



IUT
de Blois



R109 – Introduction aux Technologies Web

Cours 3 - CSS

IUT Réseaux et Télécoms, 1^{ère} année

<http://celene.univ-tours.fr/course/view.php?id=4281>

Louise PARKIN - louise.parkin@univ-tours.fr

Objectifs

- Gérer l’affichage d’une page web
 - Connaître les principaux éléments CSS (texte, couleurs, bordures)
 - Positionner les éléments sur une page
 - Structurer le code CSS
- Connaître les principes de base d’Interface Homme Machine

Plan

- Généralités sur CSS
- Les fonctionnalités de CSS
- Structuration du code
- Placement des éléments HTML
- Responsive Web Design

Historique

- 1991 : Création de HTML
- 1991 – 1996 : mise en forme uniquement en HTML
(ex : ``)
- 1996 : création de CSS (Cascading Style Sheet = feuille de style en cascade)
- 1998 : CCS2, ajout de positionnement
- Depuis 2000 : CSS3, en partie mis en œuvre par les navigateurs

Affichages différents

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4   </head>
5   <body>
6     <h1> Exemple de code HTML</h1>
7     <h2> Premier sous-titre</h2>
8     <p> Voici un paragraphe<br>
9     et bla bla bla bla bla <br>
10    bla bla bla </p>
11    <h2> Deuxième sous-titre </h2>
12    <p> Voici un second paragraphe
13    <br> et rebla bla bla </p>
14  </body>
15 </html>
```

Exemple de code HTML

Premier sous-titre

Voici un paragraphe
et bla bla bla bla bla
bla bla bla

Deuxième sous-titre

Voici un second paragraphe
et rebla bla bla

Niveau 0 - Sans CSS

Affichages différents

```
1 body{
2     color:white;
3     background-color:#11aaff;
4     font-family:sans-serif;
5 }
6
7 p{
8     font-size:12px;
9 }
10 h1{
11     font-size:20px;
12     font-style:italic;
13     font-weight:bold;
14     text-align:center;
15 }
16 h2{
17     font-size:16px;
18     font-weight:bold;
19 }
```

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

Exemple de code HTML

Premier sous-titre

Voici un paragraphe
et bla bla bla bla bla
bla bla bla

Deuxième sous-titre

Voici un second paragraphe
et rebla bla bla

Niveau 1 – Ajout de feuille de style CSS

Affichages différents

```
1 body{
2     font-family:cursive;
3 }
4
5 p{
6     font-size:12px;
7 }
8 h1{
9     font-size:24px;
10    font-style:italic;
11    font-weight:bold;
12    border:solid 2px grey;
13    padding: 5px;
14 }
15 h2{
16     font-size:16px;
17     font-weight:bold;
18     margin:20px;
19 }
```

```
1 <!DOCTYPE html>
2 <html>
3     <head>
4         <link rel="stylesheet"
5             type="text/css"
6             href="moncss2.css">
7     </head>
8     <body>
18 </html>
```

Exemple de code HTML

Premier sous-titre

Voici un paragraphe
et bla bla bla bla bla
bla bla bla

Deuxième sous-titre

Voici un second paragraphe
et rebla bla bla

Affichages différents

```
header{  
  padding:5px;  
  margin:1px;  
  width:300px;  
  background-color:#2288FF;  
  height:60px;  
  float:left;  
  text-align:center  
}  
  
section{  
  padding:5px;  
  margin:1px;  
  width:300px;  
  background-color:#66ccFF;  
  float:left;  
  clear:both;  
}
```

```
<header>  
  <h1> Exemple de code HTML</h1>  
</header>  
<section>
```

Exemple de code HTML

Premier sous-titre

Voici un paragraphe
et bla bla bla bla bla
bla bla bla

Deuxième sous-titre

Voici un second paragraphe
et rebla bla bla

Niveau 2 – Ajout de feuille de style CSS et boîtes dans le code HTML

Principes d'IHM

- Séparation du contenu (HTML) et de la mise en forme (CSS)
 - Modifier facilement l'aspect d'un site sans changer le contenu
 - Cohésion de la présentation tout au long du site
 - Réduire le temps de chargement des pages
 - Correction de certains écueils d'HTML
 - Contrôle des polices, de la distance entre les lignes, des marges et des indentations
- Moins de code, mieux structuré → sites plus faciles à maintenir et à créer

Création de style CSS

- Fichier texte : éditeur de code (ex : notepad++), éditeurs de CSS (ex : *top Style 3* gratuit)
- Définir les propriétés des éléments HTML
 - Sélecteur : le nom d'un élément HTML sur lequel on veut mettre un style
 - Propriété : par exemple couleur, fonte, bordures

```
selecteur{  
    propriété1 : valeur1;  
    propriété2 : valeur2;  
    propriété3 : valeur3;  
}
```

```
p{  
    color : red;  
    font-size:12px;  
}
```

Création de style CSS

- Dans un fichier .css à part, lié dans l'entête <head>

```
<head>
  <meta charset="utf-8"/>
  <title> Ma page</title>
  <link rel="stylesheet" href="moncss.css">
</head>
```

- Dans le fichier .html, dans l'entête <head>

```
<head>
  <style type="text/css">
    h1{color:green;}
  </style>
</head>
```

- Dans une balise de code HTML

```
<li style="color:red;">texte rouge</li>
```

Code CSS

- Commentaires en `/**/` en CSS
- `<!--` et `-->` acceptés (commentaires HTML), cela permet de cacher le CSS pour les navigateurs qui ne le supportent pas
- Aucune prise en charge des erreurs
 - Propriétés inconnues
 - Valeurs invalides (unité !)
- Compatibilité avec la plupart des navigateurs (CSS3 supporté à partir de InternetExplorer 8.0)

Plan

- Généralités sur CSS
- **Les fonctionnalités de CSS**
- Structuration du code
- Placement des éléments HTML
- Responsive Web Design

Couleurs

- Toute combinaison entre #000000 et #FFFFFF est une couleur
 - Deux premiers chiffres : rouge, deux suivants : vert, deux derniers : bleu
- color : décrit la couleur du texte d'un élément. Peut s'appliquer à tous les éléments (p, body, span, div, ...)

```
<td style="color:aqua"> Aqua </td>  
<td style="color:#FF00FF"> Fuchsia </td>  
<td style="color:rgb(255,255,0)"> Yellow </td>
```

Nom	RGB_HEX	RGB_INT	Couleur
White	#FFFFFF	rgb(255,255,255)	
Silver	#C0C0C0	rgb(192,192,192)	
Gray	#808080	rgb(128,128,128)	
Black	#000000	rgb(0,0,0)	
Red	#FF0000	rgb(255,0,0)	
Lime	#00FF00	rgb(0,255,0)	
Blue	#0000FF	rgb(0,0,255)	
Aqua	#00FFFF	rgb(0,255,255)	
Fuchsia	#FF00FF	rgb(255,0,255)	
Yellow	#FFFF00	rgb(255,255,0)	
Maroon	#800000	rgb(128,0,0)	
Green	#008000	rgb(0,128,0)	
Navy	#000080	rgb(0,0,128)	
Teal	#008080	rgb(0,128,128)	
Purple	#800080	rgb(128,0,128)	
Olive	#808000	rgb(128,128,0)	

Arrière-plan

- background-color : pour choisir la couleur d'arrière-plan d'un élément

```
h1{background-color:#FF00FF;}
```

Les fables de La Fontaine

- background-image : pour spécifier l'image d'arrière-plan d'un élément

```
header{  
  color:red;  
  background-image:url(images/CR.jpg);
```



Polices de caractères

- Attributs de fontes
 - font-family : par exemple Times, Arial, Verdana, serif
 - font-style : normal | italic
 - font-variant : normal | small-caps
 - font-weight : normal | bold "bolder" ou "lighter«
 - text-decoration : overline | underline | line-through (rayé) | blink | none (par défaut)

Normal

surligné

Italique

souligné

PETITES CAPITALES

~~rayé~~

Gras

clignotant

Polices de caractères

- font-size : permet d'indiquer la taille du texte
 - Taille absolue : en px, cm, mm, pt
 - Taille relative : en % ou en em
 - ex : font-size: 16px;
- text-align : left | right | center | justify
- vertical-align : top | middle | bottom

Conversions :
3pt = 4px
1mm \approx 3,78 px
1mm \approx 2,83 pt

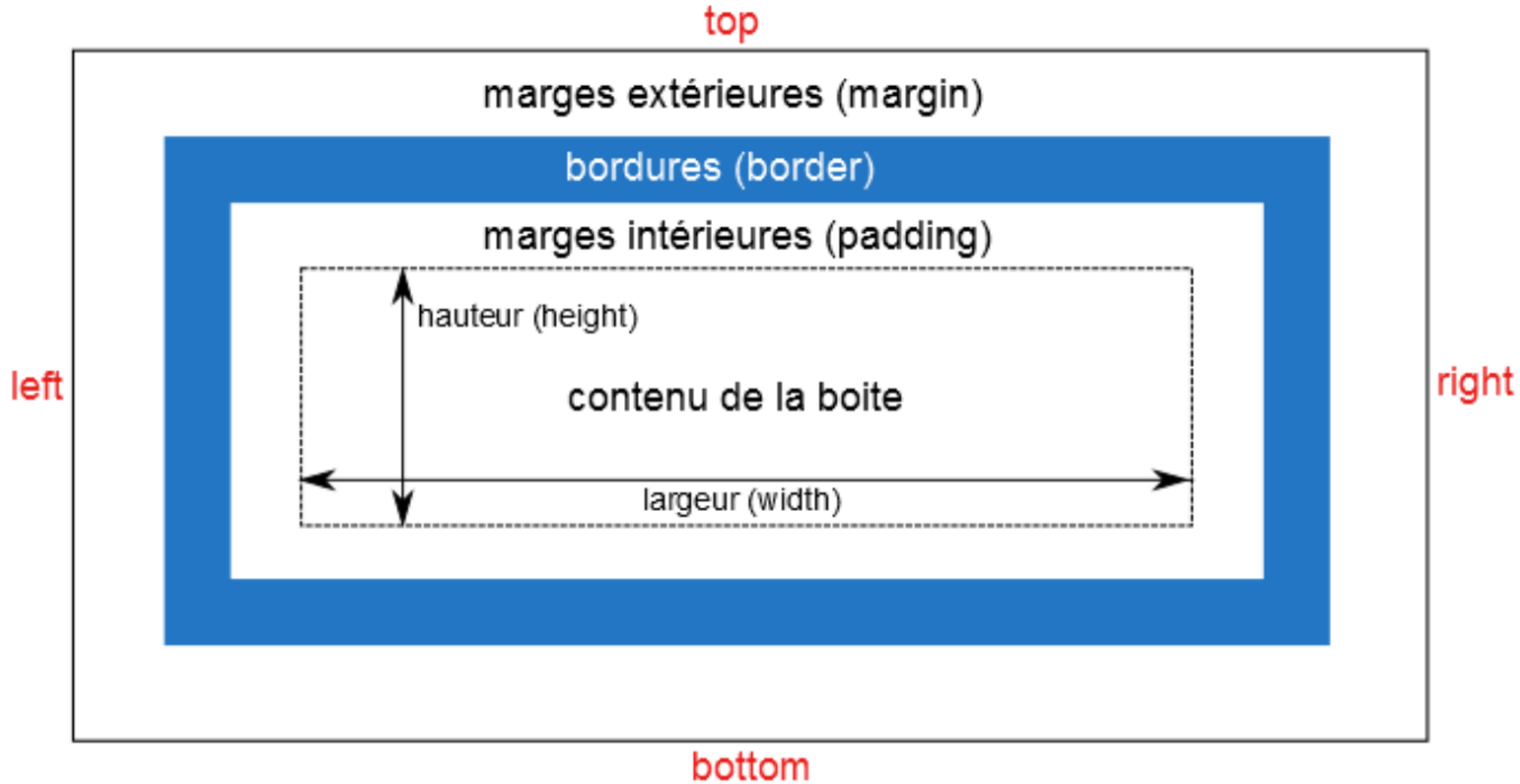
aligné à gauche

aligné à droite

centré

justifié

Contours



Bordures

- border-color, border-[top, right, bottom, left]-color : couleur de la bordure
- border-width, border-[top, right, bottom, left]-width : thin | medium | thick | longueur (px, cm) épaisseur de bordure
- border-style, border-[top, right, bottom, left]-style

```
p{  
  border-style: solid;  
  border-width: thick;  
  border-color: green;  
}
```

Texte entouré


| solid dotted | dashed double

```
p{  
  border: solid thick green;  
}
```

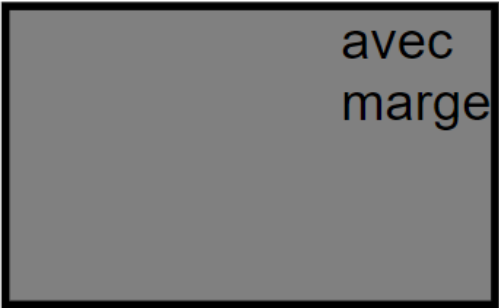
groove ridge inset outset

Marges

- margin-top, margin-right, margin-bottom, margin-left
- Valeur : longueur | pourcentage de largeur | auto | inherit
- Idem pour le padding (marge à l'intérieur de la bordure)



sans marge



avec
marge

```
<span style="background-color:grey;  
border:solid black; margin-top:50px;  
margin-bottom:50px;margin-right:150px;  
margin-left:100px"> avec marge</span>
```

Exemple de style

- Code html :

Pour faire `` un style important ``

- Style css :

```
strong{  
    font-family:Arial, sans-serif;  
    font-size:20px;  
    font-weight:bold;  
    font-decoration:underline;  
    font-variant:small-caps;  
    color:#AA0055;
```

Pour faire **UN STYLE IMPORTANT**

Plan

- Généralités sur CSS
- Les fonctionnalités de CSS
- Structuration du code
- Placement des éléments HTML
- Responsive Web Design

Définir des styles

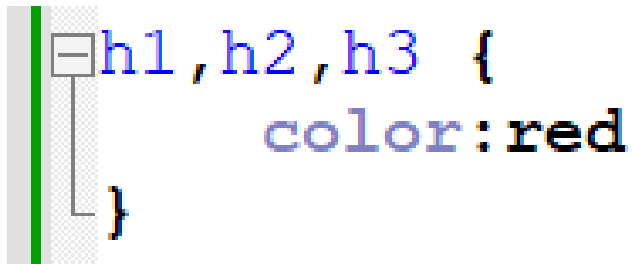
- Je veux que le style choisi pour la section soit conservé dans le paragraphe qu'elle contient → **Héritage**
- Je veux définir un même style pour les listes ordonnées et non ordonnées → **Regroupement**
- Je veux un style différent pour les tableaux s'ils sont dans un pied de page → **Sélecteur contextuel**
- Je veux le choix entre deux styles alternatifs pour mes sections → **Classe de style**
- Je veux que ces deux styles soient aussi applicables aux en-têtes → **Classe universelle**
- Je veux un changement de style au survol par la souris → **Pseudo-classes**

Héritage

- Au sens général : transmission de quelque chose d'une génération à une autre
- Au sens informatique : transmettre les caractéristiques de certains objets (parent) vers un autre (enfant) qui en dérive, permet d'organiser les objets en hiérarchie
- Au sens des CSS : permet lors de la déclaration d'un nouveau style d'y inclure les caractéristiques d'un autre

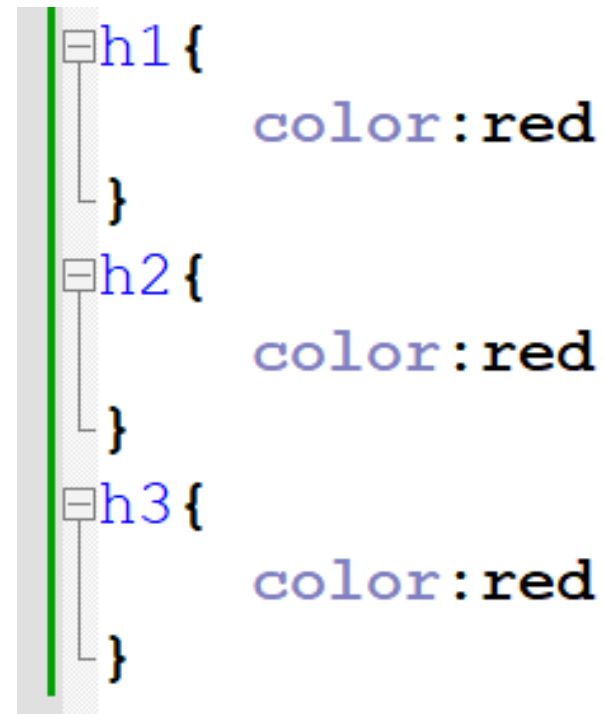
Regroupement d'éléments

- Pour appliquer un style à plusieurs balises :



A diagram illustrating a single CSS rule for multiple elements. On the left, a vertical grey bar has a green line segment. A small square icon with a horizontal line is positioned next to the green line. To the right of the icon, the text `h1, h2, h3 {` is written in blue, followed by `color:red` in black, and a closing curly brace `}` on the next line.

```
h1, h2, h3 {  
    color:red  
}
```



A diagram illustrating three separate CSS rules for multiple elements. On the left, a vertical grey bar has a green line segment. Three small square icons with horizontal lines are positioned next to the green line. To the right of the first icon, the text `h1 {` is written in blue, followed by `color:red` in black, and a closing curly brace `}` on the next line. To the right of the second icon, the text `h2 {` is written in blue, followed by `color:red` in black, and a closing curly brace `}` on the next line. To the right of the third icon, the text `h3 {` is written in blue, followed by `color:red` in black, and a closing curly brace `}` on the next line.

```
h1 {  
    color:red  
}  
h2 {  
    color:red  
}  
h3 {  
    color:red  
}
```

Sélecteurs contextuels

- Pour appliquer un style à une balise contenue dans une autre balise :
 - Ex : titres en rouge, mais titres dans un en-tête en bleu

```
<header>
  <h2> Exemple de code HTML</h2>
</header>
<section>
<h2> Premier sous-titre</h2>
```

Exemple de code HTML

Premier sous-titre


Voici un paragraphe
et bla bla bla bla bla
... ..

```
h2 {
  color:red
}

header h2 {
  color:blue
}
```

Classes de style

- Pour pouvoir appliquer différents styles à des balises identiques :
 - Ex : certains titre h2 en rouge, d'autres en bleu



```
h2.titrerouge{  
    color:red  
}  
  
h2.titrebleu{  
    color:blue  
}
```

Premier sous-titre

Voici un paragraphe

Deuxième sous-titre


Voici un second paragraphe

Troisième sous-titre

```
<h2 class="titrebleu"> Premier sous-titre</h2>  
<p> Voici un paragraphe</p>  
<h2 class="titrerouge"> Deuxième sous-titre </h2>  
<p> Voici un second paragraphe </p>  
<h2> Troisième sous-titre </h2>
```

Classes de style

- Pour pouvoir appliquer différents styles à des balises identiques :

A diagram showing a vertical bar with three colored segments: green at the top, orange in the middle, and green at the bottom. A bracket on the left side of the bar groups the orange and bottom green segments. To the right of the bracket is the CSS class selector code.

```
element.maclasse{  
    propriété1:valeur1;  
    propriété2:valeur2;  
}
```

Code CSS

```
<element class="maclasse">  
Texte avec le style de maclasse  
</element>  
<element>  
Texte avec le style de element  
</element>
```

Code HTML

Classes de style

- Plusieurs éléments peuvent utiliser une même classe de style (utilisation du regroupement)

<pre>a.bleu, h2.bleu{ color:blue }</pre>	<pre><h2> Titre</h2> <h2 class="bleu"> Titre bleu </h2> <a> Lien Lien bleu</pre>	<p>Titre</p> <p>Titre bleu</p> <p>Lien Lien bleu</p>
--	---	--

- Une balise peut avoir plusieurs classes

<pre>h2.bleu{ color:blue } h2.important{ text-decoration:underline }</pre>	<pre><h2> Titre sans style </h2> <h2 class="bleu important"> Titre avec deux styles </h2></pre>	<p>Titre sans style</p> <p><u>Titre avec deux styles</u></p>
--	---	--

Classe universelle

- On peut créer une classe sans l'associer à une balise
- Elle pourra alors être utilisée dans n'importe quelle balise

```
<header class="sombre">
  <h2> Exemple de code HTML</h2>
</header>
<section>
<h2>Titre de section </h2>
<a class="sombre"> Lien</a>
```

```
.sombre{
  color:white;
  background-color:black;
  padding:10px;
}
```

Exemple de code HTML

Titre de section

Lien

Identifiants

- L'identifiant fonctionne comme une classe
- Il est unique au sein d'une page html
 - Permet de localiser des éléments HTML grâce à javascript

```
  
<a id="lien1">Mon lien</a>
```

```
#logo{  
    width:100px;  
    border:1px dashed;  
}  
a#lien1{  
    background-color:green;  
}
```



Mon lien

Pseudo-classes

- Permet d'affiner le style appliqué à certaines balises
 - Réaction à un évènement
 - En fonction de la position relative de la balise par rapport aux autres
- Les pseudo-classes sont prédéfinies, il n'est pas possible de créer ses propres pseudo-classes
- Différents types de pseudo-classes
 - Dynamiques
 - De lien
 - De langue
 - First-child
 - De page
 - ...

Pseudo-classes dynamiques

- Modification du style selon un évènement (mouvement de souris, clic, appui clavier)
- 3 types différents
 - hover : survol par le curseur
`a: hover {font-decoration: underline;}`
 - focus : lorsque le focus est donné (par un clic dans l'élément)
`textarea: focus {color: #FF0000;}`
 - active : lorsque l'utilisateur maintient un clic sur l'élément
`a: active {color: #FF0000;}`
- Elles ne sont pas reconnues de la même façon par tous les navigateurs

Pseudo-classes de lien

- S'applique aux lien hypertexte (balises a)
 - link : si le lien n'a pas encore été consulté
 - visited : si le client a déjà visité la page cible

```
a:link {color:green;}
```

```
a:visited{color:red;}
```

Plan

- Généralités sur CSS
- Les fonctionnalités de CSS
- Structuration du code
- **Placement des éléments HTML**
- Responsive Web Design

Comportement inline

- Les éléments se placent toujours les uns à côté des autres afin de rester dans le texte (ex : lien, images)
 - Seules les propriétés de mise en forme s'appliquent (police, couleur)
 - Propriétés généralement héritées (marges, largeur)

```
<span>Logo</span>  
<span>Titre</span>  
<span>Menu</span>  
<span>Contenu</span>
```

Logo Titre Menu Contenu

Comportement bloc

- Les éléments occupent toute la largeur possible avec un saut de ligne entre chacun (ex : div, p, h1, li, etc)
 - On peut définir des marges et bordures
 - Une boîte a une marge extérieure à la bordure (margin) et une marge intérieure à la bordure (padding)
 - On peut fixer la largeur (width) et la hauteur (height) en pourcentage (%) ou pixels (px)

```
<div>Logo</div>
<div style="width:50%">
Titre</div>
<div style="margin-left:
50px">Menu</div>
<div style="padding-top:
10px">Contenu</div>
```



Overflow

- Si le contenu est plus grand que l'espace disponible
 - Overflow : visible | hidden | scroll | auto

```
div{  
  background-color:teal;  
  color:white;  
  width:200px;  
  height:100px;  
  overflow:scroll;  
}
```

Voici un paragraphe de
texte, trop long par
rapport à sa taille, donc il
faut indiquer comment
afficher le contenu avec

Display

- On peut forcer des éléments inline à se comporter comme des blocs et inversement avec display

```
li{display:inline;} strong{display:block;}
```

Voici du texte `` important `` mis en avant.

```
<ul>
```

```
  <li><a href="page1.html">HTML</a></li>
```

```
  <li><a href="page2.html">CSS</a></li>
```

```
  <li><a href="page3.html">JavaScript</a></li>
```

```
</ul>
```

Affichage sans CSS

Voici du texte **important** mis en avant.

- [HTML](#)
- [CSS](#)
- [JavaScript](#)

Affichage avec CSS

Voici du texte

IMPORTANT

mis en avant.

[HTML](#) [CSS](#) [JavaScript](#)

Positionnement des boîtes

- Propriété position
 - relative : par rapport à d'autres éléments
 - absolute : par rapport à la page
 - Coordonnées top, bottom, left, right en pixels (px)
 - A éviter car difficile à adapter à différents contextes
- Utiliser les marges et éléments flottant pour s'adapter

```
h2{  
  position:absolute;  
  left:100px;  
  top:150px;  
}
```

`<h2> mon titre</h2>`
`<p> avec la position absolue, un élément peut être placé n'importe où sur la page</p>`

Affichage sans CSS

mon titre

avec la position absolue, un élément peut être placé n'importe où sur la page

Affichage avec CSS


avec la position absolue, un élément peut être placé n'importe où

mon titre

Les grilles

Inspecteur de grille firefox sur le site de l'Université de Tours

- 3 colonnes
- 5 rangées

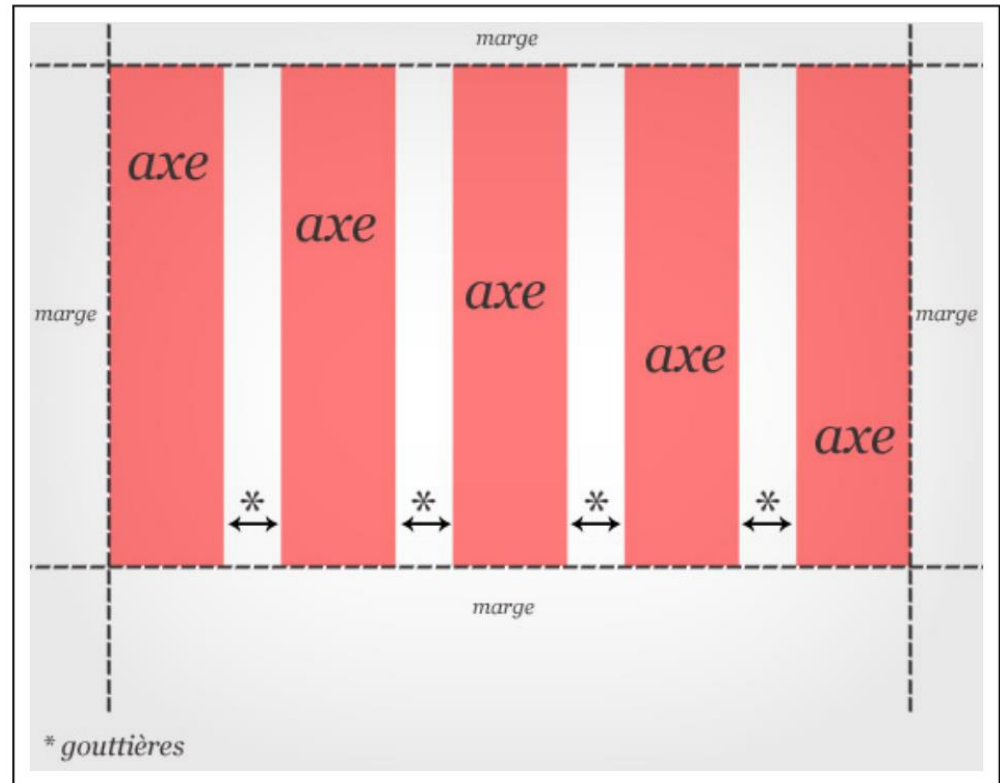
Agenda		
	26 Août - 06 Sept. Inscriptions administratives 2024-2025	Rechercher sur le site Licence, DUT, Master, Inscription
12 Sept. Fac'à vélo	18 Juin - 30 Août Candidature statut Sportif Haut ou Bon Niveau - université de Tours	
12 Juin - 04 Oct. Appel à candidature : troupe universitaire de	Connect with students	18 Sept. Faut-il construire de nouvelles relations internationales ? - Leçon
19 Sept. - 11 Oct. Passeport Vie Pro : Trouvez votre premier emploi !	28 Août - 18 Oct. Parraine un étudiant international grâce au Buddy System !	Inscription ici (ouvert à toutes et tous dans la limite des places disponibles)

Les grilles

- Ensemble de lignes horizontales et verticales qui se croisent
 - Hauteur et largeur des rangées et colonnes à taille fixe ou variable
 - Les éléments sont placés sur la grille en fonction des rangées et colonnes, grâce à leur numéro de rangée et colonne
 - Il y a un algorithme de placement automatique pour les éléments non placés
 - Création de colonnes ou rangées automatique pour du contenu
 - Contrôle de l'alignement des éléments
 - Contrôle des contenus qui se chevauchent
 - Le point de départ d'une grille est le conteneur

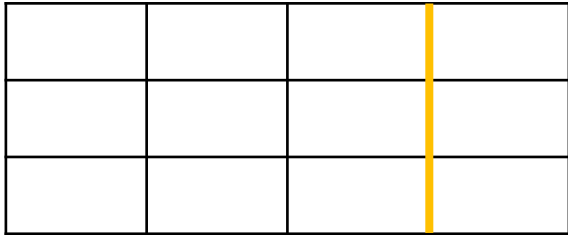
Les grilles

- Les grilles comportent généralement :
 - Des repères (axes) verticaux et/ou horizontaux, séparés par des gouttières
 - Des marges externes et internes

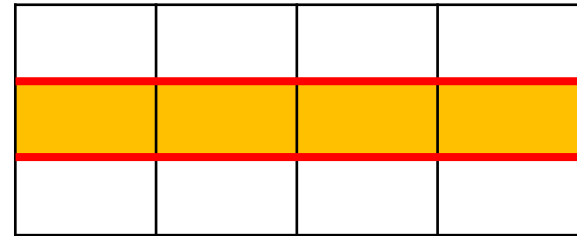


Les grilles

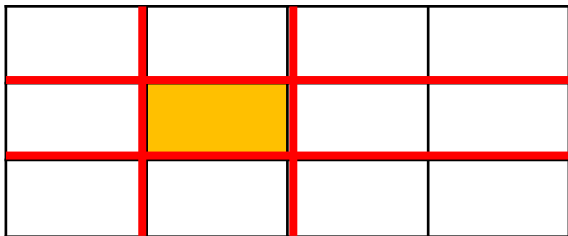
Ligne : verticale ou horizontale, délimitation des colonnes ou rangées



Piste (ou plage) de grille : espace situé entre 2 lignes adjacentes. Ce sont les colonnes ou rangées



Cellule : espace situé entre 2 lignes de rangée et 2 lignes de colonne adjacentes



Zone : espace situé entre 2 lignes de rangée et 2 lignes de colonne



Les grilles

Agencement des blocs : remplissage de la grille

- Bloc sur une zone de 2 colonnes
- Bloc sur une cellule
- Bloc sur une zone de 3 lignes

The collage consists of several posters and banners arranged in a grid-like fashion:

- Top Left:** A banner titled "Agenda" with a background image of three students in purple t-shirts.
- Top Middle:** A poster for "Inscriptions administratives 2024-2025" dated "26 Août - 06 Sept." with a background image of a building.
- Top Right:** A green banner titled "Rechercher sur le site" with a search bar containing the text "Licence, DUT, Master, Inscription" and a magnifying glass icon.
- Middle Left:** A poster for "ACTIVITÉS PHYSIQUES ET SPORTIVES VIE ÉTUDIANTE" dated "12 Sept." with the text "Fac'à vélo" and a background image of a student on a bicycle.
- Middle Center:** A poster for "ACTIVITÉS PHYSIQUES ET SPORTIVES" dated "18 Juin - 30 Août" for "Candidature statut Sportif Haut ou Bon Niveau - université de Tours" with a background image of a student in a karate gi.
- Middle Right:** A poster for "DROIT-ECONOMIE-EUROPE RECHERCHE" dated "18 Sept." titled "Faut-il construire de nouvelles relations internationales ? - Leçon" with a background image of a professor.
- Bottom Left:** A poster for "FORMATION - ORIENTATION - INSERTION VIE ÉTUDIANTE" dated "12 Juin - 04 Oct." titled "Appel à candidature : troupe universitaire de" with a background image of a student in a blue graduation gown.
- Bottom Center:** A poster for "INTERNATIONAL VIE ÉTUDIANTE" dated "28 Août - 18 Oct." titled "Parraine un étudiant international grâce au Buddy System !" with a background image of a student reading.
- Bottom Right:** A poster for "DROIT-ECONOMIE-EUROPE RECHERCHE" dated "18 Sept." titled "Faut-il construire de nouvelles relations internationales ? - Leçon" with a background image of a professor.

Les grilles - exemple

Conteneur (wrapper)
créé par display:grid

```
.wrapper{  
    display:grid}
```

En-tête

- [menu1](#)
- [menu2](#)

Titre 1

bla bla bla ...

Infos 1

Infos 2

Pub

Pied de page

```
<div class="wrapper">  
  <header class="tete"> En-tête </header>  
  <nav class="menu">  
    <ul>  
      <li><a href="#"> menu1</a></li>  
      <li><a href="#">menu2</a></li>  
    </ul>  
  </nav>  
  <section class="part1">  
    <article class="art1">  
      <h1> Titre 1 </h1>  
      <p>bla bla bla ...</p>  
    </article>  
    <aside class="info1"> Infos 1 </aside>  
    <aside class="info2"> Infos 2 </aside>  
    <aside class="pub"> Pub </aside>  
  </section>  
  <footer class="pied"> Pied de page </footer>  
</div>
```


Les grilles - exemple

- `grid-template-columns` et `grid-template-rows` pour définir les colonnes et rangées
- `fr` pour fraction de l'espace disponible, permet de la flexibilité
- `repeat()` répète des pistes : *20px repeat (6, 1fr) 20px* crée une colonne de 20 pixels de chaque côté, et divise l'espace restant en 6

```
.wrapper{  
  display:grid;  
  grid-template-columns:100px 200px 100px;  
}  
  
.wrapper{  
  display:grid;  
  grid-template-columns:1fr 2fr 1fr;  
}
```

En-tête	<ul style="list-style-type: none">• menu1• menu2	Titre 1 bla bla bla ...
Infos 1	Infos 2	Pub
Pied de page		

En-tête	<ul style="list-style-type: none">• menu1• menu2	Titre 1 bla bla bla ...
Infos 1	Infos 2	Pub
Pied de page		

Les grilles - placement

- Placement des éléments en fonction des lignes de la grille avec
 - grid-column-start, grid-column-end
 - grid-row-start, grid-row-end

	Ligne de colonne 1	Ligne de colonne 2	Ligne de colonne 3	Ligne de colonne 4
Ligne de rangée 1	En-tête	<ul style="list-style-type: none">• menu1• menu2	Titre 1	
Ligne de rangée 2			bla bla bla ...	
Ligne de rangée 3	Infos 1	Infos 2	Pub	
	Pied de page			

Les grilles - placement

```
.tete{
  background-color:yellow;
  grid-column-start:2;
  grid-column-end:4;
  grid-row-start:1;
}
.menu{
  background-color:lightgray;
  grid-column-start:1;
  grid-row-start:1;
  grid-row-end:3;
}
```

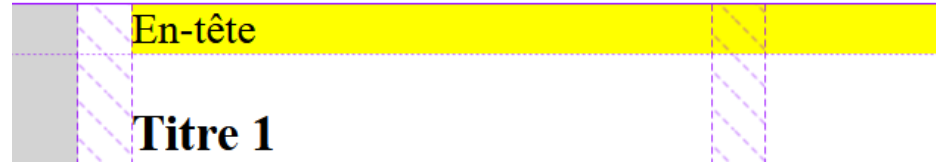
```
.part1{
  grid-column-start:2;
  grid-column-end:4;
  grid-row-start:2;
}
.pied{
  background-color:aqua;
  grid-column-start:1;
  grid-column-end:3;
  grid-row-start:3;
  grid-row-end:3;
}
```

	En-tête	
<ul style="list-style-type: none">menu1menu2	Titre 1	
	bla bla bla ...	
	Infos 1	
	Infos 2	
	Pub	
Pied de page		

Les grilles

- Gouttières = espacement entre les colonnes ou rangées
 - `grid-column-gap`, `grid-row-gap`

```
.wrapper{  
  display:grid;  
  grid-template-columns:1fr 1fr 1fr;  
  grid-auto-rows:minmax(50px,auto)  
  grid-column-gap:20px  
}
```

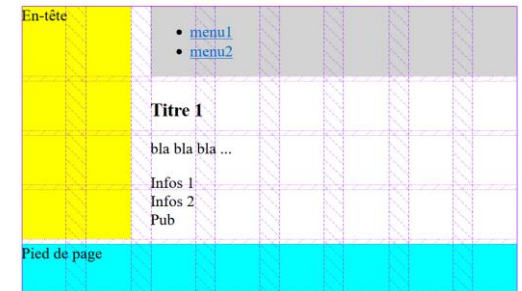


- Si des éléments sont placés en dehors de la grille, des colonnes ou rangées sont ajoutées automatiquement. `grid-auto-rows` fixe leur taille avec `minmax(50px,auto)` : taille mini de 50 pixels, maxi auto
- On peut imbriquer des grilles
 - Il n'y a pas d'héritage donc les lignes ne s'alignent pas



Les grilles - placement

- Autre moyen de placer les éléments : grid-area
 - On donne un nom à chaque zone, par exemple tt pour tête
 - On indique dans le wrapper où chaque zone se trouve dans les rangées et colonnes.
 - Ici 8 colonnes et rangées automatiques



```
.tete{
  background-color:yellow;
  grid-area:tt;}
.menu{
  background-color:lightgray
  grid-area:mn;}
.part1{grid-area:pt1;}
.pied{
  background-color:aqua;
  grid-area:ft;}
```

```
.wrapper{
  display:grid;
  grid-template-columns:repeat(8,1fr);
  grid-auto-rows:minmax(50px,auto);
  grid-column-gap:20px;
  grid-row-gap:5px;
  grid-template-areas:
    "tt tt mn mn mn mn mn mn"
    "tt tt pt1 pt1 pt1 pt1 pt1 pt1"
    "tt tt pt1 pt1 pt1 pt1 pt1 pt1"
    "tt tt pt1 pt1 pt1 pt1 pt1 pt1"
    "ft ft ft ft ft ft ft ft";}
```

Plan

- Généralités sur CSS
- Les fonctionnalités de CSS
- Structuration du code
- Placement des éléments HTML
- Responsive Web Design

Responsive Web Design

- Responsive Web Design (RWD) = web adaptatif
- On construit des interfaces auto-adaptables
- Technologie CSS3 media queries
 - Extension de la règle @media pour adapter la mise en page du site en fonction de l'outil utilisé pour le consulter (ordinateur, tablette, smartphone, télévision, imprimante, ...)
 - Permet d'utiliser des règles CSS différentes selon les caractéristiques du terminal de consultation
- Mobile first (mobile en premier)
 - Commencer par l'interface pour terminaux mobiles, avec celle pour un ordinateur
 - Car il y a une contrainte plus forte (espace d'affichage) qui nécessite d'épurer au maximum les éléments visibles

Media queries

- Fonctionnalités (préfixé de min/max pour borner)
 - aspect-ratio : ratio du périphérique de sortie (ex : 16/9)
 - resolution : résolution du périphérique (en dpi, dppx, dpcm)
 - height : hauteur de la zone d'affichage (fenêtre)
 - width : largeur de la zone d'affichage (fenêtre)
 - orientation : orientation du périphérique (portrait/paysage)
 - media : type d'écran de sortie, par exemple
 - screen : écran classique
 - handheld : périphérique mobile
 - print : impression
 - tv : télévision
 - projection : projecteur
 - all : tous types d'écran

Media queries

- Si le média est une imprimante, ne pas imprimer les parties nav, aside et footer, imprimer le texte en noir
- Si la taille de l'écran est inférieure à 1024 pixels, écrire en blanc sur fond gris, par défaut écrire en bleu

```
@media print{  
  nav,aside,footer{  
    display:none;  
  }  
  body{  
    color:black;  
  }  
}
```

```
p{color:blue}  
@media screen and  
(max-width:1024px){  
  p{ color:white;  
    background-color:gray;  
  }  
}
```

avec la position absolue, un élément peut être placé n'importe où sur la page

avec la position absolue, un élément peut être placé

Redéfinir les grilles

- Avec les grilles, on peut facilement adapter l’affichage pour les différentes résolutions
- On souhaite avoir un style différent selon qu’on soit sur un écran « large » ou un écran « mobile » type smartphone
- Définir d’abord les éléments communs
 - Zones tete, menu, pied
 - Décoration des zones art1, info1, info2, pub
 - Grid-area des éléments

```
.tete{background-color:yellow;
      grid-area:tt;}
.menu{
      background-color:lightgray;
      grid-area:mn;}
.info1{background-color:lightgray;
        grid-area:i1;}
.info2{background-color:lightgray;
        grid-area:i2;}
.pub{background-color:fuchsia;
      grid-area:pb;}
.part1{grid-area:pt1;}
.art1{grid-area:a1;}
.pied{background-color:aqua;
      grid-area:ft;}
```


Redéfinir les grilles

- Définir ensuite ce qui s'applique pour les écrans de taille supérieure à 500px
 - @media all and (min-width:500px)
 - Cette règle contient le wrapper et la .part1, ect qui seront différents selon l'écran

```
@media all and (min-width:500px) {  
  .wrapper{  
    display:grid;  
    grid-template-columns:repeat(8,1fr) ;  
    grid-auto-rows:minmax(50px,auto) ;  
    grid-column-gap:20px;  
    grid-row-gap:5px;  
    grid-template-areas:  
    "tt tt mn mn mn mn mn mn"  
    "tt tt pt1 pt1 pt1 pt1 pt1 pt1"  
    "tt tt pt1 pt1 pt1 pt1 pt1 pt1"  
    "tt tt pt1 pt1 pt1 pt1 pt1 pt1"  
    "ft ft ft ft ft ft ft ft";  
  }
```

```
.part1{display:grid;  
grid-auto-columns:minmax(100px,auto) ;  
grid-auto-rows:minmax(100px,auto) ;  
grid-gap:10px;  
grid-template-areas:  
"i1 a1 pb"  
"i2 a1 pb"  
}  
}
```



Redéfinir les grilles

- Enfin, on définit ce qui s'applique pour les mobiles
 - @media handheld
 - On adapte le wrapper pour faire 1 seule colonne

```
@media handheld{  
  .wrapper{  
    display:grid;  
    grid-template-columns:1fr;  
    grid-auto-rows:minmax(50px,auto) ;  
    grid-row-gap:5px;  
    grid-template-areas:  
      "tt"  
      "mn"  
      "a1"  
      "i1"  
      "i2"  
      "pb"  
      "pt1"  
      "ft";  
  }  
}
```

En-tête

- [menu1](#)
- [menu2](#)

Titre 1

bla bla bla ...

Infos 1

Infos 2

Pub

Pied de page