

# HTML & CSS

# Table des matières

- Introduction
- Les éléments HTML
  - Les types de balise
- Le langage CSS
  - Les propriétés
  - Les pseudo-Classes
- Les formulaires
- Les tableaux
- Le responsive design

# INTRODUCTION

# Deux langages pour le web

## HTML

### "HyperText Markup Language"

- Créé en 1991 par "Tim Berners".
- Son rôle est de gérer et d'organiser le contenu du site.

## CSS

### "Cascading Style Sheets"

- Créé en 1996 par "Hakon Wium Lie", "Bert Bos" et la W3C.
- Son rôle est de gérer l'apparence de la page web.

# Les versions de HTML

HTML 1	HTML 2	HTML3	HTML 4	HTML 5
<i>Créé en 1991</i>	<i>1994</i>	<i>1996</i>	<i>1998</i>	<i>2014</i>
<ul style="list-style-type: none"><li>✓ Titre du document</li><li>✓ Liens</li><li>✓ Titre</li><li>✓ Sous-titres</li><li>✓ Listes</li><li>✓ Texte brut</li></ul>	<ul style="list-style-type: none"><li>✓ Tables</li><li>✓ Figures</li><li>✓ Expressions mathématiques</li></ul>	<ul style="list-style-type: none"><li>✓ Tableaux</li><li>✓ Applets (JAVA)</li><li>✓ Scripts</li><li>✓ Positionnement du texte autour d'une image</li></ul>	<ul style="list-style-type: none"><li>✓ Frame</li><li>✓ Tableaux complexes</li><li>✓ Lien avec le CSS</li></ul>	<ul style="list-style-type: none"><li>✓ Vidéo</li><li>✓ Nouvelles fonctionnalités pour les formulaires</li></ul>

# Les versions de CSS

<b>CSS 1</b>	<b>CSS 2</b>	<b>CSS 3</b>
<i>1996</i>	<i>1999</i>	<i>1999</i>
✓ Couleurs ✓ Marges ✓ Polices de caractères	✓ Positionnement précis	✓ Bordures arrondies ✓ Dégradés ✓ Ombres

# Les navigateurs

Ils interprètent le code Html/Css et affichent les sites Web.

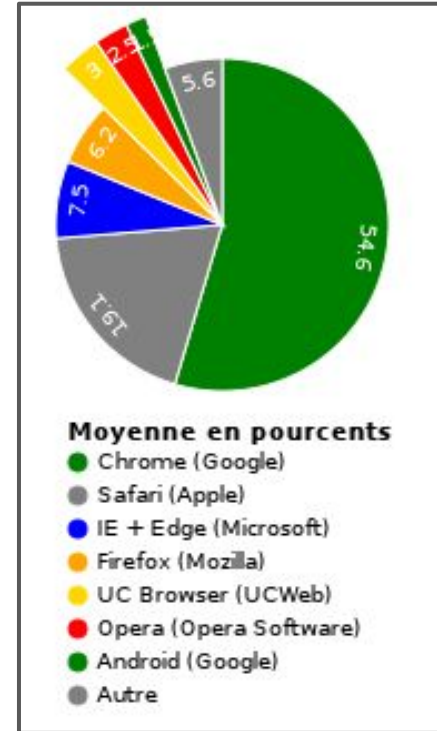
- Google Chrome
- Microsoft Edge
- Internet Explorer
- Safari
- Mozilla Firefox
- Opera
- Vivaldi
- Brave

# Les navigateurs

Chaque navigateur n'offre pas le même rendu à l'écran. Il est important de tester son site avec plusieurs navigateurs!

De plus, ils n'implémentent pas toujours les dernières fonctionnalités.

→ <https://caniuse.com>





# Écrire du code HTML & CSS

Pour créer un site web, un simple éditeur de texte suffit.

- Sublime Text
- Visual Studio Code
- Brackets
- Notepad++

# Les éléments HTML

## HTML

# Les balises

Le langage HTML est composé de balises qui se situent entre < >.

Il y a deux types de balises :

- Orphelines
- En paires

# Les balises orphelines

Cet élément est composé uniquement d'une seule balise.

Elle peut contenir des informations grâce aux attributs.

Exemple : pour afficher des images on utilise la balise "img" qui a un attribut "src" pour définir la source de l'image.

```

```

# Les balises en paires

Cet élément est composé d'une balise ouvrante et d'une balise fermante (avec un '/'). Elles peuvent contenir des données ainsi que des attributs.

Exemple : Pour définir le titre d'une page, on utilise la balise "title"

```
<title>Mon super titre!</title>
```

# Le code de base

Pour créer une page HTML vierge, le code minimal est :

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

  </body>
</html>
```

# Doctype? C'est quoi?

L'élément "Doctype" n'est pas une balise HTML, d'où la présence du « ! ».

Il indique au navigateur que notre code est du langage HTML.

Remarque: Avant le HTML5, il fallait écrire ceci...

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

# L'encodage

Pour afficher des caractères spéciaux correctement, nous pouvons ajouter la balise "meta" au head pour définir l'encodage à utiliser.

```
<meta charset="utf-8" />
```

Sans cette balise, le navigateur utilisera son encodage par défaut.



# Les commentaires

Une balise existe pour nous permettre d'ajouter des commentaires.

Celle-ci commence par "<!--" et termine par "-->".

```
<!-- Ceci est un commentaire -->
```

Attention, les commentaires seront visibles dans le code HTML reçu par le navigateur!

# Les balises de texte

## HTML

# Organiser le texte

- Les paragraphes

On délimite chaque paragraphe avec la balise `<p></p>`.

- `<p>Bonjour et bienvenue sur mon site !</p>`

  
Les sauts de ligne

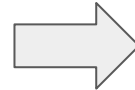
Pour ajouter un saut de ligne, on utilise la balise orpheline `<br>`

# Organiser le texte

- Les titres

On utilise la balise <h...> pour ajouter des titres au contenu du site. Il existe 6 niveaux de titre prédéfinis.

```
<h1>Titre très important</h1>  
<h2>Titre important</h2>  
<h3>Sous-titre</h3>  
<h4>Titre moins important</h4>  
<h5>Titre peu important</h5>  
<h6>Titre pas important</h6>
```



**Titre très important**  
**Titre important**  
Sous-titre  
Titre moins important  
Titre peu important  
Titre pas important

Le nombre et leur placement n'a pas d'importance particulière pour la validation w3c, retenez simplement qu'une page n'a qu'un vrai grand titre h1

# La mise en valeur

- Mettre en important : `<strong> </strong>`
- Mettre un peu en valeur : `<em> </em>`
- Marquer le texte : `<mark> </mark>`

**Attention, le navigateur appliquera un style au texte [strong → gras / em → italique / mark → surligné], mais ce n'est pas le rôle de ces balises!**

Elles permettent d'analyser votre code et d'influencer le référencement de votre site.

# Les balises de liste

## HTML

# Les listes

Il existe trois types de listes :

- Liste non ordonnée ou liste à puces
- Liste ordonnée ou liste numérotée ou énumération
- Liste de définitions

## Liste non ordonnée

On utilise la balise `<ul></ul>` pour définir la liste et la balise `<li></li>` pour chaque élément de la liste.

```
<ul>  
  <li>HTML</li>  
  <li>CSS</li>  
</ul>
```



## Liste ordonnée

On utilise la balise `<ol></ol>` pour définir la liste et la balise `<li></li>` pour chaque élément de la liste.

```
<ol>  
  <li>HTML</li>  
  <li>CSS</li>  
</ol>
```

# Liste de définitions

On utilise la balise `<dl></dl>` pour définir la liste. On utilise ensuite `<dt></dt>` pour les termes et `<dd></dd>` pour les définitions.

```
<dl>  
  <dt>Le terme 1</dt>  
  <dd>La définition 1</dd>  
  <dt>Le terme 2</dt>  
  <dd>La définition 2</dd>  
  <dt>Le terme 3</dt>  
  <dd>La définition 3</dd>  
</dl>
```

# Les balises de lien

## HTML

# Les liens

Pour créer un lien, on utilise la balise `<a></a>`. Il est nécessaire de lui ajouter un attribut "href" qui contient le lien vers la page voulue.

```
<a href="http://www.google.be">Mon lien</a>
```

Attention, pour un lien vers un autre site,  
il faut toujours commencer par « http:// »

# Navigation dans le site

Pour accéder à une autre page du site, il faut donner à l'attribut "href" le chemin vers le fichier désiré.

```
<a href="page.html">Lien vers une page</a>
```

- Cas d'une page dans un sous dossier

- ```
<a href="dossier/page.html">Page dans un dossier enfant</a>
```

```
<a href="../page.html">Page dans un dossier parent</a>
```

# Lien vers une ancre

Les ancres permettent d'accéder à une partie précise du site.  
Pour cela, vous devez :

- 1) Placer un attribut "id" dans la balise à atteindre

- 2) 

```
<h2 id="mon_ancre">Titre</h2>
```

  
Créer un lien vers le nom de l'ancre, précédé du symbole #

```
<a href="#mon_ancre">Aller vers l'ancre</a>
```

# Les liens

L'attribut "title" permet d'ajouter une infobulle au lien

```
<a href="cv.html" title="Lien vers mon cv">Mon CV</a>
```

Ajouter "mailto:..." à l'attribut "href" pour un envoi d'un e-mail

Po<a href="mailto:mon@adresse.be">Créer un e-mail</a>

```
<a href="monCV.doc">Télécharger mon CV</a>
```

# Les liens

Pour ouvrir un lien dans une nouvelle fenêtre, il faut ajouter l'attribut "target" avec la valeur "\_blank".

```
<a href="page.html" target="_blank">Lien vers ma page</a>
```

Pour éviter des failles de sécurité lors de l'utilisation de "target" pour les liens vers des sites externes, il est conseillé d'ajouter l'attribut "rel" avec la valeur "noopener".



# Les balises d'images

## HTML

# Les images

Pour insérer une image, on utilise la balise `<img>`.

Celle-ci doit comporter 2 attributs:

- `src` : La source de l'image, indiquez le chemin jusqu'à l'image.
- `alt` : Le texte alternatif, celui-ci permet d'afficher un texte si l'image ne peut pas être chargée.

Il est important de définir l'attribut "alt" car il est utilisé par les personnes malvoyantes et les robots des moteurs de recherche.

# Les images

Avant d'utiliser vos images, retirez les espaces de votre nom de fichier ou remplacez-les par "\_" ainsi que tous les accents.

```

```

L'ajout d'attribut "title" permet d'ajouter une infobulle à l'image

```

```

# Les figures

Lorsqu'une image a comme but d'illustrer un texte, on utilise la balise `<figure></figure>`.  
On place la légende entre les balises `<figcaption></figcaption>`.

```
<figure>
  
  <figcaption>Logo de Yahoo</figcaption>
</figure>
```

# Les balises universelles

## HTML

# Les balises universelles

## div

- Balise de type "block"
- Retour à la ligne
- Fortement utilisée dans la construction de design

## span

- Balise de type "inline"
- Pas de retour de chariot
- Les fonctions de dimensions CSS ne sont pas disponibles

# Les balises structurantes

## HTML

# Les balises structurantes

- En-tête

On utilise la balise « header » pour ajouter l'en-tête.

Celle-ci permet de placer :

- Un logo
- Une bannière
- Le slogan de votre site

Remarque : Il peut y avoir plusieurs balises « header » sur une page si celle-ci contient plusieurs sections



# Les balises structurantes

- Pied de page

On utilise la balise « footer » pour ajouter un pied de page.

Celle-ci permet de placer :

- Des liens de contact
- Le nom de l'auteur
- Les mentions légales
- Etc...

# Les balises structurantes

- Liens de navigation

On utilise la balise « nav » pour ajouter un menu.

Généralement, le menu regroupe les principaux liens du site sous forme d'une liste à puces à l'intérieur de la balise <nav> :

```
<nav>
  <ul>
    <li><a href="index.html">Accueil</a></li>
    <li><a href="forum.html">Forum</a></li>
    <li><a href="contact.html">Contact</a></li>
  </ul>
</nav>
```

# Les balises structurantes

- Section de page

On utilise la balise « section » pour regrouper des contenus en fonction de leur thématique.

Elle englobe généralement une portion du contenu au centre de la page.

### Actualités

A la une | Sport | Économie | People | Insolite | **Rugby 2011** | Science | Musique




- » Rugby: les All Blacks euphoriques, un point c'est tout
- » L'UE peine à définir sa réponse à la crise
- » Ski: Ligety triomphe à Sölden, juste devant Pinturault
- » Trois humanitaires européens enlevés en Algérie
- » Jean d'O, marque déposée !
- » La France ne renonce pas sur le FESF
- » La canonisation de trois religieux perturbée par un incident

- » Rugby: les All Blacks champions du monde pour un point
- » La police démantèle un gang de braqueurs "hors norme"
- » Séisme de 6,6 dans l'est de la Turquie

France | Monde | Insolite | Toute l'actu people du jour

[Toute l'actualité](#)

### FHV



La vidéo à la demande en illimité à partir de **7,99€/mois**



Nouveautés de la semaine

- 1. Beatdown
- 2. De famour à la foie
- 3. Patch adams
- 4. Les alkunés du golf
- 5. Un Amour de Coccinelle
- 6. La Belle et la Bête
- 7. Kiss Kiss, Bang Bang
- 8. Bobby Jones : naissance d'une

### Assistance



Installer votre Freebox Révolution

Pour une installation optimum, suivez notre tutonet

**<section>**

- Réseau : Freebox bloque à étape X
- Internet : Configurer le WiFi
- Télévision : Prise en main du Freebox Player
- Téléphone : Brancher son téléphone Freebox

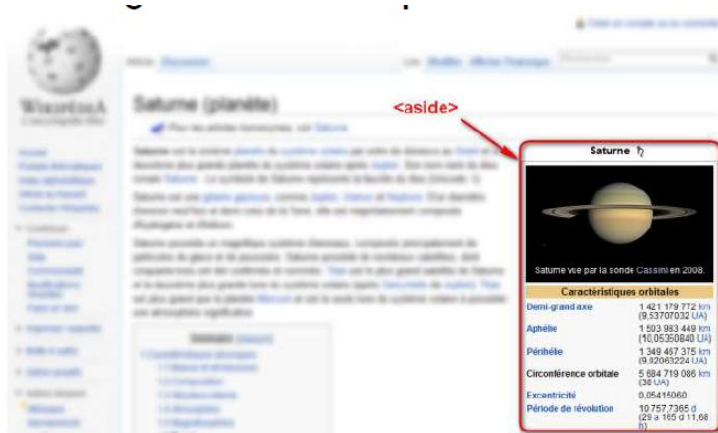
[Visitez le site de l'assistance](#)

Rchercher sur le site l'assistance de Free :

# Les balises structurantes

- Informations complémentaires

On utilise la balise « aside » pour ajouter des informations complémentaires au document que l'on visualise.



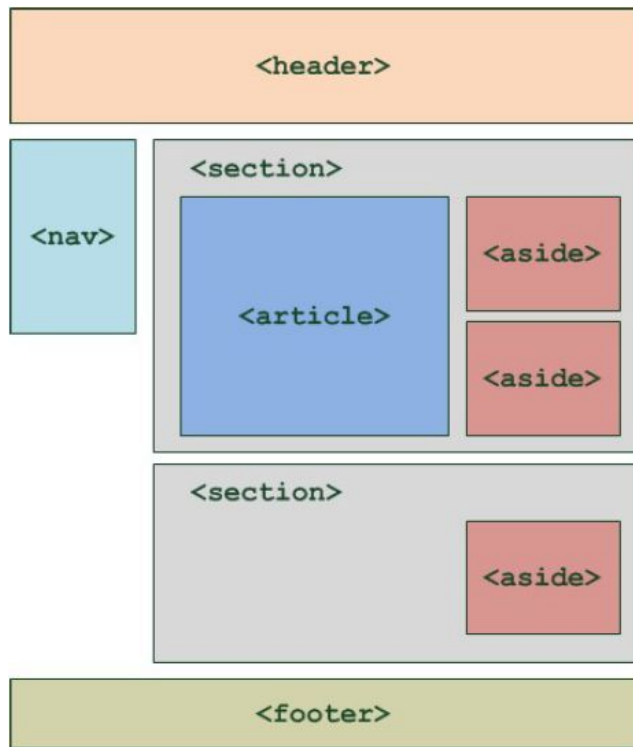
# Les balises structurantes

- Article indépendant

On utilise la balise « article » pour englober une portion généralement autonome de la page.

```
<article>
  <h1>Mon article</h1>
  <p>Lorem ipsum dolor sit amet,
    consectetur adipisicing elit. </p>
</article>
```

# Résumé



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Bobby - Le Site Web</title>
  </head>

  <body>
    <header>
      <h1>Bobby</h1>
      <h2>Carnets de voyage</h2>
    </header>

    <nav>
      <ul>
        <li><a href="#">Accueil</a></li>
        <li><a href="#">Blog</a></li>
        <li><a href="#">CV</a></li>
      </ul>
    </nav>

    <section>
      <aside>
        <h1>À propos de l'auteur</h1>
        <p>C'est moi, Bobby ! Je suis né un 23 novembre 1967.</p>
      </aside>
      <article>
        <h1>Je suis un grand voyageur</h1>
        <p>Bla bla bla bla (texte de l'article)</p>
      </article>
    </section>

    <footer>
      <p>Copyright Bobby - Tous droits réservés<br />
      <a href="#">Me contacter !</a></p>
    </footer>
  </body>
</html>
```

# Le langage CSS

CSS

# Le CSS

Le CSS est le langage qui va nous permettre de mettre en page nos sites Web.

Pour respecter la logique de séparation contenu et mise en forme, le code CSS est dans un fichier dédié avec l'extension ".css".

Il est également possible d'ajouter du CSS dans une balise "style" dans le "head" du code HTML.

Le style peut être défini directement dans les balises HTML (déconseillé).



# Le CSS

Pour lier le HTML et CSS, il faut ajouter cette balise dans le "head"

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

Le langage CSS est constitué de sélecteurs ainsi que de propriétés

Selecteur

{

propriete : valeur ;

}

# Les sélecteurs de base

- Le nom de la balises HTML

```
p { /* ... */ }
```

- La valeur de l'attribut "class" précédé d'un point

- ```
.class { /* ... */ }
```

 Le symbole #

- ```
#id { /* ... */ }
```

 Le symbole \*

# L'attribut class et id

- Class

Permet de cibler plusieurs éléments du code HTML qui doivent posséder les mêmes propriétés CSS

- Id

Permet d'identifier de manière unique un élément HTML

# Les sélecteurs avancés

- Le sélecteur descendant

`a b { /* ... */ }`

Permet de cibler toutes les balises "b" se trouvant dans une balise "a"

- Le sélecteur enfant

Même `a > b { /* ... */ }` les balises se trouvant directement dans une balise "a"

# Les sélecteurs avancés

- Le sélecteur de voisin direct

`a + b { /* ... */ }`

Permet de ne sélectionner que le premier élément "b" qui est immédiatement précédé par l'élément "a".

- Le sélecteur de voisins généraux

Même `a ~ b { /* ... */ }` au premier élément "b"

# Les sélecteurs avancés

- Sélecteur d'attribut

```
balise[attribut] { /*...*/ }
```

Permet de ne cibler que les balises qui ont un attribut.

- Sélecteur d'attribut et valeur

Perm 

```
balise[attribut="valeur"] { /*...*/ }
```

Les options possibles du sélecteur :

[https://developer.mozilla.org/fr/docs/Web/CSS/Sélecteurs\\_d\\_attribut](https://developer.mozilla.org/fr/docs/Web/CSS/Sélecteurs_d_attribut)

# Combiner des sélecteurs

Pour définir le même comportement à plusieurs sélecteurs, il suffit de créer une règle CSS avec les sélecteurs séparés d'une virgule.

```
h1 + p, h2 + p, h3 {  
    color: orangered;  
}
```

# Les propriétés du texte

## CSS



# Formatage du texte

- La taille

On utilise la propriété « font-size » suivie de sa valeur :

- Soit la taille absolue (px)
- Soit la taille relative (% , em, rem)

Différences entre les unités « em » et « rem » :

- “em” est relatif à la font-size de l'élément ou du parent le plus proche.
- “rem” se base toujours sur la valeur de la font-size de la balise « html ».

# Formatage du texte

- Mettre en gras

On utilise la propriété « font-weight » suivie de sa valeur

```
font-weight : bold;
```

- Mettre en italique

On utilise la propriété « font-style » suivie de sa valeur

```
font-style : italic;
```

# Formatage du texte

- L'alignement horizontal

On utilise la propriété « text-align » suivie d'une des valeurs suivantes :

- center
- left
- right
- justify

# Formatage du texte

- Modifier la casse du texte

On utilise la propriété « text-transform » suivie des valeurs suivantes :

- capitalize : la première lettre de chaque mot sera en capitale.
- uppercase : tous les caractères seront écrits en capitale.
- lowercase : tous les caractères seront écrits en minuscule.
- none : empêche les modifications liées à la casse.

# Formatage du texte

- Autres mises en valeur du texte

On utilise la propriété « text-decoration » suivie des valeurs suivantes :

- underline : souligné
- line-through : texte barré
- overline : ligne au-dessus
- none : aucune décoration

# Formatage du texte

- Comportement lors d'un dépassement

On utilise la propriété « overflow » pour le définir, suivie de la valeur :

- visible : Le contenu peut être affiché en dehors de l'élément
- hidden : Le contenu est rogné si besoin
- scroll : Une scrollbar verticale et horizontale sont ajoutées
- auto

# Formatage du texte

- Autoriser la césure des mots pour éviter les dépassements

On utilise la propriété « overflow-wrap », suivie de la valeur :

- normal : Utilisation des césures normales.
- break-word : Les mots peuvent être scindés.

*A l'origine, cette propriété était une extension non-standard de "Microsoft" nommée « word-wrap ». Implémentée dans la plupart des navigateurs, elle a été renommé et garde "word-wrap" en alias.*

# Formatage du texte

- Dépassement dans le cas d'un texte monoline

On peut combiner la propriété « overflow » avec les propriétés « white-space » et « text-overflow » pour obtenir un texte monoline avec des points de suspension en cas de dépassement.

```
.monoline {  
    overflow: hidden;  
    white-space: nowrap;  
    text-overflow: ellipsis;  
}
```



# La police d'écriture

Pour modifier la police, on utilise la propriété « font-family »

```
font-family : Arial;
```

Il est possible de lister plusieurs polices, dans l'ordre de souhaits.

```
font-family : "Arial Black", Arial, Verdana;
```

# La police d'écriture

Il est possible d'ajouter des polices à votre site

- Soit en utilisant des services extérieurs (Ex : [fonts.google.com/](https://fonts.google.com/))
- Soit en ajoutant la police dans les fichiers du site et en ajoutant une règle "@font-face" au fichier CSS

```
@font-face {  
  font-family: "Open Sans";  
  src: url("/fonts/OpenSans-Regular-webfont.woff2") format("woff2"),  
        url("/fonts/OpenSans-Regular-webfont.woff") format("woff");  
}
```

# Couleurs et arrières-plan

## CSS

# La couleur du texte

- Couleur prédéfinie

On utilise la propriété « color » suivie d'une valeur prédéfinie.

Il existe 140 couleurs définies en CSS3

```
color: orangered;
```

On peut utiliser les notations `rgb()` ou `hsl()` pour définir une couleur.

```
color: rgb(255, 69, 0);
```

# La couleur du texte

- Couleur personnalisée (Hexadécimale)

Il est possible de définir la couleur en une valeur hexadécimale sur 6 chiffres [#RRGGBB] ou sur 3 chiffres [#RGB] lorsque tous les couples de valeurs sont identiques.

```
/* goldenrod */  
color: #daa520;  
  
/* orangered */  
color: #ff4400; /* 6 chiffres */  
color: #f40;    /* 3 chiffres */
```

# Arrière plan

- Couleur de fond

On utilise la propriété « background-color » suivie de sa valeur

```
background-color : #428614;
```

On utilise la propriété « background-image » suivie d'un chemin url

```
background-image : url("images/monImage.png");
```

# Options de l'arrière plan

- Fixer l'image

On utilise la propriété « background-attachment » suivie de "fixed"

```
background-attachment : fixed ;
```

On utilise la propriété « background-repeat » suivie d'une des valeurs suivantes :  
no-repeat, repeat-x, repeat-y, repeat

# Options de l'arrière plan

- Position du fond

On utilise la propriété « background-position » suivie de sa position verticale et horizontale données en "pixel" ou par mot clef.

```
background-position : 10px 50px;  
background-position: top right;
```

- Horizontale : top, center, bottom
- Verticale : left, center, right



# Écriture raccourcie

Il est possible d'utiliser une écriture raccourcie pour définir l'arrière-plan en utilisant la propriété "background" suivie des valeurs : Source, Position horizontal, Position vertical, Répétition.

Cette écriture permet également de combiner plusieurs arrière-plans. Pour cela, il faut séparer d'une virgule chaque groupe de valeur.

```
div {  
    background: url("first.jpg") left top no-repeat,  
               url("second.jpg") right bottom no-repeat;  
}
```

# Transparence

Pour ajouter de la transparence, il faut modifier l'opacité des éléments, pour cela on utilise la propriété « opacity » suivie d'une valeur de 0 à 1.

```
opacity: 0.25;
```

# Couleur et Transparence

Lors de la définition d'une couleur, il est possible d'ajouter la transparence à l'aide du canal Alpha.

Pour cela, il faut utiliser les notations : `rgba()` ou `hsla()`.

```
/*orangered avec opacité de 60% */  
color: rgba(255, 69, 0, 0.6);  
color: hsla(16, 100%, 50%, 0.6);
```

# Couleur et Transparence

Il est également possible (CSS4) d'ajouter le canal Alpha à la valeur des couleurs, en définissant les valeurs hexadécimales sur 8 chiffres [#RRGGBBAA] ou 4 chiffres [#RGBA].

```
/* goldenrod avec opacité de 30% */  
color: #daa5204d;  
  
/* orangered avec opacité de 60% */  
color: #ff440099;  
color: #f409;    /*Ecriture courte*/
```

# Modifier l'apparence

## CSS

# La taille des éléments

Il est possible de modifier la hauteur et la largeur des éléments, pour cela on utilise respectivement les propriétés « height » ou « width » suivies d'une valeur en pixel (px) ou d'une valeur relative (% , em, rem).

```
div {  
    height: 250px;  
    width: 75%;  
}
```

Il est également possible de fixer la taille minimum ou maximum avec les préfixes « min-... » ou « max-... » suivis d'une valeur.

# Alignement du contenu

Pour les éléments "inline" et les cellules de tableaux, on peut utiliser la propriété « vertical-align » suivie d'une des valeurs suivantes : top, middle, bottom.

```
vertical-align : bottom;
```

Attention, cette propriété ne peut pas être utilisée pour aligner verticalement les éléments dont le type de rendu est "block".

# Le rendu

Il est possible de modifier le type de rendu des éléments HTML, pour cela on utilise la propriété « display » suivie de la valeur.

- “block” : apparaît sous l'élément précédent et au-dessus du suivant.
- “inline” : se place sur la même ligne dans le rendu.
- “none” : n'est pas présent dans le rendu.



# Les flottants

Pour rendre les éléments flottants, on utilise la propriété « float » suivie de la valeur :  
"left" ou "right"

```
float : left;  
float : right;
```

Pour arrêter cet habillage, on utilise la propriété « clear »

```
clear : both;
```

# Les marges

Pour des marges autour de nos éléments, on utilise la propriété « margin-... » suivie d'une valeur en px, em, %

```
margin-bottom: 10px;  
margin-left: 10px;  
margin-right: 10px;  
margin-top: 10px;
```

# Les paddings

Pour des marges internes à nos éléments, on utilise la propriété « padding-... » suivie d'une valeur en px, em, %

```
padding-bottom: 10px;  
padding-left: 10px;  
padding-right: 10px;  
padding-top: 10px;
```

# Écriture raccourcie

Il est possible d'utiliser une écriture raccourcie pour définir les marges et les paddings en donnant d'une à 4 valeurs

```
margin: 10px 10px;  
padding: 10px 10px 10px 10px;
```

- 2 valeurs : top & bottom → right & left.
- 4 valeurs : top → right → bottom → left.

# Les bordures

Pour mettre en forme une bordure, on utilise la propriété « border » qui contient 3 paramètres : la largeur, le style et la couleur.

Les valeurs de style possible sont :

“none, hidden, dotted, dashed, solid, double, ridge, inset, outset”.

Cette propriété est `border : 2px solid gold;` width, border-style, border-color”.

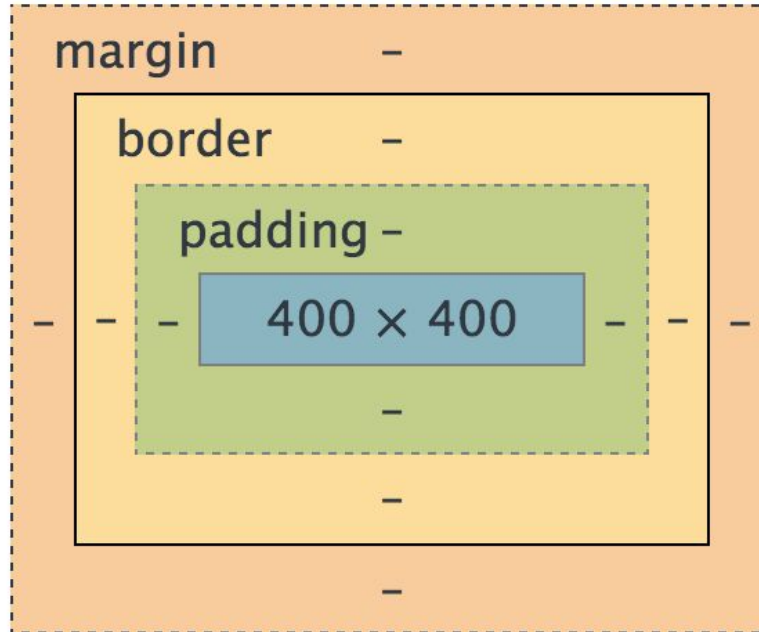
# Les bordures

Pour mettre en forme uniquement certaines bordures, on utilise les propriétés suivantes :

- border-top
- border-bottom
- border-left
- border-right

Ces propriétés sont également un raccourci qui reprend les propriétés :  
"border-X-width, border-X-style, border-X-color".

# Les bordures



# Les bordures arrondies

Pour obtenir des bordures arrondies, on ajoute la propriété « border-radius » à la bordure précédemment défini.

```
border-radius : 10px 20px 30px 40px;
```



# Calcul de la taille des éléments

Par défaut, la hauteur et la largeur des éléments est appliqué uniquement pour le contenu, sans prendre en compte le padding et la bordure.

Pour éviter cela, il est possible d'utiliser la propriété « box-sizing » avec les valeurs suivantes :

- content-box : Valeur par défaut. Calcule la taille sur base du contenu.
- border-box : Indique au navigateur de prendre en compte les valeur des bordures et du padding pour calculer la taille de l'élément.

# Élément redimensionnable

Pour rendre un élément redimensionnable (ou non) par l'utilisateur, on utilise la propriété « `resize` » suivie de la valeur :

- `none`
- `both`
- `horizontal`
- `vertical`

Par exemple, cette propriété permet de “bloquer” la taille des éléments « `textarea` ».

# Les ombres

Pour appliquer une ombre à un élément, on utilise la propriété « box-shadow » avec 4 paramètres

```
box-shadow : 5px 5px 5px red;
```

- 1) Le décalage horizontal
- 2) Le décalage vertical
- 3) L'adoucissement du dégradé
- 4) La couleur

# Les ombres

Pour appliquer une ombre à un texte, on utilise la propriété « text-shadow » avec les 4 même paramètres

```
text-shadow : 5px 5px 5px red;
```

Il est possible de définir plusieurs ombres, pour cela il faut ajouter une virgule entre chaque groupe de valeur pour la propriété "X-shadow".

# Les pseudo-classes

## CSS

# Les pseudo-classes

Les pseudo-classes permettent de définir un comportement particulier pour les éléments lors d'état spécifique.

Celles-ci sont précédé de ":" et se placeront après le sélecteur.

```
Selecteur:PseudoClasse  
{  
    propriete : valeur;  
}
```

# Les événements

- :hover

Permet de réagir au survol de l'élément

- :focus

Permet de réagir pour les éléments sélectionné

- :active

Permet de réagir au clic de l'utilisateur

# L'état des liens

- :link

Permet de modifier les liens à l'intérieur d'éléments.

- :visited

Permet de modifier les liens visité par l'utilisateur.

Ordre idéal pour styliser des liens : link → visited → hover → active



# Préciser les éléments

- :first-child

Permet de cibler uniquement le premier enfant de l'élément parent.

- :last-child

Permet de cibler uniquement le dernier enfant de l'élément parent.

- :only-child

Permet de cibler un élément s'il est le seul enfant du parent.

# Préciser les éléments

- :nth-child(...)

Permet de cibler les éléments sur base d'un motif de répétition.

Notation fonctionnelle

**A****n**+**B** → 0 où "n" est un entier incrémenté commençant à zéro

Mot-clé

**even** : Représente les éléments paires (Équivalent à "**2n**")

**odd** : Représente les éléments impaires (Équivalent à "**2n+1**")

# Préciser les éléments

- `:empty`

Permet de cibler les éléments qui n'ont aucun contenu.

- `:target`

Permet d'obtenir l'élément cible de la page (via l'ancre dans l'url)

- `:not(...)`

Négation, prend en argument un ou plusieurs sélecteurs.

# Préciser les éléments

- :only-of-type

Permet de cibler les éléments qui sont seuls de leur type parmi leurs frères.

- :first-of-type

Permet de cibler le premier élément de tel type parmi ses frères.

- :last-of-type

Permet de cibler le dernier élément de tel type parmi ses frères.

# Les formulaires

## HTML & CSS

# Créer un formulaire

Pour créer un formulaire, on utilise la balise « form » et on place à l'intérieur de celle-ci tous les champs du formulaire.

Le formulaire a deux attributs :

- method : indique la manière dont les données seront envoyées
- action : indique l'adresse du serveur qui va traiter le formulaire

```
<form method="post" action="action.php">  
    <!-- ici les différents champs du formulaire -->  
</form>
```

# Méthode d'envoi des données

- `method="get"`
  - Limitée à 255 caractères
  - Les données envoyées sont visibles dans l'adresse
- `method="post"`
  - Pas de limite du nombre de caractères
  - Invisible dans le navigateur

# Champ de saisie

Pour insérer un champ de saisie, on utilise la balise orpheline « input ». Les saisies possibles et le comportement de l'élément dépend de la valeur indiquée dans son attribut "type".

```
<input id="identifiant" name="nom" type="...">
```

L'attribut « name » permet de définir le nom du contrôle, il sera envoyé avec les données lors de l'envoi du formulaire.



# Différent type de saisie

- `type="text"`

Champ de saisie de texte sur une seule ligne.

Attributs pris en charges : autocomplete, list, maxlength, minlength, pattern, placeholder, required, size

# Différent type de saisie

- `type="number"`

Champ qui permet de saisir un nombre décimal.

Attributs pris en charges : autocomplete, list, placeholder, readonly

## Différent type de saisie

- `type="email"`

Champ qui permet de saisir et d'éditer une adresse mail ou, si l'attribut multiple est indiqué, une liste d'adresses mail.

*La valeur saisie est automatiquement validée. Les pseudo-classes « :valid » et « :invalid » sont automatiquement appliquées.*

Attributs pris en charges : autocomplete, list, placeholder, readonly

# Différent type de saisie

- `type="password"`

Champ de saisie de texte sur une ligne et dans lequel chaque caractère est remplacé par un symbole

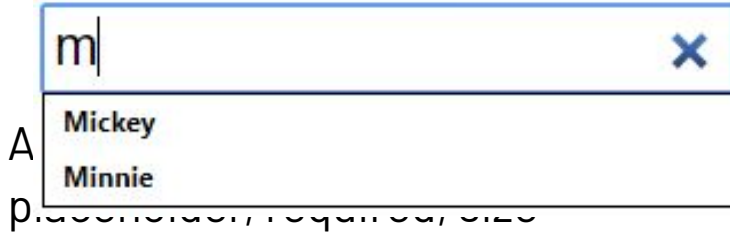


Attributs pris en charges : autocomplete, inputmode, maxlength, minlength, pattern, placeholder, readonly, required, size

# Différent type de saisie

- type="search"

Champ de saisie permettant à un utilisateur de saisir des termes de recherche. Celui-ci permet d'obtenir différent visuel en fonction du navigateurs.



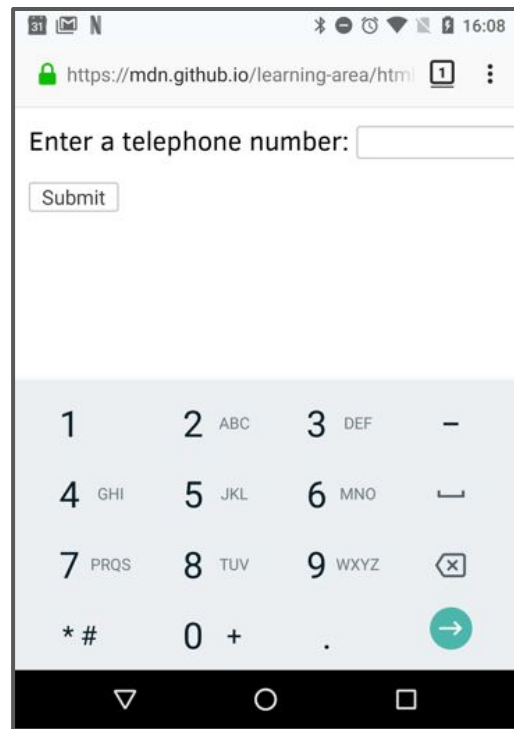
lete, list, maxlength, minlength, pattern,

# Différent type de saisie

- `type="tel"`

Champ qui permet de saisir un numéro de téléphone.

Attributs pris en charges : autocomplete, list, maxlength, minlength, pattern, placeholder, readonly et size



# Différent type de saisie

- `type="url"`

Champ qui permet de saisir et d'éditer une URL.

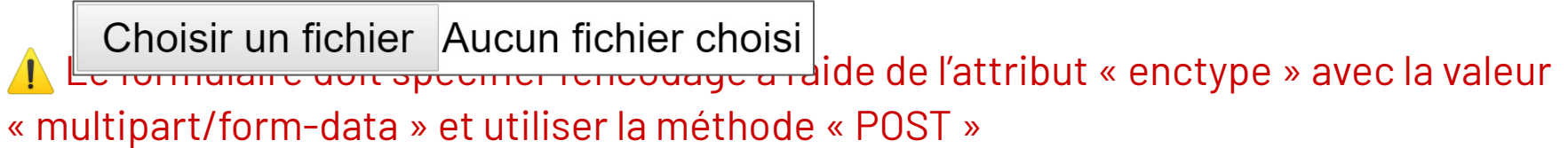
*La valeur saisie est automatiquement validée. Les pseudo-classes « :valid » et « :invalid » sont automatiquement appliquées.*

Attributs pris en charges : autocomplete, list, maxlength, minlength, pattern, placeholder, readonly et size

# Différent type de saisie

- type="file"

Champ qui permet à un utilisateur de sélectionner un ou plusieurs fichiers depuis leur appareil et de les uploader vers un serveur



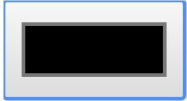
Attributs pris en charges : accept, multiple, required



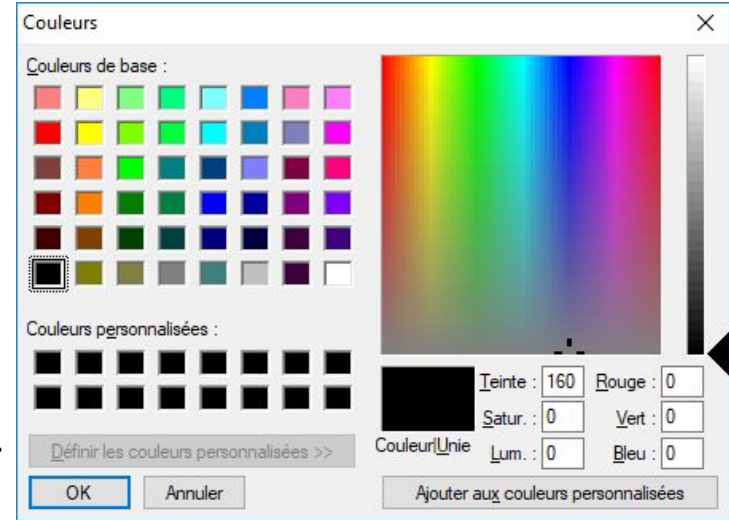
# Différent type de saisie

- `type="color"`

Champ qui permet de sélectionner une couleur avec une interface ou avec un code hexadécimal.



pris en charges : autocomplete et list.

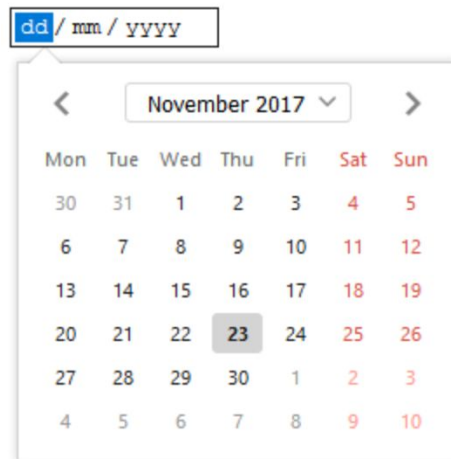


# Différent type de saisie

- type="date"

Champ qui permet de saisir une date.

Attributs pris en charges : autocomplete, list, readon



## Différent type de saisie

- `type="time"`

Champ qui permet de saisir une heure (avec des minutes et éventuellement des secondes).

Attributs pris en charges : autocomplete, list, readonly et step.

# Différent type de saisie

- type="datetime-local"

Champ qui permet de saisir une date et une heure.



*Remarque*



*Le type "datetime" est déprécié*

Attributs pris en charges : autocomplete, list, r

The image shows a web browser's date and time picker interface. At the top, a text box contains the value "24/04/2017, 10:30" with a close button (X) and a dropdown arrow. Below this, a calendar for "April 2017" is displayed. The calendar has navigation buttons (previous, current, next) and a table of dates. The date "24" is highlighted in blue.

Mon	Tue	Wed	Thu	Fri	Sat	Sun
27	28	29	30	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

# Champ de saisie multi-ligne

Pour insérer un champ qui permet d'éditer du texte sur plusieurs lignes. On utilise la balise « textarea »

```
<textarea name="textarea" rows="10" cols="50">  
...  
</textarea>
```

- cols : La largeur visible du contrôle de saisie, exprimée en largeur moyenne de caractères.

# Les labels

Pour afficher un texte qui donne une indication sur le champ, on utilise la balise « label ».

Pour lier les balises « label » et « input », il faut ajouter l'attribut « for » au label avec la valeur l'id de l'input.

```
<label for="monId">Nom : </label>  
<input type="text" name="nom" id="monId">
```

# Champ à choix - Checkbox

Pour insérer un champ dont le contenu est une case à cocher, on utilise des balises « input » avec le type "checkbox".

```
<input type="checkbox" name="css" id="cb_css">  
<label for="cb_css"> CSS</label>  
<br>  
<input type="checkbox" name="html" id="cb_html">  
<label for="cb_html"> HTML</label>
```

Pour

## Champ à choix - Bouton radio

Pour insérer un champ dont le contenu est un bouton à choisir, on utilise des balises « input » avec le type "radio" et portant la même valeur pour l'attribut "name".

```
<input type="radio" name="web" id="rb_css" value="css">  
<label for="rb_css"> CSS</label>  
<br>  
<input type="radio" name="web" id="rb_html" value="html">  
<label for="rb_html"> HTML</label>
```

Pour qu'un bouton soit pre-coché, ajoutez l'attribut « checked ».



## Champ à choix - Liste

Pour insérer une liste déroulante, on utilise la balise « select ». Chaque option sera déclarée dans une balise « option ».

```
<select name="web">
  <option value="css">CSS</option>
  <option value="html">HTML</option>
  <option value="csharp">C#</option>
  <option value="php">PHP</option>
</select>
```

Pour qu'un

## Champ à choix - Liste

Pour avoir une liste déroulante avec des sous-groupes, on utilise la balise « optgroup » avec en attribut "label" le nom du groupe.

```
<select name="web">
  <optgroup label="Front-end">
    <option value="css">CSS</option>
    <option value="html">HTML</option>
  </optgroup>
  <optgroup label="Back-end">
    <option value="csharp">C#</option>
    <option value="php">PHP</option>
  </optgroup>
</select>
```

# Regrouper des champs

Pour regrouper les champs, on utilise la balise « fieldset ». Chaque groupe est déclaré avec la balise « legend ».

```
<fieldset>
  <legend>Mes coordonnées</legend>
  <label for="nom">Nom : </label>
  <input type="text" name="nom" id="nom">
  <br>
  <label for="prenom">Prénom : </label>
  <input type="text" name="prenom" id="prenom">
</fieldset>
```

# Les boutons

La balise « input » permet également de créer les boutons de nos formulaire. Il y a 3 types de boutons différents :

- `button` : un bouton sans comportement par défaut.
- `submit` : un bouton qui envoie le formulaire.
- `reset` : un bouton qui permet de réinitialiser le formulaire.

## Quelques attributs utiles

- placeholder  
Permet de définir une indication destinée à l'utilisateur par rapport au champ.
- required  
Permet d'indiquer que le champs doit être renseignée avant que l'utilisateur puisse envoyer le formulaire.
- readonly  
Cet attribut indique que l'utilisateur ne peut pas modifier la valeur de l'élément.

## Quelques attributs utiles

- pattern  
Permet de définir une "RegExp" qui permettra de vérifier la valeur saisie.
- autocomplete  
Permet d'indiquer si la valeur saisie doit automatiquement être complétée par le navigateur.
- spellcheck  
Permet d'activer ou désactiver la vérification orthographique et grammaticale du contenu.

# Quelques attributs utiles

- autofocus  
Permet d'indiquer que l'élément doit recevoir le focus au chargement de la page.
- tabindex  
Permet de définir l'ordre de navigation au clavier (généralement avec la touche "tab"). Cet attribut peut prendre l'une des valeurs entières suivantes :
  - Positive : ordre relatif, parcourues dans l'ordre croissant.
  - 0 : ordre défini par les éléments du DOM
  - Négative : l'élément ne peut pas être atteint via la navigation au clavier

# Les pseudo-classes utiles

- `:required`  
Permet de cibler les éléments pour lequel l'attribut "required" est activé.
- `:optional`  
Permet de cibler les éléments pour lesquels l'attribut "required" n'est pas activé
- `:valid`  
Permet de cibler tous les éléments dont la validation s'effectue correctement
- `:invalid`  
Permet de cibler tout élément pour lequel la validation a échoué



# Les pseudo-classes utiles

- `:checked`  
Permet d'obtenir les boutons radio, les case à cocher ou les options d'une liste qui ont été coché ou activé
- `:enabled`  
Permet de cibler les éléments activé.
- `:disabled`  
Permet de cibler les éléments désactivé

## Remarque

Il existe d'autres types pour la balise « input », ces slides ne reprennent que les éléments les plus utiliser.

 Même si les formulaires vérifient les données de l'utilisateur grâce aux types des « input », il faudra **toujours** effectuer une vérification du **côté du serveur** qui traite le formulaire!

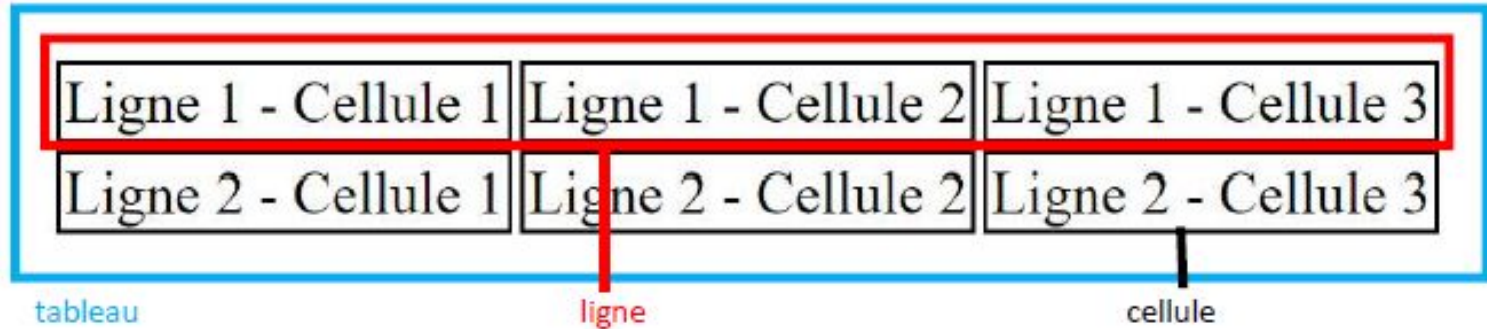
# Les tableaux

## HTML & CSS

# Les tableaux

Les tableaux HTML sont constitué principalement des 3 balises.

- La balise « table », qui permet de définir l'objet "tableau".
- La balise « tr », pour définir une ligne du tableau.
- La balise « td », pour définir une une cellule de la ligne.



## Exemple de tableau - Simple 1

```
<table>
  <tr>
    <td>Ligne 1 - Cellule 1</td>
    <td>Ligne 1 - Cellule 2</td>
    <td>Ligne 1 - Cellule 3</td>
  </tr>
  <tr>
    <td>Ligne 2 - Cellule 1</td>
    <td>Ligne 2 - Cellule 2</td>
    <td>Ligne 2 - Cellule 3</td>
  </tr>
</table>
```

# Les Tableaux

Pour définir l'entête des cellules, on utilise la balise « th »  
à la place de la balise « td »

```
<tr>
    <th>En-tête - Cellule 1</th>
    <th>En-tête - Cellule 2</th>
</tr>
```

Pour défin

## Exemple de tableau - Simple 2

```
<table>
  <caption>Résumé</caption>
  <tr>
    <th>Nom complet</th>
    <th>Abréviation</th>
  </tr>
  <tr>
    <td>HyperText Markup Language</td>
    <td>HTML</td>
  </tr>
  <tr>
    <td>Cascading Style Sheets</td>
    <td>CSS</td>
  </tr>
</table>
```

## Bordure d'un tableau

Si on ajoute dans le CSS une bordure aux éléments "table, td, th" on obtient le résultat suivant...

Résumé	
Nom complet	Abréviation
HyperText Markup Language	HTML
Cascading Style Sheets	CSS

Le CSS nous offre 2 po



# Bordure d'un tableau

- Fusionner les bordures

Pour cela, on utilise la propriété « border-collapse » avec la valeur "collapse" (valeur par défaut "separate").

- Modifier l'espacement des bordures

On utilise la propriété « border-spacing » suivis d'une ou 2 valeurs (horizontal & vertical) en px, em,...

*Fonctionne uniquement si le border-collapse vaut "separate".*

## Autre propriété CSS

- Alignement horizontal

On utilise la propriété « text-align » suivie d'une des valeurs suivante : center, left, right, justify

- Alignement vertical

On utilise la propriété « vertical-align » suivie d'une des valeurs suivante : top, middle, bottom.

## Autre propriété CSS

- Modifier la position du titre

On utilise la propriété « caption-side » suivie d'une des valeurs suivante : top, bottom

 Remarque 

Les cellules d'un tableau n'ont pas accès à la propriété « margin »

# Structurer un tableau

Lorsqu'on ne définit pas de structure, le navigateur place tous les éléments « tr » dans une balise « tbody ».

Il est possible de structurer les tableaux avec les balises suivante :

- « tbody » : forme le corps d'un tableau.
- « thead » : définit l'en-tête des colonnes.
- « tfoot » : définit les lignes qui résument les colonnes.

# Exemple de tableau - Structurer

```
<table>
  <caption>Titre</caption>
  <thead>
    <tr>
      <th>Contenu d'en-tête</th>
    </tr>
  </thead>
  <tfoot>
    <tr>
      <td>Pied de tableau</td>
    </tr>
  </tfoot>
  <tbody>
    <tr>
      <td>Contenu interne</td>
    </tr>
  </tbody>
</table>
```

# Fusion de cellules

- Fusionner des colonnes

On utilise l'attribut « colspan » dans la balise <td> suivi du nombre de cellules à fusionner.

```
<table>
  <tr>
    <td colspan="2">Fusion de colonne</td>
  </tr>
  <tr>
    <td>Cellule 1</td>
    <td>Cellule 2</td>
  </tr>
</table>
```



Fusion de colonne	
Cellule 1	Cellule 2

# Fusion de cellules

- Fusionner des lignes

On utilise l'attribut « rowspan » dans la balise <td> suivi du nombre de cellules à fusionner.

```
<table>
  <tr>
    <td rowspan="2">Fusion de ligne</td>
    <td>Cellule 1</td>
  </tr>
  <tr>
    <td>Cellule 2</td>
  </tr>
</table>
```



Fusion de ligne	Cellule 1
	Cellule 2

# Responsive design

## CSS



# Responsive design

Pour avoir un site avec un comportement responsive, on utilise la règle @ imbriquée « @media » combiné à des « requête média ».

Cela permet d'appliquer un ensemble de propriétés CSS lorsque l'appareil respecte la (ou les) condition(s) nécessaire.

- Exemple de requête média :  
min-width, max-width, aspect-ratio, orientation,...

## Exemple d'utilisation de @media

```
/* style général */  
p {  
    color : red;  
    background-color: black;  
}  
  
/* style appliqué pour les ecrans de plus de 1024 pixels */  
@media (min-width: 1024px) {  
    p {  
        color: gold;  
    }  
}
```

# Affichage pour Smartphone

Les navigateurs mobiles affichent les pages web dans une « fenêtre » virtuelle (viewport), celle-ci est souvent plus large que l'écran.

Lorsqu'on crée des interfaces responsives, il est préférable de contrôler la taille et l'échelle du viewport.



# Affichage pour Smartphone

Pour permettre de contrôler le viewport, il existe la balise « meta viewport » qui est supporté par beaucoup de navigateurs, bien qu'elle ne soit pas standardisée.

La plupart des sites optimisés pour les mobiles utilisent cette balise

```
<meta name="viewport" content="width=device-width, user-scalable=no">
```

# Attributs de la balise « meta viewport »

- width

Permet de définir la taille du viewport, elle peut avoir en argument une valeur en pixels (sans unité) ou le mot clef "device-width".

- user-scalable

Permet d'autoriser l'utilisateur à zoomer ou dézoomer sur la page. Les valeurs possible sont "yes" et "no".

# Attributs de la balise « meta viewport »

- initial-scale

Permet de définir le niveau de zoom lors du premier chargement, prend une valeur réel.  
Valeur par défaut : 1.0

- minimum-scale & maximum-scale

Permet de définir le niveau de zoom (minimum ou maximum), prend une valeur réel.

Merci pour votre attention.

