajouter une classe.

:hover - cible un élément pointé visuellement (survolé)

:focus – cible un élément qui reçoit le focus

:last-child - cible le dernier élément enfant

1 > HTML5 / CSS

Langage CSS

55

55



## Cascade

L'ordre d'apparition des règles dans le CSS a une importance.

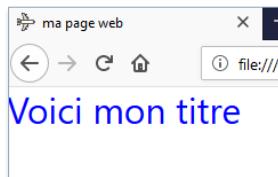
Quand deux règles applicables à un même sélecteur ont la même spécificité, c'est la dernière déclarée qui sera utilisée pour la mise en forme.

*style.css*

```
h1,h2 {color: yellow;}
h1 {color: blue;}
```

*index.html*

```
<h1>Voici mon titre</h1>
```



Résultat

1 > HTML5 / CSS

Langage CSS

56

56



# Héritage

L'héritage des CSS est fondé sur le modèle Parent-Enfant(s).

Chaque élément Enfant reçoit en héritage tous les styles de son élément Parent.

```
style.css

div{
    color: white;
    background-color: skyblue;
    font-size: 3em;
}

p {
    background-color: lightcoral;
    display: inline-flex;
    margin:0;
}
```

*index.html*

```
<div>Intro: <p> 1er paragraphe</p> : Fin </div>
```

**Intro: 1er paragraphe : Fin**

*Résultat*



# HTML5 / CSS

I. Présentation du www



II. HTML5

III. CSS / SCSS

- Définition et principes de base
- Classes et formats
- Cascades et héritage
- **Bootstrap**
- Préprocesseurs en CSS

IV. Templates responsives



# Bootstrap

Bootstrap est un framework qui organise et gère la mise en page d'un site web.

Il fait gagner beaucoup de temps aux développeurs en proposant des syntaxes CSS et HTML déjà prêtes et faciles à implémenter.

## *Installer Bootstrap:*

1. Se rendre sur le site officiel du framework : <https://getbootstrap.com/>
2. Télécharger et décompresser bootstrap dans un nouveau répertoire nommé **vendor**
3. Rajouter dans le <head> le lien vers le fichier bootstrap.min.css

```
<link rel="stylesheet" href="vendor/bootstrap-X.X.X-dist/css/bootstrap.min.css" />
```

4. Rajouter juste avant le </body> le lien vers le fichier bootstrap.bundle.min.js

```
<script src="vendor/bootstrap-X.X.X-dist/js/bootstrap.bundle.min.js"></script>
```

*Autre solution:* par lien CDN -- <https://cdnjs.com/libraries/bootstrap/5.1.3>

Toute la documentation sur: <https://getbootstrap.com/docs>

1 > HTML5 / CSS

> Langage CSS

59

59



# Les couleurs

Primary

Secondary

Success

Danger

Warning

Info

Light

Dark

## Couleur d'un composant

```
<p class="text-primary">.text-primary</p>
```

```
<button class="btn btn-secondary">.btn-secondary</button>
```

```
<button class="btn btn-outline-danger">.btn-outline-danger</button>
```

```
<div class="bg-info">.bg-info</div>
```

```
<div class="border border-success">.border-success</div>
```

```
<span class="bg-warning">.bg-warning</span>
```

The screenshot shows a browser window with a sidebar containing color swatches and their corresponding CSS classes. The sidebar has a header 'ma page web' with icons for back, forward, search, and refresh. Below the header, there are two columns of color swatches with their respective class names:

- Top row: Primary (blue), Secondary (grey), Success (green), Danger (red), Warning (yellow), Info (cyan).
- Bottom row: Light (light grey), Dark (black).

On the right side of the browser window, there are two separate boxes showing examples of how these colors are used in HTML:

- The top box shows examples for 'text-primary' and 'btn-secondary'.
- The bottom box shows examples for 'bg-info', 'border border-success', and 'bg-warning'.

## Associer les styles:

```
<p class="text-danger bg-light">.text-danger .bg-light </p>
```

```
<div class="text-white bg-dark">.text-white .bg-dark</div>
```

1 > HTML5 / CSS

> Langage CSS

60

60



## Les textes

.text-primary	<p class="text-primary fs-4">.text-primary</p>	
.text-secondary	<p class="text-secondary">.text-secondary</p>	
.text-success	<p class="text-success fs-2">.text-success</p>	
.text-danger	<p class="text-danger fst-italic">.text-danger</p>	
.text-warning	<p class="text-warning">.text-warning</p>	
.text-info	<p class="text-info">.text-info</p>	
.text-light	<p class="text-light bg-dark">.text-light</p>	
.text-dark	<p class="text-dark">.text-dark</p>	
.text-body	<p class="text-body">.text-body</p>	
.text-muted	<p class="text-muted">.text-muted</p>	
.text-white	<p class="text-white bg-dark text-decoration-line-through">.text-white</p>	
.text-black-50	<p class="text-black-50 text-decoration-underline">.text-black-50</p>	text noir semi transparent
.text-white-50	<p class="text-white-50 bg-dark">.text-white-50</p>	text blanc semi transparent

text-xxxx : couleur du texte

bg-xxxx : couleur du fond

fs-N : font-size N=[1,2,3,4,5,6]

fst-italic: italique

fw-bold  
fw-bolder  
fw-normal  
fw-light  
fw-lighter

Texte (non) souligné / barré

text-decoration-none

text-decoration-underline

text-decoration-line-through

61



## Les bordures



```
<div class="border">1</div>
<div class="border-top">2</div>
<div class="border-end">3</div>
<div class="border-bottom">4</div>
<div class="border-start">5</div>
```



```
<div class="border border-warning">1</div>
<div class="border-top border-danger">2</div>
<div class="border-end border-3 border-dark">3</div>
<div class="border-bottom border-5 border-dark">4</div>
<div class="border-start border-4 border-success">5</div>
```

Affiche les 4 côtés

Bordure en jaune

Fond en gris

Font-size 4

```
<div class="border border-top-0 border-warning border-5 bg-secondary text-white fs-4">1</div>
```

Le côté top disparaît  
(épaisseur de 0)

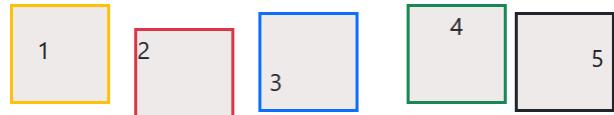
Épaisseur de 5

Texte en blanc



## Les marges

Propriétés	Côtés	Taille
<b>m</b> - margin	<b>t</b> - top	-0
<b>p</b> - padding	<b>b</b> - bottom	-1
	<b>s</b> - start = left	-2
	<b>e</b> - end = right	-3
	<b>x</b> - left and right	-4
	<b>y</b> - top and bottom	-5
	blank - sur les 4 côtés	-auto



```
<div class="border border-warning p-4">1</div>
<div class="border border-danger m-4">2</div>
<div class="border border-primary ps-2 mt-2 pe-4">3</div>
<div class="border border-success ms-5 me-2 text-center">4</div>
<div class="border border-dark text-end pe-2 pt-4 mt-2">5</div>
```

### Centrer horizontalement

```
<p class="mx-auto w-50 bg-info">
    Paragraphe centré horizontalement
</p>
```

Paragraphe centré  
horizontalement

1 > HTML5 / CSS > Langage CSS

### Centrer verticalement dans un tableau

baseline	top	middle	bottom
----------	-----	--------	--------

```
<tr>
    <td class="align-baseline">baseline</td>
    <td class="align-top">top</td>
    <td class="align-middle">middle</td>
    <td class="align-bottom">bottom</td>
</tr>
```

63

63



## Les formulaires

```
<label for="inp" class="form-label">Email</label>
<input type="email" class="form-control" id="inp">
```

Votre email

```
<input class="form-control form-control-lg" type="text">
<input class="form-control" type="text">
<input class="form-control form-control-sm" type="text">
```

.form-control-lg
.form-control
.form-control-sm

Votre choix

- Votre choix
- One
- Two
- Three

```
<select class="form-select" aria-label="xxxx">
```

Choix 1

Choix 2

Choix 1

Choix 2

```
<input class="form-check-input" id="C1" type="checkbox">
<label for="C1">Choix 1</label>
<input class="form-check-input" id="C2" type="checkbox" checked> <label for="C1">Choix 2</label>
```

```
<input class="form-check-input" id="C1" type="radio">
<label for="C1">Choix 1</label>
<input class="form-check-input" id="C2" type="radio" checked> <label for="C1">Choix 2</label>
```

1 > HTML5 / CSS > Langage CSS

64

64



## Les tableaux

#	Nom	Prénom	Âge
1	Mark	Otto	10

```
<table class="table">....</table>
```

#	Nom	Prénom	age
1	Mark	Otto	10

```
<table class="table table-primary">
</table>
```

#	Prénom	Nom	age
1	Mark	Otto	10
1	Kate	Dupond	17

```
<table class="table table-bordered">
</table>
```

#	Nom	Prénom	age
1	Mark	Otto	10

```
<table class="table table-borderless">
</table>
```

#	Prénom	Nom	age
1	Mark	Otto	10
2	Kate	Dupond	17

65

65

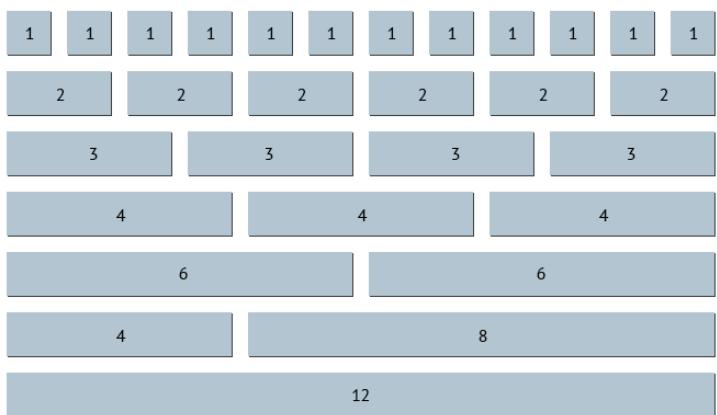


## Applications responsives

Un **site web adaptatif (responsive web design)** est un site web dont la conception vise à offrir la possibilité de consulter un même site sur différents terminaux de lecture (moniteurs d'ordinateur, smartphones, tablettes, TV ...)

### Bootstrap grid

```
<div class="container">
  <div class="row">
    <div class="col-4">
      Col1
    </div>
    <div class="col-4">
      Col2
    </div>
    <div class="col-4">
      Col3
    </div>
  </div>
</div>
```





## Tailles d'écran

	xs <576px (extra small)	sm ≥576px (small)	md ≥768px (medium)	lg ≥992px (large)	xl ≥1200px (extra large)	xxl ≥1400px (extra extra large)
Container max-width	None (auto)	540px	720px	960px	1140px	1320px
Class prefix	.col-	.col-sm-	.col-md-	.col-lg-	.col-xl-	.col-xxl-

Résultats



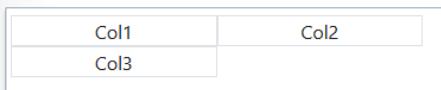
Écran lg

```

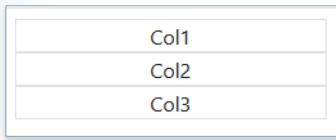
<div class="container">
  <div class="row">
    <div class="col-12 col-sm-6 col-lg-4">
      Col1
    </div>
    <div class="col-12 col-sm-6 col-lg-4">
      Col2
    </div>
    <div class="col-12 col-sm-6 col-lg-4">
      Col3
    </div>
  </div>
</div>

```

Écran sm



Écran xs



67

67



## Système d'auto-ajustement

Le class="col" détecte automatiquement au nombre de colonnes dans la ligne et calcule ainsi la longueur de chaque colonne.

1 / 2	2 / 2
1 / 3	2 / 3

```

<div class="container">
  <div class="row mb-2">
    <div class="col">1 / 2</div>
    <div class="col">2 / 2</div>
  </div>
  <div class="row">
    <div class="col">1 / 3</div>
    <div class="col">2 / 3</div>
    <div class="col">3 / 3</div>
  </div>
</div>

```

Certaines colonnes peuvent avoir une longueur imposée

1 / 2	2 / 2	3 / 3
1 / 3	2 / 3	3 / 3

```

<div class="container">
  <div class="row mb-2">
    <div class="col">1 / 2</div>
    <div class="col-6">2 / 2</div>
    <div class="col">3 / 3</div>
  </div>
  <div class="row">
    <div class="col">1 / 3</div>
    <div class="col-2">2 / 3</div>
    <div class="col">3 / 3</div>
  </div>
</div>

```



# HTML5 / CSS



## I. Présentation du www

## II. HTML5

## III. CSS / SCSS

- Définition et principes de base
- Classes et formats
- Cascades et héritage
- Bootstrap
- Préprocesseurs en CSS

## IV. Templates responsives

69

69



# Préprocesseurs CSS

Les préprocesseurs sont des **outils**, ou des programmes, qui vont vous permettre de **générer du CSS** à partir des fichiers écrits dans la syntaxe du préprocesseur que vous avez choisi.

```
test.scss

$blue: #3bbbfce;
$margin: 16px;

.content-navigation {
    border-color: $blue;
    color: darken($blue, 9%);
}

.border {
    padding: $margin / 2;
    margin: $margin / 2;
    border-color: $blue;
}
```



```
test.css

.content-navigation {
    border-color: #3bbbfce;
    color: #2ca2af;
}

.border {
    padding: 8px;
    margin: 8px;
    border-color: #3bbbfce;
}
```

- Installation de sass
- Compilation du code: sass test.scss test.css

Fichier source(.scss)

Fichier destination(.css)



## Préprocesseurs CSS

Il existe plusieurs types de préprocesseur CSS.  
Les principaux sont SASS (que l'on nomme **SassScript**) et LESS.

N.B: SASS possède 2 syntaxes possibles ⇔ extension des fichiers en .sass ou .scss

*Liste des options que les préprocesseurs offrent mais qui n'existent pas en css:*

- **Variables**
- **Fonctions mathématiques**
- **Fonctions**
- **Boucles**
- **Distinctions de cas** : les énoncés conditionnels. Ce n'est que lorsque certains états se produisent que les commandes spécifiées sont exécutées
- **Héritage** : il est possible d'hériter des propriétés d'un sélecteur à l'autre
- **Nidification** : SASS donne à l'utilisateur la possibilité de visualiser les dépendances des sélecteurs
- ...



## HTML5 / CSS



I. Présentation du www

II. HTML5

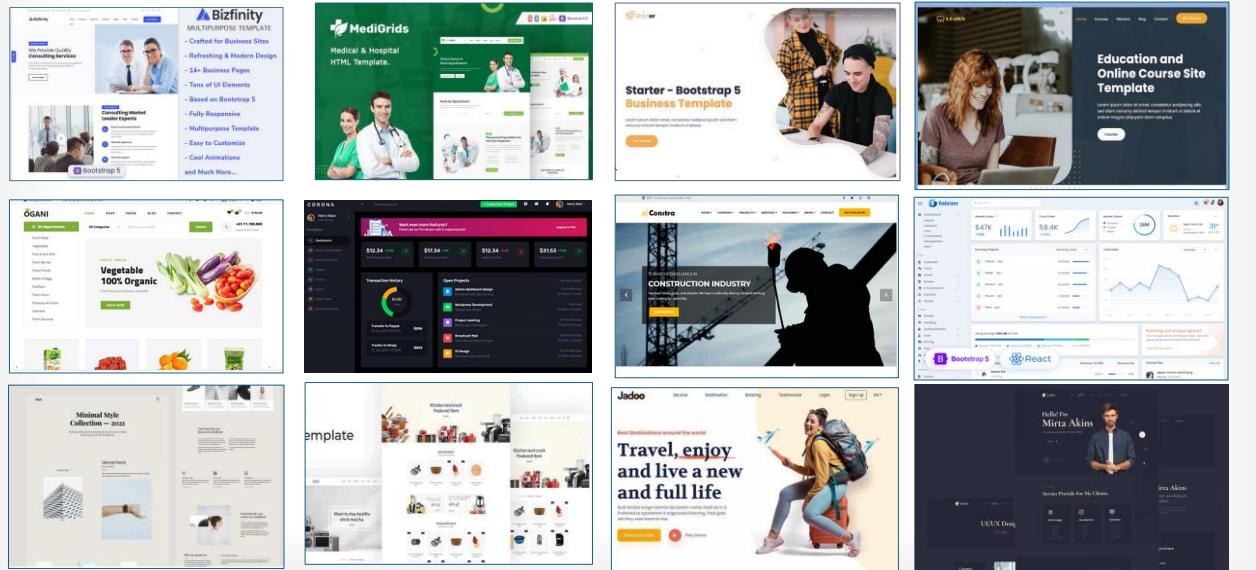
III. CSS / SCSS

**IV. Templates responsives**



## Que de choix ...

Dans votre moteur de recherche : template html5 responsive bootstrap5 gratuit

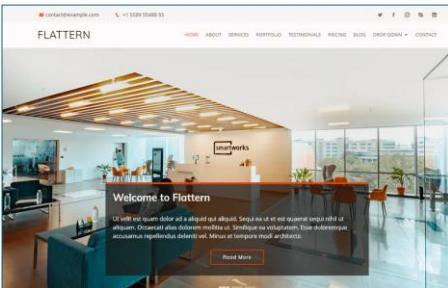


73



## Et pour le cours...

1. Télécharger : <https://bootstrapmade.com/flattern-multipurpose-bootstrap-template>



Modifier:

- Le titre du site
- Un menu de navigation
- Du texte sur la page d'accueil

2. Télécharger sur E-campus le fichier "IENAC\_sans\_flask" (version simplifiée du template flattern adaptée au cours)



## Infrastructure web



### I. Le serveur web Flask

### II. Principe de routage

### III. Passage de paramètres

### IV. Moteur de template JINJA

75

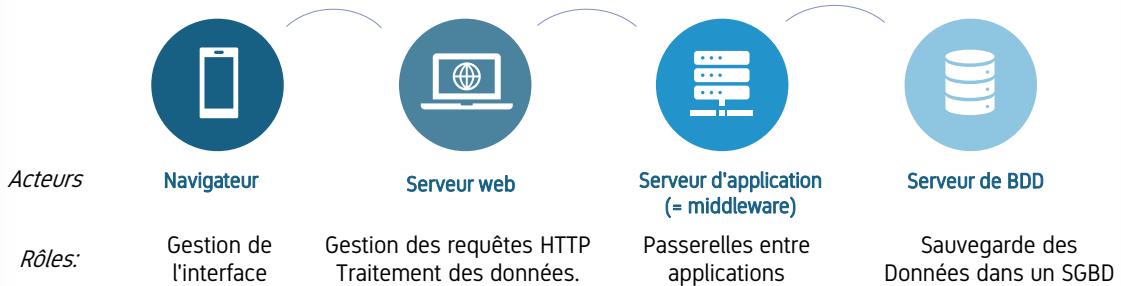
75



## Introduction

### Infrastructure web

La mise en place d'une application web fait intervenir plusieurs acteurs, chacun ayant un rôle bien particulier.



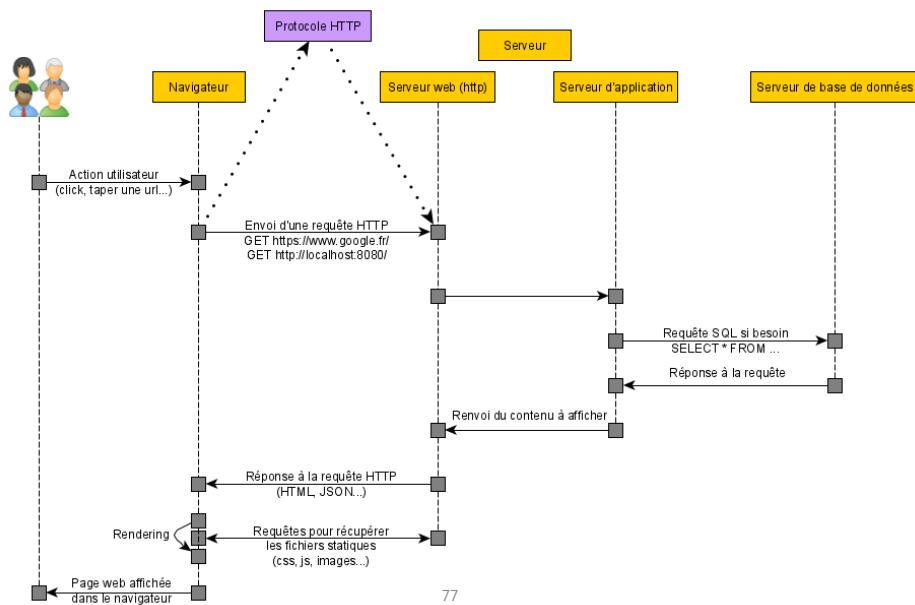
Une des principales difficultés du cours web est la compréhension des échanges entre ces différents acteurs (qui fait quoi et quand?) ainsi que l'apprentissage des différentes technologies propres à chaque acteur.

76



## Problématique: échange entre les acteurs

Diagramme de séquence



77

77



## Problématique: Technologies et langages

*Choix par acteur*



Acteurs	Navigateur	Serveur web	Middlewares	Serveur de BDD
Technologies	Chrome Firefox ....	Serveur de dév. Flask	JINJA2: Moteur de template	mysql-server
Langages	HTML, CSS, javascript	Python	Jinja	sql

FLASK est un micro framework applicatif, basé sur Python. Il permet de construire des applications web (réponses aux requêtes http, gestion du code applicatif).

Autres solutions: *Apache2 (mod-python)*, *Django*, *Cherrypy*, *Web2py* ....

JINJA est un moteur de template permettant de:

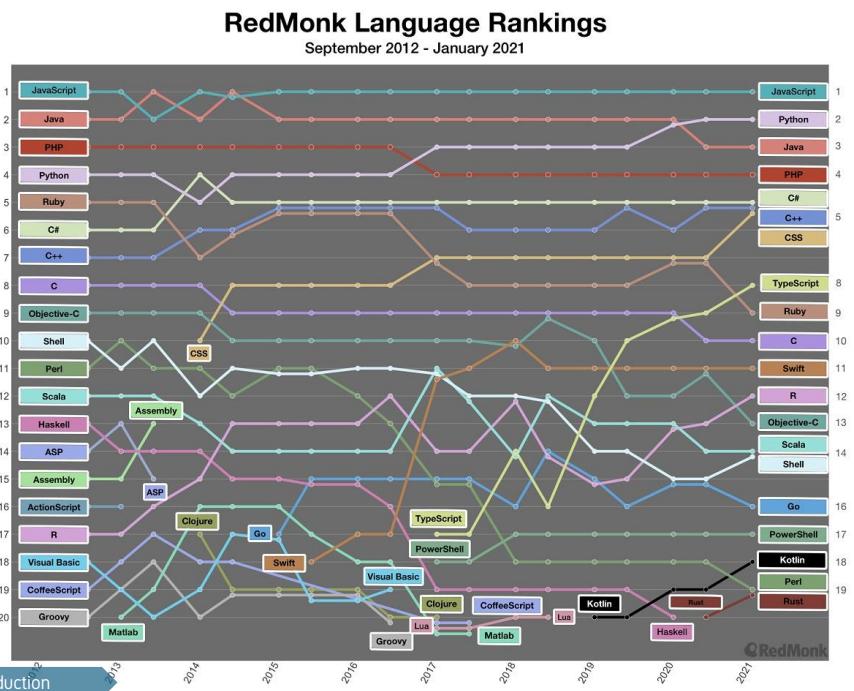
- Générer des fichiers HTML
- Traiter des paramètres envoyés par le serveur web



## Langage rankings

Source:

<https://www.blogdumoderateur.com/langages-programmation-evolution-tendances-communutes-emploi/>



79



## Premier script flask

```
from flask import Flask ..... importation de la classe Flask
app = Flask(__name__) ..... Instanciation de l'objet app

@app.route("/") ..... déclaration de la route
def index(): ..... déclaration de la vue
    return '<h2>Bienvenu sur index</h2><p>Test du serveur</p>'

if __name__ == '__main__':
    app.run(host='localhost', port=8080, DEBUG=True)
```

Contenu html affiché dans le navigateur en réponse à l'url: <http://localhost:8080/>

Nom ou adresse du serveur

Port d'écoute

2

Infrastructure web

Introduction

80

80



# Démarrer / arrêter le serveur web Flask

## Démarrer le serveur

### 1. Depuis un terminal:

- cd ..chemin..../web
- python run.py
- Dans le navigateur web:  
<http://localhost:8081>

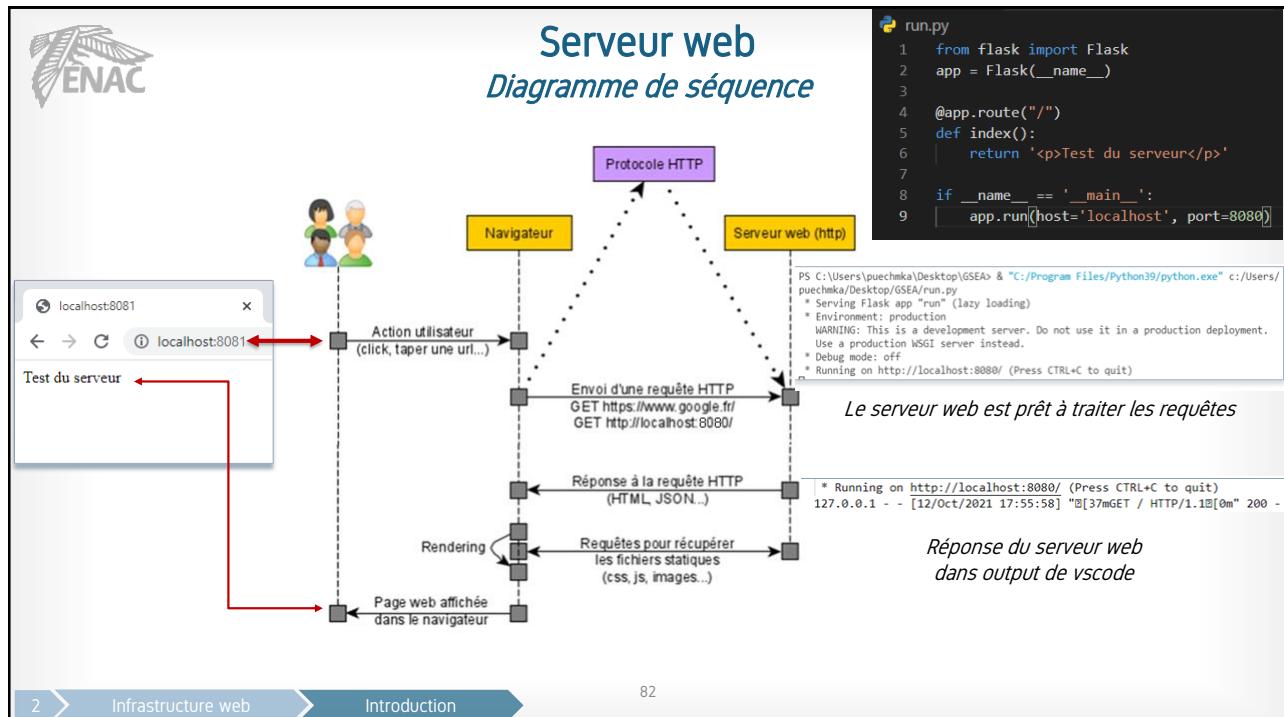
### 2. Depuis vscode:

- Ouvrir run.py
- Cliquer sur puis sur:



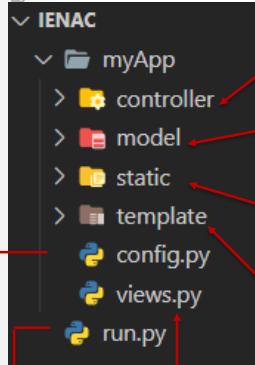
## Arrêter le serveur

Dans le terminal déjà ouvert, taper: Ctrl+C





## Fichiers et répertoires "applicatifs"

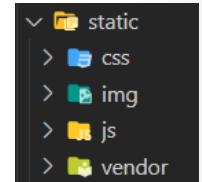


'controller' contient des fichiers python de traitement qui modifient les données du modèle et de la vue.

'model' contient des fichiers python qui gèrent le traitement des données avec le SGBD .

'static' contient les fichiers statiques de l'application, ie les fichiers qui ne seront jamais modifiés par le serveur.

'template' contient les fichiers HTML correspondants aux différentes pages de l'application.



- Le fichier 'views.py' définit les routes de l'application.
- Le fichier 'run.py' permet de lancer le serveur web et exécute l'application.
- Le fichier 'config.py' définit les variables d'environnement et de configuration



## run, views et config

```
from myApp.config import DEBUG, WEB_SERVER
from myApp.views import app
if __name__ == '__main__':
    app.run(
        host = WEB_SERVER['host'],
        port= WEB_SERVER['port'],
        debug=DEBUG
    )
from flask import Flask, render_template
app = Flask(__name__)
app.template_folder = "template"
app.static_folder = "static"
app.config.from_object('myApp.config')
@app.route("/")
def index():
    return render_template("index.html")
```

```
config.py
ENV = "development"
DEBUG = True
SEND_FILE_MAX_AGE_DEFAULT = 0 #vider le cache
SECRET_KEY="maCleSuperSecurisee"

#Configuration du serveur web
WEB_SERVER = {
    "host": "localhost",
    "port": 8080,
}

#Configuration du serveur de BDD
DB_SERVER = {
    "user": "root",
    "password": "mysql",
    "host": "localhost",
    "port": 3306, #8889 si MAC
    "database": "gsea_cours", #nom de la BDD
    "raise_on_warnings": True
}
```



## Infrastructure web



I. Le serveur web Flask

II. Principe de routage

III. Passage de paramètres

IV. Moteur de template JINJA

85

85



## Le routage

*Le routage* est la façon dont une application répond à une URL et au type de requête HTTP (GET, POST...). Les fichiers de routage peuvent être définis dans le fichier "views.py".

URL de la requête  
Méthode de la requête  
Réponse à la requête

```
@app.route("/", methods=['GET', 'POST'])
def index():
    return "Bonjour, tout le monde"
```

views.py

```
myApp > 🐍 views.py > ...
1   from flask import Flask
2   app = Flask(__name__)
3
4   @app.route("/")
5   def index():
6       return "Bonjour, tout le monde"
```

- **@app.route** est appelé "la route". Il lit l'URL déclarée dans le navigateur
- **def** est appelé "la vue". Il s'agit d'une fonction retournant la réponse du serveur web.



## Le routage

*Exemple 1: Gestion des URL*

*Fichier HTML:*  
(ex: template/index.html)

*URL du navigateur*

localhost:8080

*Routage dans views.py*

```
@app.route("/")
def index():
    return render_template("index.html")
```

<a href="ajouter">  
Ajouter  
</a>

localhost:8080/ajouter

<a href="ajouter?couleur=grey">  
Ajouter  
</a>

localhost:8080/ajouter?couleur=grey

<a href="ajouter/grey">  
Ajouter  
</a>

localhost:8080/ajouter/grey

```
@app.route("/ajouter")
@app.route("/ajouter/<couleur>")
def ajouter(couleur=''):
    return "Page ajouter";
```

2 Infrastructure web

Principe du routage

87

87



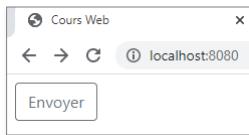
## Le routage

*Exemple 2: cas des formulaires*

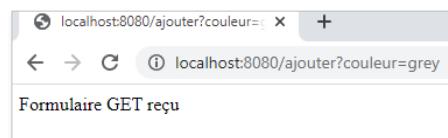
*Fichier HTML*

*Routage dans views.py*

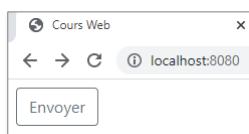
```
<form action="ajouter" method="GET">
<input type="hidden" name="couleur" value="grey" />
<button type="submit">Envoyer</button>
</form>
```



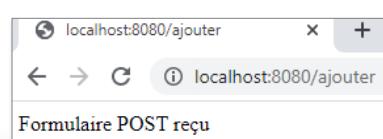
```
@app.route("/ajouter", methods=["GET"])
def ajouter():
    return "Formulaire GET reçu"
```



```
<form action="ajouter" method="POST">
<input type="hidden" name="couleur" value="grey" />
<button type="submit">Envoyer</button>
</form>
```



```
@app.route("/ajouter", methods=["POST"])
def ajouter():
    return "Formulaire POST reçu"
```



2 Infrastructure web

Principe du routage

88

88



## Les réponses du serveur

### ▪ render\_template()

La méthode render\_template() retourne une page HTM

L qui sera générée par le moteur de template JINJA.

Possibilité de rajouter un ou plusieurs paramètres.

```
from flask import Flask, render_template
@app.route("/")
def index():
    return render_template("index.html",
                           message="Bonjour", step=1)
```

### ▪ redirect()

La méthode redirect() redirige vers une autre URL.

Celle-ci peut-être:

- Une route
- Une vue
- Une URL externe

```
from flask import Flask, redirect
@app.route("/accueil")
def accueil():
    return redirect('/')
    return redirect(url_for('index'))
    return redirect("http://www.enac.fr")
```

### ▪ Return string

```
@app.route("/accueil")
def accueil():
    return "Vous êtes sur la page Accueil"
```

2 >

Infrastructure web

Principe du routage

89

89



## Les réponses du serveur

### ▪ Return JSON

```
from flask import Flask, jsonify
@app.route("/accueil")
def accueil():
    formation = ['GSEA', 'ISESA', 'IENAC', 'MCTA']
    annee = [2022, 2021, 2020]
    data = {"formation": formation, "annee": annee}
    return jsonify(data)
```

### ▪ Error handling

La méthode make\_response convertit la fonction de vue en un objet, instance de la classe Response.

```
from flask import Flask, make_response
@app.errorhandler(404)
def page_not_found():
    return make_response(render_template("404.html"), 404)

@app.errorhandler(400)
def bad_request():
    return make_response(render_template("400.html"), 400)

@app.errorhandler(500)
def server_error():
    return make_response(render_template("500.html"), 500)
```

2 >

Infrastructure web

Principe du routage

90



# Infrastructure web



I. Le serveur web Flask

II. Principe de routage

**III. Passage de paramètres**

IV. Moteur de template JINJA

91

91



## Passage de paramètres

Le passage de paramètres peut s'effectuer:

- Du client vers le serveur par les url ou les formulaires.
- Du serveur vers le client par réponse à la requête HTTP.

### Cas 1: client => serveur

Pour accéder aux données entrantes dans Flask, vous devez utiliser l'objet **request**.

L'objet request contient toutes les données entrantes de la requête, qui incluent, entre autres, le type mime, le référent, l'adresse IP, les données brutes, la méthode HTTP et les en-têtes.

Comment importer l'objet request à partir de la bibliothèque Flask:

```
from flask import Flask, request
```



## Passage de paramètres

### Cas 1: client => serveur

Client

index.html

Serveur

views.py

#### Données transmises dans URL

Ex 1: <http://localhost:8080/ajouter?couleur=grey&step=1>

Ex 2: <http://localhost:8080/ajouter/grey/1>

```
@app.route("/ajouter")
def ajouter():
    print(request.args) #Liste les données reçues
    color = request.args.get('couleur')
    step = request.args.get('step')
    return "Les données sont:"+color+" "+step

@app.route("/ajouter/<color>/<step>")
def ajouter(color="", step ""):
    print([color,step]) #Liste les données reçues
    return "Les données sont:"+color+" "+step
```

#### Données transmises par formulaire

```
<form action="ajouter" method="POST">
    <input type="text" name="couleur" value="grey" />
    <input type="hidden" name="step" value="1" />
    <button type="submit">Envoyer</button>
</form>
```

```
@app.route("/ajouter", methods=["POST"])
def ajouter():
    print(request.form) #Liste les données reçues
    color = request.form['couleur']
    step = request.form['step']
    return "Les données sont:"+color+" "+step
```

2 &gt;

Infrastructure web

Passage de paramètres

93

93



## Passage de paramètres

### Cas 2: serveur => client

views.py

- Dans le render\_template()

```
return render_template("index.html",
message="Bonjour", step=1)
```

```
{% if step==1 %}
    <div class="info">{{ message }}</div>
{% endif %}
```

- Dans le redirect()

```
views.py
Exemple 1
@app.route("/accueil")
def accueil():
    return redirect('/page2/24')

Exemple 2
@app.route("/accueil")
def accueil():
    return redirect(url_for('page2', age='24'))
```

```
@app.route("/page2/<age>")
def page2(age=""):
    return "J'ai "+age+" ans"
```

```
@app.route("/page2")
def page2():
    age = request.args.get('age')
    return "J'ai "+age+" ans"
```

2 &gt;

Infrastructure web

Passage de paramètres

94

94



# Infrastructure web



I. Le serveur web Flask

II. Principe de routage

III. Passage de paramètres

IV. Moteur de template JINJA

95

95



## Templating Moteur de template JINJA2



Le serveur Flask est basé sur le moteur de template JINJA2.

Ce moteur va nous permettre de définir le contenu des fichiers HTML au moment de l'envoi de la réponse aux requêtes HTTP adressées à notre serveur.

```
myApp > views.py > ...
1  from flask import Flask, render_template
2  app = Flask(__name__)
3
4  @app.route("/")
5  def index():
6      return render_template("index.html",
7                            message="Bonjour",
8                            step=1)
```

Génère un modèle de vue

return render\_template("index.html",  
 message="Bonjour",  
 step=1)

Nom du fichier html à afficher

Paramètre(s) transmis  
au fichier index.html



## Les fichiers de vues

Les fichiers de vue sont des fichiers HTML dans lesquels il est possible d'insérer des paramètres en provenance du serveur web (ie du fichier views.py). Tous les fichiers de vue sont enregistrés dans le répertoire "template".

```
p > 🐍 views.py > ...
from flask import Flask, render_template
app = Flask(__name__)

@app.route("/")
def index():
    return render_template("index.html", title="Accueil")
```

The diagram illustrates the rendering process. On the left, a terminal window shows the Python code in `views.py`. A red box highlights the line `return render_template("index.html", title="Accueil")`. To the right, a code editor shows the generated `index.html` file. A red box highlights the Jinja2 template code: `<h1>{{ title }}</h1>` and `<p>Bienvenu sur {{ title }}</p>`. Red arrows point from these highlighted sections to the corresponding parts in the browser window on the right. The browser window shows the title "Accueil" and the content "Bienvenu sur Accueil".

Pour insérer un paramètre provenant du serveur dans le code HTML, il faut utiliser la syntaxe du moteur de template JINJA. ex: {{ title }}

## Syntaxe JINJA

Balise de scriptlet: utilisée pour incorporer du code python pour le flux de contrôle (**for**), les conditions (**if**) ...

**{% ... %}**  
ex: `{% include('../template/header') %}`  
`{% if .... %} {% endif %}`  
`{% for ....%} {% endfor %}`

---

**{{ ... }}** renvoie la valeur de l'expression  
ex: `{{ title }}`

---

**{# ... #}** Balise de commentaire, le contenu de la balise sera automatiquement ignoré.  
ex: `{# contenu à commenter #}`

---

**{% set ... %}** Déclaration d'une nouvelle variable dans le fichier html  
ex: `{% set a = "bcd" %} {{ a }}`

2 > Infrastructure web > JINJA2

98

98

49



## Syntaxe JINJA

Exemple: Afficher les données provenant de plusieurs paramètres.

```
from flask import Flask, render_template
app = Flask(__name__)

@app.route("/")
def index():
    pays= {
        "nomPays": "France",
        "capitale": "Paris",
        "surface": 551695,
        "mers": ["Océan Atlantique", "Manche", "Méditerranée"],
        "airports": [
            { "OACI": "LFBD", "ville": "Bordeaux" },
            { "OACI": "LFBL", "ville": "Biarritz" },
            { "OACI": "LFLL", "ville": "Lyon" }
        ]
    }

    return render_template("index.html",
                           titre="Aéroports",
                           aeroports=pays[ 'airports' ])
```

OACI	Ville
LFBD	Bordeaux
LFBL	Biarritz
LFLL	Lyon

```
<h1>{{ titre }}</h1>
<!-- Si le paramètre aeroports existe --&gt;
{% if aeroports %}
    {% for ad in aeroports %}
        &lt;tr&gt;
            &lt;td&gt;{{ ad.OACI }}&lt;/td&gt;
            &lt;td&gt;{{ ad.ville }}&lt;/td&gt;
        &lt;/tr&gt;
    {% endfor %}
&lt;/tbody&gt;
&lt;/table&gt;
{% endif %}</pre>

```



## Gestion d'un template

L'interface de votre page index.html est, pour l'instant, peu développée.

Dans cette partie, nous allons voir comment rendre compatible un template HTML/CSS avec le serveur flask.



Pour transférer une template HTML/CSS sur un serveur web FLASK, vous devrez suivre 4 étapes .



## Gestion du template

**Etape 1:** Téléchargement et décompression du template HTML, transfert des fichiers.

**Etape 2:** Création des fichiers et repertoires indispensables pour une application Flask.

**Etape 3:** Modifications des liens entre fichiers.

**Etape 4:** Optimisation du code redondant.

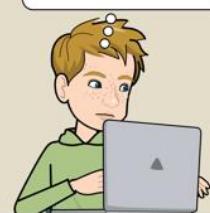


## Etape 1 et 2: Transfert des fichiers

*Le transfert des fichiers du template vers l'application web consiste à :*

1. *Copier les fichiers statiques (js, css, img, vendor) dans le répertoire "static"*
2. *Copier les fichiers HTML dans le répertoire "template" (uniquement index.html et contact.html)*
3. *Créer les fichiers run.py, config.py et views.py*

Bon, normalement, j'ai bien suivi les instructions du prof. J'ai téléchargé le template depuis Internet puis je l'ai décompressé.



A la racine, j'ai créé un répertoire "myApp", puis j'ai déplacé "static" dedans. J'ai également créé le fichier "run.py".



A la racine, j'ai créé un répertoire "myApp", puis j'ai déplacé "static" dedans. J'ai également créé le fichier "run.py".

Enfin, dans "myApp", j'ai créé "config.py" et "views.py" avec au moins une route pour accéder à "index.html". J'ai également rajouté le répertoire template dans lequel j'ai déplacé tous les fichiers HTML. Bon, tout semble correct, voyons ce que cela donne.

Horreur!  
Qu'est-ce qu'il s'est passé?

## Etape 1 et 2: Transfert des fichiers

2 > Infrastructure web > Templating 103

103

Oui, pour moi, c'est pareil. Quelle erreur retourne le serveur web?

Regarde, Maelys, mon site ne s'affiche pas correctement.

Erreur 404! Le serveur ne trouve pas le fichier dans le répertoire "assets".

/assets - 404  
/img - 404  
/vendors - 404

Bien vu! Utilisons la commande CTRL+H de vscode pour aller vite.

Mais oui, bien sûr. "assets" n'existe plus. Il a été remplacé par "static". Il suffit de le remplacer dans tous les fichiers HTML. Hum! Cela risque d'être long ...

J'ai remplacé "assets" par "/static" dans les .html et tout marche correctement!

La bonne humeur de certains fait plaisir à voir...

## Etape 3: Modification des liens entre fichiers

Pour corriger le problème d'affichage, il faut remplacer dans tous les fichiers HTML:

- assets/ par /static/

Pour plus d'efficacité, utiliser la commande CTRL+H de vscode.

2 > Infrastructure web > Templating 104

104



## Modification du menu de navigation

ACCUEIL COURS ▾ SGBD FICHIERS CONTACT SE CONNECTER

Dans index.html, repérer la balise <nav>. Tous les liens hypertextes sont redirigés vers des fichiers html. Pour que ces liens fonctionnent avec le serveur Flask:

1. Créer les routes suivantes dans `views.py`

```
#page se connecter
@app.route("/connecter")
def connecter():
    return render_template("connecter.html")
```

route	render_template
/connecter	connecter.html
#footer ( pour le menu nous contacter)	
/html	cours/html.html
/css	cours/css.html
/bootstrap	cours/bootstrap.html
/SGBD	cours/sgbd.html
/fichiers	cours/fichiers.html

2. Modifier les liens du menu dans `index.html`

```
<ul>
    <li><a class="active" href="/">Accueil</a></li>
    <li class="dropdown"><a href="#"><span>Cours</span>
        <i class="bi bi-chevron-down"></i></a>
        <ul>
            <li><a href="/HTML">HTML</a></li>
            <li><a href="/CSS">CSS</a></li>
            <li><a href="/bootstrap">Bootstrap</a></li>
        </ul>
    </li>
    <li><a href="/SGBD">SGBD</a></li>
    <li><a href="/fichiers">Fichiers</a></li>
    <li><a href="#footer">Contact</a></li>
    <li><a href="/connecter">Se connecter</a></li>
</ul>
```

105

105

Dans votre éditeur vscode, ouvrez tous les fichiers html: index, connecter..etc... Que remarquez-vous?

Il y a des parties de code identiques ou presque dans tous ces fichiers.

### Etape 4: Optimisation du code redondant

Créer un nouveau répertoire nommé "layout" dans template. Et à l'intérieur, créer 5 nouveaux fichiers: head.html, nav.html, breadcrumbs.html, footer.html et scripts.html. Vous devez obtenir l'arborescence suivante:

```
template
  cours
  layout
    head.html
    nav.html
    breadcrumbs.html
    footer.html
    scripts.html
  connecter.html
  index.html
```

106

Dans le fichier index.html,  
couper tout le code  
compris entre <head> et  
</head>. Puis, collez-le  
dans le fichier head.html.



## Etape 4: Optimisation du code redondant

```
index.htmlx index.html

C:\Users\puechmka\Desktop\IENAC21> cours > templateCourses > IENAC_sans_flask > myApp > template > index.html

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="utf-8">
6   <meta content="width=device-width, initial-scale=1.0" name="viewport">
7
8   <title>Flattern Bootstrap Template - Index</title>
9   <meta content="" name="description">
10  <meta content="" name="keywords">
11
12  <!-- Favicons -->
13  <link href="/static/img/favicon.png" rel="icon">
14  <link href="/static/img/apple-touch-icon.png" rel="apple-touch-icon">
15
16
17  <!-- Google Fonts -->
18  <link href="https://fonts.googleapis.com/css?family=Open+Sans:300,300i,400,400i,600,600i,Poppins:300,300i,400,400i,500,500i,600,600i,700,700i" rel="stylesheet">
19
20  <!-- Vendor CSS Files -->
21  <link href="/static/vendor/animate.css/animate.min.css" rel="stylesheet">
22  <link href="/static/vendor/aos/aos.css" rel="stylesheet">
23  <link href="/static/vendor/bootstrap/css/bootstrap.min.css" rel="stylesheet">
24  <link href="/static/vendor/bootstrap-icons/bootstrap-icons.css" rel="stylesheet">
25  <link href="/static/vendor/boxicons/css/boxicons.min.css" rel="stylesheet">
26  <link href="/static/vendor/lightbox/css/lightbox.min.css" rel="stylesheet">
27  <link href="/static/vendor/swiper/swiper-bundle.min.css" rel="stylesheet">
28
29  <!-- Template Main CSS File -->
30  <link href="/static/css/style.css" rel="stylesheet">
```

```
index.html <br/>  
myApp > template > index.html > html > body >  
1   <!DOCTYPE html>  
2   <html lang="fr">  
3       <head>  
4           {% set titre="Cours web" %}  
5           {% include("layout/head.html") %}  
6       </head>  
7   </html>
```

107

Procéder de la même façon pour déplacer le <header> vers le fichier nav.html. Idem pour la <section id="breadcrumbs"> vers le fichier breadcrumbs.html et <footer> vers footer.html. script.html doit aussi comprendre toutes les balises <script> présentes avant le </body>.



## Etape 4: Optimisation du code redondant

nav.html <header>....</header>

footer.html <footer>...</footer> + <a href="#" ...>...</a>

scripts.html Tous les <scripts></scripts>

breadcrumbs.html <section id="breadcrumbs"></section>

Et pour finir, dans index.html puis page-login.html, rappeler les fichiers du répertoire layout grâce à l'instruction JINJA {% include()%}



## Etape 4: Optimisation du code redondant

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    {%- set titre="Cours web" %}
    {%- include("layout/head.html") %}
  </head>

  <!-- ===== Header ===== -->
  {%- include("layout/nav.html") %}

  <!-- ===== Hero Section ===== -->
  <section id="hero">
  </section><!-- End Hero -->

  <main id="main">
  </main><!-- End #main -->

  <!-- ===== Footer ===== -->
  {%- include("layout/footer.html") %}
  {%- include("layout/scripts.html") %}

</body>
</html>
```

109



## Traitement des données



### I. Données enregistrées en BDD

- Requêtes SELECT
- Requêtes DELETE
- Requête INSERT
- Requête UPDATE

### II. Sessions et cookies

### III. Données enregistrées dans un fichier

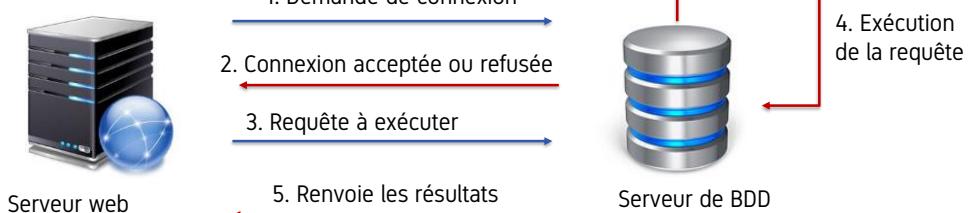
- Upload d'un fichier
- Importer les données
- Exporter les données

110

110



## Données enregistrées en BD



Transactions entre le serveur web et le serveur de BDD :

1. Demande de connexion du serveur web au serveur de BDD
2. Le serveur de BDD accepte ou refuse la connexion
3. Une fois la connexion acceptée, le serveur web envoie la requête à exécuter
4. Le serveur de BDD exécute la requête
5. Le serveur de BDD renvoie les résultats au serveur web
6. Le serveur web ferme la connexion

```
config.py
#Configuration du serveur de BDD
DB_SERVER = {
    "user": "ienac",
    "password": "ienac",
    "host": "localhost",
    "port": 3306, #8889 si MAC
    "database": "ienac_cours",
    "raise_on_warnings": True
}
```

3 > Traitement des données > BDD

111

111



## Création de la BDD

ienac_cours identification	
	idUser : int(11)
	nom : varchar(50)
	prenom : varchar(50)
	mail : varchar(100)
	login : varchar(100)
	motPasse : varchar(100)
#	statut : int(2)
	avatar : varchar(20)

1. Créer la BDD ienac\_cours
  2. Créer la table identification
- Ou
2. Importer le fichier identification.sql

Base de données : ienac_cours » Table : identification							
IDUser	nom	prenom	mail	login	motPasse	statut	avatar
1	Blériot	Louis	louis.blériot@enac.fr	a	a	0	8.png
2	Boucher	Hélène	helene.boucher@enac.fr	b	b	1	3.png
3	admin	admin	admin@enac.fr	admin	admin	0	10.png
9	Ader	Clément	clement.ader@enac.fr	ader	ader	1	2.png
10	Aubrun	Émile	emile.aubrun@enac.fr	aubrun	aubrun	1	4.png
11	Carpentier	Roger	roger.carpentier@enac.fr	carpentier	carpentier	1	6.png
12	Farman	Henri	henri.farman@enac.fr	farman	farman	1	8.png
13	Daurat	Didier	didier.daurat@enac.fr	daurat	daurat	1	9.png
14	Mermoz	Jean	jean.mermoz@enac.fr	mermoz	mermoz	1	11.png
15	Wright	Katharine	katharine.wright@enac.fr	wright	wright	1	1.png
16	Peltier	Therese	therese.peltier@enac.fr	peltier	peltier	1	3.png
17	De Laroche	Raymonde	raymonde.delaroche@enac.fr	delaroche	delaroche	1	1.png
18	Johnson	Amy	amy.johnson@enac.fr	johson	johson	1	7.png
19	Earhart	Amelia	amelia.earhart@enac.fr	earhart	earhart	1	11.png
20	Tereshkova	Valentina	valentina.tereshkova@enac.fr	tereshkova	tereshkova	1	1.png

3 > Traitement des données > BDD

112

112



## Connexion au serveur de BDD

bdd.py

```

import mysql.connector
from mysql.connector import errorcode
from ..config import DB_SERVER

# connexion au serveur de BDD
def connexion():
    cnx = ""
    try:
        cnx = mysql.connector.connect(**DB_SERVER)
        error = None
    except mysql.connector.Error as err:
        error = err
        if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
            print("Mauvais login ou mot de passe")
        elif err.errno == errorcode.ER_BAD_DB_ERROR:
            print("La Base de données n'existe pas.")
        else:
            print(err)
    return cnx, error # error: remonte problème connexion

# fermeture de la connexion au serveur de BDD
def close_bd(cursor, cnx):
    cursor.close()
    cnx.close()

```

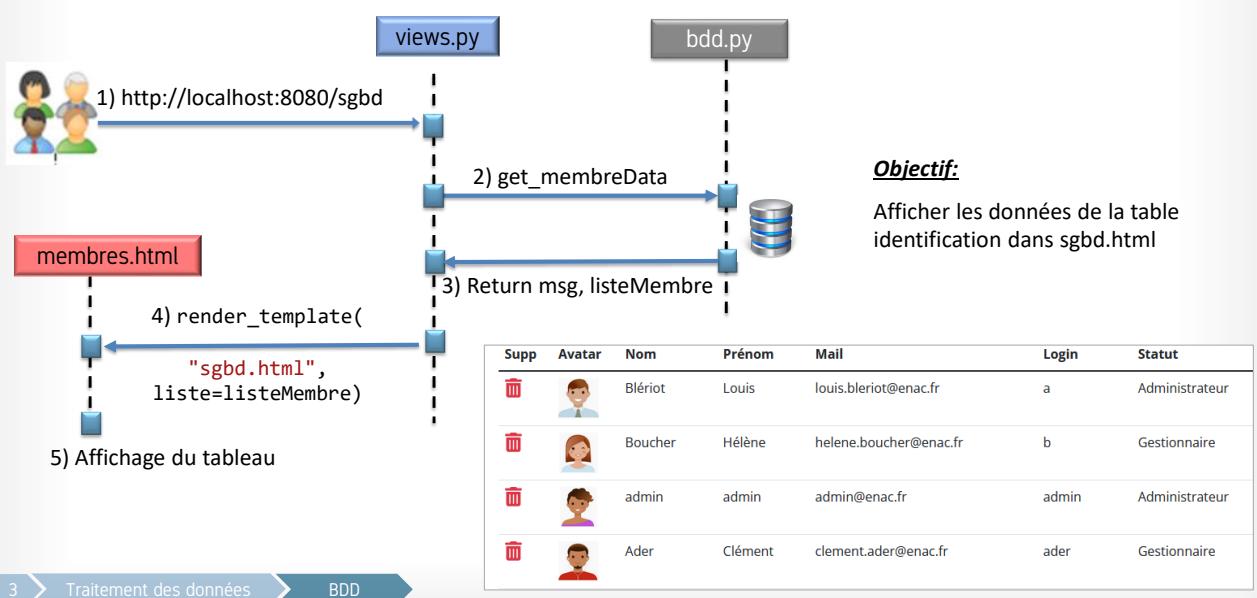
3 &gt; Traitement des données

BDD

113

113

## Rechercher des données en BDD: *requête SELECT Séquence*



114



## Rechercher des données en BDD: requête SELECT

```
# Toutes les données de la table membres
def get_membreData():
    try:
        cnx, error = connexion()
        if error is not None:
            return error, None # Problème connexion BDD
        cursor = cnx.cursor(dictionary=True)
        sql = "SELECT * FROM identification"
        cursor.execute(sql)
        listeMembre = cursor.fetchall()
        close_bd(cursor, cnx)
        msg = "OKmembres"
    except mysql.connector.Error as err:
        listeMembre = None
        msg = "Failed get membres data : {}".format(err)
    return msg, listeMembre
```

bdd.py

```
from .model import bdd as bdd
@app.route("/sgbd")
def sgbd():
    msg, listeMembre = bdd.get_membreData()
    print(msg)
    return render_template("membres.html",
                           liste=listeMembre, infoErr = msg
    )
```

views.py

```
{% if infoErr and infoErr != "OKmembres" %}
    <div class="alert alert-danger">
        {{info}}</div>
{% endif %}
{% if liste %}
    {{ liste }}
{% endif %}
```

sgbd.html

3 &gt; Traitement des données &gt; BDD

115

115



## Rechercher des données en BDD: requête SELECT

Modifier sgbd.html pour afficher le données sous forme de tableau

```
{% if liste %}
    # liste #
<table class="table">
    <thead>
        <tr>
            <th>Supp</th>
            <th>Avatar</th>
            <th>Nom</th>
            <th>Prénom</th>
            <th>Mail</th>
            <th>Login</th>
            <th>Statut</th>
        </tr>
    </thead>
```

sgbd.html

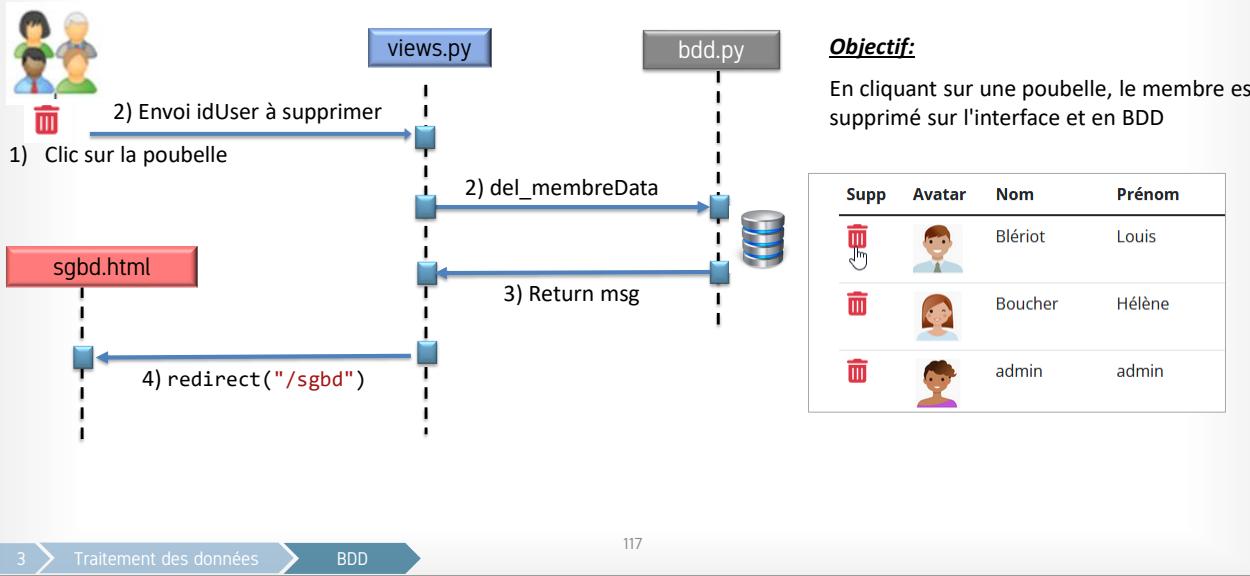
```
<tbody>
    {% for user in liste %}
        <tr>
            <td><i class='fas fa-trash-alt text-danger fs-4'></i></td>
            <td></td>
            <td>{{ user.nom }}</td>
            <td>{{ user.prenom }}</td>
            <td>{{ user.mail }}</td>
            <td>{{ user.login }}</td>
            <td>
                {% if user.statut == 0 %}
                    Administrateur
                {% else %}
                    Gestionnaire
                {% endif %}
            </td>
        </tr>
    {% endfor %}
</tbody>
</table>
{% endif %}
```

3 &gt; Traitement des données &gt; BDD

116



## Supprimer des données en BDD: requête *DELETE* Séquence

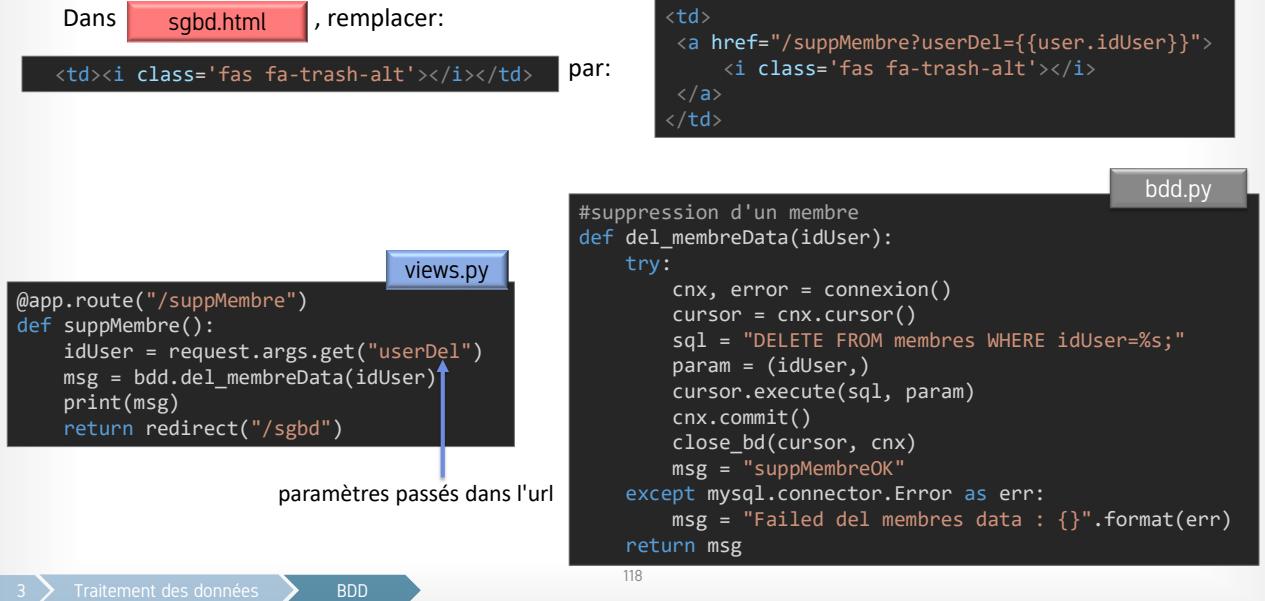


117

117



## Supprimer des données en BDD: requête *DELETE*



118

118



## Supprimer des données en BDD: requête DELETE

### Gestion des messages

Il est souvent utile d'apporter à l'internaute une information indiquant si l'action demandée s'est bien déroulée ou pas.

```
views.py
@app.route("/sgbd")
@app.route("/sgbd/<infoMsg>")
def sgbd(infoMsg=''):
    msg, listeMembre = bdd.get_membreData()
    print(msg)
    return render_template("cours/sgbd.html",
                          liste=listeMembre, infoErr=msg, info=infoMsg)

@app.route("/suppMembre")
def suppMembre():
    idUser = request.args.get("userDel")
    msg = bdd.del_membreData(idUser)
    print(msg)
    if msg == "suppMembreOK":
        return redirect("/sgbd/delOK")
    else:
        return redirect("/sgbd/delProblem")
```

```
sgbd.html
L'utilisateur a bien été supprimé

{% if info and info == "delOK" %}
    <div class="alert alert-success text-center">
        L'utilisateur a bien été supprimé
    </div>
{% endif %}

{% if info and info == "delProblem" %}
    <div class="alert alert-danger text-center">
        Problème avec la suppression de l'utilisateur
    </div>
{% endif %}
```

Problème avec la suppression de l'utilisateur

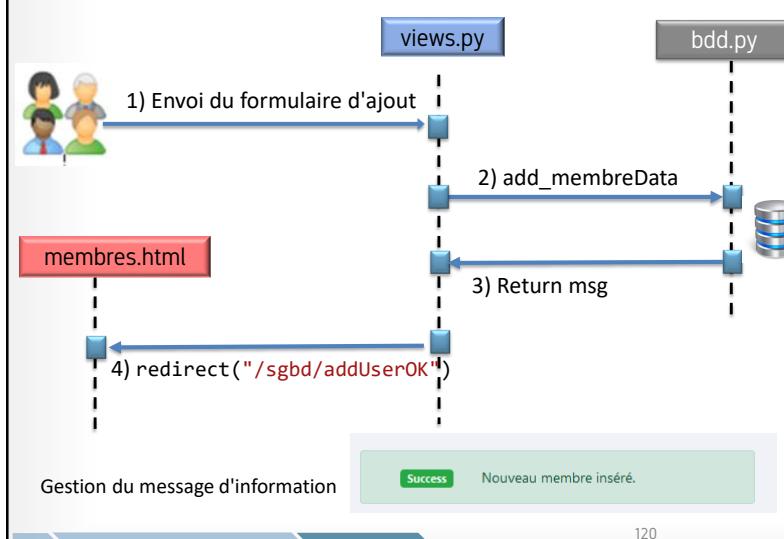
119

119



## Rajouter des données en BDD: requête insert

### Séquence



Formulaire d'ajout  
(A télécharger sur E-campus)

**S'enregistrer**

Nom

Prénom

E-mail

Login

Mot de passe

Gestionnaire  Administrateur

Choisissez un avatar

**ENREGISTRER**

120

120



## Rechercher des données en BDD: requête insert

sgbd.html

```
<form method="post" action="addMembre">

<h4>S'enregistrer</h4>
<input type="text" name="nom" required>
<input type="text" name="prenom" required>
<input type="email" name="mail" required>
<input type="text" name="login" required>
<input type="password" name="mdp" required>
<input type="radio" name="statut" value="1" checked /> Gestionnaire
<input type="radio" name="statut" value="0"> Administrateur

<h5>Choisissez un avatar</h5>
{%
    for i in range(1,13) %
        <input type="radio" name="avatar" value="{{i}}.png">
        
    {% endfor %}
}

<button type="submit">Enregistrer</button>

</form>
```

Formulaire simplifié

3 &gt; Traitement des données

&gt; BDD

121

Formulaire d'ajout  
(A télécharger sur E-campus)

S'enregistrer

  
  
  
  
  
 Gestionnaire  Administrateur  

Choisissez un avatar



























ENREGISTRER

121

```
@app.route("/addMembre", methods=['POST'])
def addMembre():
    nom = request.form['nom']
    prenom = request.form['prenom']
    mail = request.form['mail']
    login = request.form['login']
    motPasse = request.form['mdp']
    statut = request.form['statut']
    avatar = request.form['avatar']
    msg, lastId = bdd.add_membreData(nom, prenom,
    mail, login, motPasse, statut, avatar)
    if msg == "addMembreOK":
        return redirect("/sgbd/addUserOK")
    else:
        return redirect("/sgbd/addUserProblem")

{% if info and info == "addUserOK" %}
    <div class="alert alert-success text-center">
        Nouveau membre inséré
    </div>
{% endif %}

{% if info and info == "addUserProblem" %}
    <div class="alert alert-danger text-center">
        Problème insertion du nouveau membre
    </div>
{% endif %}
```

views.py

sgbd.html

## Rechercher des données en BDD: requête insert

bdd.py

```
#ajout d'un membre
def add_membreData(nom, idEquipe):
    try:
        cnx, error = connexion()
        cursor = cnx.cursor()
        sql = "INSERT INTO membres (nom, idEquipe)
VALUES (%s, %s);"
        param = (nom, idEquipe)
        cursor.execute(sql, param)
        lastId = cursor.lastrowid #dernier idUser créé
        cnx.commit()
        close_bd(cursor, cnx)
        msg = "addMembreOK"
    except mysql.connector.Error as err:
        msg = "Failed add membres data : {}".format(err)
    return msg, lastId
```

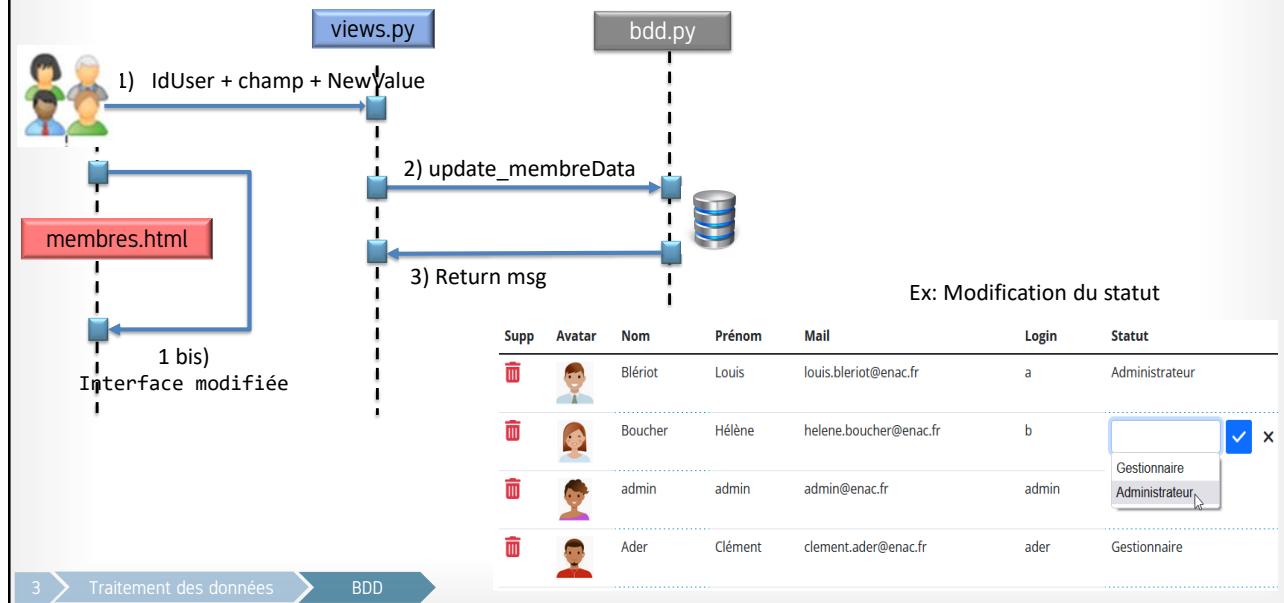
Gestion des messages d'information

122

122



## Modifier des données en BDD: requête update Séquence



123



## Modifier des données en BDD: requête update utilisation de bootstrap-x-editable

Grâce au plugin bootstrap-x-editable, il est possible de modifier facilement les données de l'idEquipe et du nom des membres.

### 1) Télécharger bootstrap-x-editable :

- Décompresser dans le répertoire vendors (<https://vitalets.github.io/x-editable>) - BS3 build
- Ou bien utiliser les liens CDN (<https://cdnjs.com/libraries/x-editable> -- version 1.5.0)

### 2) Rajouter le lien vers le CSS et JS dans membres.html

```
après le {% include("layout/head.html") %}  
<link rel="stylesheet" href="//cdnjs.cloudflare.com/ajax/libs/x-editable/1.5.0/bootstrap3-editable/css/bootstrap-editable.css" />
```

```
après le {% include("layout/scripts.html") %}  
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>  
<script src="//cdnjs.cloudflare.com/ajax/libs/x-editable/1.5.0/bootstrap3-editable/js/bootstrap-editable.min.js"></script>
```

3 > Traitement des données > BDD

124

124

 Dans **sgbd.html** remplacer

```
<td>{{ user.nom }}</td>
<td>{{ user.prenom }}</td>
<td>{{ user.mail }}</td>
<td>{{ user.login }}</td>
<td>{%
  if user.statut == 0 %
    Administrateur
  else %
    Gestionnaire
  endif %
}</td>
```

par

```
<td class='modif' data-url = 'updateMembre'
  data-name= 'nom' data-pk='{{user.idUser}}' >
  {{ user.nom }}</td>
<td>{{ user.prenom }}</td>
<td>{{ user.mail }}</td>
<td>{{ user.login }}</td>
<td class='modifSelect' data-url = 'updateMembre'
  data-name= 'statut' data-pk='{{user.idUser}}' >
  {%
    if user.statut == 0 %
      Administrateur
    else %
      Gestionnaire
    endif %
}</td>
```

- **data-url**: Les données seront envoyées vers la route updateMembre
- **data-pk**: pk=public key - idUser à modifier
- **data-name**: information concernant le champ à modifier

Initialisation du bootstrap editable pour classes modif et modifSelect. Rajouter après les <script>

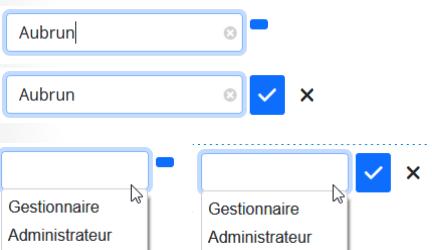
```
<script>
$.fn.editable.defaults.mode = 'inline'
$(".modif").editable()
$(".modifSelect").editable({
  type: 'select',
  source: [
    { value: 1, text: 'Gestionnaire' },
    { value: 0, text: 'Administrateur' }
  ]
})</script>
```

125

125

 **Modifier des données en BDD: requête update (suite)**

**Problème affichage icône**



Rajouter dans my.css

```
.glyphicon:before{
  font-size:1.4em;
  font-style: normal;
  font-family: 'FontAwesome';
  content: "\f00d";
}

.glyphicon-ok:before {content: "\f00c";}
```

**views.py**

```
@app.route("/updateMembre", methods=[ 'POST'])
def updateMembre():
    idUser = request.form['pk']
    champ = request.form['name']
    newvalue = request.form['value']
    msg = bdd.update_membreData(champ, idUser, newvalue)
    print(msg)
    return msg
```

**bdd.py**

```
#modification d'une donnée dans la table membre
def update_membreData(champ, idUser, newvalue):
    try:
        cnx, error = connexion()
        cursor = cnx.cursor()
        sql = "UPDATE membres SET "+champ+" = %s WHERE idUser = %s;"
        param = (newvalue, idUser)
        cursor.execute(sql, param)
        cnx.commit()
        close_bd(cursor, cnx)
        msg = "updateMembreOK"
    except mysql.connector.Error as err:
        msg = "Failed update membres data : {}".format(err)
    return msg
```

126



## Traitement des données



### I. Données enregistrées en BDD

- Requêtes SELECT
- Requêtes DELETE
- Requête INSERT
- Requête UPDATE

### II. Sessions et cookies

### III. Données enregistrées dans un fichier

- Upload d'un fichier
- Importer les données
- Exporter les données

127

127



## Les sessions et cookies

- Comment savoir si c'est la même personne qui se connecte à mon site web ?
- Comment mettre en place un système de panier ?
- Comment faire des statistiques d'accès à mon site web ?

*Plus généralement :*

Comment et où garder des informations sur les internautes qui se connectent sur un site web ?

- ✓ sous forme de **cookies** sur le **poste de l'internaute**
- ✓ sous forme de **session** sur le **serveur web**



## Les sessions et cookies

*Import du module session*

```
from flask import session
```

*Affectation de la variable de session*

```
session["id"] = valeur;
session["nom"] = valeur;
```

....

*Suppression de la session*

```
session.clear()
```

*Jinja 2: Existence et appel de la variable de session dans les fichiers html*

```
{% if session.id %}
    <p>Bonjour {{ session.nom }} {{ session.prenom }}</p>
{% endif %}
```



## Les sessions

Une session est une période délimitée pendant laquelle le serveur web peut enregistrer des données. Ces données persisteront tant que la session existe (en général jusqu'à ce que le client se déconnecte).

- Les données de session sont stockées sur le serveur.
- Le serveur Flask enregistre les sessions dans un dictionnaire.

```
<SecureCookieSession {
'IPclient': '127.0.0.1',
'UserAgent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:92.0) Gecko/20100101 Firefox/92.0',
'idUser': 2,
'mail': 'louis.blériot@enac.fr',
'nom': 'Blériot',
'prenom': 'Louis',
'statut': 'AD0'}
```



Quand je quitte un site web sur lequel je suis authentifié, je pense toujours à me déconnecter !!!

```
from flask import Flask, render_template, request, session
app = Flask(__name__)

@app.route("/")
def index():
    # déclaration des variables de sessions
    session["idUser"] = 2
    session["prenom"] = "Louis"
    session["nom"] = "Blériot"
    session["mail"] = "louis.blériot@enac.fr"
    session["statut"] = "AD0" # administrateur
    session["IPclient"] = request.remote_addr
    session["UserAgent"] = request.headers.get('User-Agent')

    # affichage des sessions dans le terminal
    print(session)

    return render_template("index.html")
```



## Les sessions

```
config.py > ...
1 ENV = "development"
2 DEBUG = True
3 SEND_FILE_MAX_AGE_DEFAULT = 0 #vider le cache
4 SECRET_KEY="maCleSuperSecurisee"
5

from flask import Flask, render_template
from .controller import function as f
views.py

app = Flask(__name__)

#page accueil
@app.route("/")
@app.route("/index/<infoMsg>")
def index(infoMsg=''):
    return render_template("index.html", info=infoMsg)

#authentification
@app.route("/login", methods=["POST"])
def testConnect():
    f.sessionTest()
    return redirect("/index/AuthOK")
```

Avec Flask, l'utilisation des sessions ne peut se faire qu'à l'intérieur d'une route.

```
from flask import session
controller/function.py

def sessionTest():
    # déclaration des variables de sessions
    session["idUser"] = 2
    session["prenom"] = "Louis"
    session["nom"] = "Blériot"
    session["mail"] = "louis.blériot@enac.fr"
    session["avatar"] = "8.png"

    # appel des variables de sessions (dans .py)
    nomComplet = session["prenom"] + session["nom"]

    # Modification d'une variable de session
    session["mail"] = "blériot@gmail.com"

    # test de l'existence de session["idUser"]
    if "idUser" in session:
        print("idUser existe dans la session")
    else:
        print("session['idUser'] n'existe pas")

    # suppression de session["mail"] uniquement
    session.pop("mail")

    # suppression de toutes les variables de session
    session.clear()
```

131



## Les sessions en JINJA2

Le moteur de template JINJA 2 permet d'accéder aux sessions générées par le serveur web.

La syntaxe peut être:

- {{ session['key'] }}

```
{% if session['idUser'] %} # si la session existe #
    
    <h5 class="pt-2 text-dark"> {{ session['prenom'] }} {{ session['nom'] }}</h5>
{% endif %}
```

ou bien

- {{ session.key }}

```
{% if session['idUser'] %} # si la session existe #
    
    <h5 class="pt-2 text-dark"> {{ session.prenom }} {{ session.nom }}</h5>
{% endif %}
```



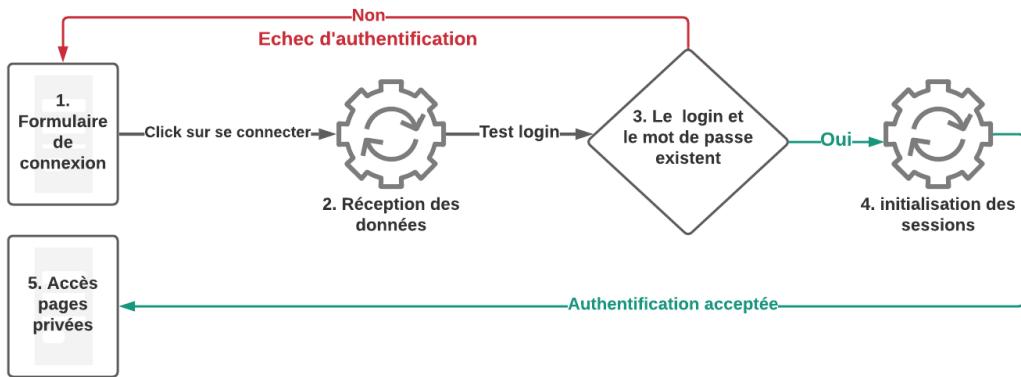
header.html

header.html



## Les sessions

exemple: authentification des utilisateurs



3 &gt; Traitement des données

&gt; Session et cookies

133



## Authentification des utilisateurs

### Etapes 1 et 2

#### 1. Formulaire de connexion

Votre login

Votre mot de passe

connecter.html

```

<form action="login" method="POST">
    <label for="login"></label>
    <input type="text" name="login" id="login" />
    <label for="mdp"></label>
    <input type="password" name="mdp" id="mdp" />
    <button type="submit" >Se connecter</button>
</form>
  
```

#### 2. Réception des données

```

from flask import Flask, render_template, session, request
from .controller import function as f
views.py

@app.route("/connecter") # menu se connecter
@app.route("/connecter/<infoMsg>")
def connecter(infoMsg=''):
    return render_template("connecter.html", info=infoMsg)

@app.route("/logout") # menu se déconnecter
def logout():
    session.clear() # suppression de la session
    return redirect("/connecter/logoutOK")

# traitement du formulaire d'authentification
@app.route("/login", methods=["POST"])
def login():
    #f.sessionTest()
    login = request.form['login']
    password = request.form['mdp']
    msg = f.verifAuth(login, password)
    print(msg)

    if "idUser" in session: # authentification réussie
        return redirect("/index/authOK")
    else: # échec authentification
        return redirect("/connecter/authEchec")
  
```

3 &gt; Traitement des données

&gt; Session et cookies

134



## Authentification des utilisateurs

Etapes 3 et 4

```
#authentification des utilisateurs
def verifAuth(login, mdp):
    session.clear()#suppression sessions précédentes
    msg, user = bdd.verifAuthData(login, mdp)
    print(msg)

    try:
        #authentification réussie
        #initialisation des sessions
        session["idUser"] = user["idUser"]
        session["nom"] = user["nom"]
        session["prenom"] = user["prenom"]
        session["mail"] = user["mail"]
        session["statut"] = user["statut"]
        session["avatar"] = user["avatar"]
        info = msg
    except TypeError as err:
        info="authEchec"
        print("Failed verifAuth : {}".format(err))

    return info
```

functions.py

```
def verifAuthData(login, mdp):
    try:
        cnx, error = connexion()
        if error is not None:
            return error, None
        cursor = cnx.cursor(dictionary=True)
        sql = "SELECT * FROM identification
               WHERE login=%s and motPasse=%s"
        param=(login, mdp)
        cursor.execute(sql, param)
        user = cursor.fetchone()
        close_bd(cursor, cnx)
        msg = "authOK"
    except mysql.connector.Error as err:
        user = None
        msg = "Failed get Auth data : {}"
        .format(err)
    return msg, user
```

bdd.py

3 > Traitement des données > Session et cookies 135

135



## Authentification des utilisateurs

### Gestion des messages

```
{% if info and info=="authOK" %}
    <div class="alert alert-success">
        <span class="badge badge-pill bg-success">Success</span>
        Authentification réussie.
    </div>
{% endif %}
```

index.html

Success Authentification réussie.

```
{% if info and info=="authEchec" %}
    <div class="alert alert-danger text-center">Authentification refusée.</div>
{% endif %}

{% if info and info=="logoutOK" %}
    <div class="alert alert-primary text-center">
        Vous êtes déconnecté. Merci de votre visite.
    </div>
{% endif %}
```

connecter.html

Authentification refusée.

Vous êtes déconnecté. Merci de votre visite.

3 > Traitement des données > Session et cookies 136

136

136



## Authentification des utilisateurs

### Gestion du menu de navigation

Le menu de navigation doit évoluer si l'utilisateur est connecté ou non

Utilisateur non connecté

ACCUEIL COURS ▾ SGBD FICHIERS CONTACT **SE CONNECTER**

Utilisateur connecté

ACCUEIL COURS ▾ SGBD FICHIERS CONTACT **SE DÉCONNEXTER**

**nav.html**

```
{% if session['idUser'] %} #{ utilisateur connecté #}
    <li><a href="/logout">Se déconnecter </a></li>
{% else %} #{ utilisateur non connecté #}
    <li><a href="/connecter">Se connecter </a></li>
{% endif %}
```

3 > Traitement des données

> Session et cookies

137



## Traitement des données



### I. Données enregistrées en BDD

- Requêtes SELECT
- Requêtes DELETE
- Requête INSERT
- Requête UPDATE

### II. Sessions et cookies

### III. Données enregistrées dans un fichier

- Upload d'un fichier
- Importer les données
- Exporter les données

138

138



## Données enregistrées dans un fichier

[Parcourir...](#) membresCoursWeb.xlsx

Dans cette partie, vous allez apprendre à:

1. Télécharger un fichier sur votre serveur
2. Importer les données dans votre BDD
3. Afficher les données sous forme de tableau
4. Modifier dynamiquement le tableau (tri, filtre...)
5. Exporter les données sous différents formats: xls, xlsx, pdf, json...



Fichier excel



Search						
Supp	Avatar	Nom	Prénom	Mail	Login	Statut
		Blériot	Louis	louis.blériot@enac.fr	a	Administrateur
		Boucher	Hélène	helene.boucher@enac.fr	b	Gestionnaire
		admin	admin	admin@enac.fr	admin	Gestionnaire
		Ader	Clément	clement.ader@enac.fr	ader	Administrateur

Showing 1 to 4 of 13 rows 4 rows per page [«](#) [1](#) [2](#) [3](#) [4](#) [»](#)

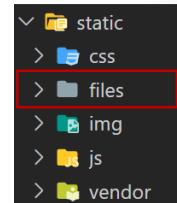
3 > Traitement des données > Gestion des fichiers

139



## Importer les données depuis un fichier

1. Créer un répertoire "files" dans static
2. Créer un formulaire pour télécharger le fichier



```
<form action="/fichiers" method="POST" enctype="multipart/form-data">
    <label for="testFile"></label>
    <input type="file" id="testFile" name="testFile" >
    <button type="submit">
        <i class="fas fa-download"></i>
    </button>
</form>
```

fichiers.html

3 > Traitement des données > Gestion des fichiers

140

140

## Importer les données depuis un fichier

### 2. Gestion de 3 routes:

- Une pour afficher la page fichier.html
- Une pour gérer le message d'information en cas d'échec du transfert
- Une autre pour gérer le fichier transféré.

### 3. Enregistrement du fichier sur le serveur (répertoire files)

### 4. Lecture des données du fichier par la bibliothèque pandas

### 5. Enregistrement des données en BDD.

*Utilisation de la bibliothèque pandas:*

```
pip install pandas
pip install xlrd
pip install openpyxl
https://pandas.pydata.org/docs/index.html
```

```
import pandas, os
from werkzeug.utils import secure_filename
#gestion des fichiers
@app.route("/fichiers")
@app.route("/fichiers/<infoMsg>")
@app.route("/fichiers", methods=['POST'])
def fichiers(infoMsg=''):
    if "testFile" in request.files:
        file = request.files['testFile']
        #enregistrement du fichier dans le répertoire files
        filename = secure_filename(file.filename)
        UPLOAD_FOLDER = 'myApp/static/files/'
        file.save(os.path.join(UPLOAD_FOLDER, filename))
        #enregistrement du fichier sur le serveur
        xls = pandas.read_excel(UPLOAD_FOLDER+file.filename)
        data = xls.to_dict('records')
        print([file.filename,data])
        #Enregistrement des données en BDD
        msg = bdd.addDataFromFile(data)
        print(msg)
    if msg=="addDataFromFileOK":
        return redirect("/sgbd/importDataOK")
    else:
        return redirect("/fichiers/importDataEchec")
    else:
        return render_template("cours/fichiers.html", info=infoMsg)
```

views.py

3 > Traitement des données > Gestion des fichiers

141



## Importer les données depuis un fichier

*Enregistrement des données en BDD*

```
def saveDataFromFile(data):
    try:
        cnx, error = connexion()
        if error is not None:
            return error
        cursor = cnx.cursor()
        sql1 = "TRUNCATE TABLE identification;" # suppression des données précédentes
        cursor.execute(sql1)
        # insertion des nouvelles données
        for d in data:
            sql = "INSERT INTO identification (idUser, nom, prenom, mail, login, motPasse, statut, avatar)
VALUES (NULL, %s, %s, %s, %s, %s, %s, %s);"
            param = (d['nom'], d['prenom'], d['mail'], d['login'], d['motPasse'], d['statut'], d['avatar'])
            cursor.execute(sql, param)
            cnx.commit()
        #changement valeur autoincrement
        sql2="ALTER TABLE identification AUTO_INCREMENT=%s;"
        param2 = (len(data),)
        cursor.execute(sql2, param2)
        cnx.commit()
        close_bd(cursor, cnx)
        msg = "addDataFromFileOK"
    except mysql.connector.Error as err:
        msg = "Failed saveDataFromFile data : {}".format(err)
    return msg
```

bdd.py

142

## Exporter les données dans un fichier

1. Créer un lien dans

sgbd.html

```
<a href="/exportToExcel"
class="btn btn-success">
Exporter les données
</a>
```

2. Utiliser la bibliothèque openpyxl pour exporter les données reçues sous forme de dictionnaire vers un fichier excel.

```
import pandas
from openpyxl import Workbook
#exporter données dans fichier
@app.route("/exportToExcel")
def exportToExcel():
    msg, listeMembre = bdd.get_membreData()
    wb = Workbook()
    sheet = wb.active

    headers = [x for x in listeMembre[0]]

    for index, value in enumerate(headers):
        sheet.cell(row=1, column=index+1).value = value

    for i, x in enumerate(listeMembre):
        for idx,value in enumerate(x.values()):
            sheet.cell(row=i+2, column=idx+1).value = value

    wb.save('myApp/static/files/export.xls')

    return redirect('/static/files/export.xls')
```

views.py

3 > Traitement des données

> Gestion des fichiers

143



## Modifier dynamiquement un tableau *utilisation de bootstrap-table*

La bibliothèque bootstrap-table permet de gérer dynamiquement un tableau HTML.

1. Télécharger bootstrap-table (<https://bootstrap-table.com/>) :
  - Décompresser dans le répertoire vendors
  - Ou bien utiliser les liens CDN
2. Rajouter le lien vers le CSS et JS dans fichiers.html

```
après le {% include("layout/head.html") %}
<link rel="stylesheet" href="https://unpkg.com/bootstrap-table@1.19.0/dist/bootstrap-table.min.css" />
```

```
après le {% include("layout/scripts.html") %} et avant le bootstrap-editable
<script src="https://unpkg.com/bootstrap-table@1.19.0/dist/bootstrap-table.min.js"></script>
```

3 > Traitement des données

> Gestion des fichiers

144

144



## Filtre et pagination d'un tableau

Quelques modifications du code HTML permettent d'obtenir rapidement des fonctionnalités intéressantes: filtrage et recherche des données, système de pagination...

```
<table class="table" data-toggle="table" data-search="true" data-pagination="true" data-page-size="8" data-page-list="[10, 25, 50, all]">
<thead>
  <tr>
    <th data-field="supp" data-align="center">Supp</th>
    <th data-field="avatar" data-sortable="true">Avatar</th>
    <th data-field="nom" data-sortable="true">Nom</th>
    <th data-field="prenom" data-sortable="true">Prénom</th>
    <th data-field="mail" data-sortable="true">Mail</th>
    <th data-field="login" data-sortable="true">Login</th>
    <th data-field="statut" data-sortable="true">Statut</th>
  </tr>
</thead>
<tbody>
```

Supp	Avatar	Nom	Prénom	Mail	Login	Statut
		Blériot	Louis	louis.blériot@enac.fr	a	Administrateur
		Boucher	Hélène	helene.boucher@enac.fr	b	Gestionnaire
		admin	admin	admin@enac.fr	admin	Gestionnaire
		Ader	Clément	clement.ader@enac.fr	ader	Administrateur

Showing 1 to 4 of 13 rows 4 rows per page

Search

1 2 3 4 >

Cette colonne peut être triée.

145



## Gestion de l'interface



I. Introduction au Javascript

II. Méthodes et propriétés

III. Gestion d'évènements

IV. HTTP requests

Cours sur E-campus

146

146

# **ANNEXES**

# Installation de la plate-forme d'hébergement

Lien E-campus : <https://e-campus.enac.fr/moodle/course/view.php?id=440#section-3>

## I. Installation d'une plate-forme d'hébergement (serveur web + serveur BDD +phpmyadmin)

Les logiciels XAMPP (windows / linux) et MAMP (MAC) permettent d'installer un serveur web, un serveur de base de données et Phpmyadmin

### Sous windows ou linux :

1. Télécharger la distribution XAMPP sur <http://www.apachefriends.org/fr/index.html>
2. Enregistrer puis exécuter le fichier proposé.

Tutorial d'installation pour window : <https://www.youtube.com/watch?v=O5sArFV7Pss>

Tutorial d'installation pour linux : <https://www.youtube.com/watch?v=HJI2ILUFBoA>

### Pour tester :

- Démarrer les serveurs Apache et Mysql ( bouton start)
- Dans votre navigateur écrire l'url : <http://localhost> (Vous devez voir apparaître la page d'accueil de xampp avec un lien vers phpmyadmin)
- Dans votre navigateur écrire l'url : <http://localhost/phpmyadmin> pour tester l'accès à une base de données

Si vous avez des problèmes avec l'installation de XAMPP, vous pouvez également essayer avec Wampserver : <http://www.wampserver.com/> (en désinstallant au préalable XAMPP)

### Sous Mac:

1. Télécharger la distribution MAMP sur <https://www.mamp.info/en/downloads/>
2. Un tutorial video d'installation est accessible ici : <https://www.youtube.com/watch?v=SWvPFCtPwJ8>

### Pour tester :

Pour tester utiliser uniquement l'application MAMP qui est gratuite et non pas celle MAMP PRO

- Démarrer des serveurs
- Accéder au site web via : <http://localhost:8888> (si le port n'a pas été changé)
- Accéder à phpmyadmin (comme dans le tutorial) depuis le panneau d'accueil de XAMPP.

## II. Import BDD + création d'un nouvel utilisateur pour la BDD

### Import d'un BDD de test

1. Télécharger le fichier test.sql sur E-campus (section Installation de la plate-forme d'hébergement web )
2. Ouvrez phpmyadmin
3. Sélectionner l'onglet « importer »
4. Cliquer sur le bouton « parcourir » pour sélectionner test.sql
5. Cliquer sur le bouton « Executer » en bas à droite



La BDD « test » doit apparaître dans le panneau de gauche avec à l'intérieur une table nommée « info »

### Création d'un nouvel utilisateur pour accéder au serveur de BDD.

Nous allons créer un utilisateur ayant pour login/mot de passe : ienac / ienac

1. Cliquer sur le mot « serveur » en haut de Phpmyadmin
2. Sélectionner l'onglet « Comptes utilisateurs »
3. Cliquer sur « Ajouter un compte utilisateur »
4. Renseigner les informations en utilisant ienac / ienac



The screenshot shows the "Informations pour la connexion" (Connection information) section of the PhpMyAdmin "Ajouter un compte utilisateur" (Add user account) page. It contains fields for:

- Nom d'utilisateur : Saisir une valeur: ienac
- Nom d'hôte : Local, localhost
- Mot de passe : Saisir une valeur: \*\*\*\*\* Strength: Strong
- Saisir à nouveau : \*\*\*\*\*
- Extension d'authentification : Authentification MySQL native
- Générer un mot de passe: Générer

5. Sélectionner la case « tout cocher »
6. Puis valider en cliquant sur le bouton

« executer » en bas à droite de l'écran

Priviléges globaux  Tout cocher

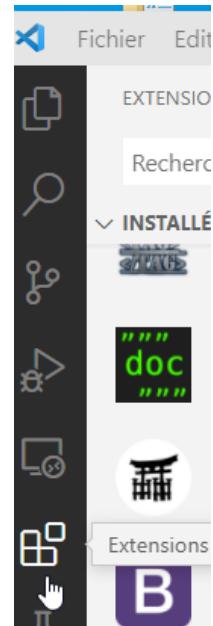
### **III. Installation de l'éditeur Visual Studio Code**

- Télécharger python sur <https://www.python.org/downloads/>. (lors de l'installation cocher les options pour rajouter le PATH et les variables d'environnement)



- Télécharger Visual Studio Code sur <https://code.visualstudio.com/>
- Liste des extensions à rajouter dans vsCode:

- French Language Pack for Visual Studio Code
- Auto Complete Tag
- Tag Inserter
- IntelliSense for CSS class names in HTML
- ESLint
- Python
- Python Extension Pack (supprimer extension Django)
- Python Indent
- Python Environnement Manager
- Path Autocomplete
- Path Intellisense
- JSON



Et d'autres selon vos besoins...

## IV. Installation et test du serveur web python Flask

Pour terminer la plate-forme d'herbergement, vous devez installer le serveur web python nommé Flask et plusieurs modules dont mysql-connector-python qui permet de faire le lien entre le serveur python et le serveur mysql.

La méthode la plus simple est d'utiliser pip (ou pip3) install.

Ouvrez un terminal et taper les commandes suivantes :

```
pip install flask
pip install mysql-connector-python
pip install openpyxl
pip install flask-Mail
pip install pandas
pip install xlrd
```

**Pour tester :**

1. Vérifier que le XAMPP ou le MAMP est bien démarré.
2. Télécharger le fichier **run.py** sur E-campus (section Installation de la plate-forme d'hébergement web )
3. Executer ce fichier python soit par les raccourcis de votre éditeur, soit soit en ligne de commande: `python3.exe runTest.py` (sous windows) ou `python3 run.py` (linux ou mac)

4. Vous devez voir apparaître un message du genre :

```
* Serving Flask app "runTest"
* Environment: production
WARNING: This is a development server. Do not use it in a production
deployment.
Use a production WSGI server instead.
* Debug mode: on
* Restarting with stat
* Debugger is active!
* Debugger PIN: 285-556-613
```

```
| * Running on http://localhost:5000/ (Press CTRL+C to quit) * Running on
```

5. Dans votre navigateur écrire l'url <http://localhost:5000>  
Si « Tout fonctionne correctement » s'affiche, c'est gagné !!!!
6. Fermer le serveur flask en tapant CTRL+C

Si vous rencontrez des problèmes lors des différentes étapes de cette installation, n'hésitez pas à contacter l'équipe enseignante

- soit par mail ([karine.puechmorel@enac.fr](mailto:karine.puechmorel@enac.fr))
- soit par teams (appel privé)

Nous vous aiderons à les résoudre grâce partage d'écran prévu sous teams.

Quand tout est terminé, pensez à renseigner sur E-campus le sondage : [Avez-vous réussi à installer la plate-forme web ?](#) .

## ANNEXE 2 : Les feuilles de styles.

### Déclaration des styles

#### Styles internes

##### A l'intérieur des balises <head></head>

```
<html>
  <head> <style type="text/css"> </style> </head>
  <body> ... </body>
</html>
```

##### A l'intérieur des balises <body></body>

```
<html>
  <head> </head>
  <body> balise { propriété de style: valeur } </body>
</html>
```

#### Styles externes

```
<html>
  <head> <link rel="stylesheet" type="text/css" href="styles.css" > <head>
  <body> ... </body>
</html>
```

```
<html>
  <head> <style type="text/css">@import:url - "monfichier.css" ;</style><head>
  <body> ... </body>
</html>
```

### Les commentaires

```
/* commentaire */
```

### Les séparateurs à virgule flottante

```
Balise1, balise2, balise3, .... { style; }
```

### Les identifiants

```
#nom_de_ID { propriété de style: valeur }
<balise id="nom_de_ID"> .... </balise>
```

### Les classes

```
.nom_de_classe { propriété de style: valeur }           classe universelle
Balise . nom_de_classe { propriété de style: valeur }
<balise class="nom_de-classe"> .... </balise>
```

## **Les pseudo-classes**

*Les pseudo-classes dynamiques*

**:hover {...}** agit lorsque le curseur de la souris pointe l'élément.

**:active {...}** agit lors de l'activation de l'élément par un clic de souris notamment.

**:focus {...}** agit lors de la réception du focus par le clavier ou par la souris.

**balise:focus:hover { propriété de style: valeur }** possibilité de combinaison

## Les pseudo-classes de liens

**a:link {...}** permet de définir le style des liens.

**a:visited {...}** permet de définir le style des liens visités.

## La pseudo-classe de texte

**:first-line {...}** permet d'affecter un style à la première ligne d'un paragraphe.

**:first-letter {...}** permet d'affecter un style à la première lettre d'un paragraphe.

## La pseudo-classe de langue

Balise:**lang-** Langue ... {Propriété1:Valeur1;...}

Ou langue peut être : **FR** pour la France, **DE** pour l'Allemagne, **ES** pour l'Espagne **EN** pour l'Angleterre, **EN-US** pour les Etats-Unis, etc...

## Les pseudo-classes de page

**@page :left { ... }** permet de formater la page de gauche.

**@page :right { ... }** permet de formater la page de droite.

**@page :first { ... }** permet de formater la première page.

## Les pseudo-éléments before et after

**:before {...}** permet d'affecter un contenu avant le contenu initial d'un élément.

**:after {...}** permet d'affecter un contenu après le contenu initial d'un élément.

## **<span>**

**<span> ... </span>** : permet d'appliquer des styles à des éléments de texte d'un paragraphe ou si vous préférez à un morceau de paragraphe.

## **<div>**

**<div> ... </div>** : permet de regrouper plusieurs paragraphes ou si vous préférez, de délimiter une zone comportant plusieurs paragraphes

## **Les styles de police**

**font :raccourci pour les différentes propriétés de police**

**font-family : définit un nom de police- Arial, Times, Helvetica... ou une famille de police - serif, sans-serif,**

**cursive, fantasy, monospace**

**font-style : définit le style de l'écriture - normal ou italique ou oblique**

**font-size : définit la taille de la police**

xx-small ou x-small ou small ou médium ou large ou x-large ou xx-large ou larger ou smaller ou taille précise - en points - pt , inches - in , centimètres - cm , pixels - px ou pourcentage - %

**font-stretch** : définit l'étirement de la police de caractères.- normal, wider, narrower, ultra-condensed, extra-condensed, condensed , semi-condensed, semi-expanded, expanded, extra-expanded, ultra-expanded

**font-variant** : définit une variante par rapport à la normale - normal ou small-caps

**font-weight** : définit l'épaisseur de la police - normal ou bold ou bolder ou lighter ou valeur numérique

soit - 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900

## Les styles du texte

**text-align** :définit l'alignement du texte - left ou center ou right

**text-indent** :définit un retrait dans la première ligne d'un bloc de texte

spécifié en inches - in ou en centimètres - cm ou en pixels - px

**text-decoration** : définit une décoration - ? du texte, soit barré, clignotant, etc.

- blink ou underline ou line-through ou overline ou none

**text-shadows** : applique des effets d'ombre au texte ou à un élément de texte.

- none , {Couleur Taille\_à\_droite Taille\_en\_bas rayon, Couleur Taille\_à\_droite, Taille\_en\_bas rayon}

**text-transform** : définit la casse du texte - majuscule, minuscule

- uppercase - met les caractères en majuscules

- ou lowercase - met les caractères en minuscules

- ou capitalize - met le premier caractère en majuscule

**color** : définit la couleur du texte

**direction** : spécifie la direction d'écriture,

- de la droite vers la gauche - "rlt" ou de la gauche vers la droite - "ltr" .

**height** : détermine la hauteur d'un élément de texte ou d'une image

- en points - pt , inches - in , centimètres - cm , pixels - px ou pourcentage - %

**letter-spacing** : définit l'espace entre les lettres spécifié

- en points - pt , inches - in , centimètres - cm , pixels - px ou pourcentage - %

**line-height** :définit l'interligne soit l'espace entre les lignes du texte

- en points - pt , inches - in , centimètres - cm , pixels - px ou pourcentage - %

**unicode-Bidi** : définit l'orientation bidirectionnelle de l'écriture

- normal ,embed ,bidi-override

**vertical-align** : spécifie l'alignement vertical des éléments HTML.

- baseline, sub, super, top ,text-top ,middle ,bottom ,text-bottom ,Longueur ,Pourcentage

**white-space** : espace ou blanc - normal ou pre ou nowrap

**width** : détermine la longueur d'un élément de texte ou d'une image

- en points - pt , inches - in , centimètres - cm , pixels - px ou pourcentage - %

**word-spacing** :définit l'espace entre les mots

-en points - pt , inches - in , centimètres - cm , pixels - px ou pourcentage - %

## Les arrière-plans

**background-color** :définit la couleur de l'arrière-plan ou transparent

**background-image** :définit l'image de l'arrière-plan - URL de l'image

**background-repeat** : définit la façon de répéter l'image d'arrière-plan

repeat ou no-repeat ou repeat-x - x = nombre de répétitions horizontales

ou repeat-y - y = nombre de répétitions verticales

**background-attachment** : spécifie si l'image d'arrière-plan reste fixe avec les déplacements de l'écran  
- scroll ou fixed

**background-position** : spécifie la position de l'image d'arrière-plan par rapport au coinsupérieur gauche de la fenêtre {top ou center ou bottom , left ou center ou right}

- ou en points - pt , inches - in , centimètres - cm , pixels - px ou pourcentage - %

**background** :raccourci pour les différentes propriétés d'arrière-plan

## Les marges

**margin-top** : détermine la valeur de la marge supérieure en unité de longueur ou auto

**margin-right** : détermine la valeur de la marge droite en unité de longueur ou auto

**margin-bottom** : détermine la valeur de la marge inférieure en unité de longueur ou auto

**margin-left** : détermine la valeur de la marge gauche en unité de longueur ou auto

**margin** : regroupe les différentes propriétés de la marge

## Les bords et les « enrobages »

**border-top-width** : donne l'épaisseur du bord supérieur -thin ou medium ou thick ou spécifié par l'auteur

**border-right-width** : donne l'épaisseur du bord droit - thin ou medium ou thick ou spécifié par l'auteur

**border-bottom-width** : donne l'épaisseur du bord inférieur - thin ou medium ou thick ou spécifié par l'auteur

**border-left-width** : donne l'épaisseur du bord gauche -thin ou medium ou thick ou spécifié par l'auteur

**border-width** : regroupe les différentes propriétés de border-width

**border-color** : détermine la couleur de la bordure

**border-style** : détermine le style du trait de la bordure

- none ou solid ou dotted ou dashed ou double

- ou groove ou ridge ou inset ou outset

**border** : regroupe toutes les propriétés des bords

**padding-top** : valeur de remplissage haut entre l'élément et le bord

- en points - pt , inches - in , centimètres - cm , pixels - px ou pourcentage - %

**padding-right** : valeur de remplissage droite entre l'élément et le bord

- en points - pt , inches - in , centimètres - cm , pixels - px ou pourcentage - %

**padding-bottom** : valeur de remplissage bas entre l'élément et le bord

- en points - pt , inches - in , centimètres - cm , pixels - px ou pourcentage - %

**padding-left** : valeur de remplissage gauche entre l'élément et le bord

- en points - pt , inches - in , centimètres - cm , pixels - px ou pourcentage - %

**padding** : regroupe les différentes propriétés de remplissage

## Les listes

**list-style-type** : détermine le type de puces ou de numérotation

- disc ou circle ou square

- decimal ou lower-roman ou upper-roman ou lower-alpha ou upper-alpha

**list-style-image** : permet de remplacer les puces par une image -url ou none

**list-style-position** : spécifie si les puces sont à l'intérieur ou à l'extérieur du texte inside ou outside

## Les tableaux

**table { border...}** : Permet de contrôler l'apparence des bordures

**table { border-collapse...}** : applique des bordures autour des cellules d'un tableau.

- collapse, separate

**table { border-color...}** : Permet de contrôler la couleur des bordures

**table { border-style...}** : Permet de contrôler le style des bordures

**table { border-left...}** : Permet de contrôler la bordure gauche

**table { border-right...}** : Permet de contrôler la bordure droite

**table { border-top...}** : Permet de contrôler la bordure supérieure

**table { border-width...}** : Permet de contrôler l'épaisseur des bordures

**table { border-left-width...}** : Permet de contrôler l'épaisseur de la bordure de

**table { border-right-width...}** : Permet de contrôler l'épaisseur de la bordure de droite

**table { border-bottom-width...}** :Permet de contrôler l'épaisseur de la bordure inférieure  
**table { margin...}** :Définit les quatres marges d'un tableau.  
**table { margin-left ...}**: Définit la marge gauche  
**table { margin-right...}** :Définit la marge droite  
**table { margin-top ...}**: Définit la marge supérieure  
**padding** Définit des espaces de remplissage autour d'un élément. `table {padding: 20%}`  
**padding-left** Définit l'espace de remplissage à gauche d'un élément `table {padding-left: 15px}`  
**padding-right** Définit l'espace de remplissage à droite d'un élément `table {padding-right: 1in}`  
**padding-top** Définit l'espace de remplissage supérieure d'un élément `table {padding-top: 2cm}`  
**padding-bottom** Définit l'espace de remplissage inférieure d'un élément `td {padding-bottom: 20pt}`  
**td { border-bottom...}** :Permet de contrôler la bordure inférieure  
**td { margin-bottom...}** : Définit la marge inférieure  
**th { border-top-width...}** :Permet de contrôler l'épaisseur de la bordure

### Contrôle du son pour sortie vocale

**volume** - volume  
**speak** - Prononciation  
**pause-before**- Pause avant un élément  
**pause-after**- Pause après un élément  
**pause** - Pause avant et après un élément  
**cue-before** - Signal de reconnaissance avant un élément  
**cue-after** - Signal de reconnaissance après un élément  
**cue** - Signal de reconnaissance avant et après un élément  
**play-during** - Son d'arrière-plan  
**azimuth** - Effet spatial gauche/droite/devant/derrière  
**elevation** - Effet spatial dessus/dessous  
**speech-rate** - Vitesse de parole  
**voice-family** - Type de parleur  
**pitch** - Timbre de la voix  
**pitch-range** , - Modulation de la voix  
**stress** - Caractère de la voix  
**richness** - Assurance de la voix  
**speak-punctuation** - Mentionner explicitement la ponctuation  
**speak-numeral** - Convention pour la lecture des chiffres

### Nouveautés CSS3

#### Font-face

# HELLO CSS 3 WORLD!

```

@font-face
{
  font-family: SketchRockwell;
  src: url(„SketchRockwell.ttf“);
}

.my_CSS3_class
{
  font-family: SketchRockwell;
  font-size: 3.2em;
}
  
```

#### Columns

## Column Width

Tincidunt urna, turpis rhoncus placerat non odio et sit integer et dolor rhoncus  
pellentesque enim urna ultrices vut cursus tortor a dis, ac nunc cum. Eu.  
Nec elementum. Turpis pulvinar in risus? Amet pid eu parturient nisi egestas proin. Cum, pulvinar nunc parturient porta, vel enim pellentesque! Lorem, scelerisque cursus in natoque vel porta?  
Magna augue tincidunt lacus rhoncus placerat, a  
Tincidunt urna, turpis rhoncus placerat non odio et sit integer et dolor rhoncus  
pellentesque enim urna ultrices vut cursus tortor a dis, ac nunc cum. Eu.  
Nec elementum. Turpis pulvinar in risus? Amet pid eu parturient nisi egestas proin. Cum, pulvinar nunc parturient porta, vel enim pellentesque! Lorem, scelerisque cursus in natoque vel porta?  
Magna augue tincidunt lacus rhoncus placerat, a  
Tincidunt urna, turpis rhoncus placerat non odio et sit integer et dolor rhoncus  
pellentesque enim urna ultrices vut cursus tortor a dis, ac nunc cum. Eu.  
Nec elementum. Turpis pulvinar in risus? Amet pid eu parturient nisi egestas proin. Cum, pulvinar nunc parturient porta, vel enim pellentesque! Lorem, scelerisque cursus in natoque vel porta?  
Magna augue tincidunt lacus rhoncus placerat, a

**column-count** : indique le nombre de colonnes utilisées pour découper la page.

**column-gap** : indique la largeur de séparation entre deux colonnes.

**column-width** : indique la largeur de chaque colonnes.

**column-rule**: Définit la bordure entre les colonnes

### Gradients



**background-image:**

**-webkit-linear-gradient ([param1], [param2 param3], [param4 param5]); /\* Safari and Chrome \*/**  
**-moz-linear-gradient ([param1], [param2 param3], [param4 param5]); /\* Firefox \*/**  
**-ms-linear-gradient ([param1], [param2 param3], [param4 param5]); /\* IE 9 \*/**  
**-o-linear-gradient ([param1], [param2 param3], [param4 param5]); /\* Opera \*/**  
**linear-gradient ([param1], [param2 param3], [param4 param5]);**

où:

param1: position de départ de votre dégradé. Ex right top

param2 : couleur de départ de votre dégradé. Ex #D60F0F

param3 : pourcentage qui correspond à l'étendue de la couleur passée au param2. Ex : 10%

param4 : couleur de fin de votre dégradé. Ex #FFDD00

param5 : pourcentage qui correspond à l'étendue de la couleur passée au param4 : Ex 100%

**Pour IE (jusqu'à IE8):**

**filter:progid:DXImageTransform.Microsoft.gradient( startColorstr='#D60F0F', endColorstr='#FFDD00', GradientType=0);**  
startColorstr : couleur de début  
endColorstr : couleur de fin

GradientType : 0 pour un dégradé vertical  
1 pour un dégradé horizontal

## Backgrounds

**background-size** spécifie la taille de l'image dans l'arrière-plan.

**background-clip** permet de définir les limites de l'arrière-plan à l'intérieur de la boîte représentée par l'élément.

3 valeurs possibles

- "border-box" : L'arrière-plan s'étend jusqu'à l'extrême limite de la bordure
- "padding-box" : Aucun arrière-plan ne sera présent en-dessous de la bordure ( l'extrême limite sera celle du padding)
- "content-box" : L'arrière-plan se limite au contenu

**background-origin** positionne le **point d'origine** de l'image d'arrière-plan.

3 valeurs possibles

- border-box : La position de l'arrière-plan est relative au bord : l'image peut donc se positionner *derrière* les bords.
- padding-box : La position de l'arrière-plan est relative au padding (marge interne)
- content-box : La position de l'arrière-plan est relative au contenu.

## Backgrounds multiples

CSS 3 rend possible l'affichage de plusieurs images dans un même arrière-plan.

```
background-image: url("image1"), url("image2");  
background-position: x y, x y;  
background-repeat: no-repeat;
```

## Les transformations

**transform:none** Définit qu'il ne devrait y avoir aucune transformation

**transform:matrix(n,n,n,n,n,n)** Définit une transformation 2D, en utilisant une matrice de six valeurs

**transform:matrix3d (n,n,n,n,n,n,n,n,n,n,n,n,n,n,n)** Définit une transformation 3D, en utilisant une matrice 4x4 de 16 valeurs

**transform:translate(x,y)** Définit une translation 2D

**transform:translate3d(x,y,z)** Définit une translation 3D

**transform:translateX(x)** Définit une translation seulement selon l'axe des X

**transform:translateY(y)** Définit une translation seulement selon l'axe des Y

**transform:translateZ(z)** Définit une translation 3d seulement selon l'axe des Z

**transform:scale(x,y)** Définit l'échelle de la transformation 2D

**transform:scale3d(x,y,z)** Définit l'échelle de la transformation 3D

**transform:scaleX(x)** Définit l'échelle de la transformation seulement selon l'axe des X

**transform:scaleY(y)** Définit l'échelle de la transformation seulement selon l'axe des Y

**transform:scaleZ(z)** Définit l'échelle de la transformation 3D en donnant une valeur à l'axe des Z

**transform:rotate(angle)** Définit une rotation 2D, l'angle est spécifiée dans le paramètre

**transform:rotate3d(x,y,z,angle)** Définit une rotation

**transform:rotateX(angle)** Définit une rotation par rapport à l'axe des X

**transform:rotateY(angle)** Définit une rotation 3D par rapport à l'axe des Y

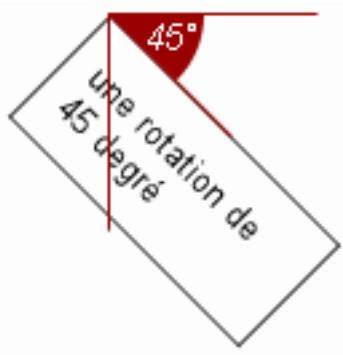
**transform:rotateZ(angle)** Définit une rotation 3D par rapport à l'axe des Z

**transform:skew(x-angle,y-angle)** permet de tordre un élément en définissant un angle para rapport à l'axe des X et des Y

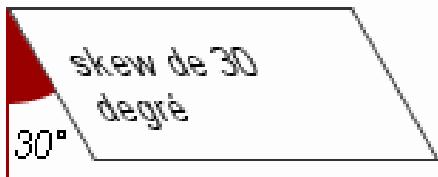
**transform:skewX(angle)** permet de tordre un élément en définissant un angle para rapport à l'axe des X

**transform:skewY(angle)** permet de tordre un élément en définissant un angle para rapport à l'axe des Y

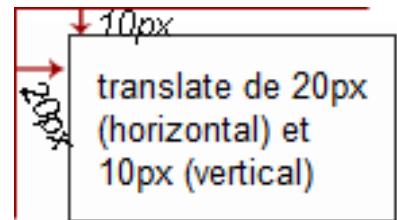
**transform:perspective(n)** Définit une vue en perspective d'un élément 3D transformé



Rotate



skew



translate

### *Les ombrages*

## box-shadow

**box-shadow** : s'applique à tout le bloc et prend quatre valeurs dans l'ordre suivant :

**box-shadow: x1 x2 x3 x4;**

1. x1: le décalage horizontal de l'ombre ;
2. x2: le décalage vertical de l'ombre ;
3. x3: l'adoucissement du dégradé ;
4. x4: la couleur de l'ombre.

### *Les bordures arrondies*

**border-radius** : arrondit les angles selon une taille d'arrondi défini

## SELECT

---

```
def get_ProduitData():
    try:
        cnx = connexion()
        cursor = cnx.cursor(dictionary=True)
        sql = "SELECT * FROM produit"
        cursor.execute(sql)
        res = cursor.fetchall()
        close_bd(cursor, cnx)
        msg = "ok"
    except mysql.connector.Error as err:
        res = None
        msg = "Failed get ProduitData: {}".format(err)

    return msg, res
```

## INSERT

---

```
def add_produitData(nom, prix):
    try:
        cnx = connexion()
        cursor = cnx.cursor()
        sql = "INSERT INTO produit (nomProduit, prix) VALUES (%s, %s);"
        param = (nom, prix)
        cursor.execute(sql, param)
        lastIdProduit = cursor.lastrowid # récupère le dernier idProduit,
généré par le serveur sql
        cnx.commit()
        close_bd(cursor, cnx)
        msg = "ok"
    except mysql.connector.Error as err:
        lastIdProduit = None
        msg = "Failed add produitData: {}".format(err)

    return msg, lastIdProduit
```

## DELETE

---

```
def del_commandeData(id_com):
    try:
        cnx = connexion()
        cursor = cnx.cursor()
        sql = "DELETE FROM fournisseur WHERE idCommande=%s;"
        param = (id_com,)    avec un seul paramètre bien penser à garder la virgule
        cursor.execute(sql, param)
        cnx.commit()
        close_bd(cursor, cnx)
        msg = "ok"
    except mysql.connector.Error as err:
        msg = "Failed del commandeData : {}".format(err)

    return msg
```

## UPDATE

---

```
def update_produitData(champ, idProduit, newValue):
    try:
        cnx = connexion()
        cursor = cnx.cursor()
        sql = "UPDATE produit SET "+champ+" = %s WHERE produit.idProduit = %s;"
        param = (newValue, idProduit)
        cursor.execute(sql, param)
        cnx.commit()
        close_bd(cursor, cnx)
        msg = "ok"
    except mysql.connector.Error as err:
        msg = "Failed update produitData : {}".format(err)

    return msg
```