

Module

Intégration web

Zeineb EL KHALFI

zeineb.EL-KHALFI@univ-amu.fr

Plan

1. Bases en CSS
2. Le modèles des boites
3. Gestion de fond
4. CSS avancé
5. Responsives design

pourquoi on a besoin des feuilles de style

- ▶ HTML a des **limitations** importantes :
 - Mise en forme de la présentation :
 - Mise en page difficile
 - Limitation des possibilités
 - Maintenance d'un site :
 - Fastidieuse
 - Prend beaucoup de temps
 - A l'origine de l'arrêt de beaucoup de sites !
- ▶ Ceci vient du mélange de données de différentes natures
 - les balises servent à structurer le document (paragraphe, etc.),
 - les balises servent comme des objets (image, etc.),
 - les balises sont utilisées pour la mise en forme. (fonte, etc.),

séparation de la présentation et des données

pourquoi on a besoin des feuilles de style

```
<html>
<head>
<title> Exemple </title>
</head>
<body>
<h1 align="center">
<font color="blue" face="arial">
<b>Ceci est un contenu bleu, en police arial, gras, centré</b>.
</font>
</h1>
<h1 align="center">
<font color="blue" face="arial">
<b>Ceci est deuxième contenu bleu, en police arial, gras, centré</b>.
</font>
</h1>
<h1 align="center">
<font color="blue" face="arial">
<b>Ceci est un troisième contenu bleu, en police arial, gras, centré</b>.
</font>
</h1>
</body>
</html>
```

Beaucoup d'information redondante, ce qui provoque :

- Des difficultés pour les mises à jours (il faut changer partout)
- Et surtout des risques d'erreurs (e.g oublier de changer une...)

Le CSS, ou Cascading StyleSheets

- ▶ CSS : feuille de style en cascade
- ▶ **Objectif :**
 - *Le CSS vient compléter le code HTML pour donner au contenu l'**apparence** que l'on souhaite.
 - *contrôle : couleur de texte, image de fond, style de police, menu à gauche ou à droite, les bordures, les marges, les transitions...
 - *rendre cohérents et homogènes les sites Web et faciliter leur maintenance.
 - séparation entre contenu et propriétés visuelles
 - un fichier HTML + une feuille de style.
 - toute modification d'une feuille de style est répercutée sur l'ensemble des pages où elle s'applique.
- ▶ **Comment ?**
 - Un ensemble de propriétés de mise en forme
 - +
 - un mécanisme qui permet de spécifier à quelle partie du document il faut les appliquer

Un peu d'histoire

- ▶ Aux débuts du Web, CSS n'existait pas. Il n'y avait initialement que le langage HTML.
- ▶ Le HTML est né en 1991 et CSS en 1996.
- ▶ Avant, on utilisait des balises de présentation directement dans le HTML
Exemple : ``
- ▶ Les pages HTML commençaient à devenir assez complexes.
→ C'est pour cela que l'on a créé le langage CSS.
- ▶ le CSS n'a pas été adopté immédiatement par les web-masters. Il fallait se défaire de certaines mauvaises habitudes et cela a pris du temps.
- ▶ Encore aujourd'hui, on peut trouver des sites web avec des balises HTML de mise en forme, anciennes et obsolètes, comme `` !
 - ▶ **CSS1**, **CSS2**, aujourd'hui la version finalisée est **CSS2.1**
 - ▶ **CSS3** : est en cours de maturation, certaines propriétés sont finalisées,
 - ▶ Les **CSS3** facilitent grandement la vie pour créer des ombres, arrondir les coins, ou encore adapter ses pages aussi bien aux grands écrans qu'aux smartphones.

Un peu d'histoire

- ▶ Aux débuts du Web, CSS n'existait pas. Il n'y avait initialement que le langage HTML.
- ▶ Le HTML est né en 1991 et CSS en 1996.
- ▶ Avant, on utilisait des balises de présentation directement dans le HTML
Exemple : ``
- ▶ Les pages HTML commençaient à devenir assez complexes.
→ C'est pour cela que l'on a créé le langage CSS.
- ▶ le CSS n'a pas été adopté immédiatement par les web-masters. Il fallait se défaire de certaines mauvaises habitudes et cela a pris du temps.
- ▶ Encore aujourd'hui, on peut trouver des sites web avec des balises HTML de mise en forme, anciennes et obsolètes, comme `` !
 - ▶ **CSS1**, **CSS2**, aujourd'hui la version finalisée est **CSS2.1**
 - ▶ **CSS3** : est en cours de maturation, certaines propriétés sont finalisées,
 - ▶ Les **CSS3** facilitent grandement la vie pour créer des ombres, arrondir les coins, ou encore adapter ses pages aussi bien aux grands écrans qu'aux smartphones.

Où appliquer un style ?

- ▶ **Le CSS “en ligne” dans la balise HTML avec l’attribut « style »**

- ▶ Attribut style = ...;

- « Je veux des titres roses » : `<h1 style="color:pink;" >`

- « des sous titres avec une couleur de fond bleu »: `<h2 style="backgroundcolor: blue;">`

- « des paragraphes en Arial »: `<p style="font-family:Arial;" >`

```
<html>
<head>
<title> Exemple </title>
</head>
<body>
<h1 style="color:blue;"> Bonjour, je suis le titre h1 en bleu </h1>
<h2 style="background-color: pink;"> Bonjour, je suis le titre h2 et mon arrière-plan est en rose </h2>
<p style="font-family: Arial;" > Bonjour, je suis un paragraphe en Arial <br/> <br/>
  Bravo: vous commencez à apprendre le CSS, continuez sur votre lancée ! </p>
</body>
</html>
```

Bonjour, je suis le titre h1 en bleu

Bonjour, je suis le titre h2 et mon arrière-plan est en rose

Bonjour, je suis un paragraphe en Arial

Bravo: vous commencer à apprendre le CSS, continuez sur votre lancée !

Intégration web

Où appliquer un style ?

- ▶ Le CSS “en ligne” dans la balise HTML → Problèmes de maintenabilité
- ▶ Et si je veux appliquer un style à tous les titres ?
- ▶ Et si je veux changer tous les titres d’un coup ?
- ▶ Long et fastidieux

```
<html>
<head>
<title> Exemple </title>
</head>
<body>
<h1 style="color:blue;"> Bonjour, je suis le titre h1 en bleu </h1>
<h2 style="background-color: pink;"> Bonjour, je suis le titre h2 et mon arrière-plan est en rose </h2>
<p style="font-family: Arial;" > Bonjour, je suis un paragraphe en Arial <br/> <br/>
  Bravo: vous commencez à apprendre le CSS, continuez sur votre lancée ! </p>
</body>
</html>
```

Bonjour, je suis le titre h1 en bleu

Bonjour, je suis le titre h2 et mon arrière-plan est en rose

Bonjour, je suis un paragraphe en Arial

Bravo: vous commencer à apprendre le CSS, continuez sur votre lancée !

Intégration web

Où appliquer un style ?

- ▶ **Le CSS « interne » dans l'entête du HTML**
- ▶ On place une balise `<style>` dans la balise `<head>` du document
 - Changer la couleur de tous les titres h2 en vert :

```
<html>
<head>
<title> Exemple </title>
<style type="text/css">
h2{
color: green;
}
</style>
</head>
<body>
<h1 style="color:blue;"> Bonjour, je suis le titre h1 en bleu </h1>
<h2 style="background-color: pink;"> Bonjour, je suis le titre h2 et mon arrière-plan est en rose </h2>
<p style="font-family: Arial;" > Bonjour, je suis un paragraphe en Arial <br/> <br/>
  Bravo: vous commencez à apprendre le CSS, continuez sur votre lancée ! </p>
</body>
</html>
```

Bonjour, je suis le titre h1 en bleu

Bonjour, je suis le titre h2 et mon arrière-plan est en rose

Bonjour, je suis un paragraphe en Arial

Bravo: vous commencez à apprendre le CSS, continuez sur votre lancée !
Intégration web

Où appliquer un style ?

- ▶ **Le CSS « interne » dans l'entête du HTML**
- ▶ On place une balise `<style>` dans la balise `<head>` du document
- ▶ Avantage :
 - Possibilité d'appliquer une règle sur tous les mêmes éléments d'un même document
- ▶ Inconvénient :
 - Ne s'applique qu'à ce document, et non au site en entier
 - Il faut donc modifier le CSS sur chaque page
(ex : si on a 10 pages, modifier les styles sur les 10 fichiers)
- MAIS certains plugins de CMS (WordPress par exemple) mal conçus l'utilisent encore.

Où appliquer un style ?

- ▶ **La feuille de style externe**
- ▶ Nous utiliserons les mêmes conventions de chemin que pour les images et pages
- ▶ Création d'un nouveau fichier appelé « styles.css » (notez l'extension **.css** ici) et on lie la feuille de styles au HTML
- ▶ On « lie » la feuille de style au document HTML avec une balise <link> dans le <head>

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

Où appliquer un style ?

► La/les feuille de style externe

Index.html

```
<html>
<head>
<title> Exemple </title>
<link rel="stylesheet" type="text/css" href="styles.css"/>
</head>
<body>
<h1 style="color:blue;"> Bonjour, je suis le titre h1 en bleu </h1>
<h2 style="background-color: pink;"> Bonjour, je suis le titre h2 et mon arrière-plan est en rose </h2>
<p style="font-family: Arial;" > Bonjour, je suis un paragraphe en Arial <br/> <br/>
  Bravo: vous commencez à apprendre le CSS, continuez sur votre lancée ! </p>
</body>
</html>
```

styles.css

```
p {
background: yellow;
}
```

Bonjour, je suis le titre h1 en bleu

Bonjour, je suis le titre h2 et mon arrière-plan est en rose

Bonjour, je suis un paragraphe en Arial

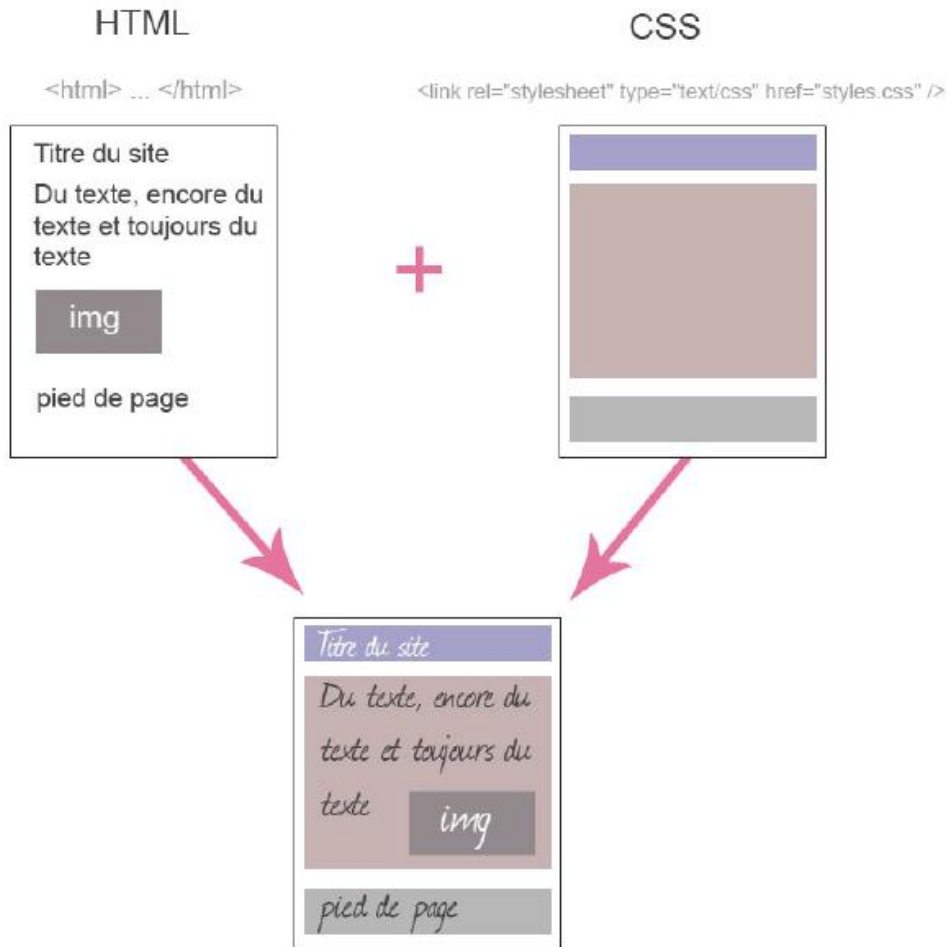
Bravo: vous commencez à apprendre le CSS, continuez sur votre lancée !

Où appliquer un style ?

► La feuille

► Avantages

- Sécurité
- d'avenir
- Portabilité
- (d'aujourd'hui)
- Coût



présentation : possibilité

de créer plusieurs pages

pour les pages futures.

La syntaxe de base

- ▶ La syntaxe est composée de 3 éléments :
 - Le **sélecteur** est l'élément sur lequel on applique les propriétés (ex balise HTML, id, etc.)
 - La **propriété** est l'effet que l'on va vouloir donner (ex couleur de texte, positionnement, couleur de fond, etc.)
 - La **valeur** de la propriété CSS (ex rouge, 10px, etc.)

sélecteur {

propriété: valeur;

propriété: valeur;

...

}

une déclaration CSS = **propriété + valeur** .

On peut en avoir plusieurs pour un même élément et elles sont séparées par le symbole point virgule

Les propriétés du CSS

- ▶ Syntaxe d'une propriété:

<Nom-de-la-propriété> : <valeur> ;
color : red;

- ▶ Le nom des propriétés est à connaître. Il en existe plusieurs :

- Gestion du texte, police, fonte, taille,
- Positionnement, alignement,
- ...

- ▶ Les valeurs dépendent des propriétés

- Peuvent être du texte **color : red;**
- Valeur numérique absolue ou relative
 - **margin-left : 100px;** px tient pour pixel (absolue)
 - **margin-left : 2em;** em est la hauteur d'une ligne (dépend du navigateur)
- Une série de valeurs
 - **Background : red url(image.jpg) left center;**

Exemple de Propriété : sur le texte

▶ Polices de caractères

- Font-family (arial, times...)
- Font-size
 - 12px
 - xx-smal... xx-large
 - Smaller, larger, 50%
- Font-style (italic, normal...)
- font-weight

▶ Divers

- Text-align
- Text-decoration (underline, none, overline...)
- Line-height (X em, X px)
- Vertical-align

Le sélecteur usuel d'élément HTML

- ▶ Syntaxe:

descriptif-d'un-élément-HTML {liste de propriété et leurs valeurs}

- ▶ On peut sélectionner n'importe quel élément HTML et lui appliquer un style.
- ▶ Une propriété appliquée sur un élément HTML, s'applique par défaut à tous les éléments présents dans le document HTML.

```
p {  
  color : blue;  
}
```

=> Tous les paragraphes auront une couleur de texte bleu

Exercice 1

Un titre sur un fond jaune

Un sous titre en bleu

La couleur du texte de ce paragraphe est le rouge et la police est Verdana.

```
<html>
<head>
<title> TD1 </title>
</head>
<body bgcolor="PowderBlue">
<h1 style="background-color:yellow"> Un titre sur un fond jaune </h1>

<h2 style="color:blue;"> Un sous titre en bleu </h2>

<p style="color:red; font-family:Verdana;"> La couleur du texte de ce paragraphe est
le rouge et la police est Verdana. </p>

</body>
</html>
```

Exercice 1

Un titre sur un fond jaune

Un sous titre en bleu

La couleur du texte de ce paragraphe est le rouge et la police est Verdana.

```
<html>
<head>
<title> TD1 </title>
<style type="text/css">
body{
background-color:PowderBlue;
}
h1{
background-color:yellow;
}
h2{
color:blue;
}
p{
color:red;
font-family: Verdana;
}
</style>
</head>
<body>
<h1> Un titre sur un fond jaune </h1>

<h2> Un sous titre en bleu </h2>

<p> La couleur du texte de ce paragraphe est
le rouge et la police est Verdana. </p>

</body>
</html>
```

Exercice 1

Un titre sur un fond jaune

Un sous titre en bleu

La couleur du texte de ce paragraphe est le rouge et la police est Verdana.

```
TD1.css x
1  body{
2    background-color:PowderBlue;
3  }
4  h1{
5    background-color:yellow;
6  }
7  h2{
8    color:blue;
9  }
10 p{
11  color:red;
12  font-family: Verdana;
13 }
```

```
<html>
<head>
<title> TD1 </title>
<link rel="stylesheet" type="text/css" href="TD1.css"/>
</style>
</head>
<body>
<h1> Un titre sur un fond jaune </h1>
<h2> Un sous titre en bleu </h2>
<p> La couleur du texte de ce paragraphe est
le rouge et la police est Verdana. </p>
</body>
</html>
```

Exercice 2

comprendre et utiliser un fichier CSS

Objectif: Nous allons nous entraîner sur cette page pour apprendre à changer l'affichage de certains éléments

Créer et intégrer un fichier styleTD.css :

choisir une image pour l'arrière plan

titres de niveau 1

couleur: DarkRed
alignement horizontal : centré
capitales: majuscules

la couleur des titres de niveau 2 est : rouge

texte des paragraphes:

gras et italique
police : Arial Black
couleur : DarkBlue

```
<html>
<head>
<title> TD1 </title>
</head>
<body>
<h1> comprendre et utiliser un fichier CSS </h1>

<p> Objectif: Nous allons nous entraîner sur cette page pour
apprendre à changer l'affichage de certains éléments </p>
<br/>
<p> Créer et intégrer un fichier styleTD.css :</p>

<h2>choisir une image pour l'arrière plan</h2>

<h2>titres de niveau 1</h2>
<p>
couleur: DarkRed <br/>
alignement horizontal : centré <br/>
capitales: majuscules
</p>

<h2> la couleur des titres de niveau 2 est : rouge</h2>

<h2> texte des paragraphes:</h2>
<p>
gras et italique <br/>
police : Arial Black<br/>
couleur : DarkBlue <br/>
</p>
</body>
</html>
```

Exercice 2

```
<html>
<head>
<title> TD1 </title>
<link rel="stylesheet" type="text/css" href="styleTD.css"/>
</head>
<body>
<h1> comprendre et utiliser un fichier CSS </h1>

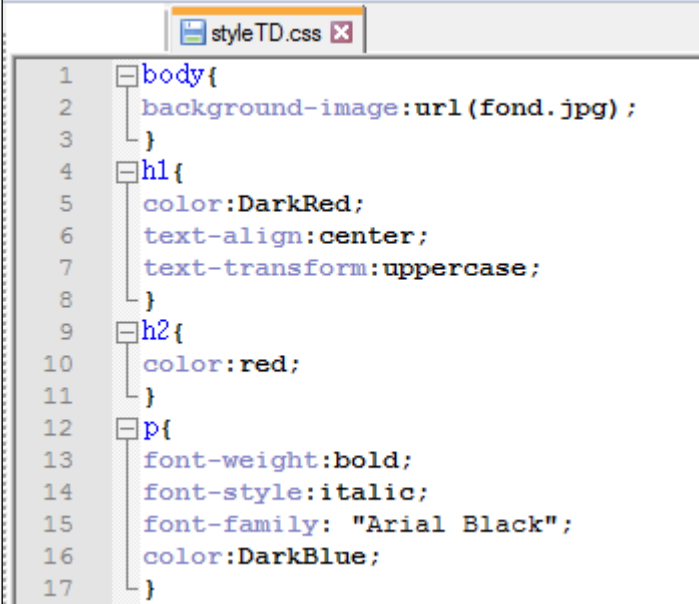
<p> Objectif: Nous allons nous entraîner sur cette page pour
apprendre à changer l'affichage de certains éléments </p>
<br/>
<p> Créer et intégrer un fichier styleTD.css :</p>

<h2>choisir une image pour l'arrière plan</h2>

<h2>titres de niveau 1</h2>
<p>
couleur: DarkRed <br/>
alignement horizontal : centré <br/>
capitales: majuscules
</p>

<h2> la couleur des titres de niveau 2 est : rouge</h2>

<h2> texte des paragraphes:</h2>
<p>
gras et italique <br/>
police : Arial Black<br/>
couleur : DarkBlue <br/>
</p>
</body>
</html>
```



```
styleTD.css x
1  body{
2     background-image:url (fond.jpg) ;
3  }
4  h1{
5     color:DarkRed;
6     text-align:center;
7     text-transform:uppercase;
8  }
9  h2{
10    color:red;
11  }
12  p{
13    font-weight:bold;
14    font-style:italic;
15    font-family: "Arial Black";
16    color:DarkBlue;
17  }
```

Exercice 2

comprendre et utiliser un fichier CSS

Objectif: Nous allons nous entraîner sur cette page pour apprendre à changer l'affichage de certains éléments

Créer et intégrer un fichier styleTD.css :

choisir une image pour l'arrière plan

titres de niveau 1

couleur: DarkRed
alignement horizontal : centré
capitales: majuscules

la couleur des titres de niveau 2 est : rouge

texte des paragraphes:

gras et italique
police : Arial Black
couleur : DarkBlue

COMPRENDRE ET UTILISER UN FICHER CSS

Objectif: Nous allons nous entraîner sur cette page pour apprendre à changer l'affichage de certains éléments

Créer et intégrer un fichier styleTD.css :

choisir une image pour l'arrière plan

titres de niveau 1

*couleur: DarkRed
alignement horizontal : centré
capitales: majuscules*

la couleur des titres de niveau 2 est : rouge

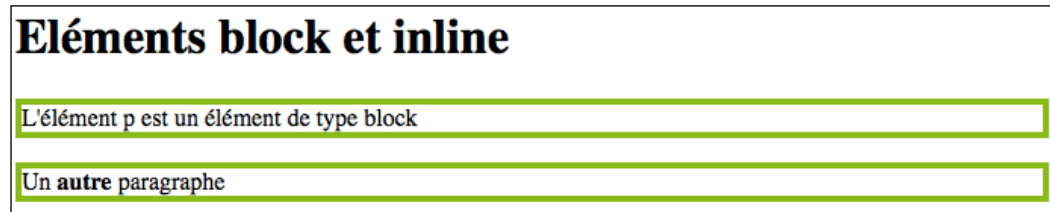
texte des paragraphes:

*gras et italique
police : Arial Black
couleur : DarkBlue*

Les types d'éléments HTML

1. Les éléments de type block

- ▶ Va toujours commencer sur une nouvelle ligne et prendre toute la largeur disponible dans la page (ex: p, div, h1, h2, ol, ul, form).
- ▶ De plus, un élément de type block peut contenir d'autres éléments de type block ainsi que des éléments de type inline.



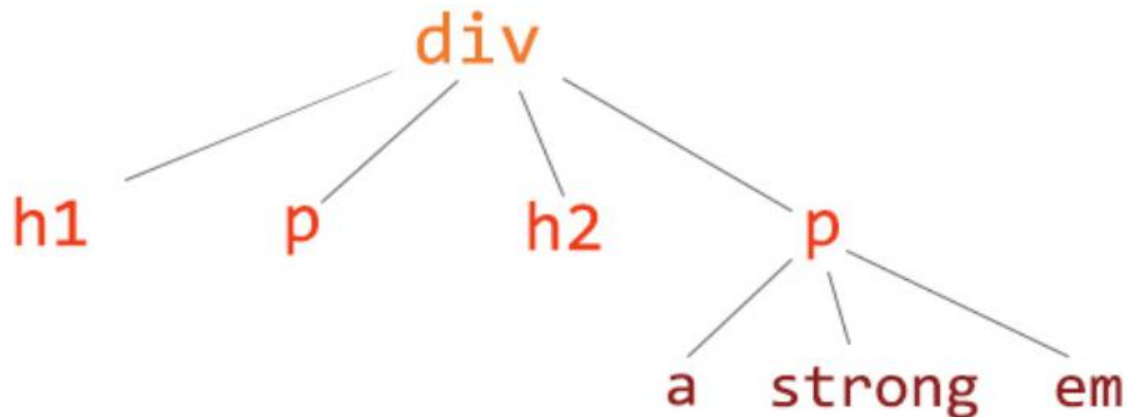
2. Les éléments de type inline

- ▶ Va toujours s'insérer dans la ligne actuelle (a, strong, img, span).
- ▶ Les éléments de type inline prennent uniquement la largeur qui leur est nécessaire (c'est-à-dire la largeur de leur contenu).



Autres types de sélecteurs

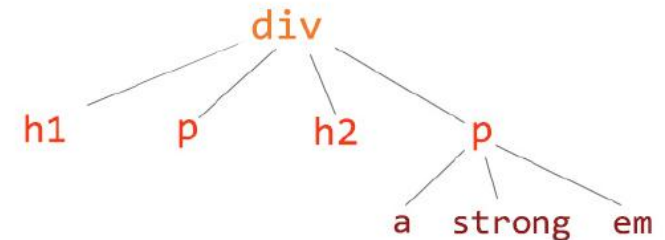
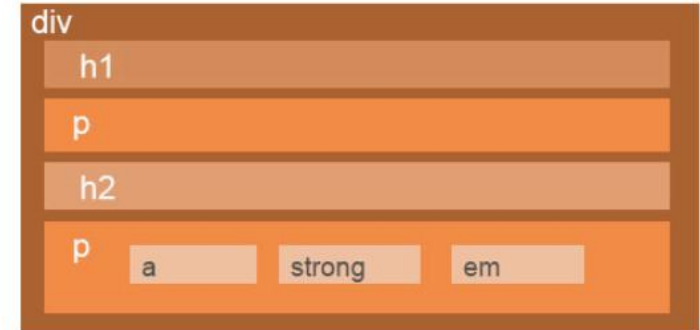
```
<div>
  <h1>Cupcake ipsum dolor sit</h1>
  <p>
    amet jelly sugar plum bear claw bonbon. Wypas dragée chocolate bar fruitcake. Gingerbread
    wypas muffin cotton candy toffee sugar plum I love danish icing.
  </p>
  <h2>Ice cream topping</h2>
  <p>
    <a href="#">Toffee gummi bears candy canes tiramisu bear</a>
    claw candy canes chocolate bar candy.
    <strong>Ice cream topping apple pie</strong>Cotton candy jelly-o gummies.
    <em>I love lollipop I love jelly</em>
  </p>
</div>
```



Autres types de sélecteurs

1. Notion d'enfant et de descendance

- h1, p, h2, p sont **enfants** de div
- a, strong et em sont **enfants** du p dans lequel ils sont contenus (mais pas de l'autre p)
- a, strong et em (et h1, p, h2 et p) sont **descendants** de div



2. Notion de parent et d'ancêtre

- div est **parent** de h1,p,h2,p
- Le 2ème p est **parent** de a, strong et em.
- div est **ancêtre** de a, strong et em (et de h1, p h2 et p)

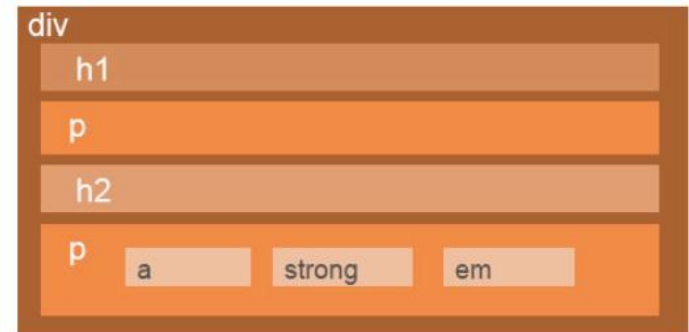
Autres types de sélecteurs

1. Sélecteur de hiérarchie

- Pour sélectionner le **a** descendant de **p**:

```
p a {  
  ...  
}
```

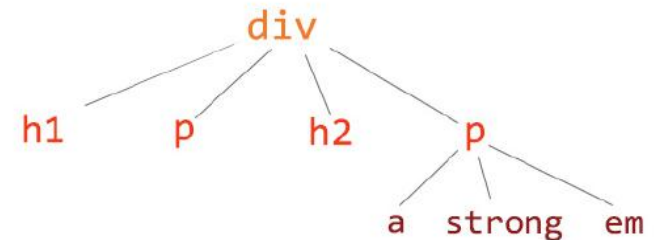
- notez l'espace entre le p et le a



2. Sélecteur de groupe

- Pour sélectionner **plusieurs éléments** et leur appliquer la **même valeur**, on les sépare par **une virgule**.

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



- Va donner la couleur rouge à tous les h1 ou h2

Les sélecteurs de classe

- **class=" "** est un attribut qui peut se mettre sur n'importe quelle balise
- Il permet de cibler de manière plus « spécifique » certains éléments

`<p class="important">` Un paragraphe spécifique qui est mis en avant `</p>`

Un <paragraphe>

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Lorem ipsum quia dolor sit amet



Les sélecteurs de classe

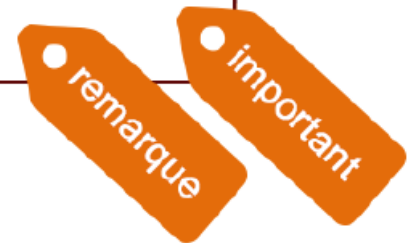
Classes multiples

- Il est possible d'avoir plusieurs classes sur un élément, on les sépare d'un espace.

`<p class="remarque important"> Une remarque mis en avant </p>`

Un <paragraphe>

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Lorem ipsum quia dolor sit amet



Les sélecteurs de classe

Classes multiples

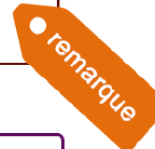
- Une même classe peut être utilisée sur plusieurs balises dans le même document

`<p class="remarque"> une remarque paragraphe</p>`

`<blockquote class="remarque"> une remarque citation</blockquote>`

Un `<paragraphe>`

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Lorem ipsum quia dolor sit amet



Un `<blockquote>`

Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Lorem ipsum quia dolor sit amet, consectetur, adipisci velit, sed quia non numquam eius modi tempora incidunt ut labore et dolore magnam aliquam quaerat voluptatem.



Cibler une classe indépendamment de la balise

- Si on veut cibler une classe, sans se préoccuper de la balise on utilise le « point ».

```
.nomdeclasse {  
  ...  
}
```

Par exemple `.remarque { ... }` va cibler

```
<p class="remarque"> ... </p>
```

```
<blockquote class="remarque"> .... </blockquote>
```

Une même classe peut être utilisée sur plusieurs balises HTML

Cibler une balise HTML avec une classe

- Pour cibler un élément doté d'une classe en CSS on procède de la manière suivante (notez qu'il n'y a pas d'espace)

element.maclasse { ... }

Par exemple : **p.remarque** { ... } va cibler tous les éléments comme celui-ci:

<p class="remarque"> ... </p>

L'id (identifiant)

- Il a le même rôle qu'une classe, mais doit être unique sur la page (donc on va moins l'utiliser)
- Un seul attribut id par balise est possible
- On le note **<element id="monid">**
- On le cible en CSS avec

```
#monid {  
  ...  
}
```

Espace ou pas ?

- Pas d'espace signifie que toutes les conditions **DOIVENT** être vraies

p.remarque{ ... } cible tous les paragraphes **ET** qui ont une classe «**remarque**»:

<p class=" remarque "> ... </p>

Espace ou pas ?

- Un espace entre deux « sélecteurs » détermine la généalogie :

p **.remarque**{ ... } cible tous les éléments qui ont pour classe «**remarque**» et qui sont descendants d'un paragraphe

<p> **** ... **** **</p>**

<p> **<h1 class=" remarque ">** ... **</h1>** **</p>**

<p> **** ... **** **</p>**

...

Convention de nommage

- Un nom de classe ou d'ID ne doit pas avoir d'accent ou caractères spéciaux
- On a tendance à éviter le « tout majuscule »
- Un nom d'ID ne prend jamais d'espace
- Un nom de classe ne prend pas d'espace, sinon c'est une nouvelle classe

Les éléments HTML: DIV et SPAN

1. L'élément HTML div

- ▶ est un élément de type block,
- ▶ Il est très commun d'attribuer un attribut class à un élément div afin de pouvoir le cibler plus facilement.

2. L'élément HTML span

- ▶ est un élément de type inline.
 - ▶ souvent servir de conteneur pour du contenu textuel: pratique pour mettre en forme une portion de texte à l'intérieur d'un élément sans pouvoir la cibler de façon simple.
- souvent utilisé avec un attribut class afin de pouvoir le cibler plus facilement en CSS.

Les sélecteurs

CSS	HTML ciblé
<code>p {}</code>	<code><p> ... </p></code>
<code>.maclasse {}</code>	<code><element class="maclasse"> ... </element></code>
<code>#monid {}</code>	<code><element id="monid"> ... </element></code>
<code>p.maclasse {}</code>	<code><p class="maclasse"> ... </p></code>
<code>p .maclasse {}</code>	<code><p> <element class="maclasse"> ... </element> </p></code>
<code>p, div {}</code>	<code><p> ... <p> <div> ... </div></code>

sélecteur	HTML
<code>.info { }</code>	<code><p id="info">C'est une histoire magnifique.</p></code>
<code>#info { }</code>	<code><p class="principal">Vraiment, crois-moi, c'est bien.</p></code>
<code>#principal { }</code>	<code><h1 class="info">La passionnante histoire</h1></code>
<code>.principal { }</code>	<code><h2 id="principal">L'histoire</h2></code>

- ▶ Ecrivez le sélecteur et le style donnant une couleur rouge et un fond noir aux éléments h1 et h3
- ▶ Ecrivez le sélecteur afin que tous les éléments ayant un attribut id « partI » soient écrits en rouge
- ▶ <https://flukeout.github.io/>

Les Pseudo-classes

- ▶ Permettent d'affecter un style à des éléments qui ne peuvent pas être représentés par un sélecteur habituel
- ▶ Il n'en existe pas encore beaucoup
- ▶ Usage :
 - Couleur des liens
 - Apparence de la première ligne d'un paragraphe
 - Apparence de la lettrine (première lettre d'un paragraphe)

Les Pseudo-classes

- ▶ Liés aux liens (balise `<a>`) :
 - **:active** : lien actuellement sélectionné
 - **:link** : lien non encore visité
 - **:visited** : lien déjà visité
 - **:hover** : lien survolé par la souris
 - Utilisation :
 - **a:link** {propriété 1; propriété 2; ...}
 - **a:visited** {propriété 1; propriété 2; ...}

- ▶ Les autres pseudo-éléments :
 - **:first-letter** :Apparence de la première lettre d'un paragraphe (lettrine)
 - **:first-line** :Apparence de la première ligne d'un paragraphe
 - Utilisation
 - **p:first-line** {propriété 1; propriété 2}
 - **p.classe l:first-letter** {propriété 3; propriété 4}

Les autres Pseudo-classes

- ▶ **:hover** peut être appliqué sur d'autres éléments que des liens.
- ▶ Exemple:
 - ▶ on peut écrire **p:hover** pour créer un effet au survol d'un paragraphe
 - ▶ [Subtle Hover effects](#)
- ▶ **:focus** appliqué sur les liens, boutons, ou inputs lorsqu'ils « ont le focus »



```
input:focus{  
background: #66FF33;  
}
```

Exercice

- ▶ Dans le fichier **index.html**, ajoutez trois liens entre les balises div. Ils peuvent pointer vers le site de votre choix.
 - ▶ Dans le fichier **style.css**, ajoutez les 3 sélecteurs de pseudo-classe pour les liens:
 - ▶ **a:link** : un lien non cliqué
 - ▶ **a:visited** : un lien qui a été cliqué
 - ▶ **a:hover** : un lien survolé par la souris et non cliqué
- Ajoutez à chaque sélecteur les règles CSS permettant de :
- ▶ Appliquez une couleur
 - ▶ Appliquez une police avec différentes styles
 - ▶ Appliquez les propriétés font-style et font-weight
 - ▶ Ajoutez la propriété text-decoration et sa valeur none

Index.html

```
<html>
  <head>
    <link type="text/css" rel="stylesheet" href="style.css" />
    <title>Liens</title>
  </head>
  <body>
    <div>
      </div>
    </div>
  </body>
</html>
```

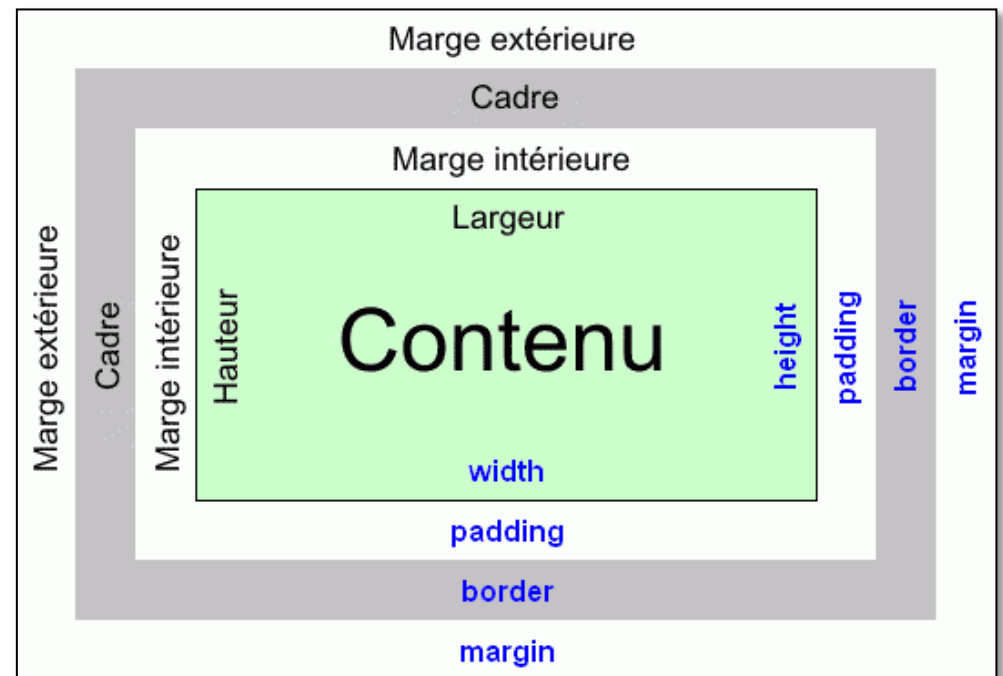
2. Le modèle des boîtes

- ▶ Une page web peut être vue comme une succession et un empilement de boîtes, qu'on appelle « blocs ».
- ▶ nous allons apprendre à manipuler ces blocs comme de véritables boîtes.
- ▶ Nous allons leur donner des dimensions, les agencer en jouant sur leurs marges, mais aussi apprendre à gérer leur contenu...

```
<h1>Titre (block)</h1>
```

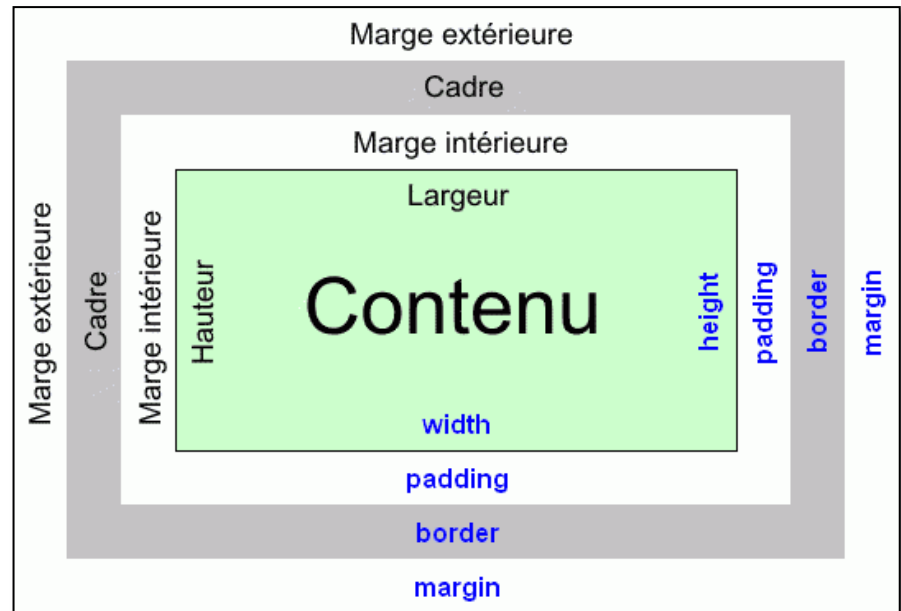
```
<p>Paragraphe blablabla blablabla  
blablabla <a>Lien inline</a> blabla  
blablabla et toujours blabla </p>
```

```
<p>Encore un paragraphe (block) </p>
```



Gestion des dimensions

- ▶ Il est possible d'établir la hauteur et la largeur des balises.
- ▶ Plusieurs unités sont utilisables :
 - ▶ px : pixels.
 - ▶ cm et in : Centimètres et inches.
 - ▶ en pourcentage (%)



- ▶ height : pour modifier la hauteur par défaut d'un élément.
- ▶ width : pour modifier la largeur par défaut d'un élément, nous allons utiliser la propriété.
- ▶ Ces propriétés peuvent prendre des valeurs absolues, relatives ou la valeur « auto ».

Gestion des bordures

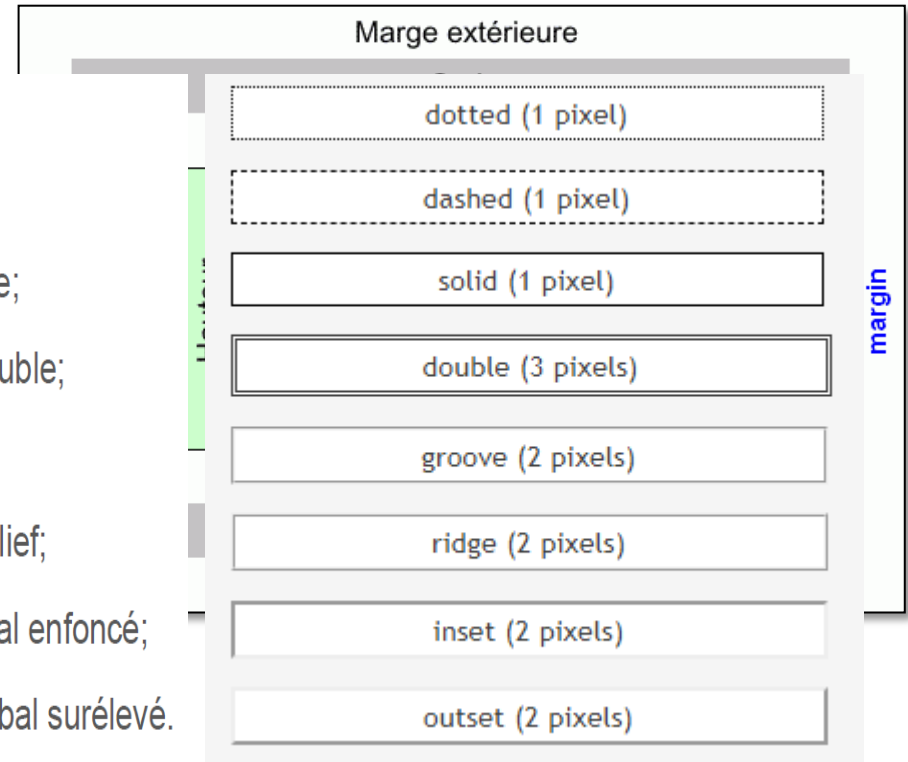
- ▶ Définition des bordures d'un élément de différentes manières en CSS :
 - ▶ en utilisant les 3 propriétés **border-width** , **border-style**, et **border-color**.
 - ▶ en utilisant directement la notation courte **border**:

border: width style color;

- **dotted** : pointillés;
- **dashed** : tirets;
- **solid** : un trait simple;
- **double** : bordure double;
- **groove** : en relief;
- **ridge** : autre effet relief;
- **inset** : effet 3D global enfoncé;
- **outset** : effet 3D global surélevé.

```
P {  
border-width: 1px ;  
border-style: solid ;  
border-color: black ;  
}
```

```
P {  
border: 1px solid black ;  
}
```



Gestion des bordures

- ▶ On peut décomposer les 4 bordures de la manière suivante :
 - ▶ `border-top: ... ;`
 - ▶ `border-left: ... ;`
 - ▶ `border-right: ... ;`
 - ▶ `border-bottom: ... ;`
- ▶ Si on ne précise pas à quelle bordure on applique la propriété, elle s'applique aux 4.
- ▶ **`border-radius : valeur ;`** permet de faire des bords arrondis.
- ▶ Valeur : une valeur en unité de mesure (px)
 - Le même arrondi partout : **`border-radius : 10px;`**
 - Un arrondi par angle (en haut à gauche, en haut à droite, en bas à droite, en bas à gauche) :
`border-radius : 2px 20px 5px 10px;`



Gestion des bordures: exercice 1

Dans le fichier [style.css](#), attribuez à tous les éléments html une bordure de 1px en trait plein avec la couleur rouge.

```
<html>
  <head>
    <link type="text/css" rel="stylesheet" href="style.css" />
    <title>Le voir pour le croire</title>
  </head>
  <body>
    <div>
      <h3>Illustration du modèle de boîte en CSS</h3>
      
      <p>IUT d'Arles: Centre international et universitaire de la culture de l'image</p>
    </div>
  </body>
</html>
```

Gestion des bordures: exercice 2

Dans le fichier style.css, définissez la bordure de l'élément div en spécifiant une bordure haute de 1px, une bordure droite de 2px, une bordure basse de 3px et une bordure gauche de 4px. Chaque bordure sera en trait double et avec la couleur bleu;

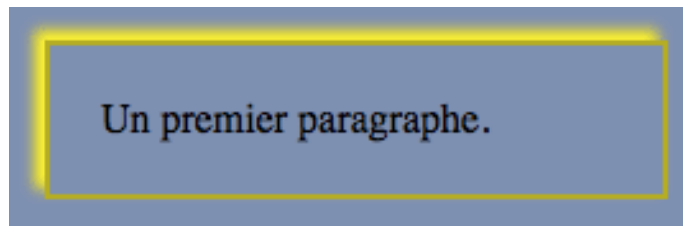
```
<html>
  <head>
    <link type="text/css" rel="stylesheet" href="style.css" />
    <title>Les bordures</title>
  </head>
  <body>
    <div>
      <p>Je suis un paragraphe.</p>
      <p>Je suis un autre paragrpahe.</p>
    </div>
  </body>
</html>
```

Les ombres

- ▶ **box-shadow** : les ombres des boîtes
- ▶ La propriété box-shadow s'applique à tout le bloc et prend quatre valeurs dans l'ordre suivant :
 - ▶ le décalage horizontal (vers la droite ou la gauche) de l'ombre ;
 - ▶ le décalage vertical (vers le bas ou le haut) de l'ombre ;
 - ▶ l'adoucissement du dégradé ;
 - ▶ la couleur de l'ombre.

Pour créer des ombres intérieures (à l'intérieur des bordures d'un élément) il faut rajouter une cinquième valeur à box-shadow : la valeur « inset ».

```
box-shadow: -4px -4px 5px yellow;
```



Les ombres

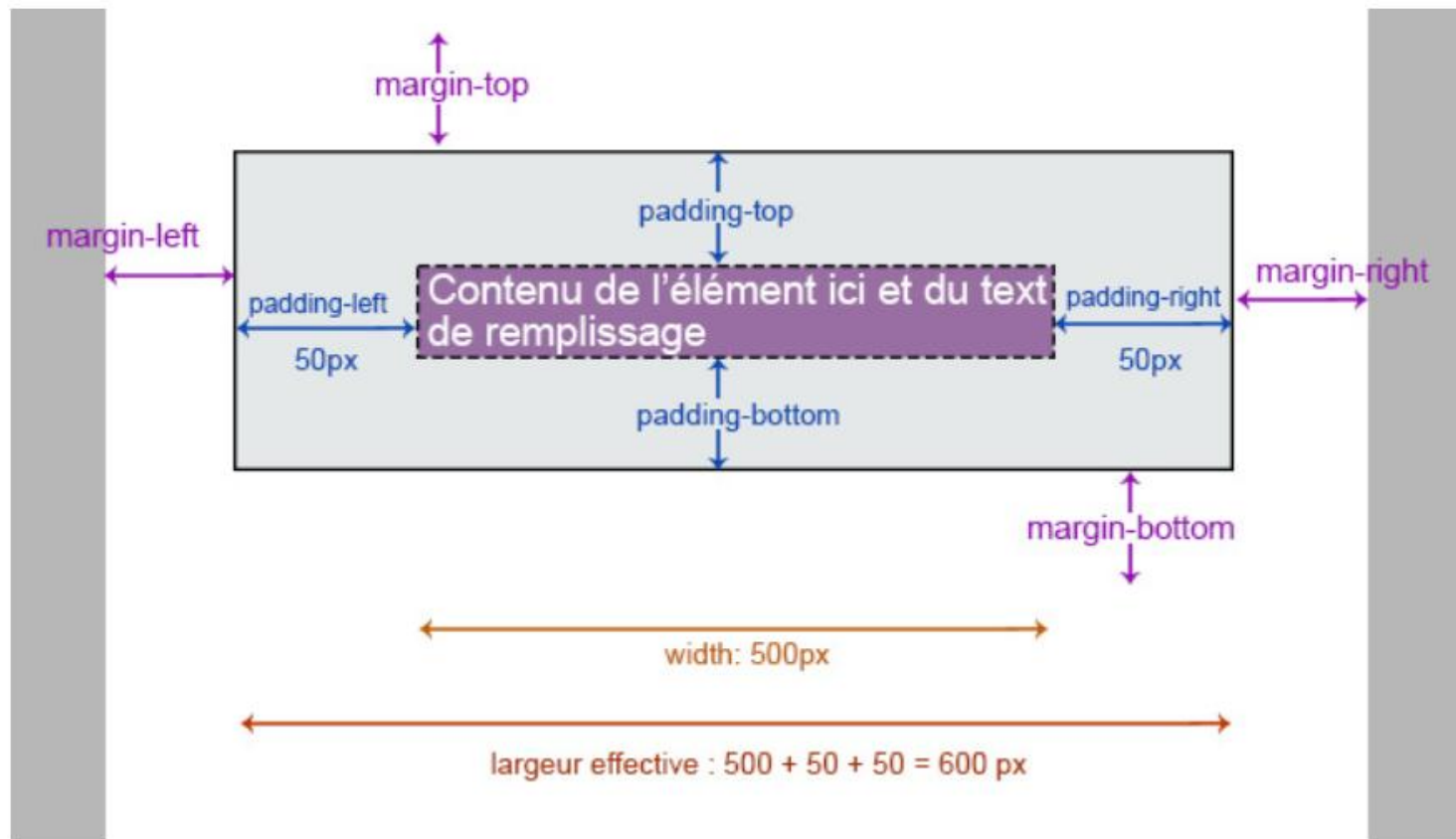
- ▶ **text-shadow** : l'ombre du texte (une ombre directement sur les lettres de votre texte)

```
text-shadow: 2px 2px 4px black;
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ullamcorper sodales elit, sit amet pellentesque lectus aliquet quis. Etiam sem ipsum, rhoncus eu aliquam nec, mattis consectetur tortor. Mauris non lectus magna, vel interdum elit. Sed fermentum commodo commodo. Fusce imperdiet vestibulum neque, id pulvinar urna ultricies ullamcorper. Donec euismod, ipsum vehicula pretium tempor, mauris odio pellentesque metus, et ultrices arcu mauris sit amet leo. Curabitur ac scelerisque sem.

Les marges

- ▶ Il existe deux types de marges : intérieure appelée **padding**, et extérieure appelée **margin**



Marge extérieure « margin »

- ▶ La propriété `margin` détermine l'espace entre le bord de l'élément, et l'élément suivant.
- ▶ Par défaut `margin: valeur;` applique la même valeur aux 4 cotés
- ▶ Valeur possible : `auto`, valeur en unité de mesure, positive ou négative
- ▶ On peut la décomposer individuellement en
 - `margin-top` : marge extérieure haute
 - `margin-right` : marge extérieure droite
 - `margin-bottom` : marge extérieure basse
 - `margin-left` : marge extérieure gauche
- ▶ Ou rassembler les 4 valeurs : `margin: 10px 5px 8px 15px;` (dans l'ordre : haut, droite, bas, gauche)
- ▶ `margin: 10px 0px;` 10px correspondent à la marge haute et basse et 0 aux marges gauches et droites
- ▶ Centrer horizontalement à l'aide de `margin`: pour centrer un élément de type bloc horizontalement dans son parent il faut lui donner une largeur (`width`), et appliquer une marge `auto` à gauche et à droite :

`Width : 200px ;`

`margin : auto ;`

Margin et éléments inline

- ▶ Par défaut, une margin **top** ou **bottom** sur un élément inline ne fonctionnera pas (puisque'il est dans le flux de la page).
- ▶ Une marge **left** ou **right** fonctionnera par contre.

Margin: exercice

- ▶ Dans le fichier `style.css`, définissez la propriété `margin` du premier paragraphe et utilisez la notation raccourcie pour appliquer une marge externe en haut de 20px, à droite de 50px, en bas de 10px et à gauche de 5px.

```
<html>
  <head>
    <link type="text/css" rel="stylesheet" href="style.css" />
    <title>Marge externe : haut, bas, gauche, droite</title>
  </head>
  <body>
    <div><p>Je suis un paragraphe.</p>
    <p>Je suis un autre paragrpahe.</p></div>
  </body>
</html>
```


Marge intérieure « padding »

- ▶ La propriété padding détermine un espacement entre le bord de la boîte et son contenu
- ▶ Par défaut **padding : valeur;** applique la même valeur aux 4 côtés
- ▶ Valeur possible : valeur en unité de mesure
- ▶ On peut la décomposer individuellement en
 - **padding-top** : marge intérieure haute
 - **padding-right** : marge intérieure droite
 - **padding-bottom** : marge intérieure basse
 - **padding-left** : marge intérieure gauche
- ▶ Ou rassembler les 4 valeurs : **padding: 10px 5px 8px 15px;** (dans l'ordre : haut, droite, bas, gauche)



Padding sur élément en inline

- ▶ Ajouter du padding sur des éléments inline permet de « changer leur taille » sans passer par un width / height (attention aux débordements)
- ▶ Pour agrandir le lien:

```
a {  
    padding: 5px 15px;  
}
```

Overflow : pour éviter que ça dépasse

- ▶ Il est possible que du contenu texte dépasse l'élément qui le contient (surtout si on définit des hauteurs fixes)
- ▶ Exemple : `p {height: 100px};`, le contenu « dépasse » du paragraphe

Chocolate sweet sweet roll. Donut oat cake gummies topping wafer caramels jelly-o. [Lemon drops lollipop sweet roll marzipan faworki](#). Gummi bears gummies pie candy canes. Wafer ice cream cookie dragée donut brownie applicake fruitcake. Wafer marshmallow sugar plum. Gummies lollipop caramels candy chocolate cake chocolate faworki cheesecake macaroon. Applicake chocolate bar soufflé marshmallow. Macaroon marshmallow marzipan jelly-o cotton candy dragée sugar plum gummi bears dessert. Jelly beans croissant danish gummi bears lemon drops. Chocolate bar biscuit pudding wafer pastry tart toffee. Caramels tart jelly.

Overflow : pour éviter que ça dépasse

▶ 1. Overflow : hidden - cacher ce qui dépasse

- **overflow: hidden;** permet de cacher tout ce qui dépasse.
- Problème : on perd tout ce qui est caché, impossible de le voir

Chocolate sweet sweet roll. Donut oat cake gummies topping wafer caramels jelly-o. [Lemon drops](#)
[lollipop](#) [sweet roll](#) [marzipan](#) [faworki](#). Gummi bears gummies pie candy canes. Wafer ice cream cookie
dragée donut brownie applicake fruitcake. Wafer marshmallow sugar plum. Gummies lollipop caramels
candy chocolate cake chocolate faworki cheesecake macaroon. Applicake chocolate bar soufflé
marshmallow. Macaroon marshmallow marzipan jelly-o cotton candy dragée sugar plum gummi bears

▶ 2. Overflow : auto - afficher une barre de défilement

- **overflow: auto;** permet d'afficher une bar de défilement si nécessaire

Chocolate sweet sweet roll. Donut oat cake gummies topping wafer caramels jelly-o. [Lemon drops lollipop](#)
[sweet roll](#) [marzipan](#) [faworki](#). Gummi bears gummies pie candy canes. Wafer ice cream cookie dragée
donut brownie applicake fruitcake. Wafer marshmallow sugar plum. Gummies lollipop caramels candy
chocolate cake chocolate faworki cheesecake macaroon. Applicake chocolate bar soufflé marshmallow.
Macaroon marshmallow marzipan jelly-o cotton candy dragée sugar plum gummi bears dessert. Jelly

Overflow : pour éviter que ça dépasse

Dans le fichier style.css, ajoutez les propriétés nécessaires à l'élément div :

- pour que le texte soit espacé de **10px** du bord du div;
- pour que le bord du div soit représenté par des **tirets** de **2px** et de couleur **bleu**;
- pour que le texte s'affiche sur un fond de couleur **bleu clair**;
- pour qu'il soit décalé de **25px** par rapport à son coin **supérieur gauche** (utilisez la notation raccourcie);

```
<html>
  <head>
    <link type="text/css" rel="stylesheet" href="style.css" />
    <title>Rappel</title>
  </head>
  <body>
    <div>
      <h1> Je suis un titre</h1>
      <p>Je suis un paragraphe.</p>
      <p>Je suis un autre paragrpahe.</p>
    </div>
  </body>
</html>
```

Positionnements CSS

- ▶ Pour des mise en page plus complexes, nous devons parler de la propriété CSS **position**.
- ▶ La propriété **position : valeur** permet de positionner les éléments dans la page
- ▶ La propriété CSS position supporte quatre valeurs différentes :
 - ▶ static (valeur par défaut)
 - ▶ relative
 - ▶ fixed
 - ▶ Absolute
- ▶ Les propriétés **top**, **right**, **bottom** et **left** (valeurs en pixels) déterminent l'emplacement final de l'élément positionné.

Position : static;

- ▶ static est la valeur par défaut de tous les éléments
- ▶ Un élément avec `position : static;` n'est positionné d'aucune manière spéciale.
- ▶ Un élément static est dit non positionné et un élément avec une propriété position ayant une valeur autre que static est dit positionné.
- ▶ Tout élément positionné de façon « static » ne pourra pas être affecté par les propriétés top, right, bottom et left.

```
position: static;  
✘ top: 100px;  
✘ left: 50px;
```

Position: relative;

- ▶ Un élément positionné avec `position : relative;` va être repositionné relativement par rapport à sa position par défaut.
- ▶ Le reste du contenu ne sera pas ajusté pour prendre la place dans l'espace laissé par l'élément.
- ▶ On peut utiliser les propriétés `top`, `left`, `right` et `bottom` pour déplacer l'élément.
- ▶ Par exemple, si on positionne un élément HTML de façon `relative` et qu'on lui ajoute `left: 50px;` l'élément sera déplacé de 50 pixels vers la droite par rapport à sa position par défaut.

```
position: relative;  
left: 50px;
```

- ▶ En positionnement relatif, une propriété avec une valeur négative peut être substituée par une valeur positive mais avec la propriété inverse. Exemples :
 - ▶ `top: -80px <==> bottom: 80px;`
 - ▶ `left: -80px <==> right: 80px;`

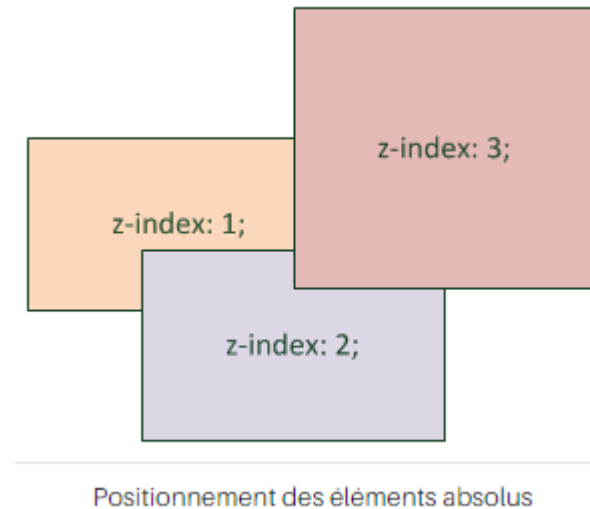
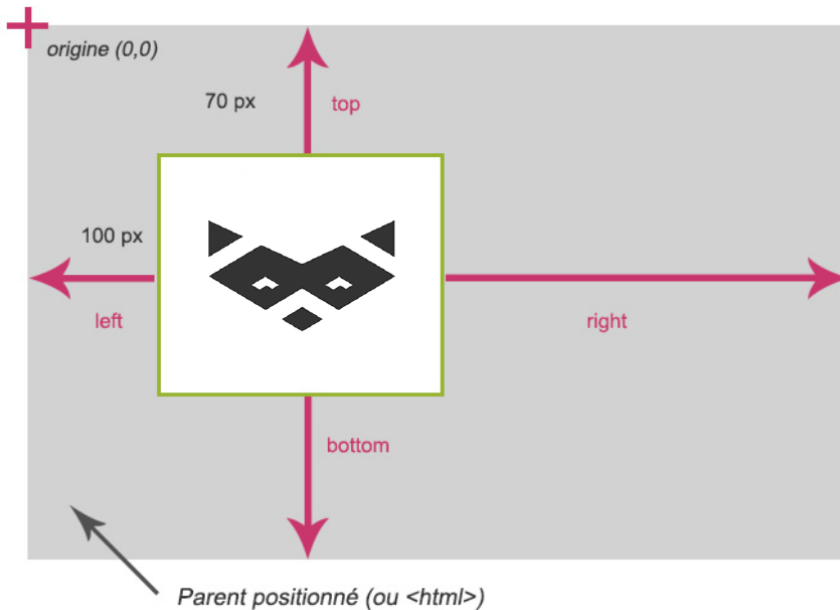
Position: fixed;

The screenshot shows a website with a fixed header and footer. The header contains the word "Marketing" in large font, a social media share bar with 43600+ marketers, a newsletter sign-up form, and a navigation menu with items like NEWS, HUBS, RESOURCES, MARKETINGTV, EVENTS, THE MAG, Join Now, and Sign In. A search bar is also present. The main content area features a large image of a person using a smartphone, followed by a text article about smartphone addiction in Australia. Below the text is a video player with the title "Woman on Hong Kong subway melts d...". The right sidebar includes a "CONTENT TYPE" section with buttons for CASE STUDY, EVENT, FEATURE, INFOGRAPHIC, INTERVIEW, NEWS, OPINION, SPONSORED, and VIDEO. An "EVENTS" section lists upcoming events like "DIGITAL MARKETING MASTERCLASS SERIES: AD" and "IABC REGIONAL LEADERSHIP INSTITUTE". A "Nir Eyal also recommends" pop-up is visible over the events section. The footer contains a social media share bar with icons for LinkedIn (34 shares), Twitter (30), Facebook (21), Google+ (9), Email, and Print.

Position: absolute;

- ▶ Un élément en **position: absolute;**:

- sort du flux (les autres éléments se réorganisent comme s'il n'était pas là).
 - Les éléments positionnés en absolu sont placés par-dessus le reste des éléments de la page.
 - l'élément vient se positionner par rapport à son dernier ancêtre positionné, à l'intérieur de celui-ci.
- ▶ Par défaut, si aucun ancêtre n'est positionné (et qu'on a modifié top, left, right ou bottom) il se positionne par rapport à l'élément racine <html>.



Le z-index

- ▶ Parfois, en modifiant la position des éléments ou pas, certains vont pouvoir se chevaucher. Par défaut, l'élément positionné en dernier apparaîtra par dessus les autres.
- ▶ Cependant, on peut choisir quel élément va apparaître au dessus de quel autre grâce à la propriété z-index.
- ▶ La propriété z-index va prendre un nombre en valeur : un nombre plus grand positionnera un élément devant un autre ayant un z-index plus petit.

```
element1 {  
  position: absolute;  
  right: 0px;  
  bottom: 0px;  
  z-index: 1;  
}
```

```
element2 {  
  position: absolute;  
  right: 30px;  
  bottom: 30px;  
  z-index: 2;  
}
```

Position et z-index

Un premier paragraphe.

Un autre paragraphe

Position et z-index

Un premier paragraphe.

Un autre paragraphe

Positionnements CSS: exercice

► Dans le fichier style.css:

1 - attribuez la propriété position et sa valeur **static** aux deux éléments div.

2 - attribuez la propriété position et sa valeur **relative** au div enfant. Précisez les valeurs de décalage top: 10px, left: 5px par rapport à sa position par défaut.

3 - attribuez la propriété position et sa valeur **absolute** au div parent. Précisez les valeurs de décalage top: 2px, left: 2px.

```
<html>
  <head>
    <link type="text/css" rel="stylesheet" href="style.css" />
    <title>Le positionnement absolu</title>
  </head>
  <body>
    <div id="parent"><div id="enfant"></div></div>
  </body>
</html>
```

```
#parent {
  height: 200px;
  width: 100px;
  border: 2px solid black;
  background-color: Red;
}

#enfant {
  height: 100px;
  width: 50px;
  border: 1px solid black;
  background-color: Blue;
}
```

ur

Mettre du texte centré sur une image



Mettre du texte sur une image

```
<body>
<div class="container">
  
  <div class="center">Texte centré.</div>
</div>
</body>
```

```
.container{
  position: relative;
}

.center{
  position: absolute;
  top: 50%;
  width: 100%;
  text-align: center;
  font-size: 18px;
}

img {
  width: 100%;
  height: 100%;
  opacity: 0.3;
}
```

Contrôle de l'affichage: JavaScript

```
<body>
<div class="Galerie">
  <a target="_blank" href="studio.jpg">
    
  </a>
  <div class="Description">Studio de MMI Arles</div>
  <p><button onclick="EliminerImage()">Eliminer image</button></p>
</div>
<div class="Galerie">
  <a target="_blank" href="amphi.jpg">
    
  </a>
  <div class="Description">Amphi de l'IUT Arles</div>
  <p><button onclick="CacherImage()">Cacher l'image</button></p>
</div>
<div class="Galerie">
  <a target="_blank" href="AmphiEtudiants.jpg">
    
  </a>
  <div class="Description">Etudiants MMI Amphi d'Arles</div>
  <p><button onclick="VoirTout()">Voir tout</button></p>
</div>
</body>
</html>
```

Contrôle de l'affichage: JavaScript

```
div.Galerie{
    margin: 5px;
    border: 1px solid Blue;
    width: 180px;
    text-align:center;
    background-color:yellow;
}
div.Galerie:hover {
    border: 8px solid LightBlue;
    background-color:red;
}
div.Galerie img{
    width: 100%;
    height: auto;
}
div.Description {
    padding: 15px;
    text-align: center;
    background-color:green;
}
```

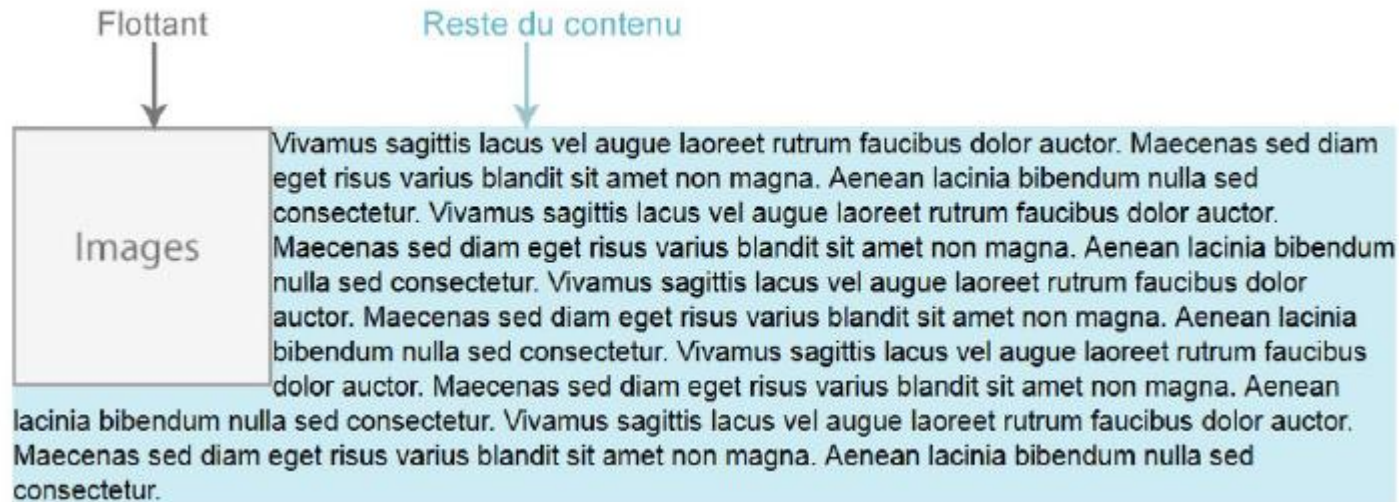

Contrôle de l'affichage: JavaScript

```
<html>
<head>
<title>Contrôle de l'affichage</title>
<link rel="stylesheet" type="text/css" href="styleTP3.css"/>
<script language="javascript">
function EliminerImage() {
    document.getElementById("studio").style.display="none";
}
function CacherImage() {
    document.getElementById("amphi").style.visibility = "hidden";
}
function VoirTout() {
    document.getElementById("studio").style.display = "block";
    document.getElementById("amphi").style.visibility = "visible";
}
</script>

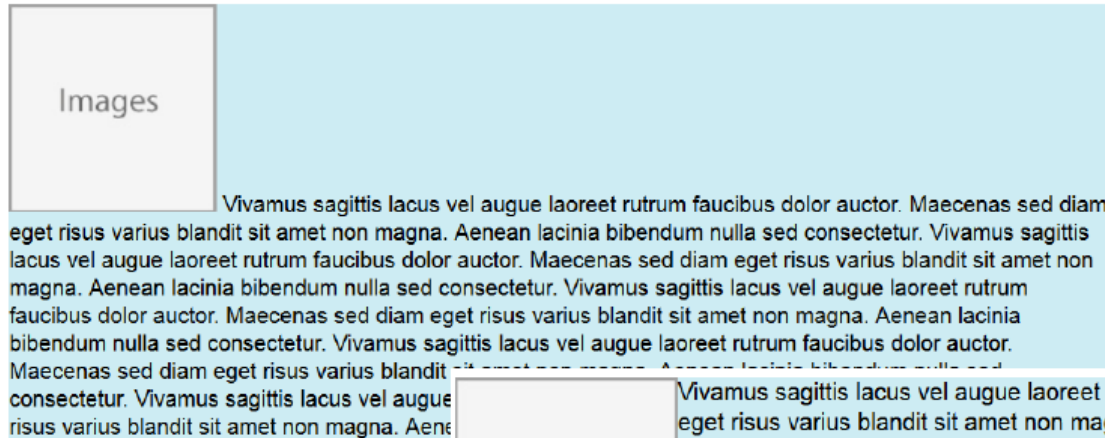
</head>
```

le positionnement flottant

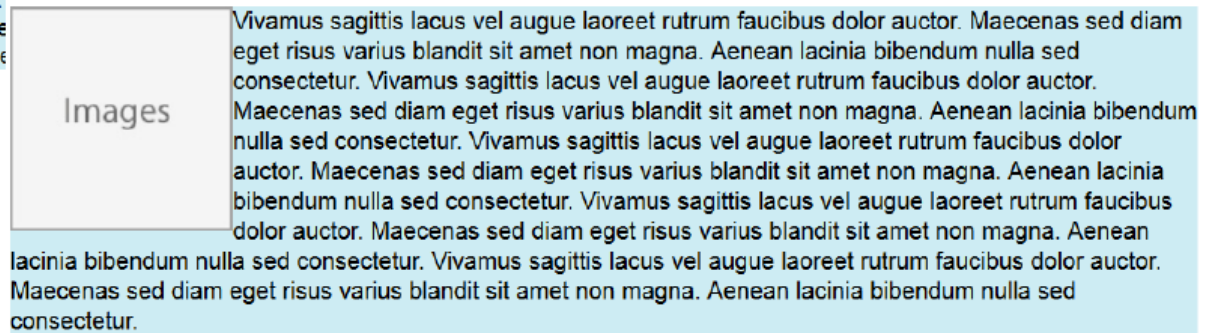
- ▶ La propriété `float:valeur` permet d'extraire des éléments du flux de la page, ce qui signifie que le reste du contenu « coule » autour
- ▶ Elle prend 3 valeurs : left, right et none (permet de remettre un élément dans le flux)



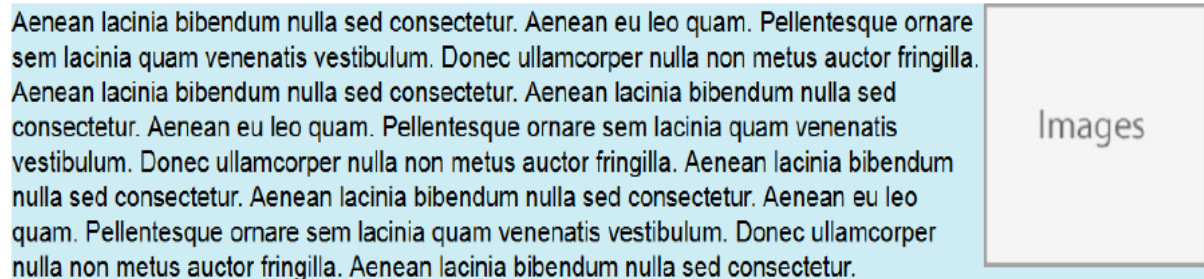
le positionnement flottant



```
img {  
  float: left;  
}
```



```
img {  
  float: right;  
}
```



Problème des flottants

Les éléments suivants un élément flottant vont se positionner à côté de celui-ci par défaut.

Clear : bloquer le dépassement des flottants

- La propriété `clear` : valeur permet à un élément de cesser le contournement des éléments flottants. Il se positionne alors sous les éléments flottants précédents comme si ces derniers étaient restés dans le flux.
- Valeurs possibles :
 - `clear: left` permet d'empêcher le contournement des blocs flottants à gauche
 - `clear: right` permet d'empêcher le contournement des blocs flottants à droite
 - `clear : both` permet d'empêcher le contournement des blocs flottants à gauche et à droite



Un titre ici

- [Home](#)
- [Service](#)
- [A propos](#)
- [Contact](#)

Soufflé gingerbread cupcake donut tiramisu cookie sweet soufflé cake.
Candy canes candy canes donut toffee candy canes cheesecake gummies
apple pie. Pie cupcake cheesecake sugar plum tart donut bear claw caramels

And then... Finally, users can insert a WordPress [gallery], which is kinda ugly and comes with some CSS stuck into the page to style it (which doesn't actually validate, nor does the markup for the gallery). The amount of columns in the gallery is also changable by the user, but the default is three so we'll work with that for our example with an added fourth image to test verticle spacing.

On l'applique sur le premier élément « suivant » dont on veut cesser le contournement.

Problème des flottants

On applique `clear:both;` sur l'élément qui ne doit plus être affecté par les flottants

```
.content {  
  clear:both;  
}
```



Un titre ici

- [Home](#)
- [Service](#)
- [A propos](#)
- [Contact](#)

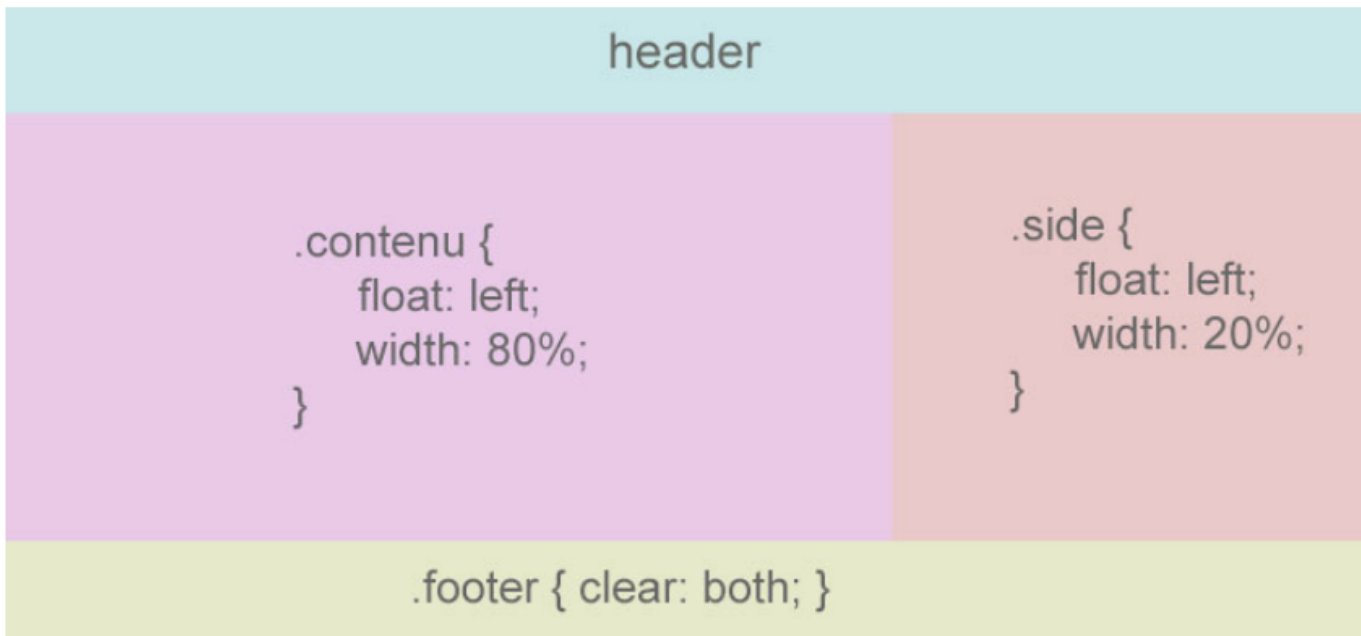
Soufflé gingerbread cupcake donut tiramisu cookie sweet soufflé cake. Candy canes candy canes donut toffe
candy canes cheesecake gummies apple pie. Pie cupcake cheesecake sugar plum tart donut bear claw
caramels

And then... Finally, users can insert a WordPress [gallery], which is kinda ugly and comes with some CSS stuck
into the page to style it (which doesn't actually validate, nor does the markup for the gallery). The amount of
columns in the gallery is also changable by the user, but the default is three so we'll work with that for our
example with an added fourth image to test verticle spacing.



Les flottants appliqués à la mise en page

Il est possible d'utiliser des flottants pour créer une mise en page en colonnes.



Les flottants appliqués à la mise en page

Dans le fichier style.css, attribuez la propriété float et sa valeur right à l'élément div.

```
<html>
  <head>
    <link type="text/css" rel="stylesheet" href="style.css" />
    <title>Le positionnement flottant (droite)</title>
  </head>
  <body>
    <div>Je flotte</div>
    <p>Je reste dans le flux courant.</p>
  </body>
</html>
```

```
div {
  height: 100px;
  width: 100px;
  border: 2px solid black;
  border-radius: 5px;
  background-color: blue;
}
```

Les flottants appliqués à la mise en page

Dans le fichier style.css, attribuez à la propriété float de div la valeur left.

```
<html>
  <head>
    <link type="text/css" rel="stylesheet" href="style.css" />
    <title>Le positionnement flottant (gauche)</title>
  </head>
  <body>
    <div>Je vais flotter à gauche</div>
    <p>
      Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche
      Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche
      Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche
      Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche
    </p>
  </body>
</html>
```

```
div {
  height: 100px;
  width: 100px;
  border: 2px solid black;
  border-radius: 5px;
  background-color: blue;
}
```


Les flottants appliqués à la mise en page

Dans le fichier stylesheet.css, attribuez au div la valeur right et au p la valeur left pour la propriété float;

```
<html>
  <head>
    <link type="text/css" rel="stylesheet" href="style.css" />
    <title>Le positionnement flottant (gauche)</title>
  </head>
  <body>
    <div>Je vais flotter à gauche</div>
    <p>
Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche
Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche
Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche
Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche Je vais flotter à gauche
    </p>
  </body>
</html>
```

```
div {
  height: 100px;
  width: 100px;
  border: 2px solid black;
  border-radius: 5px;
  background-color: blue;
}

P{
  width: 300px;
}
```

Listes à puces

1. Listes à puces

list-style-type : permet de spécifier le type de puce ou de numérotation (appliqué sur le ul / ol)

Valeurs possibles: none (pas de puce), disc, circle, square

Accueil	• Accueil	◦ Accueil	▪ Accueil
Nos services	• Nos services	◦ Nos services	▪ Nos services
A propos	• A propos	◦ A propos	▪ A propos
Nous contacter	• Nous contacter	◦ Nous contacter	▪ Nous contacter

2. Listes ordonnées

Valeurs possibles: decimal (1.), decimal-leading-zero (01.), lower-roman (vii), upper-roman (VII)...

1. Accueil	01. Accueil	i. Accueil	I. Accueil
2. Nos services	02. Nos services	ii. Nos services	II. Nos services
3. A propos	03. A propos	iii. A propos	III. A propos
4. Nous contacter	04. Nous contacter	iv. Nous contacter	IV. Nous contacter

Utiliser une image comme puce

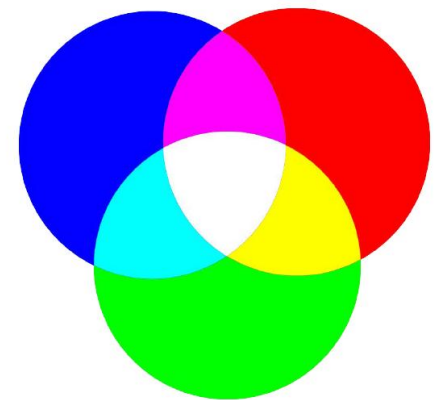
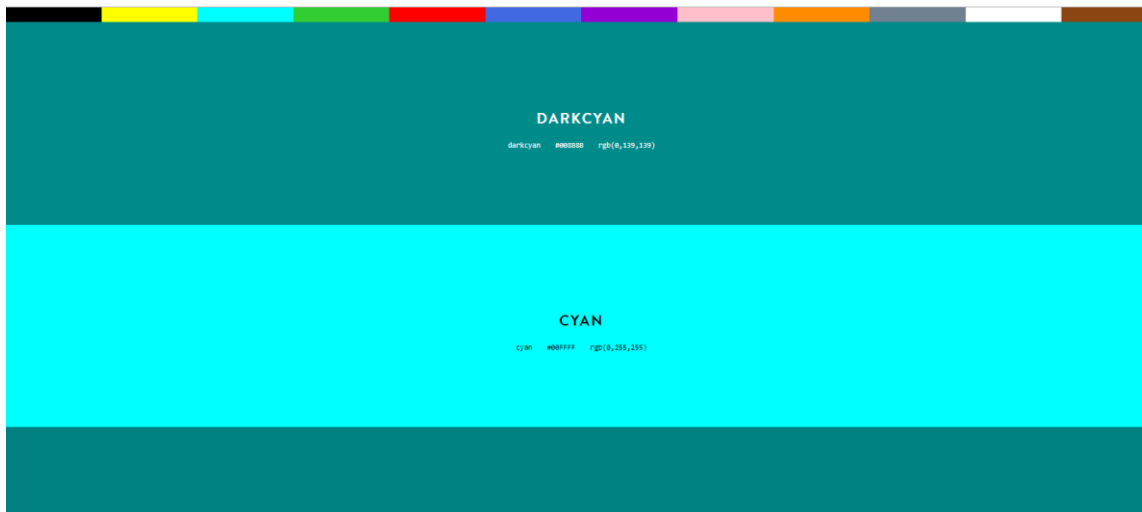
```
ul {  
    list-style-image: url("img/liste.png");  
}
```

- ★ Accueil
- ★ Nos services
- ★ A propos
- ★ Nous contacter

Les couleurs en CSS

- Notation hexadécimale : `#ffffff` (blanc)
- Notation hexadécimale courte (qui est doublée pour obtenir la version longue) : `#fff`
- Notation RGB : `rgb(255,255,255)` / `rgba(255,255,255,1)`
- Notation HSL : `hsl(0, 0%, 100%);` / `hsla(0, 0%, 100%, 1);`
- Mot clé : `white`

- <http://colours.neilorangepeel.com/>



Création d'un site web personnel

- ▶ Le site comportera quatre pages :
- ▶ Une page d'accueil : **index.html**
- ▶ Une page décrivant les expériences professionnelles : **experiences-pro.html**
- ▶ Une page décrivant les loisirs : **hobbies.html**
- ▶ Une page de contact : **contact.html**

La page d'accueil sera la page de CV : pour se présenter et mettre en avant la formation et les compétences.

- créer une page **style.css** que nous remplirons plus tard
- créer un dossier « **img** » dans le dossier contenant les différentes pages de votre site.

Création du header, footer et menu

- ▶ les parties « fixes » du site, c'est-à-dire les parties qui vont être identiques sur chaque page de notre site.
- ▶ créer ces parties dans la page index.html, puis copier et coller le code dans les autres pages.

```
<html>
<head>
<title>Zeineb EL KHALFI</title>
<meta charset="utf-8">
<link rel="stylesheet" type="text/css" href="style.css"/>
</head>
<body>
<header>
<h1> Zeineb EL khalfi </h1> <h2>Docteur en informatique</h2>
</h1>
<nav>
<div>
<ul>
<li>
<li>
<li>
<li>
</ul>
</div>
</nav>
<foot>
<p>
</p>
</foot>
</body>
</html>
```

Zeineb EL KHALFI

file:///C:/Users/zizo/ownCloud/IUT/M2206/ex/projet%20tp/index.html

Applications

Zeineb EL khalfi

Docteur en informatique

- [Accueil](#)
- [Expériences professionnelles](#)
- [Loisirs](#)
- [Contactez-moi](#)

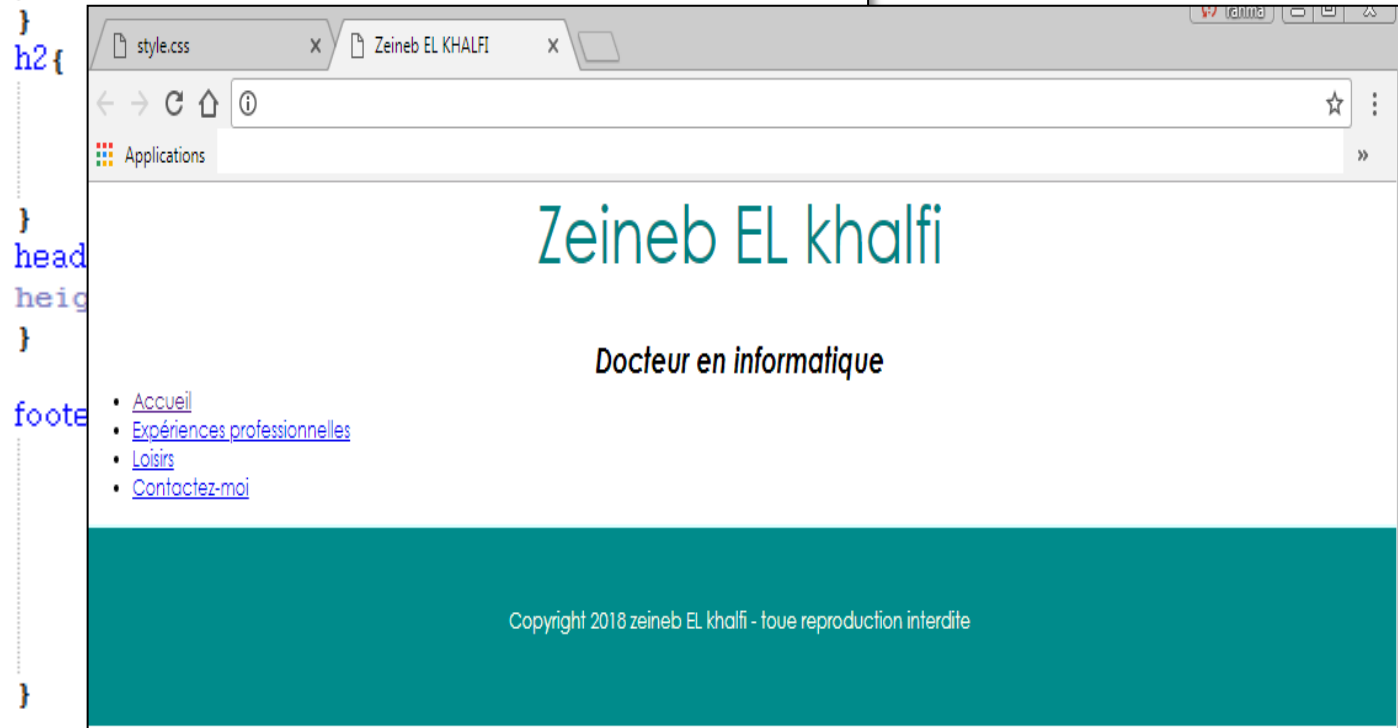
Copyright 2018 zeineb EL khalfi - toute reproduction interdite

Mise en forme CSS générale

- ▶ définir des polices de caractères
- ▶ effectuer également le même rendu sur les autres éléments (ex: marges).
- ▶ centrer les titres et sous-titres
- ▶ effectuer une mise en page

```
body{  
font-family:'Century Gothic', Arial,serif;  
font-size:13px;  
margin:0px;  
padding:0px;  
}  
h1{  
color:#008080;  
font-size:45px;  
font-weight:normal;  
text-align:center;  
}
```

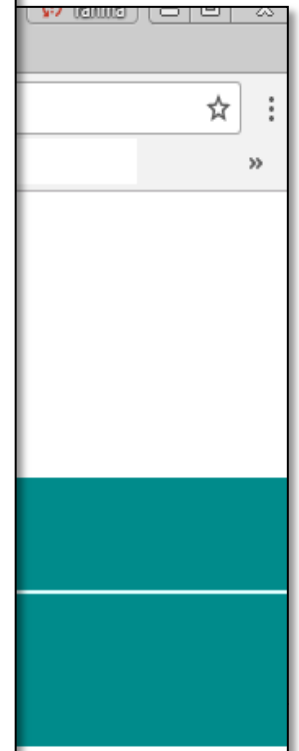
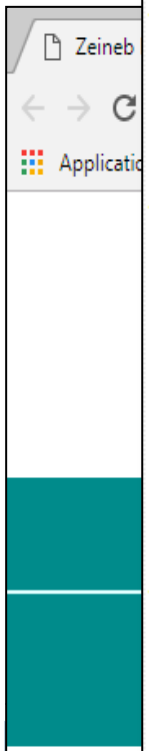
propriétés de body afin d'avoir automatiquement une petite



Mise en forme CSS générale

- ▶ mettre
- ▶ reset d
- ▶ étendr
- ▶ mettre
- ▶ centrer

```
/* mettre les éléments du menu en ligne et enlever les puces */
nav li{
    list-style-type:none;
    float: left;
}
/* reset des marges de liste */
nav ul{
    margin:0px;
    padding:0px;
}
/* étendre le nav et lui donner une couleur de fond */
nav{
    width:100%;
    background-color:#008B8B;
}
/* mettre en forme nos éléments de lien */
nav a{
    display: inline-block;
    text-decoration:none;
    padding:20px 30px;
    color:#FFFFFF0;
    text-transform:uppercase;
    font-size:15px;
}
/* centrer le menu dans la page */
.table{
    display:table;
    margin:0 auto;
}
```



Page d'accueil

- ▶ Quatre sections dans la page index.html :
- ▶ une première section de présentation
- ▶ une deuxième section sur le parcours et les expériences professionnelles
- ▶ une troisième section pour présenter les compétences
- ▶ une dernière partie contenant la lettre de recommandation

- ▶ Utiliser quatre éléments HTML **section** et placer le tout à l'intérieur d'un élément **article**, car le tout peut faire office de CV et donc se suffit à lui-même.

Section 1: présentation

- ▶ afficher une photo à gauche et un texte de présentation à droite.
- découper cette section en deux avec deux éléments div

Zeineb EL khalfi

Docteur en informatique

ACCUEL

EXPÉRIENCES PROFESSIONNELLES

LOISIRS

CONTACTEZ-MOI

Profil



Je m'appelle Zeineb El khalfi, j'ai 27 ans et je vis à Arles.

Je suis docteur en intelligence artificielle et actuellement, je suis ATER à l'université Aix-Marseille.

Domaines de Recherche: Décision sous incertitude, Théorie de décision séquentielle, Théorie des possibilités, Modèles graphiques pour la décision séquentielle.

Section 1: présentation

```
index.html x style.css x new 1 x
7 </head>
8 <body>
9 <header>
10 <h1> Zeineb EL khalfi </h1> <h2>Docteur en informatique</h2>
11 </header>
12 <nav>
<article>
<section id="présentation">
<h1> Profil </h1>
<div class="sec">
  <div class="left">
    
  </div>
  <div class="right">
    <p> Je m'appelle Zeineb El khalfi, j'ai 27 ans et je vis à Arles.</p>
    <p> Je suis docteur en intelligence artificielle et actuellement, je suis ATER à l'université Aix-Marseille.</p>
    <p> Domaines de Recherche: Décision sous incertitude, Théorie de décision séquentielle,
    Théorie des possibilités, Modèles graphiques pour la décision séquentielle. </p>
  </div>
</div>
</div>
</article>
34 </div>
35 </div>
36 </article>
37 </article>
38 <footer>
39 <p> Copyright 2018 zeineb EL khalfi - toute reproduction interdite </p>
40 </footer>
41 </body>
42 </html>
```

Section 1: présentation

```
section{
    width:100%;
    min-height:300px;
    padding-bottom:10px;
    margin-bottom:10px;
}
.sec{
    margin:0px 10%;
}
.left{
    float:left;
    width:30%;
}
.right{
    float:left;
    width:70%;
}
.sec::after{
    content:"";
    display:table;
    clear:both;
}

#présentation{
background-color: rgba (224, 255, 255, 0.3)
}

#parcours-exp{
border-top:5px solid #AFEEDD;
background-color: rgba (224, 255, 255, 0.3) ;
}
}
```

Section 2: Parcours et expériences

- ▶ séparer la section en deux avec des dates sur la gauche et les expériences correspondant aux dates sur la droite.
- ▶ afin que les dates soient bien alignées par rapport aux expériences, utiliser un div intermédiaire de plus à chaque fois pour englober la date et l'expérience correspondante ensemble.

Parcours et expérience

Cursus Universitaire

2014 - 2017

Doctorat en Informatique, spécialité Intelligence Artificielle

Université Paul Sabatier de Toulouse

Mention: très honorable

Titre: Lexicographic Refinements in Possibilistic Sequential Decision-Making Models.

Résumé: Ma thèse de doctorat porte sur la prise de décision séquentielle dans le cadre de la théorie des possibilités. Nous nous intéresserons aux arbres de décision ainsi que les processus décisionnels de Markov très étudiés en intelligence artificielle. L'objectif est d'étendre les raffinements lexicographiques, qui raffinent les utilités qualitatives possibilistes, aux problèmes séquentiels et de proposer des algorithmes pour le calcul des stratégies optimales lexicographiques dans les modèles de décision séquentielle possibilistes.

2013 - 2014

Master de recherche : Fouille de Données et Gestion des Connaissances (FDGC) Parcours Extraction des Connaissances à partir des Données

École Polytechnique de Nantes

Mention: Bien

2012 - 2013

Master de recherche, Sciences et Techniques de l'Informatique Décisionnelle

École Polytechnique de Nantes

Mention : Assez Bien

Parcours et expérience

Cursus Universitaire

2014 - 2017

Doctorat en Informatique, spécialité Intelligence Artificielle

Université Paul Sabatier de Toulouse

Mention: très honorable

Titre: Lexicographic Refinements in Possibilistic Sequential Decision-Making Models.

Résumé: Ma thèse de doctorat porte sur la prise de décision séquentielle dans le cadre de la théorie des possibilités. Nous nous intéresserons aux arbres de décision ainsi que les processus décisionnels de Markov très étudiés en intelligence artificielle. L'objectif est d'étendre les raffinements lexicographiques, qui raffinent les utilités qualitatives possibilistes, aux problèmes séquentiels et de proposer des algorithmes pour le calcul des stratégies optimales lexicographiques dans les modèles de décision séquentielle possibilistes.

2013 - 2014

Master de recherche : Fouille de Données et Gestion des Connaissances (FDGC) Parcours Extraction des Connaissances à partir des Données

École Polytechnique de Nantes

Mention: Bien

2012 - 2013

Master de recherche, Sciences et Techniques de l'Informatique Décisionnelle

École Polytechnique de Nantes

Mention : Assez Bien

- séparer la section en deux avec des dates sur la gauche et les expériences correspondant aux dates sur la droite.

- afin que les dates soient bien alignées par rapport aux expériences, utiliser un div intermédiaire de plus à chaque fois pour englober la date et l'expérience correspondante ensemble.



Section 2: Parcours et expériences

Zeineb EL khalfi

Docteur en informatique

ACCUEIL

EXPÉRIENCES PROFESSIONNELLES

LOISIRS

CONTACTEZ-MOI

Profil



Je m'appelle Zeineb El khalfi, j'ai 27 ans et je vis à Arles.

Je suis docteur en intelligence artificielle et actuellement, je suis ATER à l'université Aix-Marseille.

Domaines de Recherche: Décision sous incertitude, Théorie de décision séquentielle, Théorie des possibilités, Modèles graphiques pour la décision séquentielle.

Parcours et expérience

Cursus Universitaire

2014 - 2017

Doctorat en Informatique, spécialité Intelligence Artificielle

Université Paul Sabatier de Toulouse

Mention: très honorable

Titre: Lexicographic Refinements in Possibilistic Sequential Decision-Making Models.

Section 2: Parcours et expériences

Parcours et expérience

Cursus Universitaire

2014 - 2017

Doctorat en Informatique, spécialité Intelligence Artificielle

Université Paul Sabatier de Toulouse

Mention: très honorable

Titre: Lexicographic Refinements in Possibilistic Sequential Decision-Making Models.

Résumé: Ma thèse de doctorat porte sur la prise de décision séquentielle dans le cadre de la théorie des possibilités. Nous nous intéresserons aux arbres de décision ainsi que les processus décisionnels de Markov très étudiés en intelligence artificielle. L'objectif est d'étendre les raffinements lexicographiques, qui raffinent les utilités qualitatives possibilistes, aux problèmes séquentiels et de proposer des algorithmes pour le calcul des stratégies optimales lexicographiques dans les modèles de décision séquentielle possibilistes.

2013 - 2014

Master de recherche : Fouille de Données et Gestion des Connaissances (FDGC) Parcours Extraction des Connaissances à partir des Données

École Polytechnique de Nantes

Mention: Bien

2012 - 2013

Master de recherche, Sciences et Techniques de l'Informatique Décisionnelle

École Polytechnique de Nantes

Mention : Assez Bien

2010 - 2012

Licence fondamentale en Informatique

École Polytechnique de Nantes

Mention : Très bien

Expérience professionnelle

2013 - 2015

développeur RD

Société de consulting (MSI)

Travail dans le service informatique : Utilisation d'un ERP, Maîtrise des outils de reporting et d'analyse (Oracle Discoverer et Oracle Reports), développement d'un site Web

Section 2: Parcours et expériences

```
<section id="parcours-exp">
<h1> Parcours et expérience </h1>
<div class="sec">
<h2> Cursus Universitaire </h2>
  <div class="exp">
    <div class="left">
      <h3> 2014 - 2017 </h3>
    </div>
    <div class="right">
      <h3> Doctorat en Informatique, spécialité Intelligence Artificielle </h3>
      <p> Université Paul Sabatier de Toulouse</p>
      <p> Mention: très honorable </p>
      <p> Titre: Lexicographic Refinements in Possibilistic Sequential Decision-Making Models.</p>
      <p> Résumé: Ma thèse de doctorat porte sur la prise de décision séquentielle dans le cadre de la
        théorie des possibilités. Nous nous intéresserons aux arbres de décision ainsi que les
        processus décisionnels de Markov très étudiés en intelligence artificielle. L'objectif
        est d'étendre les raffinements lexicographiques, qui raffinent les utilités qualitatives
        possibilistes, aux problèmes séquentiels et de proposer des algorithmes pour le calcul
        des stratégies optimales lexicographiques dans les modèles de décision séquentielle
        possibilistes.</p>
    </div>
  </div>
  <div class="exp">
    <div class="left">
      <h3> 2013 - 2014 </h3>
    </div>
    <div class="right">
      <h3> Master de recherche : Fouille de Données et Gestion des Connaissances
        (FDGC) Parcours Extraction des Connaissances à partir des Données </h3>
      <p> École Polytechnique de Nantes</p>
      <p> Mention: Bien </p>
    </div>
  </div>
</div>
```

Section 2: Parcours et expériences

```
<div class="exp">
  <div class="left">
    <h3> 2010 - 2012 </h3>
  </div>
  <div class="right">
    <h3> Licence fondamentale en Informatique </h3>
    <p> École Polytechnique de Nantes</p>
    <p> Mention : Très bien </p>
  </div>
</div>

<h2> Expérience professionnelles </h2>
<div class="exp">
  <div class="left">
    <h3> 2013 - 2015 </h3>
  </div>
  <div class="right">
    <h3> développeur RD </h3>
    <p> Société de consulting (MSI) </p>
    <p> Travail dans le service informatique : Utilisation d'un ERP, Maitrise des outils
de reporting et d'analyse (Oracle Discoverer et Oracle Reports), développement d'un site We </p>
  </div>
</div>
</div>
</section>
</article>
<footer>
<p> Copyright 2018 zeineb EL khalfi - toute reproduction interdite </p>
</footer>
</body>
</html>
```

section 3: Parcours et expériences

```
<div class="exp">
  <div class="left">
    <h3> 2010 - 2012 </h3>
  </div>
  <div class="right">
    <h3> Licence fondamentale en Informatique </h3>
    <p> École Polytechnique de Nantes</p>
    <p> Mention : Très bien </p>
  </div>
</div>

<h2> Expérience professionnelles </h2>
<div class="exp">
  <div class="left">
    <h3> 2013 - 2015 </h3>
  </div>
  <div class="right">
    <h3> développeur RD </h3>
    <p> Société de consulting (MSI) </p>
    <p> Travail dans le service informatique : Utilisation d'un ERP, Maitrise des outils
de reporting et d'analyse (Oracle Discoverer et Oracle Reports), développement d'un site We </p>
  </div>
</div>
</div>
</section>
</article>
<footer>
<p> Copyright 2018 zeineb EL khalfi - toute reproduction interdite </p>
</footer>
</body>
</html>
```

section 2: Parcours et expériences

```
section{
  width:100%;
  min-height:300px;
  padding-bottom:10px;
  margin-bottom:10px;
}
.sec{
  margin:0px 10%;
}
.left{
  float:left;
  width:30%;
}
.right{
  float:left;
  width:70%;
}
.sec::after,.exp::after{
  content:"";
  display:table;
  clear:both;
}

#parcours-exp{
  border-top:5px solid #AFEEEE;
  background-color:rgba(224,255,255,0.3)
}
```

section 3: Compétences

Mention: Bien

2012 - 2013

Master de recherche, Sciences et Techniques de l'Informatique Décisionnelle

École Polytechnique de Nantes

Mention : Assez Bien

2010 - 2012

Licence fondamentale en Informatique

École Polytechnique de Nantes

Mention : Très bien

Expérience professionnelles

2013 - 2015

développeur RD

Société de consulting (MSI)

Travail dans le service informatique : Utilisation d'un ERP, Maitrise des outils de reporting et d'analyse (Oracle Discoverer et Oracle Reports), développement d'un site We

Compétences et niveau

Compétence

Niveau

Compétence

Niveau

HTML/CSS

Expert

SQL/mysql

Avancé

Javascript

Expert

JEE

moyen

section 3: Compétences

```
<h2> Expérience professionnelle </h2>
<div class="exp">
  <div class="left">
    <h3> 2013 - 2015 </h3>
  </div>
  <div class="right">
    <h3> développeur RD </h3>
    <p> Société de consulting (MSI) </p>
    <p> Travail dans le service informatique : Utilisation d'un ERP, Maitrise de reporting et d'analyse (Oracle Discoverer et Oracle Reports), développeur </p>
  </div>
</div>
</div>
</section>
<section id="competence">
<h1> Compétences et niveau </h1>
<div class="sec">
  <div class="cleft">
    <div class="sous">
      <div class="sousleft"><h3>Compétence</h3></div>
      <div class="sousright"><h3>Niveau</h3></div>
    </div>
    <div class="sous">
      <div class="sousleft"><p>HTML/CSS</p></div>
      <div class="sousright"><p>Expert</p></div>
    </div>
    <div class="sous">
      <div class="sousleft"><p>Javascript</p></div>
      <div class="sousright"><p>Expert</p></div>
    </div>
  </div>
  <div class="cright">
    <div class="sous">
      <div class="sousleft"><h3>Compétence</h3></div>
      <div class="sousright"><h3>Niveau</h3></div>
    </div>
    <div class="sous">
      <div class="sousleft"><p>SQL/mysql</p></div>
      <div class="sousright"><p>Avancé</p></div>
    </div>
    <div class="sous">
      <div class="sousleft"><p>JEE</p></div>
      <div class="sousright"><p>moyen</p></div>
    </div>
  </div>
</div>
</div>
</section>
```

```
nav li:hover a{
  padding:15px 30px 20px 30px;
}

section{
  width:100%;
  min-height:300px;
  padding-bottom:10px;
  margin-bottom:10px;
}

.sec{
  margin:0px 10%;
}

.left{
  float:left;
  width:30%;
}

.right{
  float:left;
  width:70%;
}

.sec::after, .exp::after{
  content:"";
  display:table;
  clear:both;
}

.cleft, .cright, .sousleft, .sousright{
  width:50%;
  float:left;
}

#présentation{
  background-color: rgba (224, 255, 255, 0.3)
}

#parcours-exp{
  border-top:5px solid #AFEEEE;
  background-color: rgba (224, 255, 255, 0.3) ;
}

#competence{
  border-top:5px solid #AFEEEE;
  background-color: rgba (135, 206, 250, 0.3) ;
}
```


section 4: Lettre de recommandation

Discoverer et Oracle Reports), développement d'un site We

Compétences et niveau

Compétence	Niveau	Compétence	Niveau
HTML/CSS	Expert	SQL/mySQL	Avancé
Javascript	Expert	JEE	moyen

Lettre de recommandation (à télécharger)

Lettre de recommandation



J'encadre depuis septembre 2014 la thèse de Zeineb El Khalfi, thèse effectuée à l'Université Toulouse III. Les premiers travaux qu'elle a effectués sous notre codirection sont très prometteurs, et ont été validés rapidement par plusieurs publications (LFA 2015, LFA 2016 ECAI 2016, ECSQARU 2017).

Hélène Fargier



Mademoiselle Zeineb El Khalfi a effectué son mémoire de maîtrise (de Février à Septembre 2014) s'intitulant « Extended Dynamic Programming for Optimizing Strategies in Possibilistic Decision Trees » sous ma direction.

Copyright 2018 zeineb EL khalfi - toute reproduction interdite

section 4: Lettre de recommandation

```
<section id="recommandation">
<h1> Lettre de recommandation (à télécharger)</h1>
<div class="sec">
  <div class="rec">
    <h2> Lettre de recommandation</h2>
    <div class="left">
      <a href="recommandation.pdf" target="_blank">  </a>
    </div>
    <div class="right">
      <p>J'encadre depuis septembre 2014 la thèse de Zeineb El Khalfi, thèse
        effectuée à l'Université Toulouse III. Les premiers travaux qu'elle a effectués sous notre codirection sont très
        prometteurs, et ont été validés rapidement par plusieurs publications (LFA
        2015, LFA 2016 ECAI 2016, ECSQARU 2017).</p>
    </div>
  </div>
  <div class="rec">
    <h2> Hélène Fargier </h2>
    <div class="left">
      <a href="recomZeineb.pdf" target="_blank" >  </a>
    </div>
    <div class="right">
      <p> Mademoiselle Zeineb El Khalfi a effectué son mémoire de mastère (de Février à Septembre 2014) s'intitulant
        « Extended Dynamic Programming for Optimizing Strategies in Possibilistic Decision Trees» sous ma direction.</p>
    </div>
  </div>
</div>
</section>
```

section 4: Lettre de recommandation

```
nav li:hover a{
  padding:15px 30px 20px 30px;
}

section{
  width:100%;
  min-height:300px;
  padding-bottom:10px;
  margin-bottom:10px;
}

.sec{
  margin:0px 20%;
}

.left{
  float:left;
  width:30%;
}

.right{
  float:left;
  width:70%;
}

.sec::after,.exp::after,.reco::after{
  content:"";
  display:table;
  clear:both;
}

.cleft,.cright,.sousleft,.sousright{
  width:50%;
  float:left;
}

#présentation{
background-color: rgba (224,255,255,0.3)
}

#parcours-exp{
border-top:5px solid #AFEEDD;
background-color: rgba (224,255,255,0.3) ;
}

#competence{
border-top:5px solid #AFEEDD;
background-color: rgba (135,206,250,0.3) ;
}

#recommandation{
border-top:5px solid #AFEEDD;
background-color: rgba (176,224,230,0.3) ;
}
}
```

attribut de données personnalisé data-descr pour créer une bulle d'information de type glossaire

```
<p>Voici l'exemple en action du code ci-dessus.<br />  
Nous avons un peu de <span data-descr="collection de mots et de ponctuation">texte</span>  
ici avec quelques <span data-descr="petites fenêtres surgissantes qui se cachent aussi">  
bulles d'information</span>.<br />  
Ne soyez pas timide, survolez le texte pour jeter  
un <span data-descr="à ne pas prendre au sens littéral">œil</span>.  
</p>
```

```
span[data-descr] {  
  position: relative;  
  text-decoration: underline;  
  color: #00F;  
  cursor: help;  
}  
  
span[data-descr]:hover::after {  
  content: attr(data-descr);  
  position: absolute;  
  left: 0;  
  top: 24px;  
  min-width: 200px;  
  border: 1px #aaaaaa solid;  
  border-radius: 10px;  
  background-color: #ffffcc;  
  padding: 12px;  
  color: #000000;  
  font-size: 14px;  
  z-index: 1;  
}
```

Voici l'exemple en action du code ci-dessus.
Nous avons un peu de [texte](#) ici avec quelques [bulles d'information](#).
Ne soyez pas timide, survolez le texte pour jeter un [œil](#).

à ne pas prendre au sens littéral

le Responsive Web design

- ▶ Les techniques permettant d'adapter les pages web à toutes les tailles d'écrans afin d'avoir un bon rendu.
- ▶ Implémentation
 - ▶ Uniquement avec HTML et CSS.
 - ▶ Opérations de :
 - Repositionnement.
 - Déplacement.
 - Rétrécissement.
 - Agrandissement.
 - Changement de la visibilité.
 - ▶ sur les éléments de la page.
 - ▶ la taille de la fenêtre (la largeur):
 - Smartphone: 320px (format portrait)
 - Tablettes: 768px
 - Netbook: 1024px
 - Desktop: 1600px

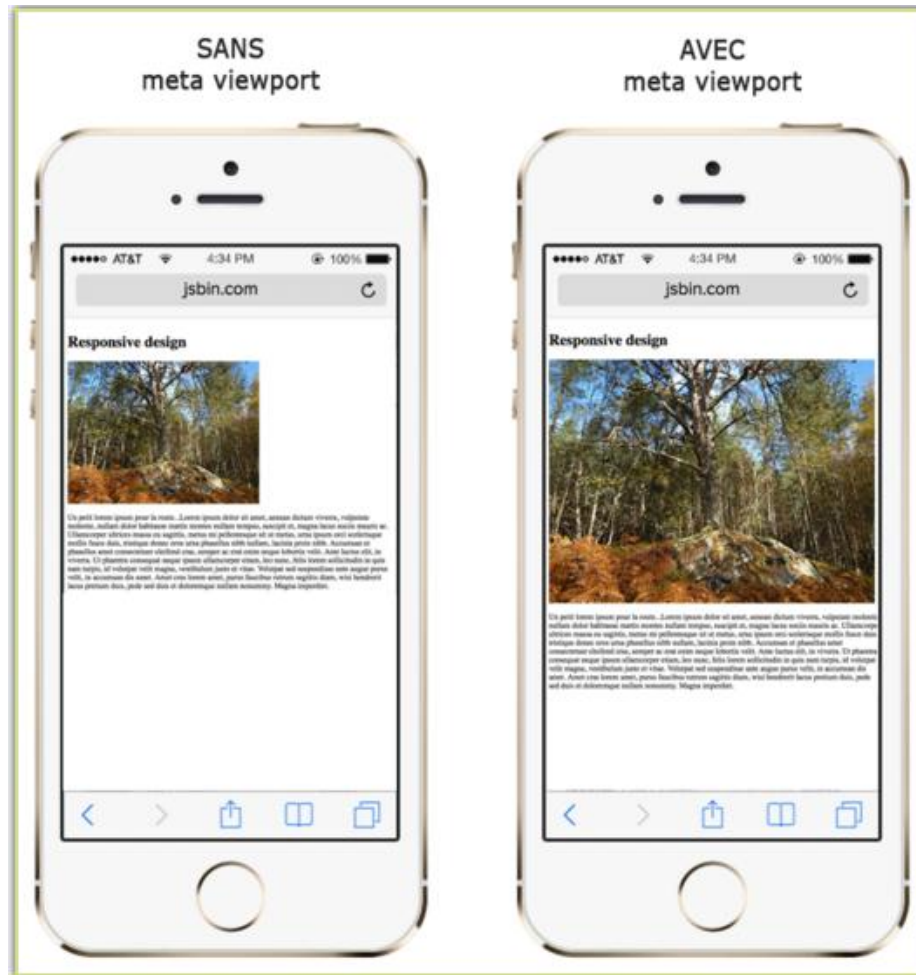
1^{ère} Solution : Ajustement automatique

- Garder la structure de la page et mettre en échelle (rétrécir) tous les éléments de manière proportionnelle afin de les rendre visibles sur un petit écran.
- Utilisation du **viewport** avec une balise **meta** :

```
<meta name =" viewport " content =" width =device –width, initial-scale =1.0 ">
```

- Le « **viewport** » correspond à la taille de la fenêtre web des utilisateurs.
- **meta viewport** : permet au navigateur de mieux contrôler la mise en échelle de la page web.
- **width=device-width** : établi que le width total de la page est équivalent à la largeur du viewport du dispositif.
- **initial-scale=1.0** : établi l'échelle initiale de la page sur un navigateur c.-à-d. le niveau de zoom initial lorsque la page est chargée par le navigateur.

1^{ère} Solution : Ajustement automatique



1ère Solution : Ajustement automatique

- Utiliser des largeurs relatives afin de mieux positionner et adapter les éléments à la largeur du viewport.
- Unités de longueur qui dépendent de la taille de la fenêtre :
 - **vw**: 1% de la largeur courante de la fenêtre (viewport width)
 - **vh**: 1% de la hauteur courante de la fenêtre (viewport height)
- On peut utiliser les propriétés CSS **max-width** et **min-width**
- Exemple:
 - pour que le body ne fasse jamais plus de 60 em de large :

```
body { max-width: 60em; }
```

- pour que les images ne dépassent jamais de leur conteneur (et donc ne soient jamais plus larges que l'écran) :

```
img { max-width: 100%; }
```


2^{ème} Solution: Requêtes média dans le HTML

```
<link rel="stylesheet" type="text/css" media="screen and (max-device-width: 480px)"  
      href="mobile.css"/>
```

→ Si l'appareil à une résolution inférieure ou égale à 480px en largeur, le fichier `mobile.css` sera lu.

Exemple:

- Quand la résolution devient petite, j'enlève la menu sur la gauche pour le placer en dessous du contenu.
- Faire disparaître certains éléments qui n'auraient par leur place sur un petit écran, avec un `display:none;`.

3^{ème} Solution: Les média Querys

Exp: Pour un affichage sur mobile:

- Afficher les différents éléments en longueur plutôt qu'en largeur.
- Ne pas afficher certains éléments pour des tailles d'écran trop petites



<https://mediaqueri.es/>

3^{ème} Solution: Les média Querys

- Ils permettent d'appliquer des propriétés de style sur des balises seulement dans le cas qu'une condition soit vérifiée.
- En général, ils sont utilisés pour vérifier les capacités du dispositif courant. Ils peuvent vérifier les propriétés suivantes :
 - Largeur et hauteur du viewport.
 - Largeur et hauteur du dispositif.
 - Orientation du viewport (landscape et portrait).
 - Résolution du dispositif.

Attention : Ils ont été introduits dans CSS3.



un titre de niveau 1

un paragraphe avec du texte important

un autre paragraphe

un titre de niveau 2

3^{ème} Solution: Les média Querys

```
<html>
<head>
<title> Les responsives </title>

<link rel="stylesheet" type="text/css" href="CSS-responsives.css"/>

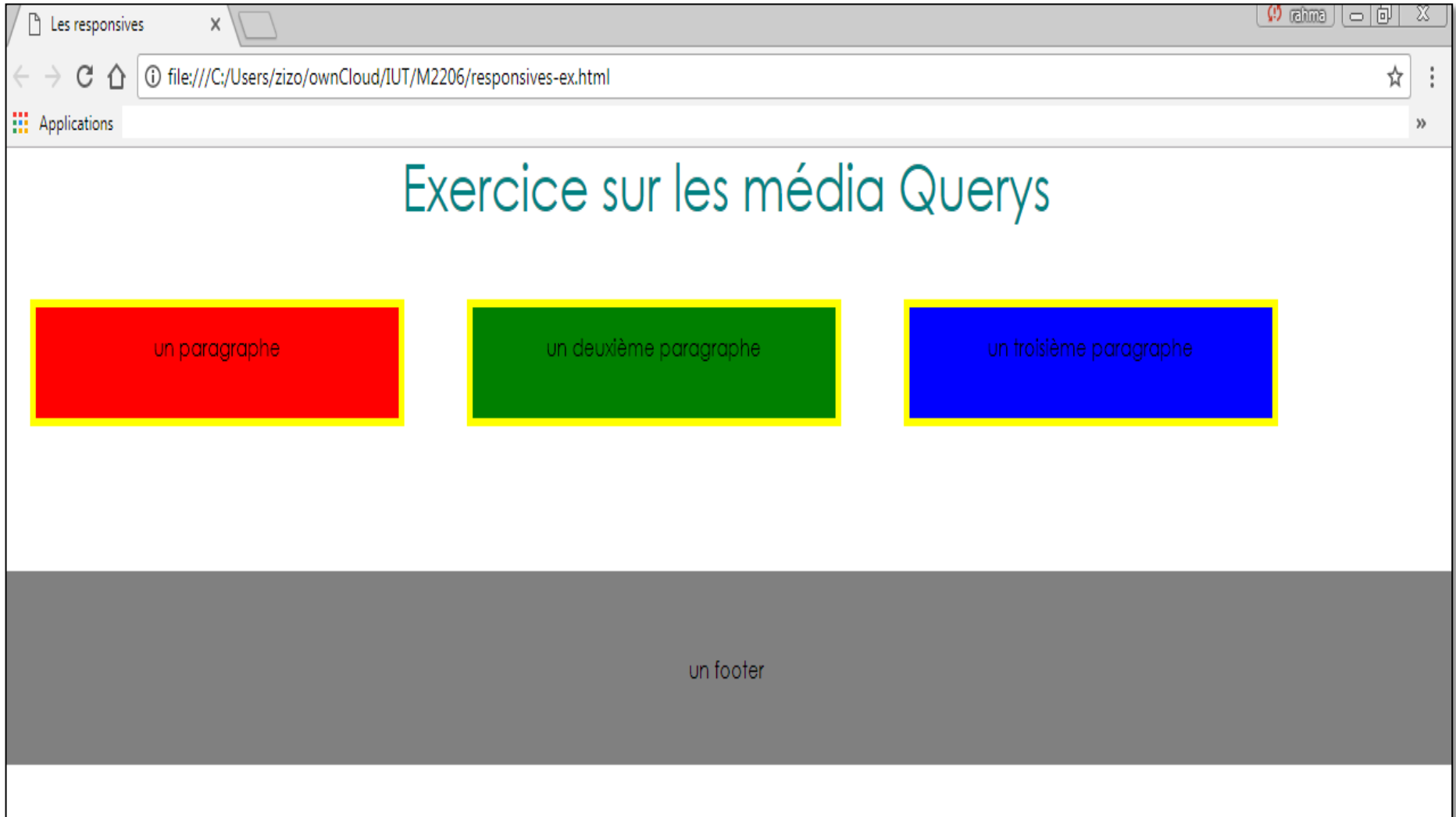
</head>
<body>
<h1> un
<p> un p
<p> un a
<h2> un
</body>
</html>

/* si la taille de l'écran d'un visiteur est inférieure à 480px, la couleur de fond
de body sera bleu */
@media screen and (max-width: 480px){
    body{
        background-color:blue;
    }
}

/* si la taille de l'écran d'un visiteur a une taille entre à 480px et 1179px, la couleur de fond
de body sera rouge */
@media screen and (min-width: 481px) and (max-width: 1179px){
    body{
        background-color:red;
    }
}

/* si la taille de l'écran d'un visiteur a une taille supérieure à 1179px, la couleur de fond
de body sera verte */
@media screen and (min-width: 1180px){
    body{
        background-color:green;
    }
}
```

Les média Querys : Exercice



Les média Querys : Exercice



Les média Querys : Exercice

```
<html>
<head>
<title> L
<link rel="stylesheet" href="style2.css"/>
</head>
<body>
<header>
  <h1>
</header>
  <div id="div1">
<p> un pa
</div>
  <div id="div2">
<p> un de
</div>
  <div id="div3">
<p> un tr
</div>
<footer>
  un foote
</footer>
</body>
</html>
```

```
@media screen and (max-width:780px){
  #div1,#div2,#div3{
    width:100%;
    border-right:none;
    border-left:none;
    margin-left:0px;
    margin-right:0px;
  }
  h1{
    text-align:left;
    margin-bottom:2%;
  }
  footer{
    display:none;
  }
}
```

```
body{
  font-family:'Century Gothic', Arial,serif;
  font-size:15px;
  margin:0px;
  padding:0px;
}
header{
  height:10%;
}
h1{
  color:#008080;
  font-size:40px;
  font-weight:normal;
  text-align:center;
}
#div1,#div2,#div3{
  display:inline-block;
  width:25%;
  height:10%;
  border: 5px solid yellow;
  margin: 2%;
  text-align:center;
}
#div1{
  background-color:red;
}
#div2{
  background-color:green;
}
#div3{
  background-color:blue;
}
footer{
  width:100%;
  height:10%;
  margin-top:5%;
  padding-top:4%;
  text-align:center;
  background-color: grey;
}
```


@media et orientation

On peut définir des règles spécifiques selon l'orientation de l'écran des visiteurs (utile pour l'affichage sur tablette ou smartphone entre autres).

Pour cela, on peut utiliser «**orientation : landscape**» pour définir des règles spécifiques pour les écrans tournés en mode «**paysage**».

@media screen and (orientation:landscape) {

```
/* les déclaration CSS */  
body {  
background - color : lightblue ;  
}
```

```
}
```

Le responsive et les images

Il est possible de charger des images différentes en dépendant du dispositif.

I. l'utilisation des médias **queries** :

```
/* Pour un viewport de moins de 400 px: */
@media only screen and (max-width:400px) {
  body {
    background-image : url('Petite_Image.jpg');
  }
}
/* Pour des viewports au-dela 400 px */
@media only screen and (min-width:400px){
  body {
    background-image : url('Grande_Image.jpg');
  }
}
```

Le responsive et les images

Il est possible de charger des images différentes en dépendant du dispositif.

2. l'utilisation de la balise **picture** :

```
<picture>
  <source srcset="Petite_Image.jpg" media="(max-width:700px)" >

  
</picture>
```

Exercice I

- Créer une page HTML avec un titre et un paragraphe (avec beaucoup de texte), puis modifier le CSS pour remplir les conditions suivantes; tester en redimensionnant la fenêtre.
- Attention, cet exercice utilise plusieurs techniques différentes (vues en cours), pas seulement des media queries !
 - La couleur de fond doit être rouge si la fenêtre fait moins de 800px de large, et bleue sinon.
 - Le paragraphe doit prendre toute la largeur de la fenêtre si celle-ci fait moins de **700 pixels** de large. Si elle est plus grande, il doit faire 700 pixels de largeur et être centré horizontalement.
 - La police du titre doit prendre 10% de la hauteur de la fenêtre, quelle qu'elle soit.

Si la fenêtre fait moins de 400px de largeur, le titre ne doit pas apparaître.
 - Le titre doit être en gras (non italique) si la fenêtre fait moins de 500px de large, en italique (non gras) si elle fait plus de 1000px de large, et en gras italique entre les deux.

Exercice I : Correction

```
/*==== Largeur des paragraphes =====*/

/* Le paragraphe doit prendre toute la largeur de la fenêtre si celle-ci
 * fait moins de 700 pixels de large. Si elle est plus grande, il doit faire
 * 700 pixels de largeur et être centré horizontalement.
 */
/*
 * NB: pas besoin de media query ici ! On dit simplement que les paragraphes
 * doivent faire au maximum 700 pixels de large.
 */
p {
    max-width: 700px;
    margin: auto;
}

/*==== Couleur de fond =====*/

/* La couleur de fond doit être rouge si la fenêtre
 * fait moins de 800px de large, et bleue sinon. */

html {
    background: navy;
    color: white;
}

@media (max-width: 800px) {
    html {
        background: darkred;
    }
}
```

Exercice I : Correction

```
/*==== Style du titre =====*/  
  
/* Le titre doit être en gras si la fenêtre fait moins de 500px de large,  
 * en italique si elle fait plus de 1000px de large,  
 * et en gras italique entre les deux. */  
  
/*  
 * Les propriétés sont indépendantes, donc on peut diviser l'énoncé en deux :  
 * - Titre en italique au-dessus de 500px  
 * - Titre en gras en-dessous de 1000px  
 *  
 * Mais attention, les titres sont en gras par défaut,  
 * donc la deuxième partie serait plutôt  
 * - Titre pas en gras au-dessus de 1000px  
 */  
  
@media (min-width: 500px) {  
    h2 {  
        font-style: italic;  
    }  
}  
  
@media (min-width: 1000px) {  
    h2 {  
        font-weight: normal;  
    }  
}  
  
/*==== Taille du titre =====*/  
  
/* La police du titre doit prendre 10% de  
 * la hauteur de la fenêtre, quelle qu'elle soit. */  
  
h2 {  
    font-size: 10vh;  
}  
  
/*==== Disparition du titre =====*/  
  
/* Si la fenêtre fait moins de 400px de largeur,  
 * le titre ne doit pas apparaître. */  
  
@media (max-width: 400px) {  
    h2 {  
        display: none;  
    }  
}
```

Exercice 2

- Créez votre fichier index.html comportant un div et un paragraphe à l'intérieur de celui-ci. Créez également votre fichier style.css.
- Dans ce dernier, voici les déclarations à y ajouter pour l'élément div:
 - Une taille (largeur) de 65%
 - Une couleur de fond gris clair
 - Une marge extérieure de 50px en haut et en bas, et automatique à gauche et droite
 - Une marge intérieure de 50px
 - Une taille de texte de 15px
- Et pour un affichage entre 0 et 700px, toujours sur le div:
 - Une couleur de fond #ff6666
 - Une taille de texte de 10px
 - Le texte en blanc

Exercice 2 : Correction

```
style.css

1  div {
2    width: 65%;
3    background-color: lightgrey;
4    margin: 50px auto;
5    padding: 50px;
6    font-size: 1.5em;
7  }
8
9  @media (max-width: 700px) {
10   div {
11     background-color: #ff6666;
12     font-size: 1em;
13     color: white;
14   }
15 }
```


Exercice 3

- Créez votre fichier index.html comportant un div et un paragraphe à l'intérieur de celui-ci. Créez également votre fichier style.css.
- Dans ce dernier, voici les déclarations à y ajouter pour l'élément div:
 - Une taille (largeur) de 50%
 - Une couleur de fond #ff6666
 - Une marge extérieure de 30px en haut et en bas, et automatique à gauche et droite
 - Une marge intérieure de 20px
 - Le texte en blanc
- Et pour un affichage entre 701px et plus, toujours sur le div:
 - Une taille (largeur) de 65%
 - Une couleur de fond gris clair
 - Une marge extérieure de 50px en haut et en bas, et automatique à gauche et droite
 - Une marge intérieure de 50px
 - Une taille de texte de 15px
 - Le texte en noir

Exercice 3 : Correction

style.css

```
1  div {
2    width: 50%;
3    background-color: #ff6666;
4    margin: 30px auto;
5    padding: 20px;
6    color: white;
7  }
8
9  @media (min-width: 701px) {
10   div {
11     width: 65%;
12     background-color: lightgrey;
13     margin: 50px auto;
14     padding: 50px;
15     font-size: 1.5em;
16     color: black;
17   }
18 }
```

Exercice 4

- Récupérer les fichiers de l'exercice « CV »: le fichier HTML, la feuille de style `style.css` déjà écrite, et créer une feuille de style **responsive.css**, que vous allez devoir remplir pour faire la mise en page et gérer le responsive.
- On va partir de la mise en page sur un grand écran, et adapter pour des écrans de plus en plus petits.
- La page doit se comporter de manière responsive en respectant les contraintes suivantes:
 - La page (c'est-à-dire le body) ne doit pas dépasser 1000 pixels de largeur; si l'écran est plus grand, des marges doivent apparaître autour de l'ensemble.
 - En-dessous de 1000 pixels de largeur, le menu doit être affiché verticalement, juste sous le header.
 - En-dessous de 900 pixels de largeur, la description de la section **profil** passe sous la photo. Pour la section **Parcours et expérience** la description de chaque expérience passe sous la dates.
 - En-dessous de 400 pixels de largeur, le menu disparaît et devienne une icône (on pourra mettre le mot « Menu » ou une image) tout en haut à gauche de l'écran.

Exercice 4

Menu

Zeineb EL khalfi

Docteur en informatique

Profil



Je m'appelle Zeineb El khalfi, j'ai 27 ans et je vis à Arles.

Je suis docteur en intelligence artificielle et actuellement, je suis ATER à l'université Aix Marseille

Zeineb EL khalfi

Docteur en informatique

- ACCUEIL
- EXPÉRIENCES PROFESSIONNELLES
- LOISIRS
- CONTACTEZ-MOI

Profil



Exercice 4 : Correction

```
/*
 * Le contenu ne doit pas dépasser 1000 pixels de largeur ;
 * si l'écran est plus grand, des marges doivent apparaître
 */
body {
    max-width: 1000px;
    margin: auto;
}

/*
 * En-dessous de 1000 pixels de largeur, le menu doit être affiché
 * verticalement.
 */
@media (max-width:1000px) {
    nav{
        width: 100%;
        text-align: center;
    }

    nav li{
        float: none;
    }

    nav a{
        display:block;
        padding: 3px;
        display: table;
    }
}
```

Exercice 4 : Correction

```
/*
 * En-dessous de 900 pixels de largeur, la description passe sous la l'image.
 */
@media (max-width: 900px) {
  .left{
    width: 100%;
  }

  .right{
    width: 100%;
  }
}

/*
 * En-dessous de 400 pixels de largeur, le menu disparaît et
 * devient une icône tout en haut à gauche de l'écran.
 */
@media (max-width: 400px) {
  /* on affiche le nav tout en haut à gauche de l'écran */
  .table{
    position: absolute;
    top: 0px;
    left: 0px;
    width: auto;
  }

  /* le menu lui-même disparaît, donc le nav est vide... */
  .table ul {
    display: none;
  }

  /* ... mais on rajoute un élément contenant Menu à la place ! */
  .table::before {
    content: "Menu";
  }

  .table: hover::after{
    content: " : Agrandir pour voir le menu";
  }
}
```