



# HTML ET CSS



<https://sdadclub.tech/>



**Ce cours est réaliser par :**  
**IHINNA HASNA**

**-Etudiante en master sciences des  
données et aide à la décision à  
l'université Cadi Ayyad  
-Membre du club SDAD**



 [hasnaihi8@gmail.com](mailto:hasnaihi8@gmail.com)

 <https://www.linkedin.com/in/hasna-ihinna-1679a7202/>



# HTML5

# HTML





## Introduction 1/6

---

- HTML est le langage de balisage standard pour la création de pages Web.
- HTML signifie Hyper Text Markup Language
- HTML décrit la structure des pages Web en utilisant le balisage
- Les éléments HTML sont les blocs de construction des pages HTML
- Les éléments HTML sont représentés par des balises
- Les balises HTML étiquettent des éléments de contenu tels que "en-tête", "paragraphe", "table", etc.
- Les navigateurs n'affichent pas les balises HTML, mais les utilisent pour rendre le contenu de la page



## Introduction 2/6

```
<!DOCTYPE html>
<html>
<head>
<title>Page Title</title>
</head>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
```

- La déclaration `<!DOCTYPE html>` définit ce document comme étant HTML5
- L'élément `<html>` est l'élément racine d'une page HTML
- L'élément `<head>` contient des méta-informations sur le document
- L'élément `<title>` spécifie un titre pour le document
- L'élément `<body>` contient le contenu de la page visible
- L'élément `<h1>` définit un gros titre
- L'élément `<p>` définit un paragraphe



## Introduction 3/6

### Balises HTML

Les balises HTML sont des noms d'éléments entourés de crochets:

```
<nom-balise> le contenu va ici ... </ nom-balise>
```

- Les balises HTML se présentent normalement par paires comme `<p>` et `</ p>`
- La première balise d'une paire est la balise de début, la seconde balise est la balise de fin
- La balise de fin est écrite comme la balise de début, mais avec une barre oblique avant le nom de la balise

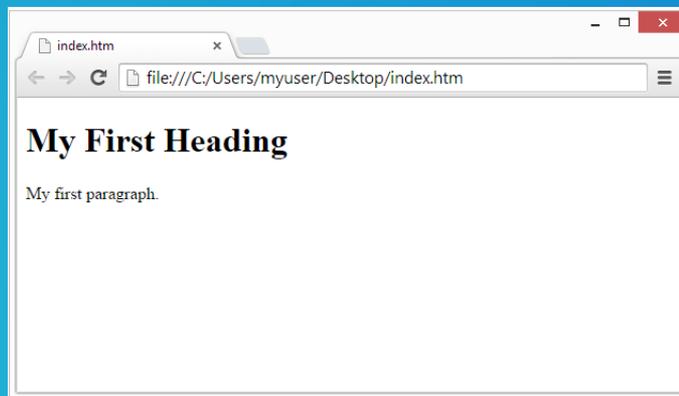
La balise de début est également appelée la balise d'ouverture et la balise de fin la balise de fermeture.



# Introduction 4/6

## Navigateurs Web

- Un navigateur Web (Chrome, IE, Firefox, Safari) a pour but de lire des documents HTML et de les afficher.
- Le navigateur n'affiche pas les balises HTML, mais les utilise pour déterminer comment afficher le document:





## Introduction 5/6

### Structure d'une page HTML

Ci-dessous une visualisation d'une structure de page HTML:

```
<html>
  <head>
    <title>Page title</title>
  </head>
  <body>
    <h1>This is a heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
  </body>
</html>
```

- La déclaration `<! DOCTYPE>` représente le type de document et aide les navigateurs à afficher correctement les pages Web.
- Il ne doit apparaître qu'une seule fois, en haut de la page (avant toute balise HTML).
- La déclaration `<! DOCTYPE>` n'est pas sensible à la casse.
- La déclaration `<! DOCTYPE>` pour HTML5 est: `<!DOCTYPE html>`

Seul le contenu de la section `<body>` (la zone blanche) est affiché dans un navigateur.



## Introduction 6/6

### Versions HTML

Version	Année
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014



## Editeur HTML

---

- Les pages Web peuvent être créées et modifiées à l'aide d'éditeurs HTML professionnels.
- Cependant, pour apprendre le HTML, nous recommandons un éditeur de texte simple comme Notepad (PC) ou TextEdit (Mac).
- Nous pensons que l'utilisation d'un éditeur de texte simple est un bon moyen d'apprendre le langage HTML.
- Suivez les quatre étapes ci-dessous pour créer votre première page Web avec Notepad ou TextEdit.



## Editeur HTML

### Etape 1

Ouvrir Notepad sous Windows  
ou TextEdit sous Mac

### Etape 2

Écrivez ou copiez du code  
HTML dans le Notepad.

### Etape 3

Enregistrez le fichier sur votre  
ordinateur. Sélectionnez  
« Fichier> Enregistrer sous»  
dans le menu Notepad.

### Etape 4

Ouvrez le fichier HTML enregistré dans votre  
navigateur préféré (double-cliquez sur le fichier  
ou cliquez avec le bouton droit de la souris - et  
choisissez "Ouvrir avec").



## Élément HTML?

Un élément HTML consiste généralement en une balise de début et une balise de fin, le contenu étant inséré entre:

```
<nom-balise> le contenu va ici ... </ nom-balise>
```

exemple

```
<h1> le contenu va ici ... </ h1>
```

Element HTML

Les éléments HTML sans contenu sont appelés éléments vides. Les éléments vides n'ont pas de balise de fin, comme l'élément `<br>` (qui indique un saut de ligne).



## Attributs HTML?

Les attributs fournissent des informations complémentaires sur les éléments HTML

- Tous les éléments HTML peuvent avoir des attributs
- Les attributs fournissent des informations supplémentaires sur un élément
- Les attributs sont toujours spécifiés dans la balise de début
- Les attributs viennent généralement en paires nom / valeur comme: nom= « valeur »





# Elément et Attributs de base HTML

## Les en-têtes HTML

- Les en-têtes sont définis avec les balises `<h1>` à `<h6>`.
- `<h1>` définit l'en-tête le plus important. `<h6>` définit l'en-tête le moins important.

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

Interprétation

```
en-tête 1  
en-tête 2  
en-tête 3  
en-tête 4  
en-tête 5  
en-tête 6
```



# Elément et Attributs de base HTML

## Les paragraphes HTML

- L'élément HTML `<p>` définit un paragraphe
- Les navigateurs ajoutent automatiquement un espace blanc (une marge) avant et après un paragraphe.

```
<p> Ceci est un paragraphe </p>  
<p> Ceci est un autre paragraphe </p>
```

Utiliser l'élément HTML `<br>` pour définir un saut de ligne.

- Vous ne pouvez pas être sûr de la manière dont HTML sera affiché.
- De grands ou petits écrans et des fenêtres redimensionnées créeront des résultats différents.
- Avec HTML, vous ne pouvez pas modifier la sortie en ajoutant des espaces supplémentaires ou des lignes supplémentaires dans votre code HTML.
- Le navigateur supprimera tous les espaces supplémentaires et les lignes supplémentaires lorsque la page est affichée.



# Elément et Attributs de base HTML

## Les Liens HTML

- Les liens se trouvent dans presque toutes les pages Web. Les liens permettent aux utilisateurs de cliquer sur leur chemin de page en page.
- Les liens HTML sont des hyperliens.
- Vous pouvez cliquer sur un lien et accéder à un autre document.
- Lorsque vous déplacez la souris sur un lien, la flèche de la souris se transforme en petite main.

Un lien ne doit pas nécessairement être du texte. Ce peut être une image ou tout autre élément HTML.



# Elément et Attributs de base HTML

## Les Liens HTML

- En HTML, les liens sont définis par la balise `<a>`:

```
<a href=" https://www.google.com " > text du lien </a>
```

- L'attribut `href` spécifie l'adresse de destination du lien (`https://www.google.com`) .
- Le texte du lien est la partie visible (*text du lien*).
- En cliquant sur le texte du lien, vous serez envoyé à l'adresse indiquée (`https://www.google.com`).

L'exemple ci-dessus utilise une URL absolue (une adresse Web complète). Un lien local (lien vers le même site Web) est spécifié avec une URL relative (sans `https://www` ...).



# Elément et Attributs de base HTML

## Les Liens HTML- L'attribut « target »

L'attribut « target » spécifie où ouvrir le document lié.

L'attribut « target » peut avoir l'une des valeurs suivantes:

- `_blank` : Ouvre le document lié dans une nouvelle fenêtre ou un nouvel onglet
- `_self` : Ouvre le document lié dans la même fenêtre / le même onglet que celui sur lequel vous avez cliqué (ceci est la valeur par défaut)
- `_parent` : Ouvre le document lié dans le cadre parent
- `_top` : Ouvre le document lié dans le corps entier de la fenêtre
- `_frameName` : Ouvre le document lié dans un cadre nommé

```
<a href="https://www.google.com/" target="_blank" >aller à google!</a>
```



# Élément et Attributs de base HTML

## Les images

En HTML, les images sont définies avec la balise `<img>`.

La balise `<img>` est vide, elle contient uniquement des attributs et n'a pas de balise de fermeture.

L'attribut `src` spécifie l'URL (adresse Web) de l'image:

```

```

L'attribut `alt`  
fournit un texte  
alternatif pour une  
image

les attributs `width` et `height`  
définissent toujours la largeur et la  
hauteur de l'image en pixels.

2014



# Elément et Attributs de base HTML

## Les listes HTML

Il y a deux types de liste:

Liste non ordonnée

- Item
- Item
- Item
- Item

Liste ordonnée

1. Item
2. Item
3. Item
4. Item



# Élément et Attributs de base HTML

## Les listes non ordonnées

- Une liste non ordonnée commence par la balise `<ul>`. Chaque élément de la liste commence par la balise `<li>`.
- Les éléments de la liste seront marqués par des puces (petits cercles noirs) par défaut:

```
<ul>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ul>
```

Interprétation

- Coffee
- Tea
- Milk



# Élément et Attributs de base HTML

## Les Listes non ordonnées - le marqueur d'élément de liste 1/2

La propriété **list-style-type** est utilisée pour définir le style du marqueur d'élément de liste:

Valeur	Description
disc	Définit le marqueur d'élément de liste sur une puce (par défaut)
circle	Définit le marqueur d'élément de liste sur une cercle
square	Définit le marqueur d'élément de liste sur un carré
none	Les éléments de la liste ne seront pas marqués





# Élément et Attributs de base HTML

## Les Listes ordonnées

- Une liste ordonnée commence par la balise `<ol>`. Chaque élément de la liste commence par la balise `<li>`.
- Les éléments de la liste seront marqués par des numéros par défaut:

```
<ol>  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

Interprétation

1. Coffee
2. Tea
3. Milk



# Elément et Attributs de base HTML

## Les Listes ordonnées - L'attribut de type 1/2

L'attribut **type** de la balise `<ol>` définit le type du marqueur d'élément de liste::

Type	Description
1	Les éléments de la liste seront numérotés avec des numéros (par défaut)
A	Les éléments de la liste seront numérotés avec des lettres majuscules
a	Les éléments de la liste seront numérotés avec des lettres minuscules
I	Les éléments de la liste seront numérotés avec des chiffres romains en majuscule
i	Les éléments de la liste seront numérotés avec des chiffres romains en minuscules



# Elément et Attributs de base HTML

## Les Listes ordonnées - L'attribut de type 2/2

```
<ol type="1">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

```
<ol type="A">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

```
<ol type="a">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```

```
<ol type="I">  
  <li>Coffee</li>  
  <li>Tea</li>  
  <li>Milk</li>  
</ol>
```



# Elément et Attributs de base HTML

## Tableau HTML

- Un tableau HTML est défini avec la balise `<table>`.
- Chaque ligne de table est définie avec la balise `<tr>`. Un en-tête de tableau est défini avec la balise `<th>`. Par défaut, les en-têtes de tableau sont en gras et centrés.
- Chaque cellule est définie avec la balise `<td>`.

```
<table style="width:100%"border= "1px">  
  <tr>  
    <th>Firstname</th>  
    <th>Lastname</th>  
  </tr>  
  <tr>  
    <td>Jill</td>  
    <td>Smith</td>  
  </tr>  
</table>
```

Interprétation

Firstname	Lastname
Jill	Smith



## Elément et Attributs de base HTML

### Tableau HTML- cellules de plusieurs colonnes

Pour faire qu'une cellule avec plusieurs colonnes, utilisez l'attribut **colspan**:

```
<table style="width:100%" border="1px">
<tr>
  <th>Name</th>
  <th colspan="2">Telephone</th>
</tr>
<tr>
  <td>Bill Gates</td>
  <td>55577854</td>
  <td>55577855</td>
</tr>
</table>
```

Interprétation

Name	Telephone	
Bill Gates	55577854	55577855



## Elément et Attributs de base HTML

### Tableau HTML- cellule de plusieurs lignes

Pour faire qu'une colonne avec plusieurs lignes, utilisez l'attribut **rowspan**:

```
<table style="width:100%">
  <tr>
    <th>Name:</th>
    <td>Bill Gates</td>
  </tr>
  <tr>
    <th rowspan="2">Telephone:</th>
    <td>55577854</td>
  </tr>
  <tr>
    <td>55577855</td>
  </tr>
</table>
```

Interprétation

<b>Name:</b>	Bill Gates
<b>Telephone:</b>	55577854
	55577855





# Elément et Attributs de base HTML

## Formulaires HTML

attribut	Description
action	Spécifie une adresse (URL) où soumettre le formulaire (par défaut: la page de soumission)
method	Spécifie la méthode HTTP utilisée lors de la soumission du formulaire (par défaut: GET).
name	Spécifie un nom utilisé pour identifier le formulaire (pour une utilisation DOM: document.forms.name).
novalidate	Spécifie que le navigateur ne doit pas valider le formulaire.
target	Spécifie la cible de l'adresse dans l'attribut action (par défaut: <code>_self</code> ).



# Elément et Attributs de base HTML

## Formulaires HTML- l'élément <input>

- L'élément `<input>` est l'élément de forme le plus important.
- L'élément `<input>` peut être affiché de plusieurs manières, en fonction de l'attribut type.
- Voici quelques exemples:

Type	Description
<code>&lt;input type="text"&gt;</code>	Définit un champ de saisie de texte d'une ligne
<code>&lt;input type="radio"&gt;</code>	Définit un bouton radio (pour sélectionner l'un des nombreux choix)
<code>&lt;input type="submit"&gt;</code>	Définit un bouton de soumission (pour soumettre le formulaire)



# Elément et Attributs de base HTML

## Formulaires HTML- l'élément `<input>`

`<input type="text">` définit un champ de saisie de texte d'une ligne:

`<input type = "password">` définit un champ de mot de passe:

`<input type = "reset">` définit un bouton de réinitialisation qui réinitialisera toutes les valeurs de formulaire à leurs valeurs par défaut:

```
<form action="">  
User name:<br>  
<input type="text" name="userid">  
<br>  
User password:<br>  
<input type="password" name="psw">  
<input type = "reset">  
</form>
```

User name:

User password:

Réinitialiser

Chaque zone de saisie doit avoir un attribut **name**. Si l'attribut **name** est omis, les données de ce champ d'entrée ne seront pas envoyées du tout.



# Elément et Attributs de base HTML

## Formulaires HTML- l'élément <input>

- `<input type = "radio">` définit un bouton radio.  
Les boutons radio permettent à l'utilisateur de sélectionner SEULEMENT UN parmi un nombre limité de choix:
- `<input type = "checkbox">` définit une case à cocher.  
Les cases à cocher permettent à un utilisateur de sélectionner les options ZÉRO ou PLUS parmi un nombre limité de choix.

```
<form>  
<input type="radio" name="gender" value="male" checked> Male<br>  
<input type="radio" name="gender" value="female"> Female<br>  
<input type="checkbox" name="vehicle1" value="Bike">I have a bike<br>  
<input type="checkbox" name="vehicle2" value="Car">I have a car <br>  
</form>
```

- Male
- Female
- I have a bike
- I have a car



# Elément et Attributs de base HTML

## Formulaires HTML- l'élément `<input>`

`<input type = "submit">` définit un bouton pour soumettre les données du formulaire à un gestionnaire de formulaire.

Le gestionnaire de formulaire est généralement **une page de serveur** avec un script pour traiter les données d'entrée.

Le gestionnaire de formulaire est spécifié dans l'attribut d'action du formulaire:

```
<form action="/action_page.php">  
  First name:<br>  
  <input type="text" name="firstname" value="Mickey"><br>  
  Last name:<br>  
  <input type="text" name="lastname" value="Mouse"><br>  
  <input type="submit" value="Submit">  
</form>
```

First name:  
Mickey  
Last name:  
Mouse  
Submit



# Elément et Attributs de base HTML

## Formulaires HTML- l'élément <fieldset>

L'élément `<fieldset>` est utilisé pour regrouper des données associées dans un formulaire.

L'élément `<legend>` définit une légende pour l'élément `<fieldset>`.

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personal information:</legend>
    First name:<br>
    <input type="text" name="firstname" value="Mickey"><br>
    Last name:<br>
    <input type="text" name="lastname" value="Mouse"><br><br>
    <input type="submit" value="Submit">
  </fieldset>
</form>
```

Personal information:

First name:  
Mickey

Last name:  
Mouse

Submit



# Elément et Attributs de base HTML

## Formulaires HTML- l'élément `<select>`

- L'élément `<select>` définit une liste déroulante:
- Les éléments `<option>` définissent une option pouvant être sélectionnée.
- Par défaut, le premier élément de la liste déroulante est sélectionné.
- Pour définir une option présélectionnée, ajoutez l'attribut `selected` à l'option:

`<option value="fiat" selected>Fiat</option>`

- Utilisez l'attribut `size` pour spécifier le nombre de valeurs visibles:

`<select name="cars" size="3">`

- Utilisez l'attribut `multiple` pour permettre à l'utilisateur de sélectionner plusieurs valeurs:

`<select name="cars" size="4" multiple>`

```
<select name="cars">
  <option value="volvo">Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

Volvo ▾
Volvo
Saab
Fiat
Audi

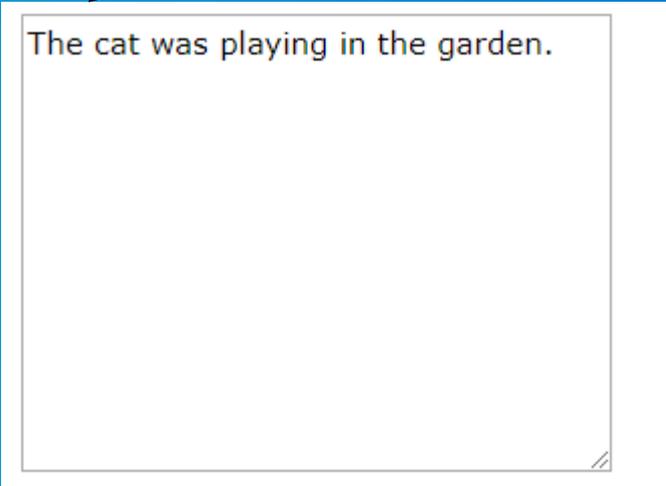


# Elément et Attributs de base HTML

## Formulaires HTML- l'élément <textarea>

- L'élément `<textarea>` définit un champ de saisie
- L'attribut **rows** spécifie le nombre visible de lignes dans une zone de texte.
- L'attribut **cols** spécifie la largeur visible d'une zone de texte.

```
<textarea name="message" rows="10" cols="30">  
The cat was playing in the garden.  
</textarea>
```



The cat was playing in the garden.

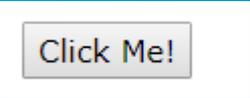


# Elément et Attributs de base HTML

## Formulaires HTML- l'élément <button>

- L'élément <button> définit un bouton cliquable:

```
<button type="button" onclick="alert('Hello World!')>Click Me!</button>
```



Click Me!



# Élément et Attributs de base HTML

## Formulaires HTML- l'élément <datalist>

- L'élément <datalist> spécifie une liste d'options prédéfinies pour un élément <input>.
- Les utilisateurs verront une liste déroulante des options prédéfinies lors de la saisie des données.
- L'attribut list de l'élément <input>, doit faire référence à l'attribut id de l'élément <datalist>.

```
<form action="/action_page.php">  
<input list="browsers">  
<datalist id="browsers">  
  <option value="Internet Explorer">  
  <option value="Firefox">  
  <option value="Chrome">  
  <option value="Opera">  
  <option value="Safari">  
</datalist>  
</form>
```



## Elément et Attributs de base HTML

### Formulaires HTML- l'élément <datalist>

- HTML5 a ajouté plusieurs nouveaux types d'entrée:

**color** est utilisé pour les champs de saisie qui doivent contenir une couleur.

**date** est utilisé pour les champs de saisie qui doivent contenir une date

**datetime-local** spécifie un champ de saisie de la date et de l'heure

**email** est utilisé pour les champs de saisie qui doivent contenir une adresse électronique

**file** définit un champ de sélection de fichier et un bouton "Parcourir" pour les téléchargements de fichiers..

**Month** permet à l'utilisateur de sélectionner un mois et une année.

**Number** définit un champ de saisie numérique.

**Range** définit un contrôle pour la saisie d'un nombre dont la valeur exacte n'a pas d'importance

**Search** est utilisé pour les champs de recherche (un champ de recherche se comporte comme un champ de texte normal).



# Élément et Attributs de base HTML

## Layouts HTML

- Les sites Web affichent souvent le contenu sur plusieurs colonnes.
- HTML5 offre de nouveaux éléments sémantiques qui définissent les différentes parties d'une page Web:

`<header>` - Définit un en-tête pour un document ou une section

`<nav>` - Définit un conteneur pour les liens de navigation

`<section>` - Définit une section dans un document

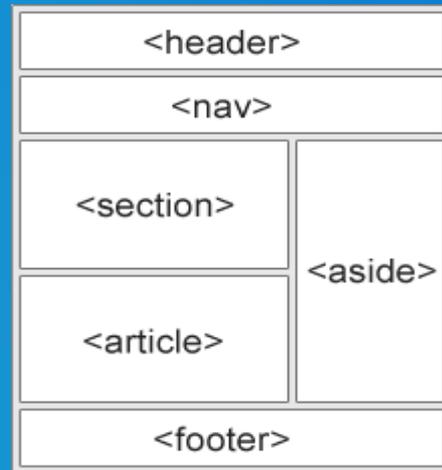
`<article>` - Définit un article indépendant

`<aside>` - Définit le contenu en dehors du contenu (comme une barre latérale)

`<footer>` - Définit un pied de page pour un document ou une section.

`<détails>` - Définit des détails supplémentaires

`<summary>` - Définit un en-tête pour l'élément `<details>`



# Elément et Attributs de base HTML

## Layouts HTML Exemple



**Zozor**  
Carnets de voyage

`<nav>`

[ACCUEIL](#) [BLOG](#) [CV](#) [CONTACT](#)

---

`<header>`



`<div>`

Retour sur mes vacances aux États-Unis... Voir l'article ▶

---

`<article>`

### JE SUIS UN GRAND VOYAGEUR

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris metus massa, luctus in tincidunt a, porttitor ac leo. Maecenas at mi feugiat turpis elementum ornare. Sed tempor rutrum lorem, in vestibulum felis elementum ac. Fusce purus orci, scelerisque ut tincidunt in, dignissim vel augue. Nulla laculis ultrices sagittis. Nulla vitae neque dignissim enim tempor scelerisque at quis tellus. In hac habitasse platea dictumt. Aenean elit elit, pellentesque nec venenatis id, convallis eu sem. Mauris eu leo nec arcu volutpat euismod nec eu dolor. Morbi aliquet, mauris quis porttitor dapibus, odio enim viverra eros, quis interdum massa urna at velit. Integer tempor facilisis libero non accumsan. Aliquam diam felis, dapibus sed condimentum quis, molestie vel odio. Maecenas eget ante massa, a sagittis quam. Cras posuere magna ac urna molestie vitae luctus lacus lobortis. Quisque leo neque, vulputate at semper non, varius porta enim.

Præsent sit amet luctus eros, ac pellentesque nisi. Donec consequat magna sed libero condimentum vitae aliquet elit ornare. Nunc at nulla purus. Aliquam sit amet sapien sit amet nisi aliquet rutrum vel nec mi. Mauris ultrices felis egestas mi varius molestie molestie sapien tristique. Cras lacus lacus, rutrum id sagittis sit amet, malesuada nec odio. Nulla consectetur lobortis libero, ac convallis massa consectetur in. Nam facilisis posuere sagittis. Sed a ligula id dui vulputate congue quis at tortor. Nunc pellentesque faucibus felis, eu venenatis massa interdum in. Donec venenatis lacus id tortor vestibulum id accumsan est lobortis. Morbi turpis quam, tincidunt in accumsan quis, ullamcorper quis orci. Quisque nisi magna, egestas eget consectetur non, mollis ac ante. Donec elit felis, blandit at auctor in, lacinia et dolor.

Ut blandit, diam id aliquam volutpat, quam libero euismod neque, ut volutpat nunc ipsum a magna. Donec hendrerit sem in dolor egestas lobortis. Etiam bibendum lobortis interdum. Etiam ac felis vitae neque sodales sodales. Nunc tempus dignissim dapibus. Duis sit amet tellus vitae elit suscipit convallis. Sed et tincidunt velit. Donec congue elementum ante eu consectetur. Morbi lectus mauris, sodales a euismod id, dapibus sollicitudin urna. Sed sagittis sagittis placerat. Etiam at lorem risus. Quisque imperdiet elementum tortor nec viverra. Tempus dignissim dapibus. Duis sit amet tellus vitae elit suscipit convallis. Sed et tincidunt velit. Donec congue elementum ante eu consectetur. Morbi lectus mauris, sodales a euismod id, dapibus sollicitudin urna. Sed sagittis sagittis placerat.

`<aside>`

#### À PROPOS DE L'AUTEUR



Laisse-moi le temps de me présenter : je m'appelle Zozor, je suis né un 23 Novembre 2005.

Bien maigre, n'est-ce pas ? C'est pourquoi, aujourd'hui, j'ai décidé d'écrire ma Biographie (ou ZBiographie, comme vous voulez !) afin que les autres sachent enfin qui je suis réellement.







`<section>`

---

`<div>`

**MON DERNIER TWEET**

Hii haaaaan !

le 12 mai à 23h12

**MES PHOTOS**



**MES AMIS**

- ▶ Pupi le lapin
- ▶ Mr Baobab
- ▶ Kahwall
- ▶ Perceval.eu
- ▶ Belette
- ▶ Le concombre masqué
- ▶ Ptit prince
- ▶ Mr Fan

`<footer>`



# Elément et Attributs de base HTML

## HTML et CSS

- CSS est l'acronyme de Cascading Style Sheets.
- CSS décrit comment les éléments HTML doivent être affichés à l'écran
- CSS économise beaucoup de travail. Il peut contrôler la mise en page de plusieurs pages Web en même temps.
- CSS peut être ajouté aux éléments HTML de 3 manières:

**inline**- en utilisant l'attribut style dans les éléments HTML

**inteme**- en utilisant un élément <style> dans la section <head>

**exteme**- en utilisant un fichier CSS externe



# Elément et Attributs de base HTML

## HTML et CSS – CSS inline

- Un CSS inline est utilisé pour appliquer un style unique à un seul élément HTML
- Un CSS en ligne utilise l'attribut style d'un élément HTML
- Cet exemple définit la couleur du texte de l'élément `<h1>` sur bleu:

### Exemple

```
<h1 style="color:blue;">This is a Blue  
Heading</h1>
```

**This is a Blue Heading**

```
<h1 style="color:blue;background-color:red;">This is a Blue Heading</h1>
```

**This is a Blue Heading**



# Elément et Attributs de base HTML

## HTML et CSS – CSS interne

- Un CSS interne est utilisé pour définir un style pour une seule page HTML.
- Un CSS interne est défini dans la section <head> d'une page HTML, au sein d'un élément <style>:

### Exemple

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color: powderblue;}
h1 {color: blue;}
p {color: red;}
</style>
</head>
<body>

<h1>This is a heading</h1>
<p>This is a paragraph.</p>

</body>
</html>
```

**This is a heading**

This is a paragraph.



# Elément et Attributs de base HTML

## HTML et CSS – CSS externe

- Une feuille de style externe est utilisée pour définir le style de nombreuses pages HTML.
- Avec une feuille de style externe, vous pouvez changer l'apparence d'un site Web entier, en changeant un fichier!
- Pour utiliser une feuille de style externe, ajoutez un lien dans la section <head> de la page HTML:
- **Exemple**

```
<!DOCTYPE html>
<html>
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```



# Elément et Attributs de base HTML

---

## HTML et CSS – CSS externe

- Une feuille de style externe peut être écrite dans n'importe quel éditeur de texte. Le fichier ne doit contenir aucun code HTML et doit être enregistré avec une extension .css.
- Voici à quoi ressemble « styles.css » de l'exemple précédent:

```
body {  
    background-color: powderblue;  
}  
h1 {  
    color: blue;  
}  
p {  
    color: red;  
}
```



# Elément et Attributs de base HTML

## HTML et CSS – Attribut id (#)

- Pour définir un style spécifique pour un élément spécial, ajoutez un attribut id à l'élément:

```
<p id="p01">I am different</p>
```

- puis définissez un style pour l'élément avec l'identifiant spécifique:

```
#p01 {  
  color: blue;  
}
```



# Elément et Attributs de base HTML

## HTML et CSS – Attribut id

### Exemple

```
<!DOCTYPE html>
<html>
<head>
<style>
#p01 {
  color: blue;
}
</style>
</head>
<body>
<p>This is a paragraph.</p>
<p>This is a paragraph.</p>
<p id="p01">I am different.</p>
</body>
</html>
```

This is a paragraph.

This is a paragraph.

I am different.



# Élément et Attributs de base HTML

## HTML et CSS – Attribut class (.)

- Pour définir un style pour des types spéciaux d'éléments, ajoutez un attribut de classe à l'élément:

```
<head>
<style>
.error {
  color: red;
}
</style>
</head>
<body>
<H1 class="error">This is a paragraph.</H1>
<p>This is a paragraph.</p>
<p class="error">I am different.</p>
<p>This is a paragraph.</p>
<p class="error">I am different too.</p>
```

**This is a paragraph.**

This is a paragraph.

I am different.

This is a paragraph.

I am different too.



CSS3

CSS





# Introduction

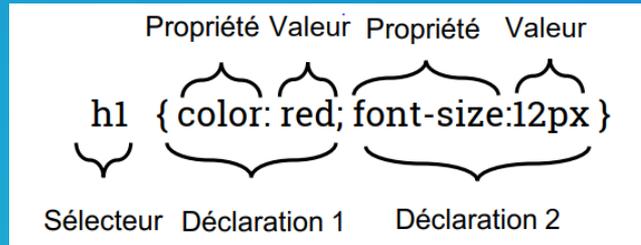
---

- CSS est l'acronyme de Cascading Style Sheets.
  - CSS décrit comment les éléments HTML doivent être affichés à l'écran, sur papier ou sur un autre support.
  - CSS économise beaucoup de travail. Il peut contrôler la mise en page de plusieurs pages Web en même temps
  - Les feuilles de style externes sont stockées dans des fichiers CSS (.css)
- CSS permet de définir les styles de vos pages Web, notamment la conception, la disposition et les variations d'affichage pour différents appareils et tailles d'écran.
  - Le World Wide Web Consortium (W3C) a créé CSS pour supprimer la mise en forme du style de la page HTML.
  - CSS économise beaucoup de travail! Les définitions de style sont normalement enregistrées dans des fichiers .css externes. Avec une feuille de style externe, vous pouvez changer l'apparence d'un site Web entier en changeant un seul fichier!



## Syntaxe et sélecteurs CSS

Syntaxe CSS :



Exemple:

```
p {  
  color: red;  
  text-align: center;  
}
```

- Le sélecteur pointe sur l'élément HTML que vous souhaitez styler.
- Le bloc de déclaration contient une ou plusieurs déclarations séparées par des points-virgules.
- Chaque déclaration inclut un nom de propriété CSS et une valeur, séparés par deux points.
- Une déclaration CSS se termine toujours par un point-virgule et les blocs de déclaration sont entourés d'accolades.



## Sélecteurs CSS(1/ 4):

Les sélecteurs CSS permettent de "rechercher" (ou de sélectionner) des éléments HTML en fonction de leur nom, id, classe, attribut, etc. :

### ❑ Le sélecteur d'élément:

- Le sélecteur d'élément sélectionne les éléments en fonction du nom de l'élément.
- Vous pouvez sélectionner tous les éléments <p> sur une page comme ceci (dans ce cas, tous les éléments <p> seront alignés au centre, avec une couleur de texte rouge):

```
p {  
text-align: center;  
color: red;  
/* ceci est un commentaire */  
}
```

- Les commentaires sont utilisés pour expliquer le code et peuvent vous aider lorsque vous modifiez le code source à une date ultérieure. Un commentaire CSS commence par /\* et se termine par \*/.



## Sélecteurs CSS (2/4) :

### ❑ Le sélecteur id :

- Le sélecteur id utilise l'attribut id d'un élément HTML pour sélectionner un élément spécifique.

- id = L'identifiant d'un élément et il doit être unique dans une page, le sélecteur d'identifiant sert donc à sélectionner un élément unique!

Exemple: élément HTML avec un id `<p id="para1">mon text</p>`

- En CSS, pour sélectionner un élément avec un identifiant spécifique, écrivez un caractère dièse (#), suivi de l'identifiant de l'élément.

- La règle de style ci-dessous sera appliquée à l'élément HTML avec id = "para1":

```
#para1 {  
text-align: center;  
color: red;  
}
```



## Sélecteurs CSS (3/4) :

### Le sélecteur class

- Le sélecteur class sélectionne les éléments avec un attribut de classe spécifique.
- En CSS, pour sélectionner des éléments avec une classe spécifique, écrivez un caractère point (.), Suivi du nom de la classe.
- Dans l'exemple ci-dessous, tous les éléments HTML avec class = "center" seront en rouge et centrés:

```
.center {  
text-align: center;  
color: red;  
}
```

- Vous pouvez également spécifier que seuls des éléments HTML spécifiques doivent être affectés par une classe.
- Dans l'exemple ci-dessous, seuls les éléments <p> avec class = "center" seront alignés au centre:

```
p.center {  
text-align: center;  
color: red;  
}
```

- Les éléments HTML peuvent également faire référence à plus d'une classe.

```
<p class="center large">This paragraph refers to two classes.</p>
```



## Sélecteurs CSS (4/4) :

- Si vous avez des éléments avec les mêmes définitions de style, comme ceci:

```
h1 {  
  text-align: center;  
  color: red;  
}  
h2 {  
  text-align: center;  
  color: red;  
}  
p {  
  text-align: center;  
  color: red;  
}
```

- Il sera préférable de regrouper les sélecteurs pour minimiser le code.
- Pour regrouper les sélecteurs, séparez chaque sélecteur par une virgule.

Exemple: sélecteurs regroupés

```
h1, h2, p {  
  text-align: center;  
  color: red;  
}
```



# Comment utiliser le CSS

- Lorsqu'un navigateur lit une feuille de style, il formate le document HTML en fonction des informations contenues dans la feuille de style.

- Il y a trois façons d'insérer une feuille de style:

1. Style en ligne

2. Feuille de style interne

3. Feuille de style externe

- Style en ligne

Un style en ligne peut être utilisé pour appliquer un style unique à un seul élément.

Pour utiliser les styles en ligne, ajoutez l'attribut style à l'élément concerné. L'attribut style peut contenir n'importe quelle propriété CSS.

L'exemple ci-dessous montre comment changer la couleur et la marge gauche d'un élément <h1>:

```
<h1 style="color:blue;margin-left:30px;">This is a heading</h1>
```



# Comment utiliser le CSS

## ❑ Feuille de style interne

- Une feuille de style interne peut être utilisée si une seule page a un style unique.
- Les styles internes sont définis dans l'élément `<style>`, dans la section `<head>` d'une page HTML:

```
<html>
<head>
<style>
body {
background-color: linen;
}
h1 {
color: maroon;
margin-left: 40px;
}
</style>
</head>
<body>...</body>
</html>
```



# Comment utiliser le CSS

## ❑ Feuille de style externe

- Avec une feuille de style externe, vous pouvez changer l'apparence d'un site Web entier en ne modifiant qu'un seul fichier!
- Chaque page doit inclure une référence au fichier de feuille de style externe à l'intérieur de l'élément `<link>`. L'élément `<link>` va à l'intérieur de la section `<head>`:

```
<head>
```

```
<link rel="stylesheet" type="text/css" href="mystyle.css">
```

```
</head>
```

- Une feuille de style externe peut être écrite dans n'importe quel éditeur de texte. Le fichier ne doit contenir aucune balise HTML. Le fichier de feuille de style doit être enregistré avec une extension `.css`.
- Voici à quoi ressemble "mystyle.css":

```
body {  
background-color: lightblue;  
}  
h1 {  
color: navy;  
margin-left: 20px;  
}
```



# Couleurs CSS

## ☐ Noms de couleur

- En HTML, une couleur peut être spécifiée en utilisant un nom de couleur:



## ☐ Couleur de fond

- Vous pouvez définir la couleur d'arrière-plan des éléments HTML par la propriété background-color:

```
<h1 style="background-color:DodgerBlue;">Hello World</h1>  
<p style="background-color:Tomato;">Lorem ipsum...</p>
```





# Couleurs CSS

## ☑ Couleur du texte

- Vous pouvez définir la couleur du texte par la propriété color:

```
<h1 style="color:Tomato;">Hello World</h1>
```

```
<p style="color:DodgerBlue;">Lorem ipsum...</p>
```

```
<p style="color:MediumSeaGreen;">Ut wisi enim...</p>
```

Hello World

Lorem ipsum...

Ut wisi enim...

## ☑ Couleur de la bordure

- Vous pouvez définir la couleur des bordures par la propriété border:

```
<h1 style="border:2px solid Tomato;">Hello World</h1>
```

```
<h1 style="border:2px solid DodgerBlue;">Hello World</h1>
```

```
<h1 style="border:2px solid Violet;">Hello World</h1>
```

Hello World

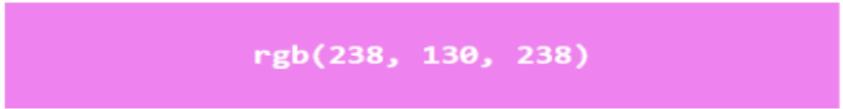
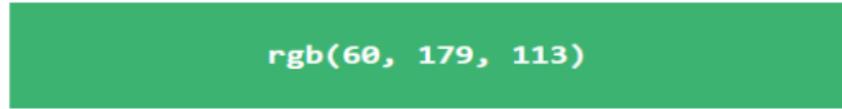
Hello World

Hello World



# Couleurs CSS

☐ Valeurs de couleur : \_\_\_\_\_



```
<h1 style="background-color:rgb(255, 99, 71);">...</h1>
```

```
<h1 style="background-color:#ff6347;">...</h1>
```

```
<h1 style="background-color:hsl(9, 100%, 64%);">...</h1>
```



# Arrière-plans CSS

Les propriétés background de CSS permettent de définir les effets d'arrière-plan des éléments.

Propriété	Description
<a href="#">background</a>	Définit toutes les propriétés d'arrière-plan dans une déclaration
<a href="#">background-attachment</a>	Définit si une image de fond est fixe ou défile avec le reste de la page
<a href="#">background-clip</a>	Spécifie la zone de peinture de l'arrière-plan
<a href="#">background-color</a>	Définit la couleur d'arrière-plan d'un élément
<a href="#">background-image</a>	Définit l'image d'arrière-plan d'un élément
<a href="#">background-position</a>	Définit la position de départ d'une image d'arrière-plan
<a href="#">background-repeat</a>	Définit comment une image d'arrière-plan sera répétée
<a href="#">background-size</a>	Spécifie la taille des images d'arrière-plan

## Exemple:

```
body {  
background-image: url("img_tree.png");  
background-repeat: no-repeat;  
background-position: right top;  
background-attachment: fixed;  
}
```

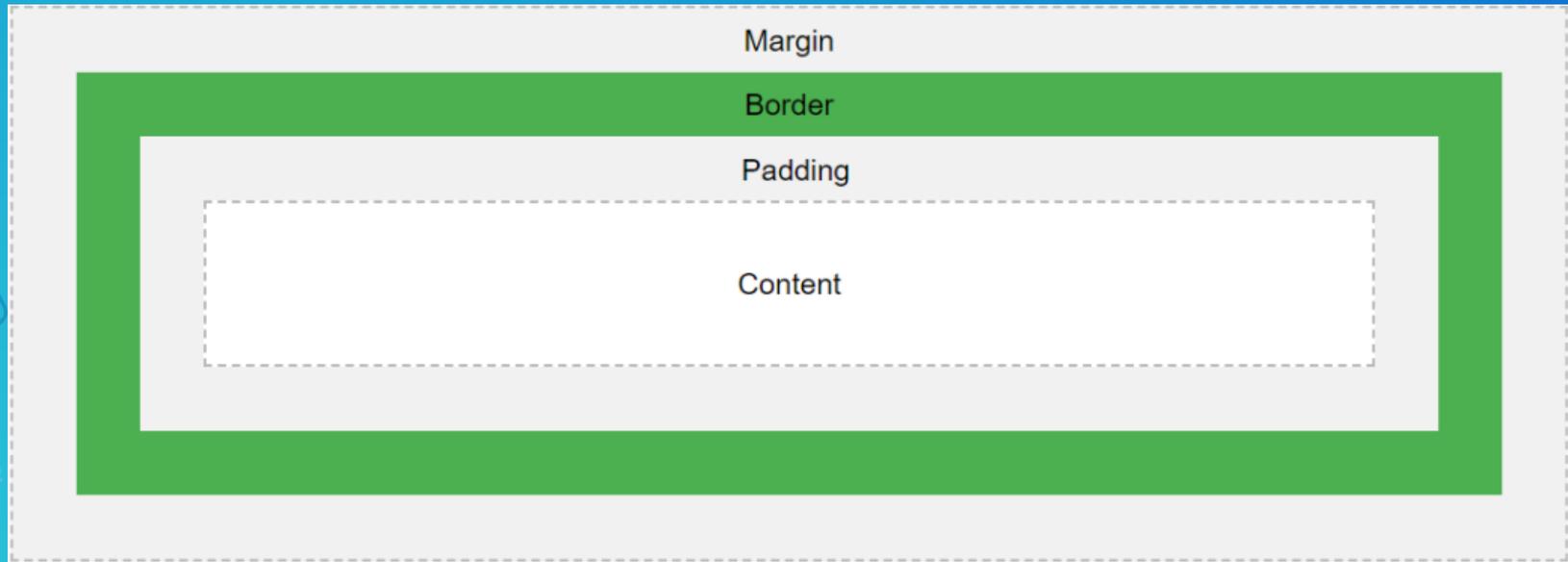
```
body {  
background: #ffffff url("img_tree.png") no-repeat  
right top;  
}
```



# Modèle de boîte CSS

Tous les éléments HTML peuvent être considérés comme des boîtes. En CSS, le terme "modèle de boîte" est utilisé quand on parle de design et de mise en page.

Le modèle de boîte CSS est essentiellement une boîte qui entoure chaque élément HTML. Il comprend: les marges, les bordures, le remplissage et le contenu réel. L'image ci-dessous illustre le modèle de boîte:





# Bordures (border) CSS

Les propriétés border de CSS vous permettent de spécifier le style, la largeur et la couleur de la bordure d'un élément

propreté	Description
<a href="#">border</a>	Définit toutes les propriétés de la bordure dans une déclaration
<a href="#">border-bottom</a>	Définit toutes les propriétés de la bordure inférieure dans une déclaration
<a href="#">border-bottom-color</a>	Définit la couleur de la bordure inférieure
<a href="#">border-bottom-style</a>	Définit le style de la bordure inférieure
<a href="#">border-bottom-width</a>	Définit la largeur de la bordure inférieure
<a href="#">border-color</a>	Définit la couleur des quatre bordures
<a href="#">border-radius</a>	Définit les quatre propriétés de bordure - * - pour les coins arrondis
<a href="#">border-style</a>	Définit le style des quatre bordures
<a href="#">border-width</a>	Définit la largeur des quatre bordures

A la place bottom on peut avoir aussi top, left, right pour les bordures supérieur, gauche et droite

## Exemples :

```
p {  
border-left: 6px solid red;  
background-color: lightgrey;  
}
```

```
p {  
border-top-style: dotted;  
border-right-style: solid;  
border-bottom-style: dotted;  
border-left-style: solid;  
}
```

```
p {  
border: 5px solid red;  
}
```



# Marges (margin) CSS

- Les propriétés margin CSS sont utilisées pour créer un espace autour des éléments, en dehors de toute bordure définie.
- Avec CSS, vous avez un contrôle total sur les marges. Il existe des propriétés permettant de définir la marge de chaque côté d'un élément (haut, droite, bas et gauche).
- CSS possède des propriétés permettant de spécifier la marge de chaque côté d'un élément:

margin-top

margin-right

margin-bottom

margin-left

- Toutes les propriétés de la marge peuvent avoir les valeurs suivantes:

auto - le navigateur calcule la marge

longueur - spécifie une marge en px, pt, cm, etc.

% - spécifie une marge en% de la largeur de l'élément conteneur

inherit - spécifie que la marge doit être héritée de l'élément parent

Exemple: 

```
p {  
margin-top: 100px;  
margin-bottom: 100px;  
margin-right: 150px;  
margin-left: 80px;  
}
```

**Exemple:** la marge gauche de l'élément p est 100px hérité de l'élément père (div)

```
<div>  
<p class="ex1"> Ce paragraphe a une marge gauche héritée (de l'élément div) </p>  
</div>
```

```
div {  
border: 1px solid red;  
margin-left: 100px;  
}  
p.ex1 {  
margin-left: inherit;  
}
```



# Remplissage (Padding) CSS

- Les propriétés padding de CSS permettent de créer un espace autour du contenu d'un élément, à l'intérieur de tout bordure.
- Avec CSS, vous avez un contrôle total sur le remplissage. Il existe des propriétés permettant de définir le remplissage pour chaque côté d'un élément (haut, droite, bas et gauche).

- CSS possède des propriétés permettant de spécifier le remplissage de chaque côté d'un élément:

padding-top  
padding-right  
padding-bottom  
padding-left

- Toutes les propriétés de remplissage peuvent avoir les valeurs suivantes:
  - longueur - spécifie un remplissage en px, pt, cm, etc.
  - % - spécifie un remplissage en% de la largeur de l'élément conteneur
  - inherit - spécifie que le remplissage doit être hérité de l'élément parent

Exemple 1: 

```
div {  
padding-top: 50px;  
padding-right: 30px;  
padding-bottom: 50px;  
padding-left: 80px;  
}
```

Exemple 2: 

```
div {  
padding: 25px 50px 75px 100px;  
/*top right bottom left*/  
}
```



# hauteur et largeur CSS

- Les propriétés `height` et `width` de CSS permettent de définir la hauteur et la largeur d'un élément.
- La hauteur et la largeur peuvent être définies sur `auto` (valeur par défaut. Indique que le navigateur calcule la hauteur et la largeur) ou spécifiées en valeurs de longueur, telles que `px`, `cm`, etc. ou en pourcentage (%) du bloc qui le contient.

Exemple:

```
div {  
  height: 200px;  
  width: 50%;  
  background-color: powderblue;  
}
```

Propriété	Description
<a href="#"><u>height</u></a>	Définit la hauteur d'un élément
<a href="#"><u>max-height</u></a>	Définit la hauteur maximale d'un élément
<a href="#"><u>max-width</u></a>	Définit la largeur maximale d'un élément
<a href="#"><u>min-height</u></a>	Définit la hauteur minimale d'un élément
<a href="#"><u>min-width</u></a>	Définit la largeur minimale d'un élément
<a href="#"><u>width</u></a>	Définit la largeur d'un élément



# Position CSS

- La propriété position spécifie le type de méthode de positionnement utilisé pour un élément (static, relative, fixed, absolute or sticky).

**static (par défaut):** Les éléments positionnés statiques ne sont pas affectés par les propriétés top, bottom, left et right.

**relative:** Si vous définissez les propriétés haut, droit, bas et gauche d'un élément relativement positionné, celui-ci sera ajusté pour s'éloigner de sa position normale. Les autres contenus ne seront pas ajustés pour s'insérer dans les espaces laissés par l'élément.

**fixed:** Un élément avec position: fixed; est positionné par rapport à la fenêtre d'affichage, ce qui signifie qu'il reste toujours au même endroit même si la page est défilée. Les propriétés top, right, bottom et left sont utilisées pour positionner l'élément.

**absolute:** Un élément avec position: absolute; est positionné par rapport à son père « positionné » le plus proche (au lieu de positionné par rapport à la fenêtre d'affichage, comme fixed). Toutefois; Si un élément absolu positionné n'a pas de père positionnés, il utilise le corps du document et se déplace avec le défilement de page.

**sticky:** Un élément avec position: sticky; est positionné en fonction de la position de défilement de l'utilisateur.



# Position CSS

```
div {  
  position: fixed;  
  bottom: 0;  
  right: 0;  
  width: 300px;  
  border: 3px solid #73AD21;  
}  
div.relative {  
  position: relative;  
  width: 400px;  
  height: 200px;  
  border: 3px solid #73AD21;  
}
```

```
div.absolute {  
  position: absolute;  
  top: 80px;  
  right: 0;  
  width: 200px;  
  height: 100px;  
  border: 3px solid #73AD21;  
}  
div.sticky {  
  position: -webkit-sticky; /* Safari */  
  position: sticky;  
  top: 0;  
  background-color: green;  
  border: 2px solid #4CAF50;  
}
```



# Débordement (overflow ) CSS

- La propriété overflow de CSS contrôle ce qui arrive au contenu trop volumineux pour le mettre dans une zone.

**visible** - Par défaut. Le débordement n'est pas coupé. Il rend en dehors de la boîte de l'élément

**hidden** - Le débordement est tronqué et le reste du contenu sera invisible.

**scroll** - Le débordement est tronqué, mais une barre de défilement est ajoutée pour afficher le reste du contenu.

**auto** - Si le débordement est tronqué, une barre de défilement doit être ajoutée pour afficher le reste du contenu.

```
div {  
width: 200px;  
height: 50px;  
background-color: #eee;  
overflow: visible;  
}
```

```
div {  
overflow: scroll;  
}
```

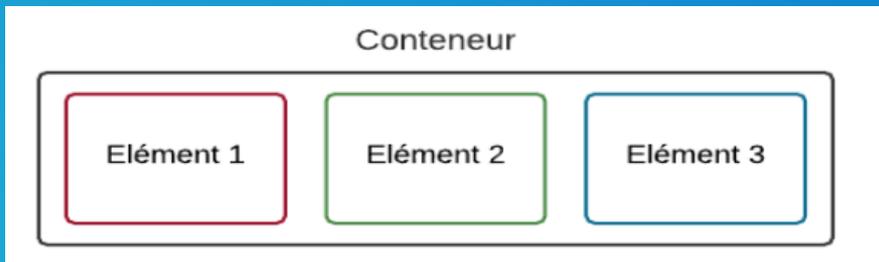
```
div {  
overflow: hidden;  
}
```

```
div {  
overflow-x: hidden; /* Hide horizontal scrollbar */  
overflow-y: scroll; /* Add vertical scrollbar */  
}
```



# Flexbox CSS

- Le principe de la mise en page avec Flexbox est simple : vous définissez un conteneur, et à l'intérieur vous placez plusieurs éléments.



- Le conteneur est une balise HTML, et les éléments sont d'autres balises HTML à l'intérieur

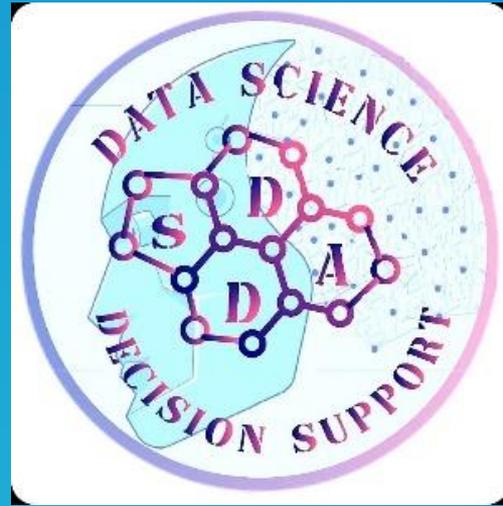
## HTML

```
<div id="conteneur">  
<div class="element">Elément 1</div>  
<div class="element">Elément 2</div>  
<div class="element">Elément 3</div>  
</div>
```

## CSS

```
#conteneur  
{  
  display: flex;  
}
```

# THANK YOU



<https://sdadclub.tech/>