HTML 5 et CSS 3

Sacha RESTOUEIX

Principe du Web

Définitions

Réseau : machines connectées ensemble

Internet : réseau connectant tous les réseaux

Web : application permettant de consulter des pages Web à l'aide d'un navigateur.

- Caractérisé par des hyperliens
- Distinction entre client et serveur
- Utilise Internet si nécessaire

Navigateur : logiciel de visualisation de pages Web

HTML : langage de balises permettant d’écrire des pages Web

Qu'est-ce qu'une URL (Uniform Resource Locator ou Adresse Web) ?

```
protocole://adresse_site/chemin_repertoire/fichier
```

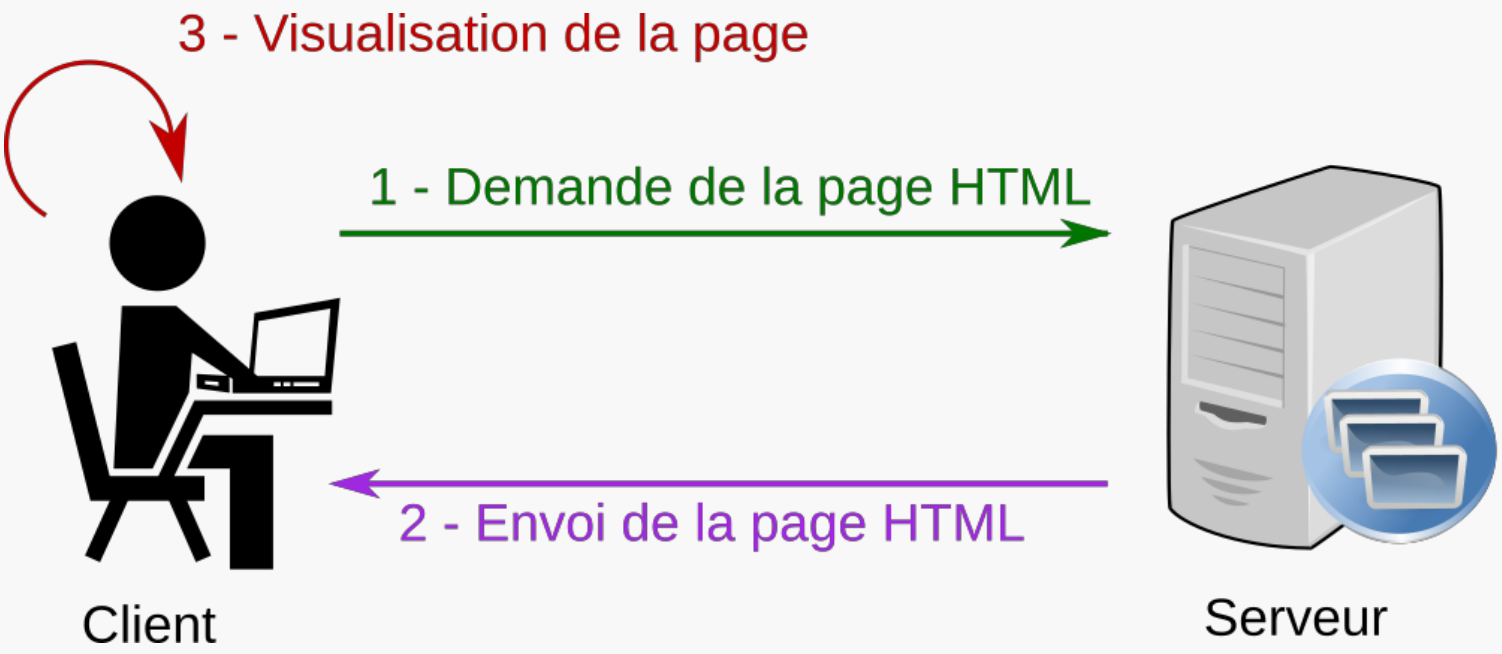
- **protocole :** langage utilisé pour la communication entre ordinateurs. Protocole pour le Web : **http** (**H**yper **T**ext **T**ransfer **P**rotocol)
- **adresse_site :** Adresse du serveur (numéro IP ou nom de domaine) contenant le fichier
- **chemin_repertoire :** répertoire où se trouve le fichier sur le serveur
- **fichier :** nom du fichier que l'on veut afficher

Exemples d'url :

```
http://afpa.fr/~restoueixsacha/Documents/Web/interaction.html
http://168.240.21.38/index.php

http://www.facebook.com/
```

Fonctionnement du Web



Fonctionnement du Web

Étape 1

```
GET /~restoueixsacha/Documents/Web/interaction.html HTTP/1.1
Host: afpa.brive.fr
```

Fonctionnement du Web

Étape 2

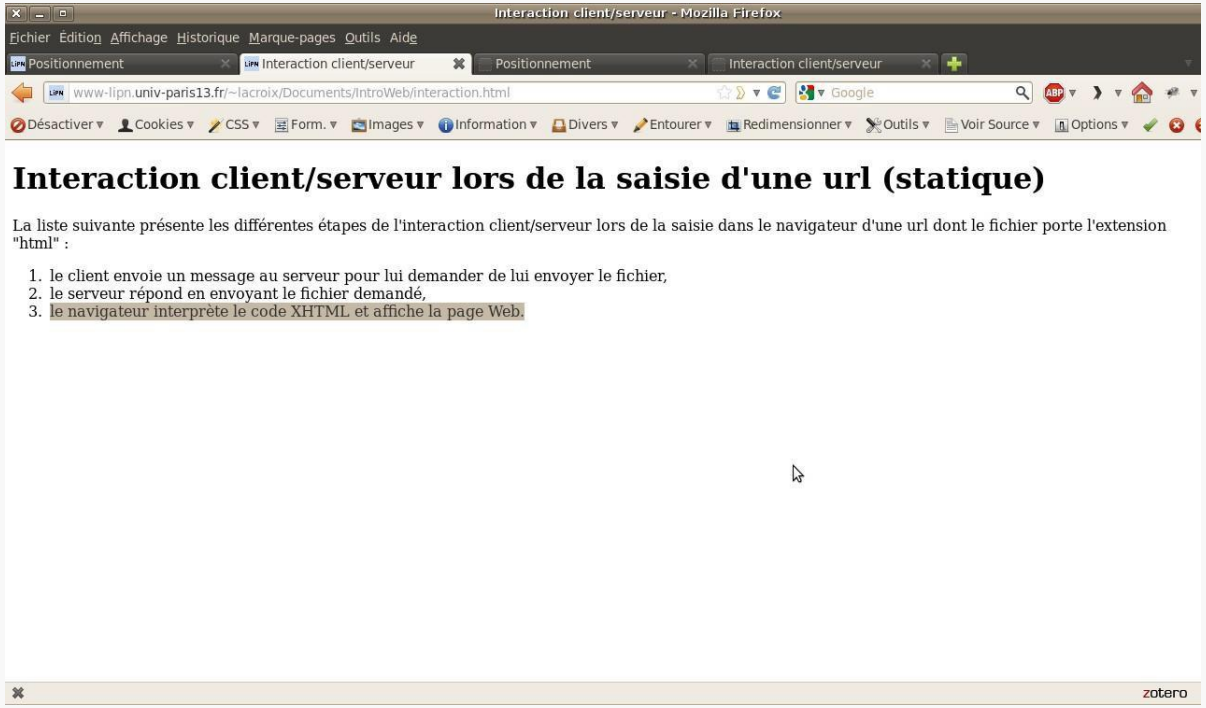
```
HTTP/1.1 200 OK
Content-Type: text/html

<!DOCTYPE html>
<html lang="fr">
<head>
    <title>Interaction client/serveur</title>
    <meta charset="utf-8">
</head>
<body>
<h1> Interaction client/serveur lors de la saisie d'une url
  (statique) </h1>

...
```

Fonctionnement du Web

Étape 3



Langage HTML

Principe

- Définit la structure d'une page Web :
  - Paragraphes,
  - titres,
  - etc
- Ne fait pas la mise en forme de la page
- Langage de balises

```
<balise attribut1="valeur1" attribut2="valeur2"> ... </balise>
<balise attribut1="valeur1" attribut2="valeur2" />
```

Les principales balises

Je suis un titre de niveau 1

Je suis un titre de niveau 2 (titre de sous-section)

On peut aller jusqu'à des titres de niveau 6

Un paragraphe avec *emphase faible* et **emphase forte** et un retour
à la ligne. On trace ensuite une ligne horizontale :

- liste non ordonnée, item 1
- liste non ordonnée, item 2

1. liste ordonnée, item 1
2. liste ordonnée, item 2

Paragraphe avec une image 

On peut insérer des hyperliens pointant sur des pages Web/documents du site même site (chemin relatif) ou sur un autre serveur (chemin absolu).

On peut aussi insérer des tableaux :

Balise	Signification
p	paragraphe
h1	Titre de niveau 1
hr	ligne horizontale

Les principales balises

```
<h1>Je suis un titre de niveau 1</h1>
<h2>Je suis un titre de niveau 2 (titre de sous-section)</h2>
<h6>On peut aller jusqu'à des titres de niveau 6</h6>
<p>Un paragraphe avec <em>emphase faible</em> et <strong>emphase forte</strong>
et un retour <br/> à la ligne. On trace ensuite une ligne horizontale :</p>
<hr/>
<ul>
    <li>liste non ordonnée, item 1</li>
    <li>liste non ordonnée, item 2</li>
</ul>
<ol>
    <li>liste ordonnée, item 1</li>
    <li>liste ordonnée, item 2</li>
</ol>
```

Les principales balises

```
<p>Paragraphe avec une image
<img src="img/requeteHtml_petite.png" alt="requete HTML"></p>
<!-- Ici un commentaire -->
<p>On peut insérer des <a href="https://afpa.brive.fr/">hyperliens</a> pointant sur des
pages Web/documents du site même site (chemin relatif) ou sur un autre serveur
(chemin absolu).</p>

<p>On peut aussi insérer des tableaux :</p>
<table>
    <tr>  <th>Balise</th>  <th>Signification</th>      </tr>
    <tr>  <td>p</td>       <td>paragraphe</td>         </tr>
    <tr>  <td>h1</td>      <td>Titre de niveau 1</td>  </tr>
    <tr>  <td>hr</td>      <td>ligne horizontale</td>  </tr>
</table>
```

Les principales balises

Pour diviser logiquement une page :

- **`article`** : regroupe un contenu ayant un sens propre indépendant
  Exemple : article de journal, commentaire, etc.
- **section** : regroupe des éléments ayant une même thématique.
- **header** : en-tête d'article, section, page, etc.
- **footer** : pied-de-page d'article, section, page, etc.
- **main** : partie principale de la page Web (1 seul)
- **nav** : menu de navigation.

Structure d'une page Web

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <title>TITRE OBLIGATOIRE</title>
    <meta charset="utf-8" />
  </head>
  <body>
    ... code HTML de la page (titres, paragraphes, etc)...
  </body>
</html>
```

Représentation en arbre

```
<balise1> ... <balise2>  .... </balise2> ... </balise1>
```

Ascendants/descendants

- `balise2` est *descendant* de `balise1`
- `balise1` est *ascendant* de `balise2`

Enfants/parents

- `balise2` est *enfant* de `balise1` s'il est descendant de `balise1` et qu'il n'existe pas d'élément ascendant de `balise2` et descendant de `balise1`
- `balise1` est parent de `balise2` si cette dernière est enfant de `balise1`

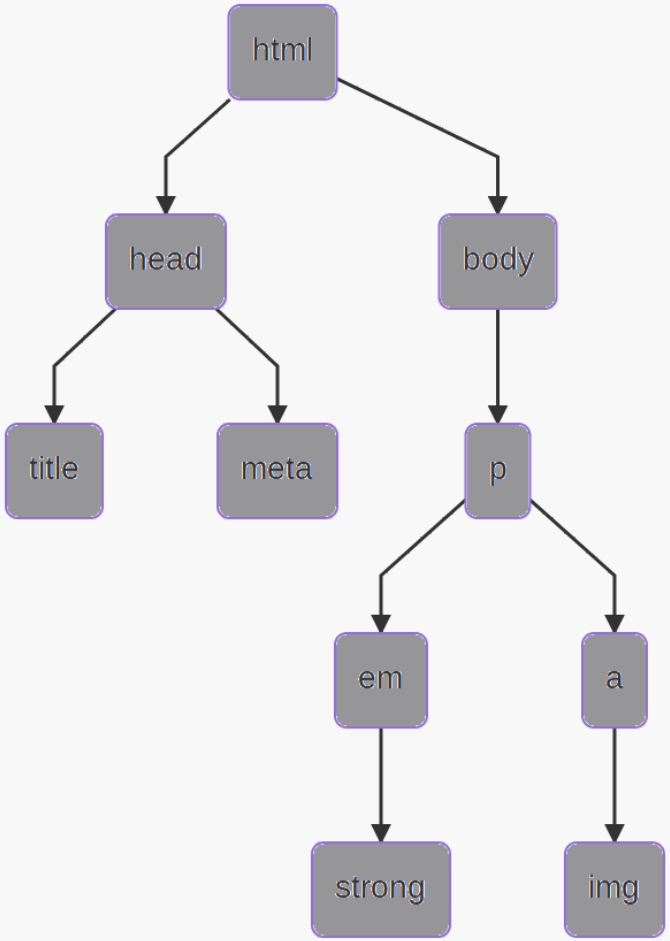
```
<p> texte <a href=""> texte <em> texte </em> texte </a> texte </p>
```

- `em` est enfant de `a` et descendant de `p`
- `a` est parent de `em` et enfant de `p`

Représentation en arbre

- **Sommets :** balises HTML du document
- **Arcs:** arc `b1` à `b2` si `b1` est parent de `b2`

```
<DOCTYPE html>
<html lang="fr">
<head>
    <title>Presentation</title>
    <meta  charset="utf-8"/>
</head>
<body>
<p> Texte <em> avec emphase <strong> forte
</strong> </em> et hyperlien <a href="">
<img src="" alt=""/> </a>
</body>
</html>
```



Balises block et inline

Balises de type inline

- Balises `br`, `em`, `strong`, `span`, `img` et `a`
- Pas de retour à la ligne avant et après
- Prennent uniquement la place dont elles ont besoin

Balises de type block

- Toutes les autres balises
- Retour à la ligne avant et après
- Prennent toute la largeur du conteneur parent

```
<p> du texte <em> sur la même </em> </p>
<p> Nouveau bloc ! </p>
```

Du texte *sur la même* ligne.

Nouveau bloc !

Règles à respecter

Pas d'imbrication de balises

```
<balise1> ... <balise2> ... </balise1> ... </balise2>
```

Enfants et parents possibles

Certaines balises sont interdites comme enfants ou parents d'une autre balise.

Exemples :

- un titre ne peut pas être fils de paragraphe et inversement
- un paragraphe ne peut pas être fils d'un autre paragraphe
- les seuls fils possibles pour `ul` et `ol` sont `li`
- les balises `inline`, excepté `a`, ne peuvent être parents d'une balise `block`

Langage CSS

Règles CSS

Langage CSS : ensemble de règles

```
selecteur {
  propriete1: valeur;
  propriete2: valeur;

  ...
}
```

Exemple :

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

Lier le fichier CSS au HTML (dans la partie head):

```
<link href="fichier.css" rel="stylesheet" />
```

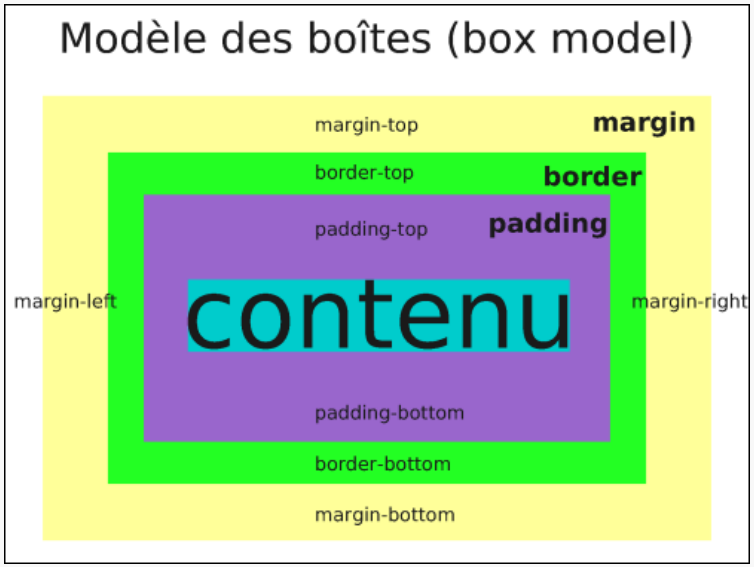
Propriétés CSS

Police et couleurs

- `font-family` : police utilisée (importer des polices avec `@font-face`) `font-size` : taille de la police en pixels `px`, taille relative `em`
- `font-weight` : `bold` pour mettre en gras (`normal` sinon)
- `font-style` : `italic` ou `normal`
- `text-align` : `left`, `right`, `center` ou `justify`
- `text-decoration` : `underline` ou `none`
- `color` : couleur de police
  - Nom de couleur : `red`, `black`, etc
  - Code hexadécimal : #123526
  - `rgba(rouge, vert, bleu, transparence)` avec :
  - `rouge`, `vert` et `bleu` : nombres entre 0 et 255
  - transparence : nombre entre 0 et 1

Marges et bordures

- largeur : `width`
- hauteur : `height`
- marge intérieure : `padding`
- marge extérieure : `margin`
- bordure : `border: type size color;`
  - `type` : `solid`, `double`, `inset`, `outset`
  - `size` : épaisseur de la bordure
  - `color` : n'importe quelle couleur
- arrondis : `border-radius : taille;`



`padding`, `border` et `margin` peuvent être différents pour chaque côté (`-left`, `-right`, `-top` et `-bottom`)

`width` et `height` donnent uniquement la taille du contenu. Pour prendre en compte les marges intérieures et les bordures => `box-sizing:border-box`.

Arrière-plan

- `background-color` : couleur d'arrière-plan (n'importe quelle couleur)
- `background-image:url("fichier_image")` : image d'arrière-plan
- `background-repeat` : si l'image de fond se répète (`repeat`) ou pas (`no-repreat`)
- `background-position` : position de l'image par rapport à l'élément.
  - Positionnement vertical : `top`, `bottom`, `center`
  - Positionnement horizontal : `left`, `right`, `center`

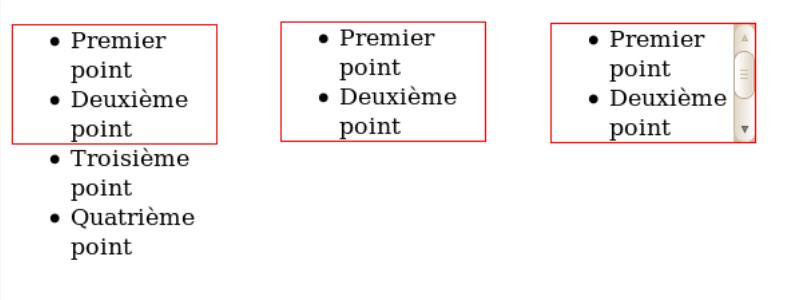
```
background-position : bottom right;
```

Listes et tableaux

- `border-collapse:collapse` pour coller les cellules d'un tableau
- `list-style` : style des puces et numéros
  liste non-ordonnée : `square`, `disc`, `circle`, `none`
  - liste ordonnée : `decimal`, `lower-roman`, `upper-roman`, `none`

Dépassement et type de balises

- `overflow` : comportement en cas de dépassement
  - `visible` : ce qui dépasse est affiché
  - `hidden` : caché
  - `auto` : barre de défilement (scroll)



- `display` : modification du comportement d'affichage
  - `block` : s'affiche comme un élément de type `block`
  - `inline` :s'affiche comme un élément de type `inline`
  - `inline-block` : taille, marge mais pas de retour à la ligne
  - `none` : supprime l'élément

Classes et identifiants

Utilisation

Classes et identifiants permettent de ne modifier que certains éléments.

```
<balise class="nom_classe" id="nom_identifiant"> ... </balise>
<balise class="nom_classe" id="nom_identifiant" />
```

```
.nom_classe{
    ...
}
#nom_identifiant{
  ...
}
```

Les règles ne s'appliquent qu'aux éléments ayant la classe (et/ou l'identifiant).

L'identifiant est unique.

Pseudo-classes

Les pseudo-classes précisent un état de la cible.

- `:hover` : quand la cible est survolée par la souris
- `:nth-child(n)` : quand la cible est le $n^{ème}$ enfant
- `:active`: quand la cible est cliquée
- `:visited` : quand la cible (hyperlien) a été visité

Exemple :

```
p:hover {
  color: red;
}
```

Quand on survole un paragraphe, il s'affiche en rouge.

Balises universelles

- Permettre de modifier l'apparence de code non délimité par des balises
- Deux balises universelles :
  - `div` : balise de type `block`
  - `span` : balise de type `inline`

Exemple

```
<p><span class="bleu">Paris</span>, <span class="bleu">Lyon</span> et
<span class="bleu">Marseille</span> sont les trois plus grandes villes de
France.</p>
```

```
.bleu { color: blue;}
```

Paris, Lyon et Marseille sont les trois plus grandes villes de France.

Combinaisons et modifications en cascade

Combiner les sélecteurs

Combinaison	Explication
E	un élément E
*	n'importe quel élément
E F	un élément F descendant de E
E > F	un élément F enfant de E
E.ma_classe	un élément E ayant la classe ma_classe
E#id	un élément E ayant l'identifiant id
E, F	un élément E ou un élément F

Exemples

```
p.rouge { /* les éléments de classe rouge contenus dans un paragraphe */}

p.rouge { /* les paragraphes de classe rouge */}

p > em.rouge { /* les em de classe rouge et enfants de paragraphe */ }
```

Styles en cascade

Plusieurs règles peuvent s'appliquer à un même élément.

```
<div id="monId"> <p class="rouge"> ... </p> </div>
```

```
p {
  font-size: small;
}
.rouge {
  color: red;
}
#monId p {
  text-align: right;
}
```

=> Le paragraphe est écrit en petit, rouge et à droite.

Priorité des règles

Lorsque plusieurs règles s'appliquent, si des valeurs sont différentes pour une propriété, on choisit la règle prioritaire.

- On associe un nombre à trois chiffres pour chaque sélecteur :
  - centaines : nombre d'identifiants dans le sélecteur,
  - dizaines : nombre de classes et pseudo-classes,
  - unités : nombre de balises.
- La règle prioritaire est celle dont le nombre associé au sélecteur est le plus grand.
- En cas d'égalité, on choisit celle écrite en dernier dans le fichier.

```
p{ color: yellow; } /* Priorité : 001 */
.rouge { color: red; } /* Priorité 010 */
#monId p { color: green; } /* Priorité 101 */
```

=> Le paragraphe est écrit en vert.

Héritage

Propriété supportant l'héritage : si la valeur de cette propriété n'est pas spécifiée pour un élément, celle-ci prend comme valeur celle de l'élément parent.

```
body{ color: red; }
```

=> Tout le document (sauf hyperliens) est en rouge.

```
<p class="rouge"> Du texte <em> mis en valeur </em> puis <strong> fortement
mis en valeur. </strong> </p>
```

```
.rouge { color: red; }
em{ color: blue; }
```

- `Du texte` et `puis` => rouge d'après la règle 1
- `mis en valeur` => bleu d'après la règle 2
- `fortement mis en valeur` => rouge par héritage.

Positionnement en CSS

Propriété position

Types de positionnement

Positionnement statique (`static`)

- Les éléments sont affichés dans l'ordre où ils apparaissent dans le HTML (flux).
- Positionnement par défaut.

Positionnement relatif (`relative`)

- déplace l'élément par rapport à la position qu'il avait dans le flux.
- `top`, `bottom`, `left`,`right` indiquent le **décalage** (`left:50px;` ajoute 50 pixels à gauche de l'élément)

Types de positionnement

Positionnement absolu (absolute)

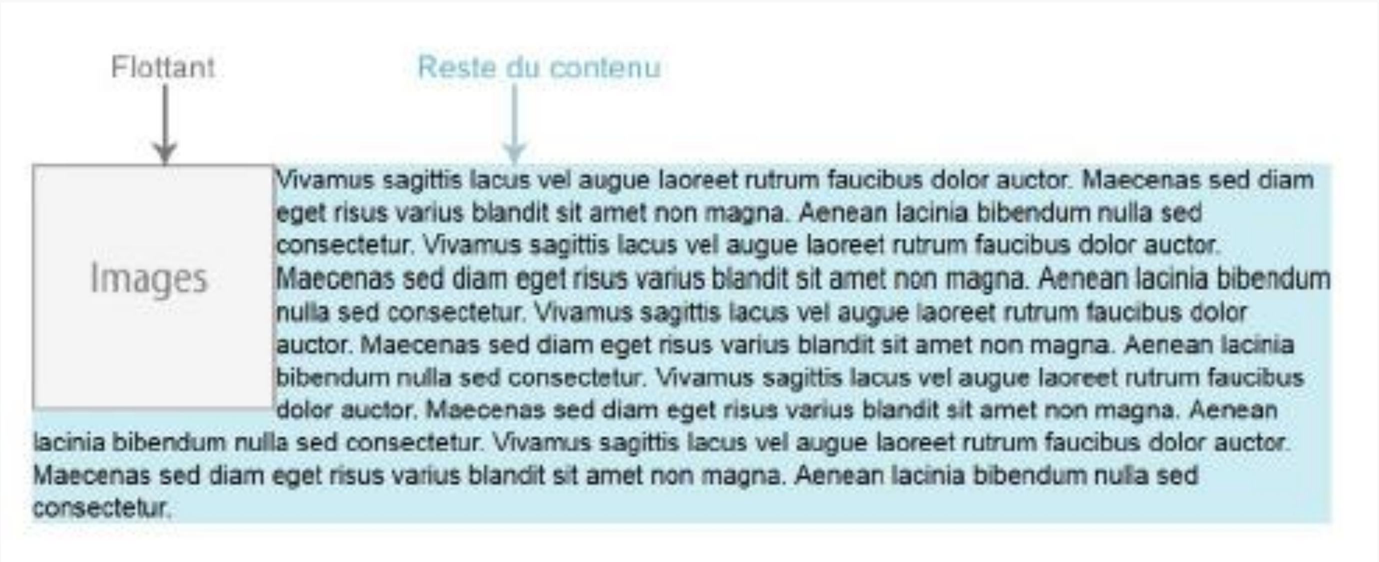
- L'objet est retiré du flux.
- Il est ajouté par dessus les autres éléments.
- `top`, `bottom`, `left`,`right` indiquent la **distance** par rapport au premier ascendant qui n'a pas de positionnement statique (page Web si aucun).

Positionnement fixe (fixed)

- Similaire au positionnement absolu.
- La distance est toujours par rapport au navigateur.
- Insensible au scrolling.

Les « flottants » et le flux

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)

Float:left;

Un exemple de **float : left** sur une image :

img.floatleft { **float: left;** }

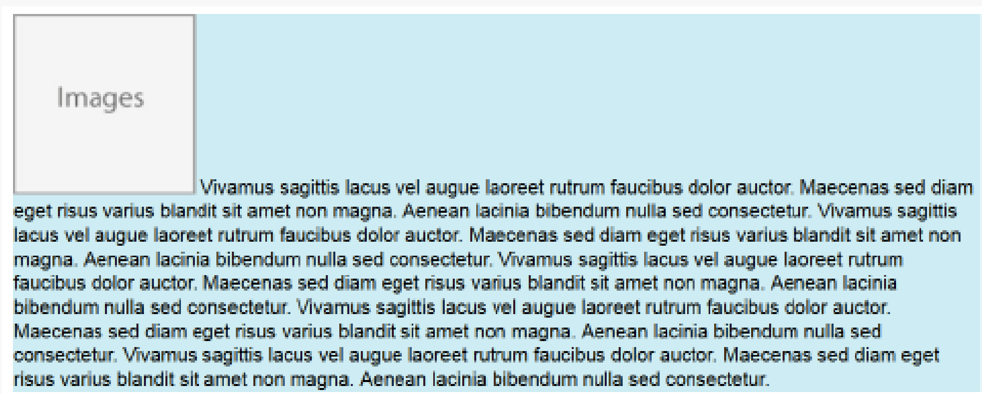
Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Maecenas sed diam eget risus varius blandit sit amet non magna. Aenean lacinia bibendum nulla sed consectetur. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Maecenas sed diam eget risus varius blandit sit amet non magna. Aenean lacinia bibendum nulla sed consectetur. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Maecenas sed diam eget risus varius blandit sit amet non magna. Aenean lacinia bibendum nulla sed consectetur. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Maecenas sed diam eget risus varius blandit sit amet non magna. Aenean lacinia bibendum nulla sed consectetur.

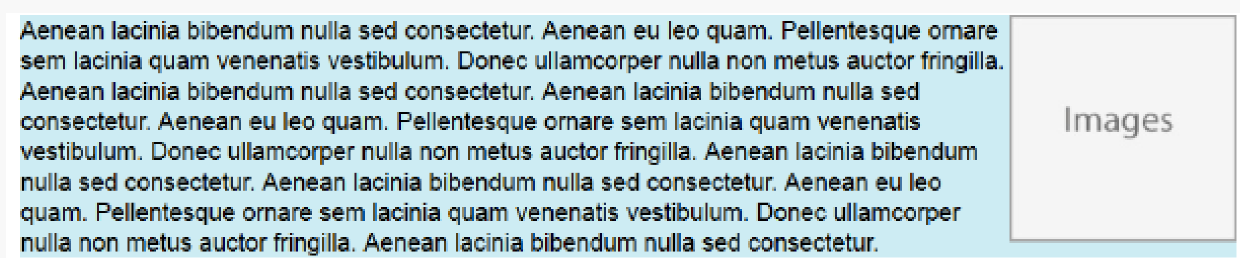
Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Maecenas sed diam eget risus varius blandit sit amet non magna. Aenean lacinia bibendum nulla sed consectetur. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Maecenas sed diam eget risus varius blandit sit amet non magna. Aenean lacinia bibendum nulla sed consectetur. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Maecenas sed diam eget risus varius blandit sit amet non magna. Aenean lacinia bibendum nulla sed consectetur. Vivamus sagittis lacus vel augue laoreet rutrum faucibus dolor auctor. Maecenas sed diam eget risus varius blandit sit amet non magna. Aenean lacinia bibendum nulla sed consectetur.

Float right;

Un exemple de **float: right** sur une image :

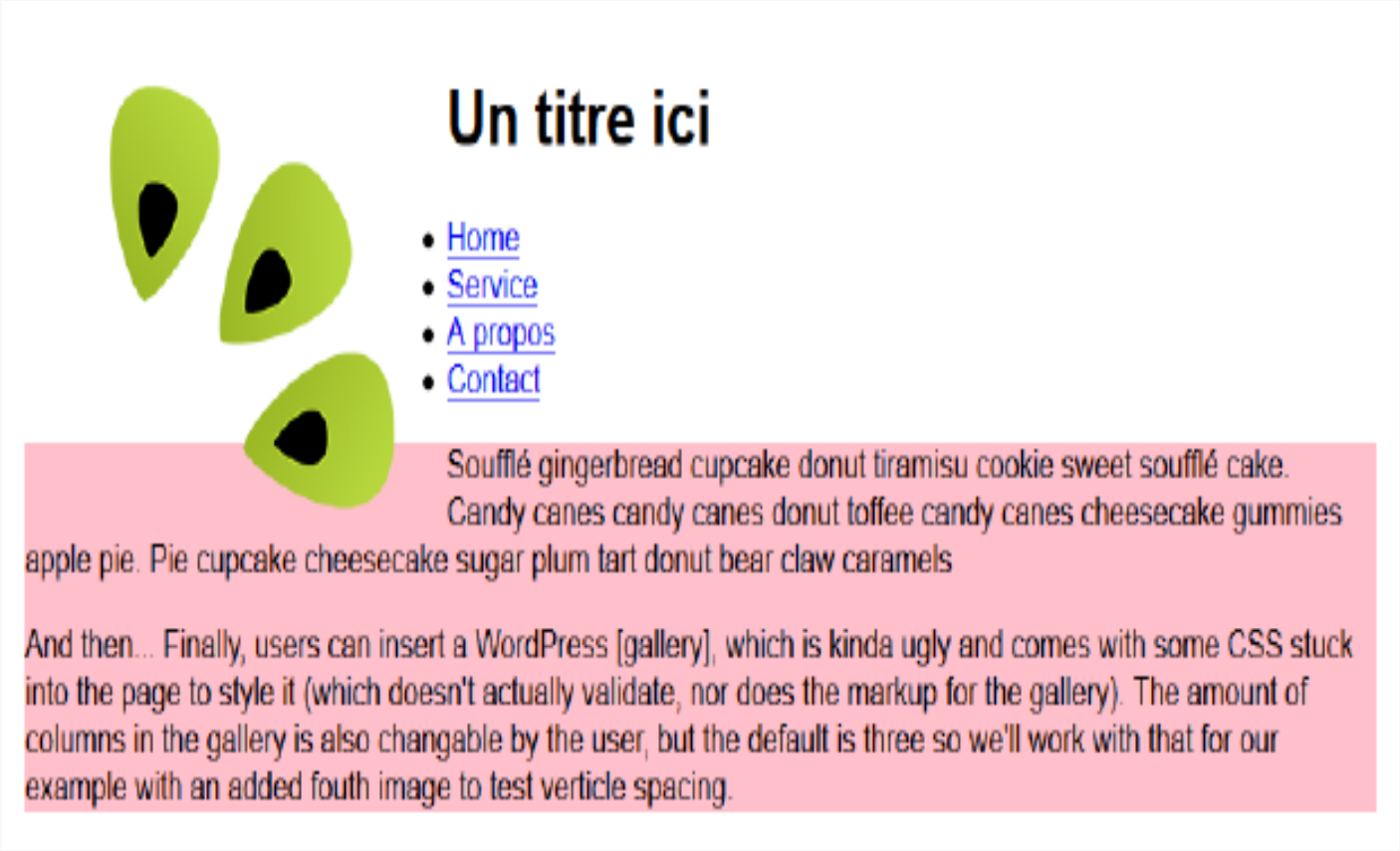
img.floatright { **float: right;** }





Problème des flottants

Comme le contenu « **coule** » autour, on se retrouve avec ce genre de problèmes



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**

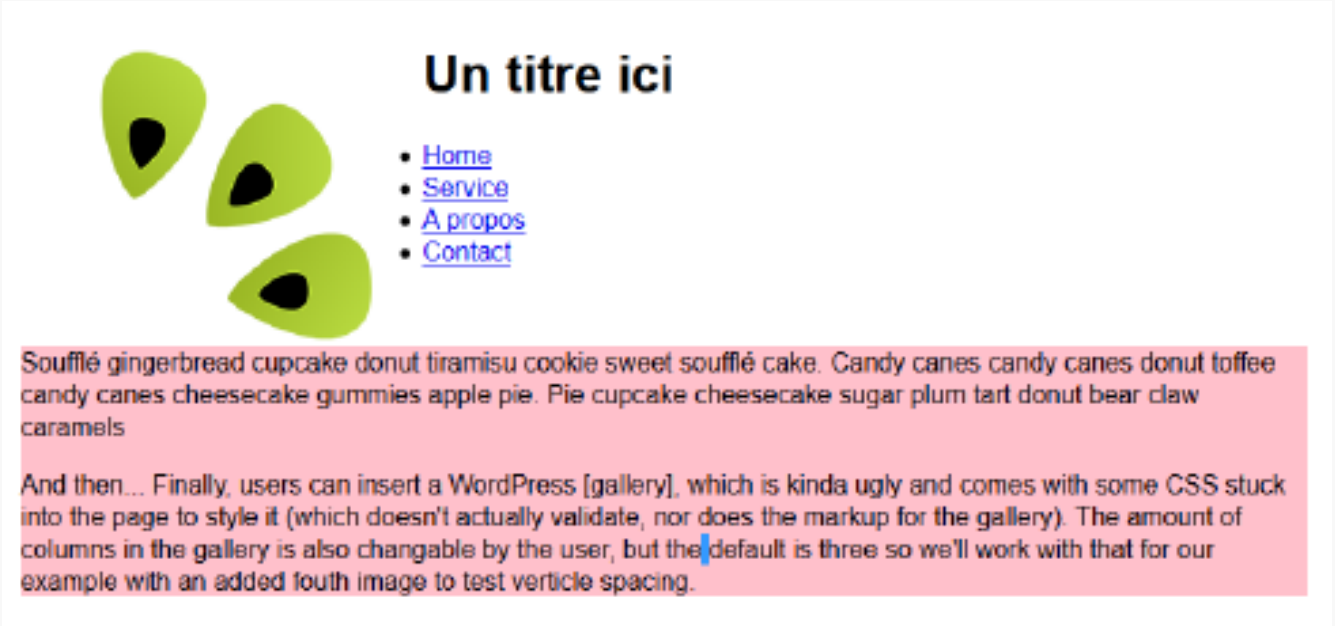
– **clear: right** permet d'empêcher le contournement des blocs flottants à **droite**

– **clear : both** permet d'empêcher le contournement des blocs flottants à **gauche et à droite**

Clear : bloquer le dépassement des flottants

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

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

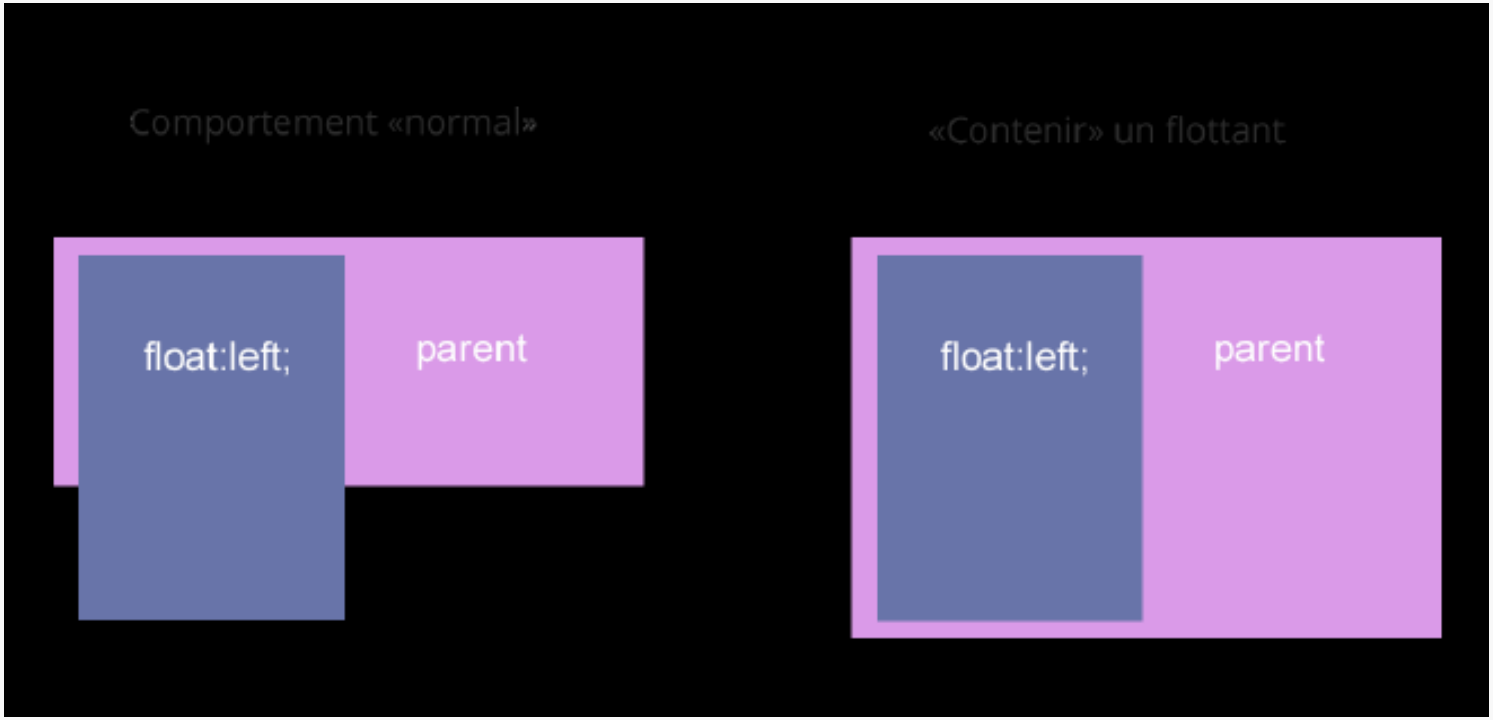


BFC et contenir les flottants dans un bloc

Les éléments flottants peuvent « **dépasser** » de leur parent si le contenu de celui-ci n'est pas suffisant

BFC et contenir les flottants dans un bloc

- Un « **contexte de formatage de bloc** » (ou **BFC**) est un élément avec des « **super pouvoirs** » :
  - Il peut « contenir » les flottants (qui ne peuvent plus en dépasser)
  - Il ne s'écoule pas autour des flottants



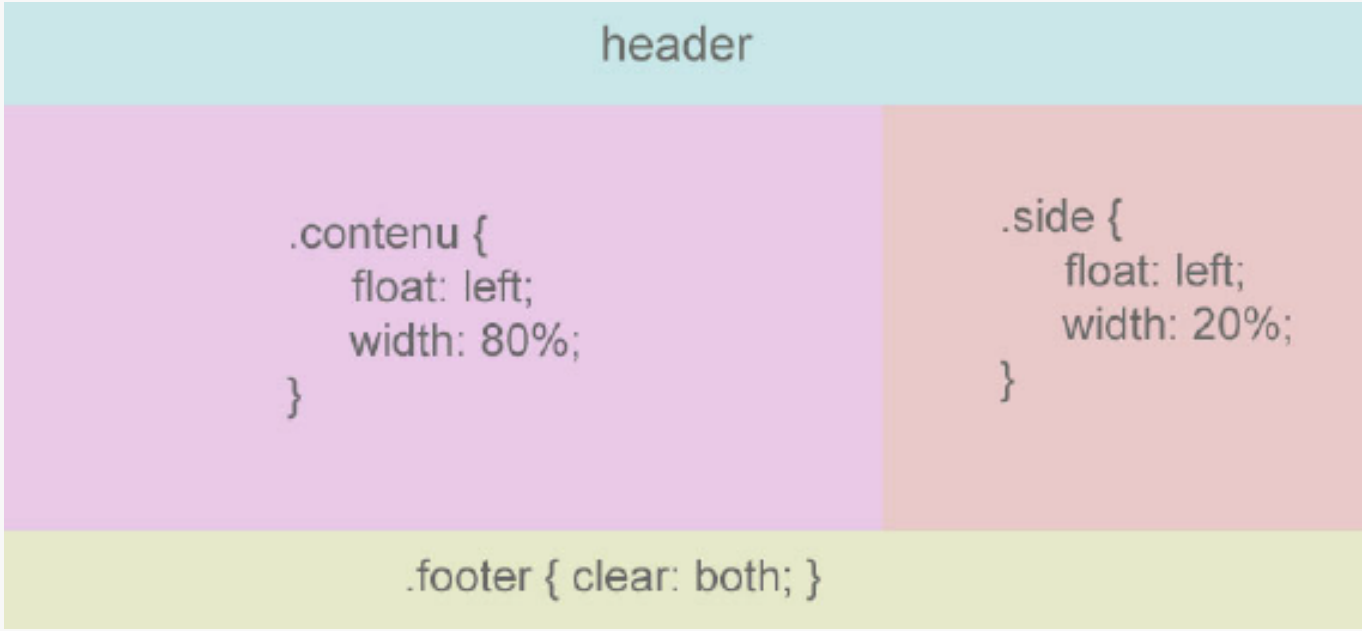
Overflow : hidden

Ajouter **overflow:hidden** au **parent** des flottants permet de créer un « **contexte de formatage de bloc** » et résoudre notre problème.

```
.parent { overflow:hidden; }
```

Les flottants appliqués à la mise en page

Il est possible d'utiliser des **flottants** pour créer une mise en page en colonnes.
C’est notamment utilisé dans d’anciennes versions de **Bootstrap** et ressemble à ça.
On privilégie cependant **Flexbox** aujourd’hui pour une mise en page de site web.



Responsive design

Principe

Affichage "confortable" du site quel que soit le support (taille) Pas de modification du fichier HTML sauf ajout dans le head:

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

Du code CSS s'applique pour certaines tailles d'écran seulement

Utilisation des *media queries*

```
@media (condition){
/* Règles CSS qui sont prises en compte si la condition est vérifiée */
}
```

conditionest définie par min-width:tailleet/ou max-width:taille

Pour une expérience utilisateur optimale, il faut un CSS compatible pour (presque) tous les navigateurs, quel que soit leur version.