

61

HTML : LES FORMULAIRES

3^{ÈME} ANNÉE LICENCE EN INFORMATIQUE

FSM 2020-2021

Les formulaires en HTML

- Les formulaires ont pour objectif d'interagir avec l'utilisateur tout en permettant d'obtenir des réponses de ce dernier
 - Pour définir un formulaire en HTML, on utilise l'élément `<form>...</form>`
 - La balise `<form>` peut avoir comme attributs :
 - `method="get" | "post"` : permettant de préciser la méthode de l'envoi des informations
 - `action="URL"` : fourni l'adresse de la page ou du code qui va traiter les informations
- ```
<form method="post" action="traitement.php">
 <p>Texte à l'intérieur du formulaire</p>
</form>
```

# Zone de texte monoligne

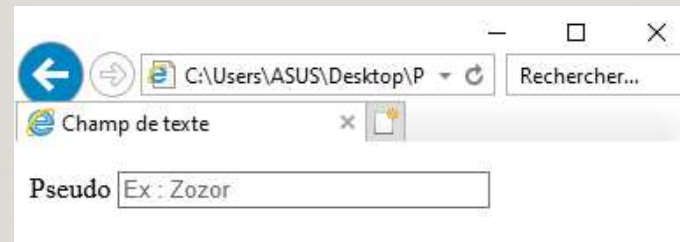
- **Rôle** : créer une zone de texte à une ligne
- **Syntaxe** :

```
<input type="text" name="nom de la zone" id="identifiant" value="texte par défaut"
size="taille" maxlength="nb" placeholder="exemple" >
```

- **name** : donner un nom à un élément de formulaire
- **id** : identifier l'élément HTML pour pouvoir y accéder et le manipuler. Il est unique
- **value** : une valeur par défaut pour préremplir le champ de texte
- **size** : agrandir le champ
- **maxlength** : peut limiter le nombre de caractères que l'on peut saisir
- **placeholder** : donner une indication sur le contenu du champ

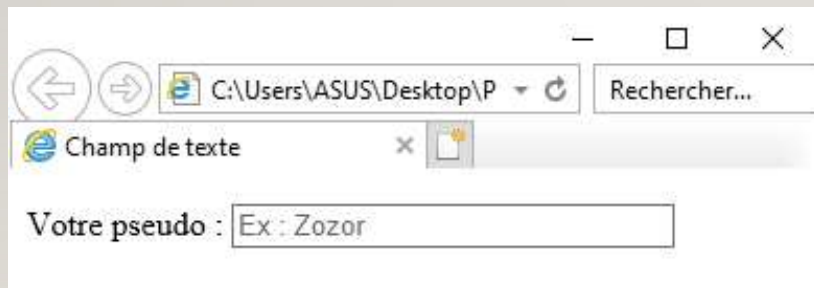
# Zone de texte monoligne : Exemple

```
<body>
 <form method="post" action="traitement.php">
 <p>
 Pseudo <input type="text" name="pseudo" id="pseudo"
 placeholder="Ex : Zozor" size="30" maxlength="10">
 </p>
 </form>
</body>
```



# Les libellés

- **Rôle** : donne une indication sur ce que le visiteur doit écrire dans la zone de texte
- **Syntaxe** : `<label for="identifiant du champ">....</label>`
- L'attribut **for** permet de lier le label au champ correspondant , il a la même valeur que l' *id* du champ



```
<body>
 <form method="post" action="traitement.php">
 <p>
 <label for="pseudo">Votre pseudo :</label>
 <input type="text" name="pseudo" id="pseudo"
 placeholder="Ex : Zozor" size="30" maxlength="10">
 </p>
 </form>
</body>
```



# Zone de texte multiligne

- **Rôle:** permet de définir une zone de texte pour saisir des données de taille plus importante que dans une case de texte monoligne
- **Syntaxe :**

`<textarea name="nom de la zone" id="identifiant"`

`rows="r" cols="c"> valeur_par_défaut</textarea>`

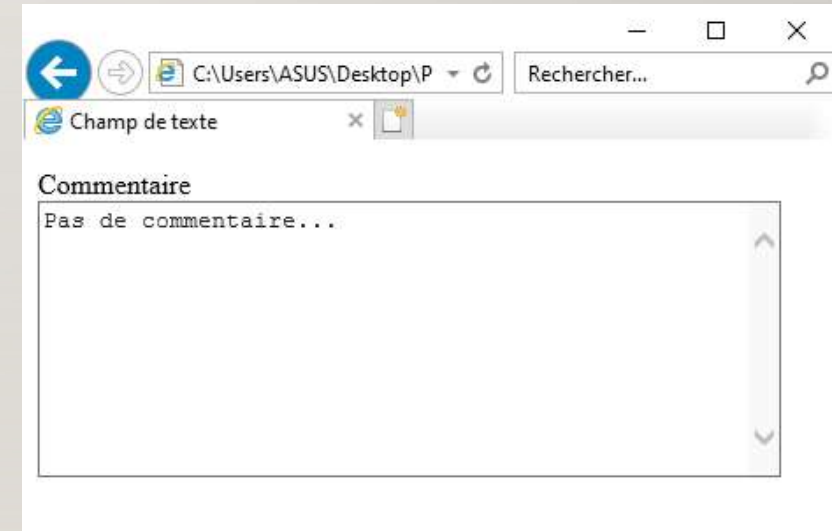
- **rows="r"** : taille de la zone de saisie en nombre de lignes
- **cols="c"** : taille de la zone de saisie en nombre de colonnes

67

# Zone de texte multiligne : Exemple

```
<body>
 <form method="post" action="traitement.php">
 <p>
 <label for="commentaire">Commentaire</label>

 <textarea name="commentaire" id="commentaire" rows="10"
 cols="50">Pas de commentaire...</textarea>
 </p>
 </form>
</body>
```



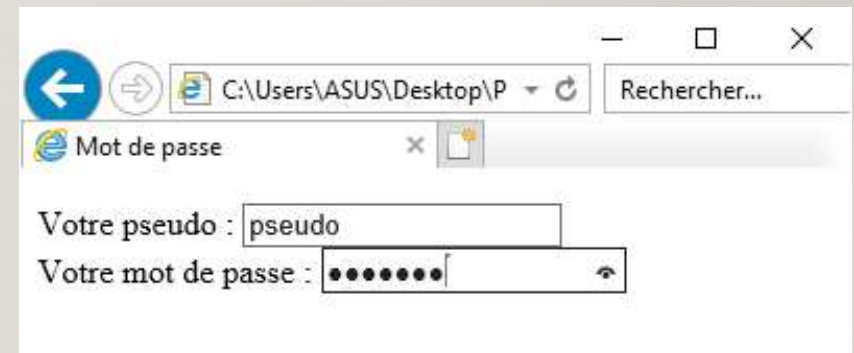
A screenshot of a web browser window. The address bar shows the file path C:\Users\AŞUS\Desktop\P. The page title is "Champ de texte". The form contains a label "Commentaire" and a text area with the placeholder text "Pas de commentaire...". The text area is a rectangular box with a vertical scrollbar on the right side.

# Zone de mot de passe

- **Rôle:** permet de définir une zone de saisie où on ne voit pas à l'écran les caractères saisis
- **Syntaxe :** `<input type="password" name="nom" id="identifiant">`

```
<body>
 <form method="post" action="traitement.php">
 <p>
 <label for="pseudo">Votre pseudo :</label>
 <input type="text" name="pseudo" id="pseudo">

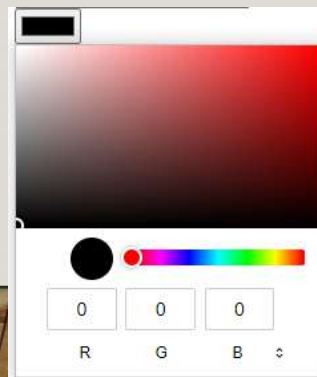
 <label for="pass">Votre mot de passe :</label>
 <input type="password" name="pass" id="pass">
 </p>
 </form>
</body>
```





# Nouveaux types de champs

- HTML5 apporte de nouveaux types de champs
- Il suffit de donner à l'attribut type de la balise **<input>** l'une des nouvelles valeurs suivantes :
  - **"e-mail"** : demander à saisir une adresse e-mail
  - **"url"** : demander à saisir une adresse absolue (commençant généralement par http:// )
  - **"tel"** : saisie de numéros de téléphone
  - **"color"** : saisir une couleur



# Nouveaux types de champs

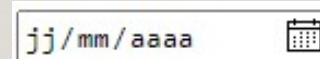
- Autres valeurs de l'attribut type de la balise **<input>**
  - **"number"** : saisir un nombre entier. Ce type de champ peut être personnalisé par les attributs : min = valeur minimale ; max = valeur maximale ; step = pas de déplacement

A standard HTML number input field, which is a white rectangular box with a small upward and downward arrow icon on the right side.

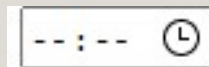
- **"range"** : sélectionner un nombre avec un curseur (on peut utiliser les attributs min , max et step)

A range input field, consisting of a horizontal slider bar with a dark grey track and a black circular thumb in the center.

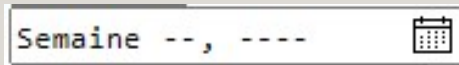
- **"date"** : pour la date (05/08/1985 par exemple)

A date input field, showing a white box with the placeholder text 'jj/mm/aaaa' and a small calendar icon on the right.

- **"time"** : pour l'heure (13:37 par exemple)

A time input field, showing a white box with the placeholder text '--:--' and a clock icon on the right.

- **"week"** : pour la semaine

A week input field, showing a white box with the placeholder text 'Semaine --, ----' and a small calendar icon on the right.

- **"month"** : pour le mois

A month input field, showing a white box with the placeholder text '-----' and a small calendar icon on the right.

# Nouveaux types de champs

- Si votre formulaire grossit et comporte beaucoup de champs, il peut être utile de les regrouper au sein de plusieurs balises `<fieldset>` .
- Chaque `<fieldset>` peut contenir une légende avec la balise `<legend>`

```
<body>
 <form method="post" action="traitement.php">
 <p>
 <fieldset>
 <legend>Informations personnelles</legend>
 <label for="pseudo">Votre pseudo :</label>
 <input type="text" name="pseudo" id="pseudo"/>

 <label for="pass">Votre mot de passe :</label>
 <input type="password" name="pass" id="pass" />

 <label for="date-nais">Date de naissance :</label>
 <input type="date" name="date-nais" id="date-nais"/>

 </fieldset>
 </p>
 </form>
</body>
```



The screenshot shows a web browser window with a form titled "Informations personnelles". The form contains three input fields: "Votre pseudo :" followed by a text input field, "Votre mot de passe :" followed by a password input field, and "Date de naissance :" followed by a date input field showing "jj/mm/aaaa" and a calendar icon.

# Case à cocher

- **Syntaxe**

```
<input type="checkbox" name="nom" id="identifiant"
value="valeur attachée au bouton " checked>
```

- **Principes**

- Plusieurs cases peuvent avoir l'option **checked**
- Les 4 cases n'ont pas le même nom
- **value** : indique la valeur qui sera envoyée au serveur si la case est cochée
- **checked** : Un attribut booléen qui indique si la case est cochée.



# Case à cocher : Exemple

```
<body>
 <form method="post" action="traitement.php">
 <fieldset>
 <legend> Sport </legend>
 Quel(s) sport(s) pratiquez vous?

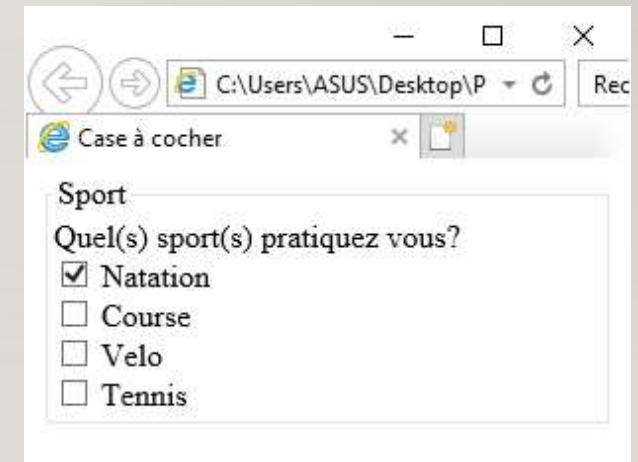
 <input type="checkbox" name="natation" id="natation" value="natation"
 checked>
 <label for="natation">Natation</label>

 <input type="checkbox" name="course" id="course" value="course">
 <label for="course">Course</label>

 <input type="checkbox" name="velo" id="velo" value="velo">
 <label for="velo">Velo</label>

 <input type="checkbox" name="tennis" id="tennis" value="tennis">
 <label for="tennis">Tennis</label>

 </fieldset>
 </form>
</body>
```



The screenshot shows a web browser window with the address bar displaying 'C:\Users\ASUS\Desktop\P'. The page title is 'Case à cocher'. The form content is as follows:

Sport

Quel(s) sport(s) pratiquez vous?

- ☒ Natation
- ☐ Course
- ☐ Velo
- ☐ Tennis



# Bouton radio

- **Syntaxe**

```
<input type="radio" name="nom du groupe" id="identifiant"
value="valeur attachée au bouton " checked>
```

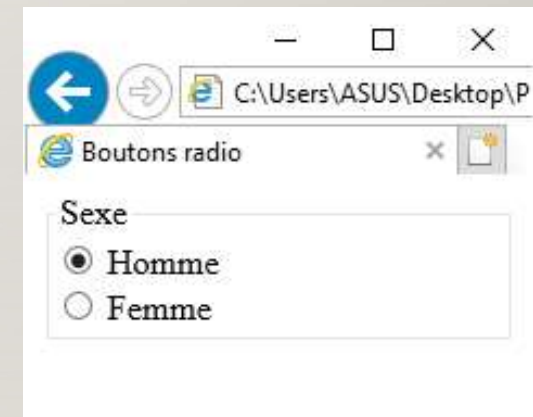
- **Principes**

- une seule case peut avoir l'option **checked**
- les 4 cases ont le même nom
- value: indique la valeur qui sera envoyée au serveur si le bouton est coché

# Bouton radio : Exemple

```
<body>
 <form method="post" action="traitement.php">
 <fieldset>
 <legend> Sexe </legend>
 <input type="radio" name="sexe" id="homme" value="Homme" checked>
 <label for="homme">Homme</label>

 <input type="radio" name="sexe" id="femme" value="Femme">
 <label for="femme">Femme</label>
 </form>
 </body>
```



- **Bouton générique**

- C'est un bouton qui n'aura (par défaut) aucun effet. En général, ce bouton est géré en JavaScript pour exécuter des actions sur la page
- **Syntaxe :** `<input type="button" name="nom" value="texte du bouton ">`

- **Bouton Submit**

- Le bouton *Submit* permet de transmettre les données du formulaire. Les données seront envoyées à la méthode précisée dans l'attribut *method* de la balise `<form>` à l'adresse précisé en valeur de l'attribut *action* de cette même balise
- **Syntaxe :** `<input type="submit" value="texte du bouton " >`

# Les boutons

- **Bouton Reset**
  - Le bouton *Reset* permet d'annuler les modifications apportées dans un formulaire et de restaurer les valeurs par défaut.
  - **Syntaxe** : `<input type="reset" value="texte du bouton " >`

# Liste déroulante

- L'élément `<select>...</select>` permet un choix dans une liste de plusieurs options présentées sous forme de liste déroulante
- Chaque élément de la liste est défini en utilisant l'élément `<option>...</option>`
- **Syntaxe :** `<select name="nom de la liste" id="identifiant">>`  
    `<option value="valeur de l'option" selected>texte de la ligne 1</option>`  
    `</select>`
  - **selected :** indique que la valeur correspondante est sélectionnée par défaut



# Liste déroulante : Exemple

```
<body>
 <form method="post" action="traitement.php">
 <p>
 <label for="pays">Dans quel pays habitez-vous ?</label>

 <select name="pays" id="pays">
 <option value="france">France</option>
 <option value="espagne" selected>Espagne</option>
 <option value="italie">Italie</option>
 <option value="royaume-uni">Royaume-Uni</option>
 <option value="canada">Canada</option>
 <option value="etats-unis">États-Unis</option>
 <option value="chine">Chine</option>
 <option value="japon">Japon</option>
 </select>
 </p>
 </form>
</body>
```

Dans quel pays habitez-vous ?

Espagne ▼

Dans quel pays habitez-vous ?

Espagne ▼

France

Espagne

Italie

Royaume-Uni

Canada

États-Unis

Chine

Japon

# Liste déroulante : <optgroup>

- Dans une liste déroulante, on peut regrouper les options avec l'élément <optgroup> </optgroup>

```
<body>
 <form method="post" action="traitement.php">
 <p>
 <label for="pays">Dans quel pays habitez-vous ?</label>

 <select name="pays" id="pays">
 <optgroup label="Europe">
 <option value="france">France</option>
 <option value="espagne">Espagne</option>
 <option value="italie">Italie</option>
 <option value="royaume-uni">Royaume-Uni</option>
 </optgroup>
 <optgroup label="Amérique">
 <option value="canada">Canada</option>
 <option value="etats-unis">États-Unis</option>
 </optgroup>
 <optgroup label="Asie">
 <option value="chine">Chine</option>
 <option value="japon">Japon</option>
 </optgroup>
 </select>
 </p>
 </form>
</body>
```

Dans quel pays habitez-vous ?

France ▼

Europe

France

Espagne

Italie

Royaume-Uni

Amérique

Canada

États-Unis

Asie

Chine

Japon

# Améliorer un formulaire

- On peut améliorer l'ergonomie ou la précision d'un formulaire en utilisant différents attributs HTML:
  - L'attribut **autocomplete** : permet de finir automatiquement le remplissage d'un champ en se basant sur les informations entrées précédemment par les utilisateurs. Il peut prendre la valeur :
    - "on" : le champ s'autocomplete
    - "off" : le champ ne s'autocomplètera pas
  - L'attribut **autofocus** : permet d'attribuer automatiquement le focus sur un élément lors du chargement de la page
  - L'attribut **required** : permet de rendre un champ de formulaire obligatoire, c'est-à-dire que si l'utilisateur n'a pas rempli ce champ, le formulaire ne pourra pas être envoyé.

82

# HTML : AUDIO ET VIDÉO

---

3<sup>ÈME</sup> ANNÉE LICENCE EN INFORMATIQUE

FSM 2020-2021





# Audio en HTML

- Pour insérer de l'audio dans un document HTML, on utilise l'élément `<audio>`
- Il existe plusieurs formats audio (MP3, AAC, WAV,...) mais aucun de ces formats n'est reconnu de façon universelle par tous les navigateurs
- Il faut proposer donc plusieurs versions du fichier audio. Dans ce cas, on va construire notre balise comme ceci :

`<audio controls>`

`<source src="nom_fichier.mp3">`

`<source src="nom_fichier.ogg">`

Il serait temps de mettre à jour votre navigateur

`</audio>`



# Audio en HTML

- Vous pouvez compléter la balise `<audio>` par les attributs suivants :
  - **controls** : pour ajouter les boutons « Lecture », « Pause » et la barre de défilement.
  - **width** : pour modifier la largeur de l'outil de lecture audio
  - **loop** : la musique sera jouée en boucle
  - **autoplay** : la musique sera jouée dès le chargement de la page

# Vidéo en HTML

- Pour insérer une vidéo dans un document HTML, on utilise l'élément **<video>**
- Comme pour le cas des fichiers audio, il faut proposer plusieurs versions du fichier vidéo afin d'assurer une bonne compatibilité avec les navigateurs. Dans ce cas, on va construire notre balise comme ceci :

**<video controls>**

**<source src="nom\_fichier.mp4">**

**<source src="nom\_fichier.webm">**

**<source src="nom\_fichier.ogv">**

Il serait temps de mettre à jour votre navigateur

**</video>**

# Vidéo en HTML

- Vous pouvez compléter la balise `<video>` des attributs suivants :
  - **poster** : image à afficher à la place de la vidéo tant que celle-ci n'est pas lancée.
  - **controls** : pour ajouter les boutons « Lecture », « Pause » et la barre de défilement.
  - **width** : pour modifier la largeur de la vidéo
  - **height** : pour modifier la hauteur de la vidéo
  - **loop** : la vidéo sera jouée en boucle
  - **autoplay** : la vidéo sera jouée dès le chargement de la page

87

# HTML : BALISES DE STRUCTURE EN HTML5

---

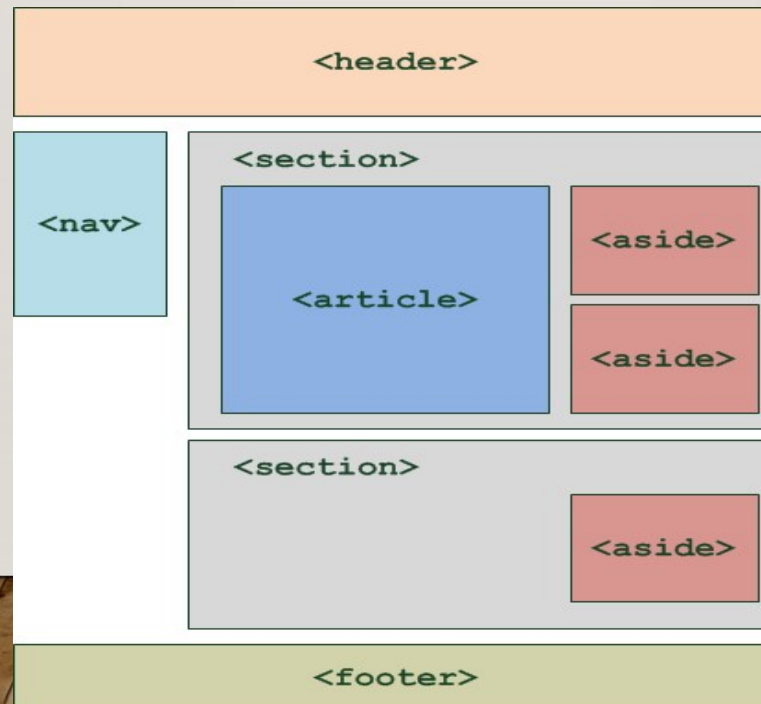
3<sup>ÈME</sup> ANNÉE LICENCE EN INFORMATIQUE

FSM 2020-2021



# Balises de structure en HTML5

- HTML5 propose de nouvelles balises pour mieux structurer le document web
- Ces balises sont très utiles au développeur pour comprendre l'architecture de la page





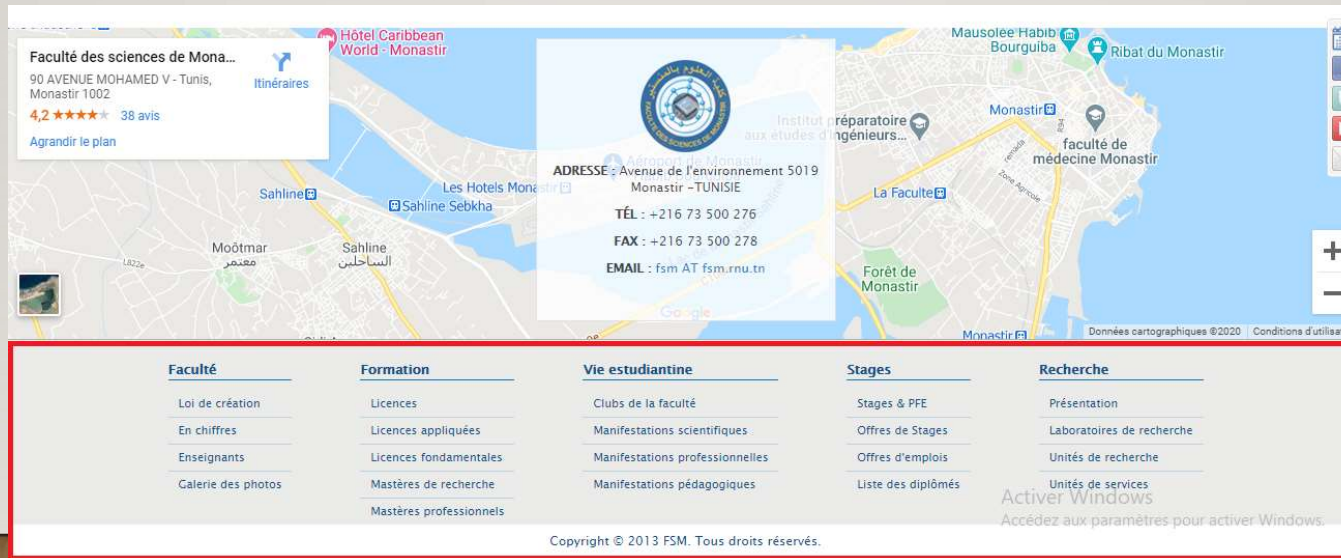
# Balise <header>

- **<header>** : permet de définir la partie haute (entête) d'une page web. On y trouve généralement, le logo, une bannière, ...



# Balise <footer>

- **<footer>** : définit le pied de page qui se trouve en général tout en bas du document. On y trouve des informations comme des liens de contact, le nom de l'auteur, etc.



# Balise <nav>

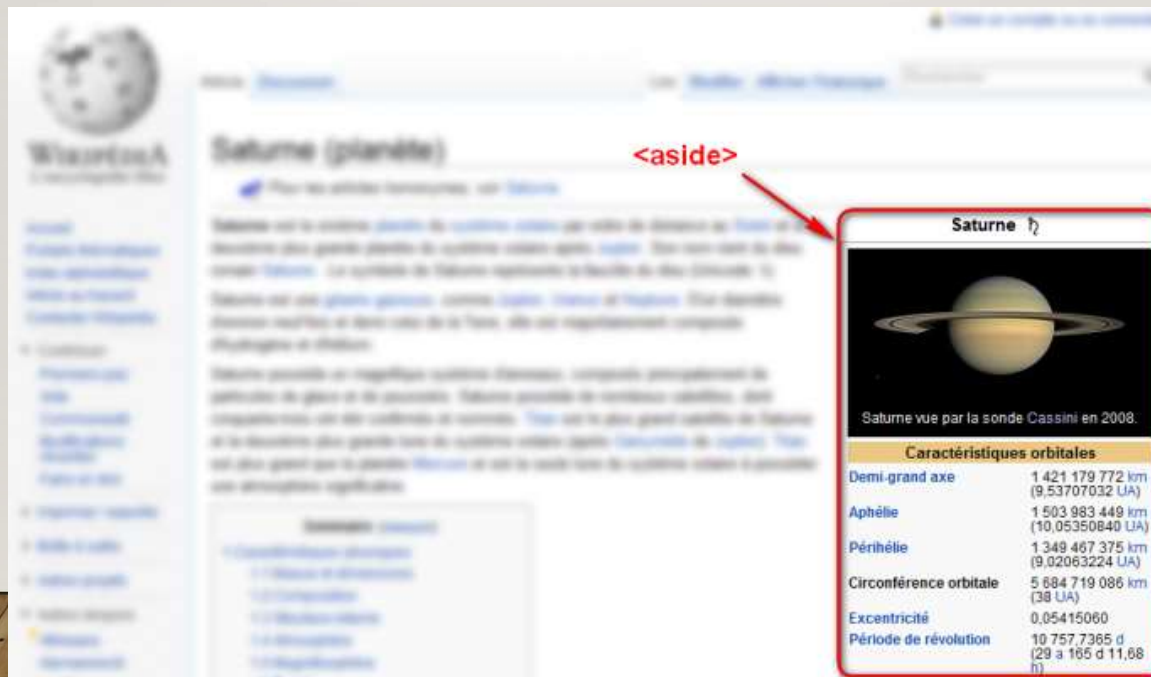
- **<nav>** : regroupe tous les principaux liens de navigation du site. On y trouve généralement le menu principal du site.





# Balise <aside>

- **<aside>** : conçue pour contenir des informations complémentaires au document que l'on visualise. Ces informations sont généralement placées sur le côté.



The image shows a screenshot of the French Wikipedia page for "Saturne (planète)". A red box highlights the "Caractéristiques orbitales" sidebar on the right, which is labeled with a red arrow and the text "<aside>". The sidebar contains a photograph of Saturn and a table of orbital data.

**Saturne  $\eta$**

Saturne vue par la sonde Cassini en 2008.

Caractéristiques orbitales	
Demi-grand axe	1 421 179 772 km (9,53707032 UA)
Aphélie	1 503 983 449 km (10,05350840 UA)
Périhélie	1 349 467 375 km (9,02063224 UA)
Circonférence orbitale	5 684 719 086 km (38 UA)
Excentricité	0,05415060
Période de révolution	10 757,7365 d (29 a 165 d 11,68 h)

## Les balises <section> et <article>

---

- Le contenu principal est organisé en deux parties:
  - **<section>** : sert à regrouper des contenus en fonction de leur thématique. Elle englobe généralement une portion du contenu au centre de la page
  - **<article>** : définit un contenu indépendant du document qui possède une identité à part comme l'article d'un blog, un post dans un forum ou un produit dans un site e-commerce



# Remarques

- Les positions des balises **<nav>** et **<aside>** peuvent changer selon vos besoins
- On peut imaginer une seconde balise **<header>**, placée à l'intérieur d'une **<section>**. Dans ce cas-là, elle sera considérée comme étant l'en-tête de la section.
- Une section ne doit pas forcément contenir un **<article>** et des **<aside>**. Utilisez ces balises uniquement si vous en avez besoin.

95

# CSS : FEUILLES DE STYLE

---

3<sup>ÈME</sup> ANNÉE LICENCE EN INFORMATIQUE

FSM 2020-2021



# Introduction

---

- HTML sert à structurer le contenu, CSS sert à la mise en forme
- Le CSS (*Cascading Style Sheet*) permet de mettre en place un contenu et de changer son apparence en lui appliquant des styles
- Il permet d'appliquer des styles différents au même document et surtout d'appliquer le même style à de nombreux documents !
- Le CSS est un ensemble de mises en forme génériques (position des éléments, aspect des textes à afficher, ...) associés à des balises.

# Sélecteurs et propriétés CSS

---

- Pour appliquer un style à un élément HTML, il faut le sélectionner ou le cibler
- Un sélecteur permet de cibler un ou plusieurs éléments HTML afin de pouvoir appliquer un style particulier
- Il existe différents types de sélecteurs:
  - Sélecteurs simples
  - Sélecteurs complexes

# Sélecteurs et propriétés CSS

- La grande majorité des sélecteurs CSS simples sont appelés des sélecteurs éléments
- Ils prennent le nom du type de l'élément HTML à cibler
- Syntaxe

**sélecteur {propriété: valeur; propriété: valeur; ...}**

- Les propriétés CSS permettent de choisir quels aspects ou styles d'un élément HTML qu'on souhaite modifier
- Une propriété CSS peut être accompagné d'une ou plusieurs **valeurs** qui vont définir le comportement de cette propriété



# Sélecteurs et propriétés CSS

- Exemples

**h1 {text-decoration: underline; color: red;}**

h1 est le sélecteur qui définit le style qui sera appliqué à la balise html <h1>, le titre <h1> sera ainsi souligné et de couleur rouge

**body {background-color: orange;}**

body est le sélecteur qui définit le style qui sera appliqué à la balise html <body>, la déclaration attribue la couleur orange au fond de la page

# Où mettre le code CSS?

- Trois méthodes pour lier le code (ou le style) CSS à un document HTML :
  - Dans un élément du fichier HTML (la méthode la moins recommandée)
  - Dans l'en-tête **<head>** du fichier HTML
  - Dans un fichier **.css** séparé (la méthode la plus recommandée)

# CSS dans un élément HTML

- Cette méthode consiste à ajouter un attribut *style* à n'importe quelle balise et insérer le code CSS directement dans cet attribut :

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8" />
 <title>Les bases en CSS</title>
 </head>

 <body>
 <h1>Je suis un titre</h1>

 <p style="color: red;">Je suis un paragraphe</p>
 <p>Je suis un deuxième paragraphe</p>
 </body>
</html>
```

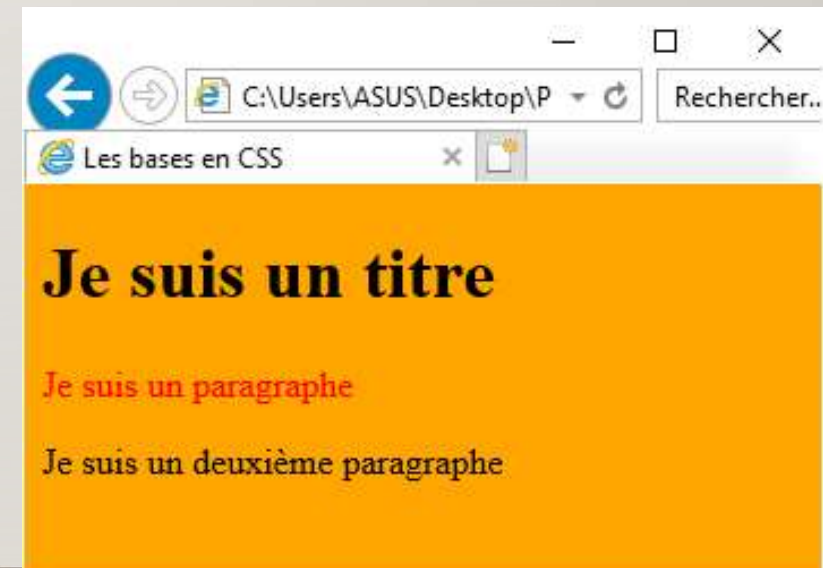


# CSS dans l'entête <head>

- Cette méthode consiste à insérer le code CSS directement dans une balise **<style>** à l'intérieur de l'en-tête **<head>**

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8" />
 <title>Les bases en CSS</title>
 <style>
 body{background-color: orange;}
 </style>
 </head>

 <body>
 <h1>Je suis un titre</h1>
 <p style="color: red;">Je suis un paragraphe</p>
 <p>Je suis un deuxième paragraphe</p>
 </body>
</html>
```



## CSS dans un fichier séparé (.css)

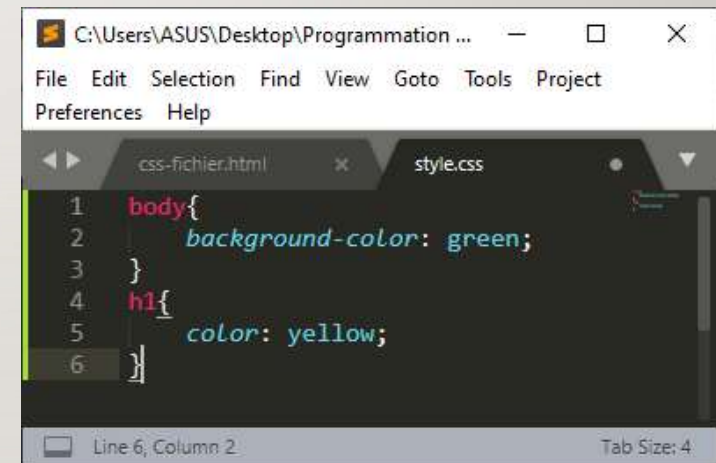
- Les deux méthodes précédentes ne sont généralement pas recommandées pour des raisons de performance et de maintenabilité du code
  - Il faut écrire un code CSS pour chaque page HTML (assez long pour les sites qui comportent plusieurs pages HTML)
- On recommande alors la méthode qui consiste à écrire le code CSS dans un fichier séparé
  - C'est la méthode la plus pratique et la plus souple
  - Le code CSS est écrit dans un fichier ayant l'extension `.css`
  - Il suffit d'ajouter la balise `<link rel="stylesheet" href="URL du fichier css">` dans l'entête (`<head>`) du fichier HTML



# CSS dans un fichier séparé (.css)

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8" />
 <title>Les bases en CSS</title>
 <link rel="stylesheet" href="style.css">
 </head>

 <body>
 <h1>Je suis un titre</h1>
 <p>Je suis un paragraphe</p>
 <p>Je suis un deuxième paragraphe</p>
 </body>
</html>
```



# Les commentaires en CSS

- Comme en HTML, il est possible de mettre des commentaires en CSS.
- Le commentaire est écrit entre ***/\*Ceci est un commentaire\*/***
- Les commentaires peuvent être écrits sur une ou plusieurs lignes

```
/*Ceci est un commentaire*/
body{
 background-color: green;
}
h1{
 color: yellow;
}

/*Ceci est
un commentaire sur
plusieurs lignes*/
```

# Limitation des sélecteurs simples

- Les sélecteurs simples sont des sélecteurs de type élément
- Un sélecteur simple c'est un sélecteur qui cible un élément HTML qui se base sur le nom de la balise
- Il est possible de grouper plusieurs sélecteurs ayant une règle de style identique, séparés par une virgule : `h1, h2, h3 { color: red }`
- **Limitation** : Appliquer deux couleurs différents à deux paragraphes différents. C'est possible avec les déclarations CSS dans la balise ouvrante des éléments HTML, mais cette méthode n'est pas recommandée
- **Solution** : les attributs id et class

## Les attributs *class* et *id*

- Les attributs **class** et **id** permettent de cibler un élément particulier du document HTML
- Ils sont définis sous forme d'attributs de la balise ouvrante de l'élément cible en leur attribuant une valeur
- Cette valeur va être utilisé par la suite dans le fichier CSS pour cibler l'élément :
  - On utilise un **#** avant la valeur de l'attribut **id**
  - On utilise un **.** avant la valeur de l'attribut **class**

# Les attributs *class* et *id*

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8" />
 <title>Les bases en CSS</title>
 <link rel="stylesheet" href="style.css">
 </head>
 <body>
 <h1>Je suis un titre</h1>
 <p id="p1">Je suis un paragraphe</p>
 <p class="p2">Je suis un deuxième paragraphe</p>
 </body>
</html>
```

```
#p1{
 color:blue;
}

.p2{
 color:red;
}
```





# Les attributs *class* et *id*

- Chaque attribut **id** doit avoir une valeur unique
- Plusieurs attributs **class** peuvent avoir la même valeur ce qui permet d'appliquer le même le style à différents types d'éléments HTML ciblés

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8" />
 <title>Les bases en CSS</title>
 <link rel="stylesheet" href="style.css">
 </head>

 <body>
 <h1>Je suis un titre</h1>
 <p id="p1">Je suis un paragraphe</p>
 <p class="p2">Je suis un deuxième paragraphe</p>
 <p class="p2">Je suis un troisième paragraphe</p>
 </body>
</html>
```



# Concept d'héritage en CSS

- L'héritage est une notion centrale et fondamentale du langage CSS
- L'héritage en CSS signifie que tout élément HTML enfant va hériter en cascade des styles de ses parents, c'est de là que vient le nom de CSS (*Cascading StyleSheet*)
- Par exemple, tous les éléments à l'intérieur de l'élément **<body>** vont être des enfants de cet élément vont hériter des styles de l'élément **<body>**

```
<!DOCTYPE html>
<html>
 <head>
 <meta charset="utf-8" />
 <title>Les bases en CSS</title>
 <link rel="stylesheet" href="style.css">
 </head>
 <body>
 <h1>Je suis un titre</h1>
 <p id="p1">Je suis un paragraphe</p>
 <p class="p2">Je suis un deuxième paragraphe</p>
 <p class="p2">Je suis un troisième paragraphe</p>
 </body>
</html>
```

```
body{
 color: blue;
}
```





# Concept d'héritage en CSS

- En CSS, la règle est simple : s'il y a conflit, c'est-à-dire s'il y a un élément qui reçoit plusieurs fois la même propriété avec des valeurs différentes, le style le plus proche de l'élément est le style le plus prioritaire

```
body{
 color: blue;
}

p{
 color: red;
}

#p1{
 color: green;
}
```



## Les balises *div* et *span*

- Il arrive parfois qu'on aura besoin de mettre en forme uniquement un mot dans un paragraphe et non pas tout le paragraphe.
- Vu que ce mot n'est pas entouré par une balise, on ne pourra pas appliquer les attributs **id** et **class**
- Pour remédier à ce problème, deux balises universelles ont été créées, à savoir les balises **<div>** et **<span>**
- Elles sont un peu spéciales puisqu'elles ne possèdent aucune valeur sémantique (qui ne servent pas à donner un sens à un contenu) contrairement aux autres balises
- Ils servent de conteneurs pour faciliter la mise en forme des pages web



# La balise `<span>`

- `<span>` `</span>` : c'est une balise de type **inline**, c'est-à-dire une balise que l'on place au sein d'un paragraphe de texte pour sélectionner certains mots uniquement.
- Un élément de type *inline* est un élément qui va s'insérer dans la ligne actuelle et il prend juste la largeur nécessaire tels que `<strong>`, `<em>`, `<a>`, `<img>`

```
<body>
 <h1>Div et span</h1>
 <p>L'élément em
 est un élément de type inline</p>
</body>
```

```
.span-em{
 color:red;
 text-decoration: underline;
 font-weight: bold;
}
```





# La balise `<div>`

- `<div>` `</div>` : c'est une balise de type **block**, qui entoure un bloc de texte.
- Un élément de type *block* commence toujours sur une nouvelle ligne et prend toute la largeur disponible dans une page tels que `<p>`, `<h1>`...`<h6>`, `<ol>`,`<ul>`, `<form>`
- Un élément de type *block* peut contenir d'autres éléments de type block et inline

```
<body>
 <h1>Div et span</h1>
 <p>L'élément em
 est un élément de type inline</p>
 <div class="div-para">
 <p>L'élément p est un élément de type block
 </p>
 <p>Un autre paragraphe</p>
 </div>
</body>
```

```
.span-em{
 color:red;
 text-decoration: underline;
 font-weight: bold;
}

.div-para{
 background-color: #88BB11;
 font-weight: bold;
}
```

