

Concevoir des interfaces graphiques avec HTML5, CSS3 et JavaScript



CSS

Programme détaillé ou sommaire

Introduction

Utilisation de CSS

Les sélecteurs

Les principales propriétés

Les sélecteurs avancés

Affichage et positionnement (HTML4)

Affichage et positionnement (HTML5)

Les media queries



Qu'est-ce que c'est ?

- ❑ CSS signifie **C**ascading **S**tyle **S**heets
- ❑ Dans la conception web il est d'usage de séparer le fond de la forme



Structure



Présentation



Qu'est-ce que CSS ?

- ❑ HTML s'occupe du fond et CSS de la forme
- ❑ Avec CSS on va pouvoir appliquer du style (couleur, police, forme) et/ou des effets (animations) à tous les éléments HTML en fonction :
 - De leur id (id=xxx)
 - De leur type (input, div, select, h1, h2, span)



- **France**

65 millions d'habitants

- **Belgique**

10 millions d'habitants

- **Allemagne**

83 millions d'habitants



France

65 millions d'habitants



Belgique

10 millions d'habitants



Allemagne

83 millions d'habitants

- ❑ **1996**: apparition de CSS pour le navigateur Netscape
 - 1 an après javascript
- ❑ **1997**: début du travail sur les spécifications de CSS2
 - Introduit de nombreuses propriétés, notamment de positionnement
- ❑ **1999**: début du travail sur les spécifications de CSS3
 - La finalisation prend plus de 10 ans
 - Son implémentation dans les navigateurs est finalisée à 99%
 - Beaucoup d'améliorations notamment au niveau des sélecteurs CSS





Utilisation de CSS

Méthode #1

- ❑ CSS **in-line** qui permet d'ajouter du style directement dans le composant:

```
<h1 style="color: blue;">Bonjour !</h1>
```

Bonjour !

- ❑ Fonctionne sur toutes les balises HTML avec l'attribut **style**



Méthode #1

❑ Syntaxe:

- Les différentes propriétés sont séparées par des ;
- La propriété et sa valeur sont séparées par un :

```
<tag style="prop1: val1; prop2: val2; etc...">...</tag>
```

- ## ❑ Problème:
- si vous voulez avoir un style similaire pour une famille de balises donnée (exemple h1), vous devrez copier/coller l'attribut style avec toutes ses propriétés et valeurs sur chacune de ces balises.



Méthode #2

❑ CSS déclaré dans le **head** dans une **balise style**:

```
<head>
  <style>
    h1 {
      color: blue;
    }
  </style>
</head>
<body>
  <h1>Bonjour !</h1>
</body>
```

On précise le **type de balises** auxquelles le style s'applique puis on ouvre les accolades pour déclarer les propriétés et leurs valeurs

Bonjour !



Méthode #2

❑ **Problème:** pour avoir une charte graphique cohérente vous devrez copier/coller ces styles dans l'ensemble de vos pages !



```
<head>
  <style>
    h1 {
      color: blue;
    }
  </style>
</head>
<body>
  <h1>Bonjour !</h1>
</body>
```

Bonjour !



Méthode #3

- ❑ Le **fichier CSS** qu'on pourra importer dans l'ensemble des pages
- ❑ Page HTML



```
<head>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1>Bonjour !</h1>
</body>
```

Bonjour !

- ❑ Fichier styles.css:

```
h1 {
  color: blue;
}
```



Exemple de fichier css

❑ Exemple:

```
styles.css x
55
56 h1 {
57     font-family: 'SmallCaps', sans-serif;
58     font-size: 2em;
59     font-weight: bold;
60     font-variant: small-caps;
61     text-align: center;
62     margin-top: 1.0em;
63     margin-bottom: 0.5em;
64 }
65
66 h2 {
67     font-family: 'SmallCaps', sans-serif;
68     font-size: 1.6em;
69     font-weight: bold;
70     font-variant: small-caps;
71     text-align: center;
72     margin-top: 1.0em;
73     margin-bottom: 0.5em;
74 }
75 }
```





Les sélecteurs CSS

Qu'est ce qu'un sélecteur CSS

- ❑ Un **sélecteur CSS** permet de définir **comment** et **sur quoi le style va s'appliquer**.
- ❑ Il y a **plusieurs manières** d'appliquer un style:
 - Par élément
 - Par attribut
 - Par affiliation
 - Par action
 - Par état



Syntaxe

☐ Par défaut

```
sélecteur {  
    propriété1: valeur1;  
    propriété2: valeur2;  
}
```

☐ Sélecteur multiple

```
sélecteur1, sélecteur2 {  
    propriété1: valeur1;  
    propriété2: valeur2;  
}
```



Exemple

- ❑ Style qui s'applique à tous les éléments HTML de type **input**

```
input {  
  propriété1: valeur1;  
  propriété2: valeur2;  
}
```



Style nommé

- ❑ Définition d'un style avec un nom particulier.
- ❑ Le nom du style nommé doit être précédé du caractère point (.)
- ❑ Par défaut ce style ne sera appliqué à aucun élément.

```
.nom {  
    propriété1: valeur1;  
    propriété2: valeur2;  
}
```

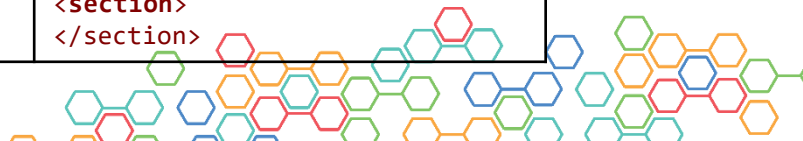
- ❑ Pour appliquer ce style à un élément il faudra utiliser l'attribut **class**

```
<h1 class="nom">Bonjour !</h1>
```



Les principaux sélecteurs

Une famille de balise donnée	nom_balise { }	<code><h1>Bonjour</h1></code>
Un élément dont l'identifiant est X	#X { }	<code><h1 id="X">Bonjour</h1></code>
Tous les éléments ayant la classe nom	.nom { }	<code><h1 class="nom">Bonjour !</h1></code>
Tous les éléments element2 contenus dans element1	element1 element2 { }	<code><section> <article> <h1>Bonjour</h1> </article> </section></code>
Tous les éléments element2 dont le parent est élément1	element1 > element2 { }	<code><section> <h1>Bonjour</h1> </section></code>
Tous les éléments element2 placés juste après élément1	element1 + element 2 { }	<code><h1>Bonjour</h1> <section> </section></code>
Tous les éléments element2 précédés par élément1	element1 ~ element 2 { }	<code><h1>Bonjour</h1> <section> </section></code>





Les principales propriétés

Les propriétés de texte et de couleur

Couleur de fond	background-color	<code>background-color: mediumblue;</code> <code>background-color: rgb(63, 155, 44);</code> <code>background-color: #330050;</code>
Couleur de la police	color	<code>color: mediumblue;</code> <code>color: rgb(63, 155, 44);</code> <code>color: #330050;</code>
Choix de la police, par ordre de priorité. Si un navigateur ne supporte pas la 1 ^{ère} alors la 2 ^{nde} est appliquée et ainsi de suite.	font-family	<code>font-family: Verdana, Geneva, Tahoma, sans-serif;</code>
Taille de la police	font-size	<code>font-size: 1.2em;</code> <code>font-size: 1.2rem;</code> <code>font-size: 20px;</code> <code>font-size: medium;</code>
Type de la police	font-style	<code>font-style: italic;</code>
Poids de la police	font-weight	<code>font-weight: bold;</code>



Les unités de taille de police: rem

- ❑ Les tailles en **rem** représentent un coefficient par rapport à la taille de la police par défaut, ou celle définie pour la page HTML
- ❑ Exemple de taille de police par défaut pour la page

```
html {  
    font-size: 16px;  
}  
h1 {  
    font-size: 0.9rem; /* 90% de la taille de la police html */  
}  
h2 {  
    font-size: 0.8rem; /* 80% de la taille de la police html */  
}
```

- ❑ En changeant la taille de la police par défaut, je modifie toutes les autres



Les unités de taille de police: em

- ❑ Les tailles en **em** fonctionnent comme celles en *rem* mais par rapport à la taille de la police de l'élément parent
- ❑ C'est un peu plus complexe à utiliser que les *rem*.



Les propriétés de bordure et d'espacement

Epaisseur de la bordure	border-width	<code>border-width: 1px;</code>
Style de la bordure	border-style	<code>border-style: solid;</code>
Couleur de la bordure	border-color	<code>border-color: lightcoral;</code>
Les 3 précédentes déclarées en 1 seule fois	border	<code>border: 1px solid lightcoral;</code>
Espaces à l'intérieur d'un élément	padding	<code>padding: 5px 5px 5px 5px; /* top right bottom left */</code>
Espaces entre 1 élément et ses voisins	margin	<code>margin: 0px 0px 0px 0px; /* top right bottom left */</code>
Arrondir les coins	border-radius	<code>border-radius: 3px;</code>

❑ Un **sélecteur CSS** permet de définir **comment** et **sur quoi le style va s'appliquer**.



Exemple

❑ Exemple pour encadrer un texte

```
.text1 {  
    border: 1px solid lightcoral;  
    border-radius: 3px;  
    padding: 5px 5px 5px 5px; /* top right bottom left */  
}
```

`Exemple avec un span de texte`

`Autre exemple`

Exemple avec un span de texte

Autre exemple



Avec et sans padding

☐ Avec

Exemple avec un span de texte

Autre exemple

☐ Sans

Exemple avec un span de texte

Autre exemple



Padding vs margin

- ❑ **Padding** : 4 espaces entre le contenu de l'élément et son bord

Exemple avec un span de texte

Autre exemple

- ❑ **Margin**: 4 espaces entre un élément et les éléments voisins.
- ❑ Ci-dessous c'est la **margin-right** qui a été modifiée

Exemple avec un span de texte

Autre exemple



- OBJECTIFS :
- Définir des CSS afin d'obtenir un résultat attendu
- DESCRIPTION : Dans ce TP n°1 vous allez devoir appliquer du style afin de reproduire un visuel particulier





Les sélecteurs avancés

Sélecteur d'attribut

- ❑ Comment appliquer un style uniquement aux éléments html qui ont un attribut donné avec une valeur donnée ?
- ❑ Dans le sélecteur CSS, on ouvre les crochets, et on définit le nom de l'attribut ainsi que la valeur recherchée.
- ❑ Exemple pour appliquer un style uniquement aux éléments **input** de **type button**:

```
input[type="button"] {  
    propriété1: valeur1;  
    propriété2: valeur2;  
}
```



Les divers types de sélecteur (2/2)

Tous les éléments avec l'attribut attr	[attr] { }	<input type ="text" name ="nom">
Tous les éléments avec l'attribut attr et dont la valeur vaut val	[attr=val] { }	<input type ="text" name ="nom">
Tous les éléments avec l'attribut attr et dont la valeur contient val	[attr*=val] { }	<input type ="text" name ="nom"> <input type ="number" name ="age">
Tous les éléments avec l'attribut attr et dont la valeur commence par val	[attr^=val] { }	<input type ="text" name ="nom"> <input type ="time" name ="heure">
Tous les éléments avec l'attribut attr et dont la valeur termine par val	[attr\$=val] { }	<input type ="date" name ="nom"> <input type ="time" name ="heure">



Sélecteur de pseudo-classe

- ❑ On peut sélectionner un élément en fonction d'un évènement ou d'un état
- ❑ Syntaxe

:pseudo-classe

- ❑ Par exemple, je veux donner un style particulier à un lien HREF lorsque je survole celui-ci avec la souris

```
<a href="https://fr.wikipedia.org/" target="_blank">Lien</a>
```

```
a:hover {  
    color: red;  
    font-style: italic;  
}
```

Lien

Lien



On peut combiner les sélecteurs

- ❑ On peut combiner les sélecteurs.
- ❑ Exemple pour appliquer un style lorsqu'on survole un lien HREF qui a la classe highlight

```
<a class="highlight" href="https://fr.wikipedia.org/"  
target="_blank">Lien</a>
```

```
a.highlight:hover {  
    color: red;  
    font-style: italic;  
}
```

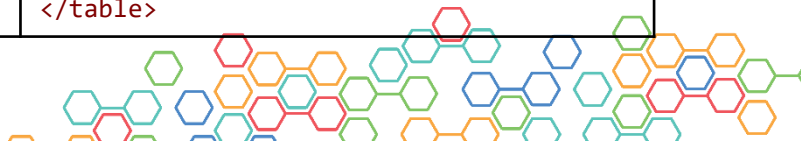
Lien

Lien



Les pseudo-classes structurelles

Tous les éléments premier enfant de leur parent	<code>element:first-child {</code> <code>}</code>	<pre><table> <tr></tr> <tr></tr> <tr></tr> </table></pre>
Tous les éléments dernier enfant de leur parent	<code>element:last-child {</code> <code>}</code>	<pre><table> <tr></tr> <tr></tr> <tr></tr> </table></pre>
Tous les éléments qui sont le ième enfant d'un parent	<code>element:nth-child(i) {</code> <code>}</code>	<pre><table> <tr></tr> <tr></tr> <tr></tr> <tr></tr> </table></pre>
Tous les éléments pairs ou impairs	<code>element:nth-child(even) {</code> <code>}</code> <code>element:nth-child(odd) {</code> <code>}</code>	<pre><table> <tr></tr> <tr></tr> <tr></tr> <tr></tr> </table></pre>



Pays	Population (M)	Superficie (km2)
France	65.2	551 000
Espagne	46.7	510 000
Suède	10.1	450 000
Allemagne	83.8	357 000
Finlande	5.5	338 000
Norvège	5.4	323 000
Italie	60.4	301 000
Royaume-Uni	67.9	243 000

Pays	Population (M)	Superficie (km2)
France	65.2	551 000
Espagne	46.7	510 000
Suède	10.1	450 000
Allemagne	83.8	357 000
Finlande	5.5	338 000
Norvège	5.4	323 000
Italie	60.4	301 000
Royaume-Uni	67.9	243 000



Les pseudo-classes événementielles

Tous les éléments actifs Element sur lequel il y a la focus Element survolé par la souris	<code>element:active {</code> <code>}</code> <code>element:focus {</code> <code>}</code> <code>element:hover {</code> <code>}</code>
Lien non visité	<code>a:link {</code> <code>}</code>
Lien visité	<code>a:visited {</code> <code>}</code>
Eléments cochés	<code>input:checked {</code> <code>}</code>

- ❑ L'état **focus** est atteint par un élément lorsqu'il est sélectionné avec la touche **tab** ou lorsqu'on **clique dessus avec la souris**.
- ❑ L'état **active** est atteint par un élément uniquement lorsqu'on **clique dessus avec la souris**.
- ❑ Un élément peut être à la fois dans l'état focus et l'état active.



Les pseudo-classes positionnelles

Ajoute du contenu avant un élément	<code>element:before {</code> <code>}</code>
Ajoute du contenu après un élément	<code>element:after {</code> <code>}</code>

- ☐ L'état **focus** est atteint par un élément lorsqu'il est sélectionné avec la touche **tab** ou lorsqu'on **clique dessus avec la souris**.
- ☐ L'état **active** est atteint par un élément uniquement lorsqu'on **clique dessus avec la souris**.
- ☐ Un élément peut être à la fois dans l'état focus et l'état active.



Exemple

```
.box {  
  background-color: mediumblue;  
  width: 300px;  
  border: solid 1px lightblue;  
  padding: 10px;  
  color: white;  
}  
.box:before {  
  content: "#";  
  border: solid 1px lightblue;  
  padding: 2px;  
  margin: 0 10px 0 0;  
}
```

```
<p class="box">Hello !</p>
```

Hello !



- En se basant sur le code html page suivant: <https://www.exelib.net/css/affectation-d-un-style-sur-des-selecteurs-combinateurs.html>
- Utiliser les combinateurs et appliquer les styles CSS ci dessous :
- Un retrait gauche de 20px pour tous les paragraphes descendants de la classe "article".
- Un background de couleur jaune pour les paragraphes frères directs du titre "h2" (qui se trouvent immédiatement après h2) .
- Une bordure noir et solide pour les paragraphes adjacents du titre "h2".
- Une couleur verte pour les paragraphes fils des éléments "li".



Le résultat à obtenir:

Titre1

- Item1:

résumé item1.....

- Item2:

résumé item2.....

- Item3:

résumé item3.....

Lorem ipsum dolor sit amet, consectetur adipisicing elit

Lorem ipsum dolor sit amet, consectetur adipisicing elit

Titre2

Lorem ipsum dolor sit amet, consectetur adipisicing elit

Lorem ipsum dolor sit amet, consectetur adipisicing elit

exemple1

Lorem ipsum dolor sit amet, consectetur adipisicing elit





Affichage et positionnement (HTML4)

Créer une mise en page

- ❑ Appelée également **layout** en anglais.
- ❑ Correspond à la **structure d'une page**
- ❑ i.e. le **positionnement** des divers blocs les uns par rapport aux autres.



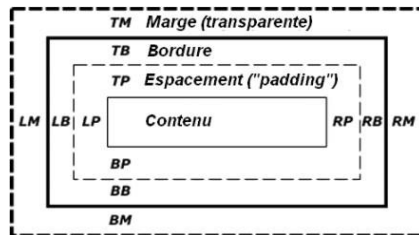
Le positionnement par défaut des éléments

- ❑ En HTML il existe **2 types** d'éléments en terme **de positionnement**
- ❑ Les éléments de type **block**
- ❑ Les éléments de type **inline**
 - Ce sont les éléments qui s'affichent sur la même ligne que l'élément précédent.
 - Exemple: a, span, img, input



Les éléments de type block

- ❑ Ce sont les éléments qui se comportent comme des **boîtes**.
- ❑ **En tant que boîtes**, ils peuvent avoir une hauteur, une largeur, des marges intérieures (padding) et extérieures (margin)



- ❑ Les blocks s'empilent verticalement.
- ❑ **Exemple:** `div`, `h1`, ..., `h6`, `p`, `form`, `header`, `footer`, `section`, `nav`, `article`, `li`



Les éléments de type inline

- ❑ Ce sont les éléments qui se comportent plutôt comme du contenu qu'on place sur une ligne.
- ❑ **En tant que inline**, à l'exception de l'image (img), ils ne peuvent pas avoir de hauteur, de largeur, de marge extérieure ou intérieure.
- ❑ Les **inline** s'organisent horizontalement.
- ❑ **Exemples:** span, input, img, a



Les éléments de type inline-block

- ❑ Ce sont les éléments qui se comportent comme des boîtes et se placent en ligne.
- ❑ **L'avantage est qu'on va pouvoir leur définir une** hauteur, une largeur, des marges extérieures ou intérieures.
- ❑ Un attribut CSS va permettre de modifier le type de l'élément (block, inline ou inline-block).



Attribut de style display

- ❑ Il est possible de modifier le comportement de positionnement d'un élément avec **display**.

```
span {  
  display: block;  
}
```

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



Application

- ❑ Par défaut l'élément **label** est **inline**. On ne peut donc pas lui attribuer de propriété de boîte.
- ❑ Voici un formulaire brut avec quelques labels
 - Si je veux par exemple mettre une largeur de 100px à chaque label pour avoir un alignement sur la droite je ne peux pas.

Nom

E-mail

Sujet

Message



Application: avec inline-block et largeur 100px

- ❑ Pour pouvoir mettre une largeur aux labels sans leur changer leur comportement inline, je dois passer en inline-block

```
label {  
    display: inline-block;  
    width: 100px;  
}
```

Nom	<input type="text"/>
E-mail	<input type="text"/>
Sujet	<input type="text"/>
Message	<input type="text"/>



Application: avec alignements

- ❑ Comme les labels ont désormais un comportement de boîte, on peut également utiliser des propriétés d'alignement

```
label {  
    display: inline-block;  
    text-align: right;  
    vertical-align: top;  
    width: 100px;  
}
```

Nom	<input type="text"/>
E-mail	<input type="text"/>
Sujet	<input type="text"/>
Message	<input type="text"/>



Application: avec une marge à droite

- ❑ On peut également ajouter une marge à droite afin de « décoller » un peu les **labels** des champs **inputs**.

```
label {  
    display: inline-block;  
    text-align: right;  
    vertical-align: top;  
    margin-right: 5px;  
    width: 100px;  
}
```

Nom	<input type="text"/>
E-mail	<input type="text"/>
Sujet	<input type="text"/>
Message	<input type="text"/>

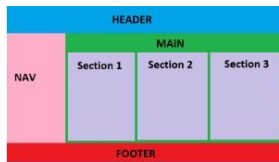


Propriété float

- ❑ Elle sert à faire par exemple flotter une image sur la gauche et faire s'enrouler le texte autour

```
<div class="parent">  
  <div class="child"></div>  
  Lorem ipsum dolor sit amet, consectetur adipiscing elit...  
</div>
```

```
.parent {  
  width: 50%;  
}  
  
.child {  
  float: left;  
}
```



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas ligula massa, varius a, semper congue, euismod non, mi. Proin porttitor, orci nec nonummy molestie, enim est eleifend mi, non fermentum diam nisl sit amet erat. Duis semper. Duis arcu massa, scelerisque vitae, consequat in, pretium a, enim. Pellentesque congue. Ut in risus volutpat libero pharetra tempor. Cras vestibulum bibendum augue. Praesent egestas leo in pede. Praesent blandit odio eu enim. Pellentesque sed dui ut augue blandit sodales. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam nibh. Mauris ac mauris sed pede pellentesque fermentum. Maecenas adipiscing ante non diam sodales hendrerit. Ut velit mauris, egestas sed, gravida nec, ornare ut, mi. Aenean ut orci vel massa suscipit pulvinar. Nulla sollicitudin. Fusce varius, ligula non tempus aliquam, nunc turpis ullamcorper nibh, in tempus sapien eros vitae ligula. Pellentesque rhoncus nunc et augue. Integer id felis. Curabitur aliquet pellentesque diam. Integer quis metus vitae elit lobortis egestas. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi vel erat non mauris convallis vehicula. Nulla et sapien. Integer tortor tellus, aliquam faucibus, convallis id, congue eu, quam. Mauris ullamcorper felis vitae erat. Proin feugiat, augue non elementum posuere, metus purus iaculis lectus, et tristique ligula justo vitae magna. Aliquam convallis sollicitudin purus. Praesent aliquam, enim at fermentum mollis, ligula massa adipiscing nisl, ac euismod nibh nisl eu lectus. Fusce vulputate sem at sapien. Vivamus leo. Aliquam euismod libero eu enim. Nulla nec felis sed leo placerat imperdiet. Aenean suscipit nulla in justo. Suspendisse cursus rutrum augue. Nulla tincidunt tincidunt mi. Curabitur iaculis, lorem vel rhoncus faucibus, felis magna fermentum augue, et ultricies lacus lorem varius purus. Curabitur eu amet



- OBJECTIFS :
- Définir des CSS afin d'obtenir un résultat attendu
- DESCRIPTION : Dans ce TP n°3 vous allez devoir appliquer du style afin de reproduire un visuel particulier

ACCUEIL

EXERCICES

EXAMENS

CONTACT

CONNEXION





Affichage et positionnement (HTML5)

Introduction aux flexbox

- ❑ Depuis HTML5, un nouveau système pour structurer les pages HTML (layouts) est apparu
- ❑ Le concept de **flexbox**.
- ❑ Ce concept améliore notablement le positionnement des composants principaux de la page.



Le conteneur flexible

- ❑ Le **conteneur flexible** est une **div**, qui par essence est un conteneur, avec la propriété CSS **display: flex**
- ❑ Par défaut les éléments du conteneur sont ajoutés sur une ligne les uns derrière les autres

```
.parent {  
  display: flex;  
  background-color: turquoise;  
}
```

```
.item {  
  background-color: wheat;  
}
```

```
<div class="parent">  
  <div class="item">Item1</div>  
  <div class="item">Item2</div>  
  <div class="item">Item3</div>  
</div>
```

Item1Item2Item3



La direction du conteneur

- ❑ Il est possible de modifier la direction principale du conteneur avec la propriété **flex-direction**
- ❑ **Par défaut**, comme vu précédemment, cette propriété vaut **row**.
- ❑ Il existe 4 valeurs possibles pour **flex-direction**.
 - row
 - row-reverse
 - column
 - column-reverse



flex-direction: row

❑ Exemple

- .parent {
 - display: flex;
 - flex-direction: row;
 - background-color: turquoise;
- }
- .item {
 - background-color: wheat;
- }

```
<div class="parent">  
  <div class="item">Item1</div>  
  <div class="item">Item2</div>  
  <div class="item">Item3</div>  
</div>
```

Item1Item2Item3



flex-direction: row-reverse

❑ Exemple

- .parent {
 - display: flex;
 - flex-direction: row-reverse;
 - background-color: turquoise;
- }
- .item {
 - background-color: wheat;
- }

```
<div class="parent">  
  <div class="item">Item1</div>  
  <div class="item">Item2</div>  
  <div class="item">Item3</div>  
</div>
```

Item3Item2Item1



flex-direction: column

❑ Exemple

- .parent {
 - display: flex;
 - flex-direction: column;
 - background-color: turquoise;
- }
- .item {
 - background-color: wheat;

```
<div class="parent">  
  <div class="item">Item1</div>  
  <div class="item">Item2</div>  
  <div class="item">Item3</div>  
</div>
```

Item1
Item2
Item3



flex-direction: column-reverse

❑ Exemple

```
.parent {  
  display: flex;  
  flex-direction: column-reverse;  
  background-color: turquoise;  
}
```

```
.item {  
  background-color: wheat;  
}
```

```
<div class="parent">  
  <div class="item">Item1</div>  
  <div class="item">Item2</div>  
  <div class="item">Item3</div>  
</div>
```

Item3
Item2
Item1



L'occupation de l'espace

- ❑ Il est possible de préciser comment les éléments se partagent l'espace du conteneur
- ❑ Pour cela il existe 2 propriétés :
 - **flex-grow** qui permet d'indiquer une proportion, avec plusieurs valeurs possibles
 - 1
 - 2
 - 0.6
 - **flex-basis** qui permet d'exprimer la taille initiale d'un élément flexible
 - auto | fill | max-content | min-content | fit-content
 - 10em | 3px
 - 33%



flex-grow

❑ Exemple 1

- .parent {
 - display: flex;
 - background-color: turquoise;
- }
- .item {
 - background-color: wheat; flex-grow: 1;
 - margin: 5px;
- }

```
<div class="parent">  
  <div class="item">Item1</div>  
  <div class="item">Item2</div>  
  <div class="item">Item3</div>  
</div>
```



flex-grow

❑ Exemple 2

- .parent {
 - `display: flex;`
`<div class="parent">`
 - `background-color: turquoise;`
`<div class="item flex1">Item1</div>`
 - } `<div class="item flex2">Item2</div>`
 - .item { `<div class="item flex1">Item3</div>`
`</div>`
 - `background-color: wheat; margin: 5px;`
 - }
- .flex1 { `flex-grow: 1;` }
- .flex2 { `flex-grow: 2;` }



flex-basis

- ❑ Comme vous avez pu le constater précédemment, les éléments positionnés avec flex-grow occupent le maximum d'espace.
- ❑ Il est possible de fixer une pour les éléments, **avec flex-basis**



flex-basis

❑ Avec une taille **en pixels**

- .parent {

❑ Exemple: flex;

- background-color: turquoise;
- }<div class="parent">
- .item {<div class="item flex1">Item1</div>
- }<div class="item flex2">Item2</div>
- background-color: wheat; margin: 5px;<div class="item flex1">Item3</div>
- }</div>
- .flex1 { flex-basis: 100px; }
- .flex2 { flex-basis: 200px; }

Item1

Item2

Item3



flex-basis

❑ Avec une taille **en %** de la largeur totale

❑ Exemple

```
<div class="parent">  
  <div class="item flex1">Item1</div>  
  <div class="item flex2">Item2</div>  
  <div class="item flex1">Item3</div>  
</div>
```

```
.parent {  
  display: flex;  
  background-color: turquoise;  
}  
.item {  
  background-color: wheat;  
  margin: 5px;  
}  
  
.flex1 { flex-basis: 15%; }  
.flex2 { flex-basis: 30%; }
```

Item1

Item2

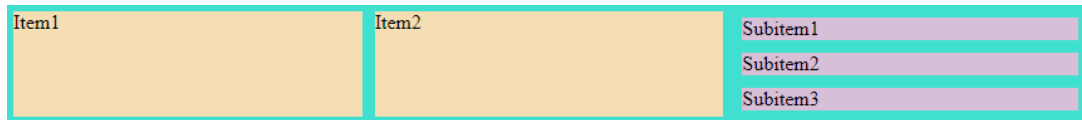
Item3



Un enfant peut également être un conteneur flexible

❑ Exemple

```
<div class="parent">
  <div class="item">Item1</div>
  <div class="item">Item2</div>
  <div class="subparent item">
    <div class="subitem">Subitem1</div>
    <div class="subitem">Subitem2</div>
    <div class="subitem">Subitem3</div>
  </div>
</div>
```

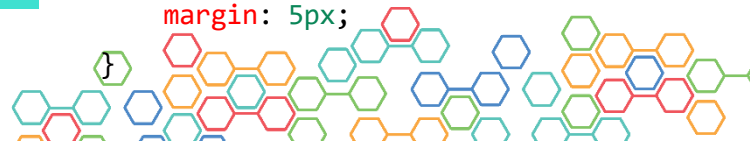


```
.parent {
  display: flex;
  background-color: turquoise;
}

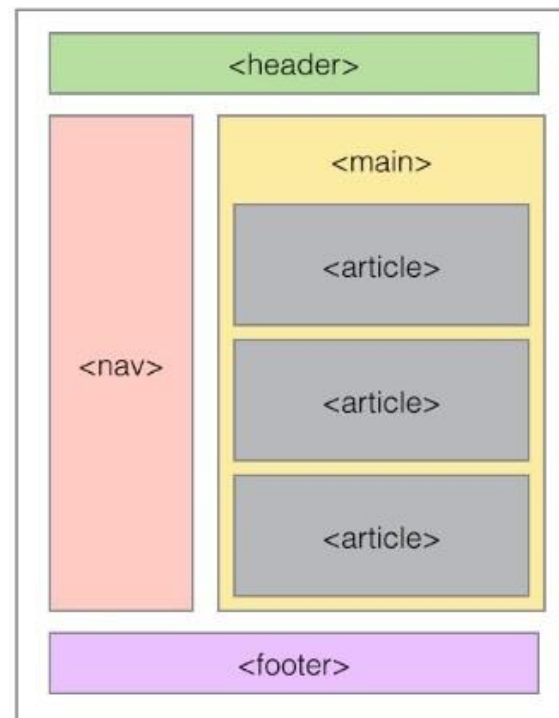
.item {
  background-color: wheat;
  margin: 5px;
  flex-basis: 32.6%;
}

.subparent {
  display: flex;
  flex-direction: column;
  background-color: turquoise;
}

.subitem {
  background-color: thistle;
  margin: 5px;
```



- OBJECTIFS :
- Utilisez ce que vous savez des flexbox
- DESCRIPTION : Dans ce TP n°4 vous allez devoir mettre
- en œuvre ce que vous savez des flexbox
- pour reproduire un layout.





Les media queries

A quoi servent les media queries ?

- ❑ Elles servent à créer des pages dites « responsive » dont le contenu s'adapte au périphérique et à la taille disponible.
- ❑ On va s'en servir pour définir des styles qui ne s'appliqueront que pour des périphériques ayant des tailles comprises entre une largeur minimum et une largeur maximum
- ❑ On peut également s'en servir pour définir des styles qui s'appliqueront uniquement dans le cas d'une impression.

```
@media print {  
    #menu, #footer, aside {  
        display: none;  
    }  
    body {  
        font-size: 120%;  
        color: black;  
    }  
}
```



Exemple de media queries

- ❑ Voici un exemple pour un écran dont la taille maximum est de 640px de largeur.
- ❑ Dans cet écriture d'une media queries on commence par préciser la nature du media, ici un écran (screen), puis avec un ET logique on va ajouter des contraintes
- ❑ On peut utiliser des AND et des OR avec si besoin un système de parenthèses

```
@media screen and (max-width: 640px) {  
  .bloc {  
    font-size: 0.8em;  
  }  
}
```



Usage des AND

- ❑ On peut utiliser des AND afin d'ajouter des contraintes

```
@media screen and (min-width: 200px) and (max-width: 640px) {  
    .bloc {  
        font-size: 0.8em;  
    }  
}  
  
@media screen and (min-width: 640px) and (max-width: 820px)  
{  
    .bloc {  
        font-size: 1em;  
    }  
}
```



Comment faire un OU ?

- ❑ Il n'y a pas d'opérateur OR, on utilise à la place la virgule

```
@media screen and (max-width: 600px), screen and (min-width: 900px)
{
    .bloc1{
        color: blueviolet;
    }
}
```



L'orientation

- ❑ Il est possible d'ajouter une contrainte pour prendre en compte l'orientation de l'écran: **portrait** ou **landscape**

```
@media (orientation: landscape) {  
    body {  
        flex-direction: row;  
    }  
}  
  
@media (orientation: portrait) {  
    body {  
        flex-direction: column;  
    }  
}
```



Media queries à l'import

- ❑ Il est possible également de préciser la media queries dans le lien d'import du fichier CSS avec l'attribut **media**
- ❑ **Avantage:** éviter des fichiers CSS trop gros avec toutes les media queries

```
<link rel="stylesheet" media="screen and (max-width:600px)" href="styles-small.css">
```

```
<link rel="stylesheet" media="screen and (min-width:600px) and (max-width:1200px)" href="styles-medium.css">
```

```
<link rel="stylesheet" media="print" href="styles-print.css">
```



Pour Version tablette:

- Il y a peu de travail pour la version tablette.
- Suivez les règles suivantes,
- à partir d'une résolution de 768px de large :
- Augmenter légèrement la taille du texte dans
- le menu, le aside et le corp de l'article
- Faire que la balise <aside> soit placé en dessous
- de l'article (attention a ce que le footer ne cache pas le contenu si la hauteur n'est pas suffisante)
- L'image dans l'article doit prendre 50% de la largeur



- Version Smartphone:
- Pour la version smartphone il reste encore
- un peu de travail. Suivez les règles suivantes, à partir de 630px de large
- Le menu doit disparaître
- L'image de l'article doit prendre toute la largeur.
- Le texte en dessous de l'article doit disparaître

