

## M1105 - Cours n°3

*Le langage CSS (niveau 2)  
Design d'un site web*



1

## Les principales propriétés CSS

Le cours de la semaine précédente (cours n°2)

- ① Propriétés de mise en forme du texte
- ② Propriétés de couleur et de fond
- ③ Propriétés des listes
- ④ Propriétés des tableaux
- ⑤ Propriétés des liens (pseudo-classes)

Aujourd'hui (cours n°3)

- ⑥ Propriétés des boîtes  
*width, height, margin, margin-left, margin-right, ... padding, border,*
- ⑦ Propriétés de positionnement et d'affichage des boîtes  
*display, float, clear, overflow, position, ...*

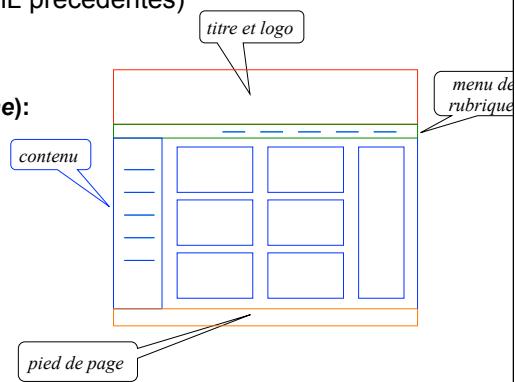
2

## I - Une page web = des boîtes qui s'emboîtent !

Structuration d' une page en plusieurs parties (ou boîtes/divisions) :

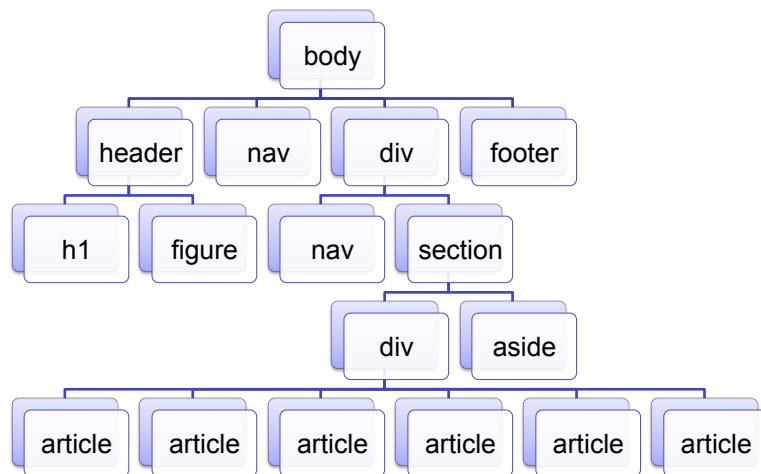
- Construction de la page à partir des éléments HTML5 : *header, nav, section, aside, article, footer*
- (ou construction à partir de l'élément générique *div* pour les générations HTML/XHTML précédentes)

Exemple de **maquette filaire (wireframe)**:  
schéma en général « **fil de fer** »,  
pour représenter les principaux blocs  
et leur disposition dans la page



3

## Exemple de l'arbre pour la maquette



4

## Le modèle de boîte "classique"

- Tous les éléments HTML définissent des **boîtes**.
- Mais... certains sont dit « éléments de type **bloc** » et d'autres sont dit « éléments de type **en ligne** ».
  
- **Eléments de type bloc (*block*)**  
`<header> <section> <footer> <aside> <article> <div> <h1> <h2> <h3> <h4> <h5> <h6> <p> <table> <tr> <td> <th> <ol> <ul> <li> <dd> <dl> <dt> <form> <pre> <address> <blockquote> <hr /> ...`
  
- **Eléments de type en-ligne (*inline*)**  
`<img> <a> <em> <strong> <kbd> <br /> <cite> <sub> <sup> <span>...`

5

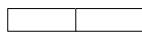
## II - Elément bloc ? en-ligne ? Quelle différence?

Les éléments, s'ils sont de type **bloc** ou s'ils sont de type **en-ligne** se positionnent différemment les uns par rapport aux autres:

- **Éléments « bloc/block »** : Ces éléments se positionnent les uns sous les autres ; ils peuvent contenir d'autres éléments « bloc » ou des éléments « en-ligne ».



- **Éléments « en-ligne/inline »** : Ces éléments se positionnent les uns à côté des autres ; ils peuvent s'imbriquer entre eux mais ne peuvent pas contenir d'éléments « bloc ».

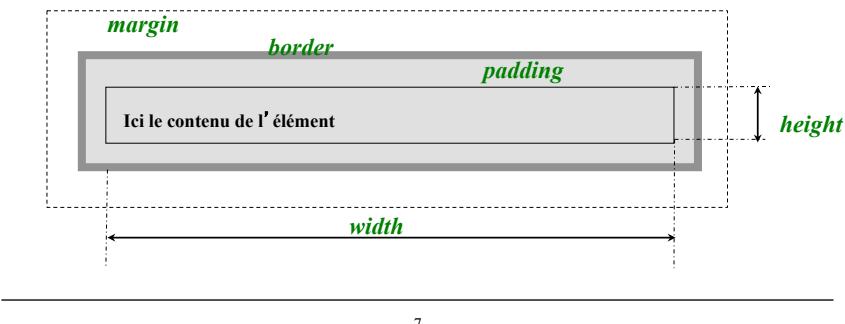


6

### III – Le modèle de boîte "classique"

Tout élément HTML est donc une « boîte » et possède:

- **Un contenu** (de largeur **width**)
- **Une marge interne** (**padding**)
- **Une marge externe** (**margin**)
- **Une bordure** (**border**)

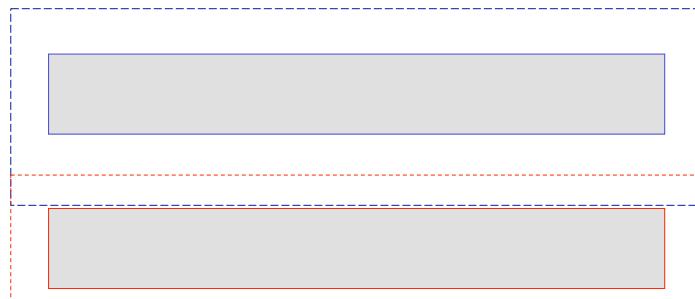


7

### Les marges extérieures (margin) des éléments

Remarques:

- Le navigateur met des marges par défaut à certains éléments (par exemple pour les éléments p, figure, h1, h2,...)
- Attention aux marges entre les boîtes! (max)



8

## Largeur / Hauteur d'un élément?

Par convention:

- **Largeur de l'élément = width** (largeur du contenu) + **left padding** (marge intérieure gauche) + **right padding** (marge intérieure droite) + **left border** + **right border**



- **Idem pour hauteur**

9

## Exemple avec un paragraphe

The screenshot shows a browser window with developer tools open. At the top, there is a pink rectangular box labeled "Un paragraphe". Below it is a larger purple rectangular box also labeled "Un paragraphe". A callout bubble points to the purple box with the text "p 883x126". At the bottom of the screen, the browser's inspection panel is visible, showing the HTML structure and a detailed "Style" panel for the paragraph element. The style panel displays various properties like border, padding, and width.

10

## IV - Les principales propriétés CSS (niveau 2)

---

### ⑥ Propriétés des boîtes

*width, height, margin, margin-left, margin-right, ...  
padding, border,*

### ⑦ Propriétés de positionnement et d'affichage des boîtes

*display, overflow, float, clear, position, ...*

11

## ⑥ Propriétés des boîtes

---

Propriété	Valeurs (exemples)	Description
width	150px, 80...	Largeur
height	150px, 80...	Hauteur
min-width	150px, 80...	Largeur minimale
max-width	150px, 80...	Largeur maximale
min-height	150px, 80...	Hauteur minimale
max-height	150px, 80...	Hauteur maximale
margin-top	23px	Marge en haut
margin-left	23px	Marge à gauche
margin-right	23px	Marge à droite
margin-bottom	23px	Marge en bas
	23px 5px 23px 5px (haut, droite, bas, gauche)	Super-propriété de marge.
margin		Combine : margin-top, margin-right, margin-bottom, margin-left.

Tableau extrait du site <http://www.siteduzero.com>

12

## ⑥ Propriétés des boîtes (suite)

Tableau extrait du site <http://www.siteduzero.com>

Propriété	Valeurs (exemples)	Description
padding-left	23px	Marge intérieure à gauche
padding-right	23px	Marge intérieure à droite
padding-bottom	23px	Marge intérieure en bas
padding-top	23px	Marge intérieure en haut
padding	23px 5px 23px 5px (haut, droite, bas, gauche)	Super-propriété de marge intérieure.
border-width	3px	Combine : padding-top, padding-right, padding-bottom, padding-left.
border-color	nom, rgb(rouge,vert,bleu), rgba(rouge,vert,bleu,transparence), #CF1A20...	Épaisseur de bordure
border-style	solid, dotted, dashed, double, groove, ridge, inset, outset	Couleur de bordure
border	3px solid black	Type de bordure
		Super-propriété de bordure. Combine border-width, border-color, border-style.
border-radius	5px	Existe aussi en version border-top, border-right, border-bottom, border-left.
box-shadow	6px 6px 0px black (horizontale, verticale, fondu, couleur)	Bordure arrondie
		Ombre de boîte

13

## ⑥ Exemples pour les marges: margin/padding

### Marges intérieures :

```
p {                                     ou      p {  
    padding: 10px 10px 10px 10px;          padding: 10px;  
}                                         }           }
```

### Marges extérieures :

```
article {                                ou      article {  
    margin: 10px 15px 10px 15px;          margin: 10px 15px;  
}                                         }           }
```

### D'autres exemples :

```
section {                                section {  
    margin-left: 100px;                  margin-bottom: 20px;  
}                                         padding: 15px;  
}                                         }
```

14

## ⑥ Exemples pour les bordures: border

Différents styles de bordure: solid, dotted, dashed, double, groove, ridge, inset, outset

```
p {  
    border-style:solid;  
    border-width:1px;  
    border-color:red;  
}
```

*ou*

```
p {  
    border: 1px solid red;  
}
```

Bordures arrondies :

```
p {  
    border-radius:10px 10px 10px 10px;      ou      p {border-radius: 10px;}
```

Ombre : décalage horizontal de l'ombre, décalage vertical, adoucissement, couleur

```
p {  
    border-shadow: 5px 5px 5px blue;  
}
```

15

## ⑦ Propriétés de positionnement et d'affichage des boîtes

Tableau extrait du site <http://www.siteduzero.com>

Propriété	Valeurs (exemples)	Description
display	block, inline, flex, inline-block, table, table-cell, none...	Type d'élément (block, inline, inline-block, none...)
visibility	visible, hidden	Visibilité
clip	rect (0px, 60px, 30px, 0px) rect (haut, droite, bas, gauche)	Affichage d'une partie de l'élément
overflow	auto, scroll, visible, hidden	Comportement en cas de dépassement
float	left, right, none	Flottant
clear	left, right, both, none	Arrêt d'un flottant
position	relative, absolute, static	Positionnement
top	20px	Position par rapport au haut
bottom	20px	Position par rapport au bas
left	20px	Position par rapport à la gauche
right	20px	Position par rapport à la droite
z-index	10	Ordre d'affichage en cas de superposition.  La plus grande valeur est affichée par-dessus les autres.

16

## ⑦ Quelques exemples

---

```

Exemples :          nav ul{
nav ul li {           display: flex;
    display: inline-block; }
}
article {           img
{
    width: 150px;   position: absolute;
    height: 150px;  top: 30px;
    overflow: scroll; right: 30px;
}
}
img {               footer{
    float: right;   clear: right;
}
}

```

---

17

## V - Principe du positionnement des éléments

Si on ne précise rien:

- Les éléments de type « bloc » (div, section, article, aside, header, footer, nav, p, h1, h2,..., ul, ol, li, table,...) s’affichent les uns en dessous des autres et prennent toute la largeur
- Les éléments de type « en-ligne » se placent au fil du texte

Comment positionner les éléments les uns par rapport aux autres?

par exemple :

- pour créer un menu sur le côté,
- pour que le corps de la page occupe 80% de l’ espace,
- pour faire une zone d’ images
- ...

=> Utilisation les techniques de positionnement CSS

18

## Quel positionnement ?

---

7 catégories:

1. ( Positionnement en flux )
  2. Positionnement « inline-block »
  3. Positionnement « flexbox »
  4. Positionnement « relative »
  5. Positionnement « absolute »
  6. Positionnement « fixed »
  7. Positionnement « float »
- 

19

## Le positionnement CSS – flux (normal)

---

### 1. ( Positionnement en flux )

- Lorsque aucun positionnement n'est spécifié, les éléments se placent dans la page selon l'ordre dans lequel ils apparaissent et en fonction de leur type (bloc ou en-ligne)
- C'est le flux « normal » !

Rappel:

- **Éléments « bloc/block » :** Ces éléments se positionnent les uns sous les autres  
ex: h1, h2,..., p, ul, ol, table, section, article, div,...
  - **Éléments « en-ligne/inline » :** Ces éléments se positionnent les uns à côté des autres  
ex: img, a, span, em, strong, br,...
- 

20

## Le positionnement CSS – "inline-block"

### 2. Positionnement « inline-block »

**display: inline-block;**

- Les éléments de type inline-block se positionnent les uns à côté des autres (comme les éléments inline) et on peut leur donner des dimensions (comme les éléments blocs)
- Est utilisé pour faire le positionnement des boîtes
- Aujourd'hui, on préfère le positionnement "flex"

21

## Exemple positionnement « inline-block »

```
<body>
<article>
    <h2>Photo n°1</h2>
    
</article>
<article>
    <h2>Photo n°2</h2>
    
</article>
<article>
    <h2>Photo n°3</h2>
    
</article>
</body>

article:nth-child(1){
    width:200px;
    border:1px solid #DDD;
    padding:10px;
    display:inline-block;
}

article:nth-child(2){
    width:300px;
    border:1px solid #DDD;
    padding:10px;
    display:inline-block;
}

article:nth-child(3){
    width:400px;
    border:1px solid #DDD;
    padding:10px;
    display:inline-block;
}
```

22

---

**Photo n°1**



**Photo n°2**



**Photo n°3**



---

23

## Exemple (2) positionnement « inline-block »

---

```
body article:nth-child(1){  
width:200px;  
border:1px solid #DDD;  
padding:10px;  
display:inline-block;  
vertical-align:top;}  
  
body article:nth-child(2){  
width:300px;  
border:1px solid #DDD;  
padding:10px;  
display:inline-block;  
vertical-align:top;}  
  
body article:nth-child(3){  
width:400px;  
border:1px solid #DDD;  
padding:10px;  
display:inline-block;  
vertical-align:top;}
```

**Photo n°1**



**Photo n°2**



**Photo n°3**



---

24

## Attention: inline-block et white-space!

Il existe un espace entre les éléments de type inline-block...

Voir article :

<http://www.alsacreations.com/astuce/lire/1432-display-inline-block-espaces-indesirables.html>

Solution (exemple précédent) :

```
<article>
  ...
</article><!--
--><article>
  ...
</article><!--
--><article>
  ...
</article>
```



Démo

25

## Le positionnement CSS – flexbox CSS3

### 3. Positionnement "flexbox"

#### display : flex

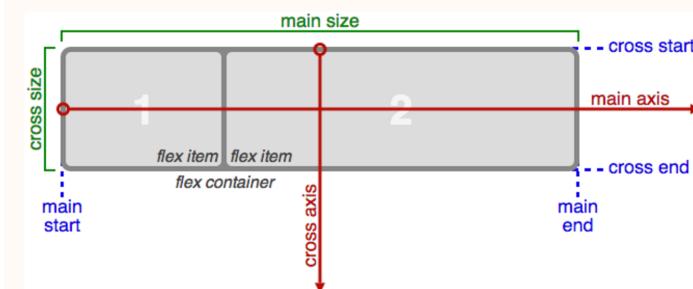
- un nouveau modèle de boîte : **le modèle de boîte flexible**
- ≠ du modèle de boîte classique et des positionnements vus précédemment
- principe:
  - Un **conteneur flex** (**flex container**) en **display : flex | inline-flex**
  - Des **éléments flex** (**flex items**) à l'intérieur du conteneur, que l'on peut agencer de différentes façons
  - propriétés : flex, flex-basis, flex-grow, flex-shrink, flex-flow, flex-direction, flex-wrap, align-items, align-self, align-content, justify-content et order.

⇒ Un mode de positionnement récent (attention aux différents navigateurs): **W3C Candidate Recommendation**, 19 October 2017 : <https://www.w3.org/TR/css-flexbox-1/>

26

## flexbox CSS3 : principe

- Le modèle de boîte classique : le positionnement est basé sur des directions de flux *block*, *inline*, *inline-block*
- Le modèle de boîte flexible** : le positionnement flex est basé sur les directions *flex-flow*

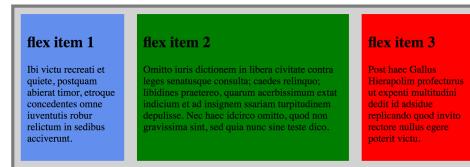


27

## Exemple de positionnement flex

```
<div id="flex-container">
  <div class="flex-item">
    <h2>flex item 1</h2>
    <p>Ibi victu ... </p>
  </div>
  <div class="flex-item">
    <h2>flex item 2</h2>
    <p>Omitto ... </p>
  </div>
  <div class="flex-item">
    <h2>flex item 3</h2>
    <p>Post haec ...</p>
  </div>
</div>
```

Essai positionnement flex



28

## Exemple de positionnement flex

```
#flex-container {
```

```
  display: flex;
```

```
  background-color: lightgrey;
  border: 1px solid gray;}
```

```
.flex-item {
```

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

```
.flex-item:nth-child(1) {
```

```
  background-color: cornflowerblue;}
```

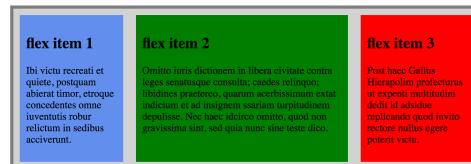
```
.flex-item:nth-child(2) {
```

```
  background-color: green;}
```

```
.flex-item:nth-child(3) {
```

```
  background-color: red;}
```

### Essai positionnement flex



29

## Principales propriétés - au niveau conteneur (1)

### 1. **flex-direction : row | row-reverse | column | column-reverse**

- définit la **direction de l'axe principal** du conteneur flex, le long duquel les éléments-flex sont disposés.
- spécifie si les éléments-flex sont affichés de droite à gauche, de gauche à droite, de haut en bas ou de bas en haut.

### 2. **flex-wrap : nowrap | wrap | wrap-reverse**

- Spécifie la façon dont les éléments-flex vont dynamiquement s'agencer selon la taille de la fenêtre du navigateur

Propriété raccourci : **flex-flow : <'flex-direction'> || <'flex-wrap'>**

- par défaut: **: row nowrap;**

Démo

30

## Propriétés d'alignement – au niveau conteneur (2)

3. **justify-content: flex-start | flex-end | center | space-between | space-around**
  - contrôle **l'alignement** de tous les éléments-flex sur **l'axe principal**
4. **align-items: stretch | flex-start | flex-end | center | baseline**
  - contrôle **l'alignement** de tous les éléments-flex sur **l'axe secondaire (cross axis)**
5. **align-content : normal | stretch | flex-start | flex-end | center | space-between | space-around | baseline | ...**
  - Similaire à align-items mais au lieu d'aligner les éléments-flex, aligne les lignes. (plusieurs lignes nécessaires)

 Démo

31

## Propriétés – au niveau d'un élément-flex

6. **align-self : stretch | flex-start | flex-end | center | baseline**
  - contrôle **l'alignement d'un élément-flex particulier** le long de l'axe secondaire
7. **Order : 1 | 2 | 3 | ...**
  - Pour cibler des éléments-flex en particulier et modifier leur ordre de visualisation

 Démo

32

## Flexibilité des éléments-flex

### 8. `flex: <'flex-grow'> || <'flex-shrink'> || <'flex-basis'>`

- **flex-grow** : quelle proportion de l'espace libre peut-on allouer en supplément à l'élément-flex ?
- **flex-shrink** : quelle proportion de l'espace peut être retirée à l'élément-flex ?
- **flex-basis** : quelle est la taille de l'élément-flex avant tout agrandissement/réduction ?
- par défaut les éléments-flex occupent la taille minimale de leur contenu (**0 1 auto**)

⇒ Contrôle **la taille des éléments-flex le long de l'axe principal**

Quelques exemples intéressants :

- **flex: 1 1 0; (ou 1 1 0px)** flexible, taille identique
- **flex: 0 1 200px;** non flexible, taille de 200px

 Démo

33

## Pour aller plus loin

Positionnement flex: simple et très pratiques pour les dispositions "classiques", mais peut être compliqué à utilisé car très puissant

- <http://www.alsacreations.com/tuto/lire/1493-css3-flexbox-layout-module.html>
- [https://developer.mozilla.org/fr/docs/Web/CSS/CSS\\_Flexible\\_Box\\_Layout/](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Flexible_Box_Layout/)
- <https://openclassrooms.com/courses/apprenez-a-creer-votre-site-web-avec-html5-et-css3/la-mise-en-page-avec-flexbox>
- <http://flexboxfroggy.com/>  
(a game where you help Froggy and friends by writing CSS code)!

34

## Quel positionnement ?

7 catégories:

1. Positionnement en flux
2. Positionnement « inline-block »
3. Positionnement « flexbox »
4. Positionnement « relative »
5. Positionnement « absolute »
6. Positionnement « fixed »
7. Positionnement « float »

35

## Le positionnement CSS – relatif

### 4. Positionnement « relative »

`position: relative;`

et il faut préciser le décalage : top, bottom, left, right

- un contenu en flux normal peut être décalé horizontalement et/ou verticalement. Le contenu suivant n'est pas affecté par ce déplacement.

36

## Exemple positionnement « relative »

```

<p>                                p { background : #00FF00;
    Un bloc normal                  margin:0;
</p>                                width: 200px;}
<p class="test">                    .test { background:#FF0000;
    Un bloc de test                position: relative;
</p>                                top: 0.5em; }
<p>                                Un autre bloc normal
</p>                                
```

Démo →

37

## Le positionnement CSS – absolu

### 5. Positionnement « absolute »

**position: absolute;**

et on précise *top, bottom, left, right*

- L’ élément est placé (top, right, bottom, left) n’importe où dans la page! Il est **totalement retiré du flux normal**.
- Il est positionné par rapport au dernier parent ou ancêtre positionné (en absolu ou en relatif), ou par rapport aux limites de la zone de visualisation (la fenêtre du navigateur) s’il n’y a pas de parent ou d’ancêtre positionné.

38

## Le positionnement CSS –fixé

### 6. Positionnement « fixed »

**position: fixed;**

et on précise *top, bottom, left, right*

- Comme en positionnement absolu, l'élément est totalelement retiré du flux normal.
- Mais il est cette fois positionné uniquement par rapport aux limites de la zone de visualisation. Le défilement de la page n'a aucun effet sur un contenu en position fixée.

=> **Attention !** Les positionnements "absolute" et "fixed" sont à utiliser uniquement dans des cas bien précis (voir exemple ci-après)

Démo

39

## Exemple positionnement « relative/absolute » combiné

```
<article>
  
  <div>
    Photo n°1
  </div>
</article>
```

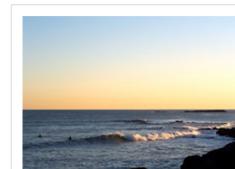


Photo n°1

```
article {
  position: relative;
  width: 200px;
  border: 1px solid #DDD;
  padding: 10px;
  margin: 10px;
}
div {
  position: absolute;
  left: 10px;
  top: 10px;
}
```



40

## Le positionnement CSS – flottant et enligne

### 7. Positionnement « float »

#### **float,: left ; (ou right)**

et on utilise `clear : left (ou right)` sur l'élément suivant pour "arrêter" le flottant précédent

- Positionne un élément à droite ou à gauche dans son parent
- Permet à un élément de « flotter », une image par exemple avec du texte qui l'entoure
- A utiliser dans des cas bien précis
- Difficile à maîtriser !

41

## Exemple positionnement « float »

Faire « flotter » une image avec du texte l'entoure :

```
<p>
  
  Le Lorem Ipsum est simplement du faux texte employé dans la composition et la
  mise en page avant impression...
</p>
```

```
img {
  float: left;
  margin-right:10px;
  margin-bottom:5px;}
```



Le Lorem Ipsum est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte. Il n'a pas fait que survivre cinq siècles, mais s'est aussi adapté à la bureautique informatique, sans que son contenu n'en soit modifié. Il a été popularisé dans les années 1960 grâce à la vente de feuilles Letraset contenant des passages du Lorem Ipsum, et, plus récemment, par son inclusion dans des applications de mise en page de texte, comme Aldus PageMaker. (<http://fr.lipsum.com/>).

Démo

42

## Quelques conseils "positionnement"

1. Pour positionner les grandes divisions d'une page (faire la mise en page / layout):
  - Conseillé :
    - le positionnement flex (`display: flex`) qui est l'avenir!
2. le positionnement inline-block (`display: inline-block`) mais attention au "white space problem" et ne pas oublier `vertical-align:top` qui est souvent nécessaire
3. Le positionnement « relative/absolute » peut se révéler très pratique pour des "incrustations"
4. Le positionnement « float » est à utiliser dans des cas particuliers, par exemple pour une image avec du texte qui l'entoure

43

## Conclusion CSS

Cascading style sheets (CSS) : feuilles de style en cascade

- CSS est un langage de définition de style
- Une feuille de style est une collection de règles de style s'appliquant sur les éléments HTML d'une page. Ces règles sont prioritaires par rapport aux règles de style des navigateurs
- Les propriétés CSS sont héritées en cascade : attention! C'est compliqué et pièges! (voir cours suivant)
- On utilise 1 ou plusieurs feuilles de style pour spécifier les caractéristiques de la page web.

44

## Conclusion CSS

---

Avantages des feuilles de style:

- La garantie d'une présentation homogène, cohérente de son site
  - La possibilité de changer globalement la présentation d'un site en modifiant uniquement le (les) fichier(s) CSS!
  - Des fonctionnalités puissantes permettant une grande variété de designs
  - La (quasi!) compatibilité multi-navigateurs  
⇒ attention... les navigateurs n' interprètent pas encore tout à fait de la même manière les spécifications CSS... (en particulier le positionnement "flex")
- 

45

## Conclusion CSS

---

- Beaucoup de nouveautés pour CSS3
- CSS3 est en cours d'évolution
- Un « bon » site web a plusieurs feuilles de style en fonction des médias, en particulier:
  - **écran**  
`<link rel="stylesheet" type="text/css" href="style.css" media="screen" />`
  - **Imprimante**  
`<link rel="stylesheet" type="text/css" href="styleprint.css" media="print" />`



Pour les feuilles de style pour imprimante, voir:  
<http://www.alsacreations.com/tuto/lire/586-feuille-css-impression.html>

46

## Exemples du site [www.csszengarden.com](http://www.csszengarden.com)

Un unique et même fichier HTML...  
... Une grande variété de designs



Floral Touch  
JADAS JIMMY, France



Elegance in Simplicity  
MANI SHERIAR, United States



Dazzling Beauty  
DENY SRI SUPRIYONO,  
Indonesia



Dark Rose  
ROSE FU, United States



Leggo My Ego  
JON TAN, United Kingdom



LuGoZee  
VIALLON PIERRE-ANTOINE,  
France

47

## INFORMATION

Séance de cours M1105 de la semaine prochaine

**Fin cours 3**  
**+ cours 4 en "cours inversé"**



- ⇒ **Chez vous** : lire et étudier le cours n°4 (en ligne), trouver des compléments d'informations si besoin
- ⇒ **En séance** : QCM, temps pour interagir et travailler ensemble: résumé, points importants à retenir, questions/réponses...

48