



Présentation du langage CSS

- Appartient aux langages de *description* (définissent un contenu statique) != langages de programmation (génèrent un contenu variable, dynamique). Même si on peut faire quelques animations selon la position de la souris en CSS...
- Contrairement au HTML, qui structure l'information, le CSS permet de *mettre en forme* l'information (couleurs, bordures, positionnement des éléments...).
- Version actuelle : CSS 3



Présentation du langage CSS

Ce que permet le CSS

Le HTML gère uniquement la structure et la sémantique d'un site web ; le CSS gère intégralement son design. Exemples :

- Taille du texte
- Couleur/image du fond de la page (ou d'autres éléments)
- Bordures
- Mise en page (positionnement du menu, de l'header, largeur des éléments, display mobile => *responsive...*)



Présentation du langage CSS

Avant

- [Services](#)
- [Portfolio](#)
- [Our Team](#)
- [Blog](#)
- [Contact](#)
- [We're Hiring](#)

Home ▼

We create digital chemistry

We're proud of what we do. All our projects are carefully handcrafted, tailored and refined by us from our offices in Southsea. Our clients choo



Robust web applications

Whatever your requirements are, we're here to infuse our technical and digital expertise into everything we do. We develop with great care and thrive and function better.



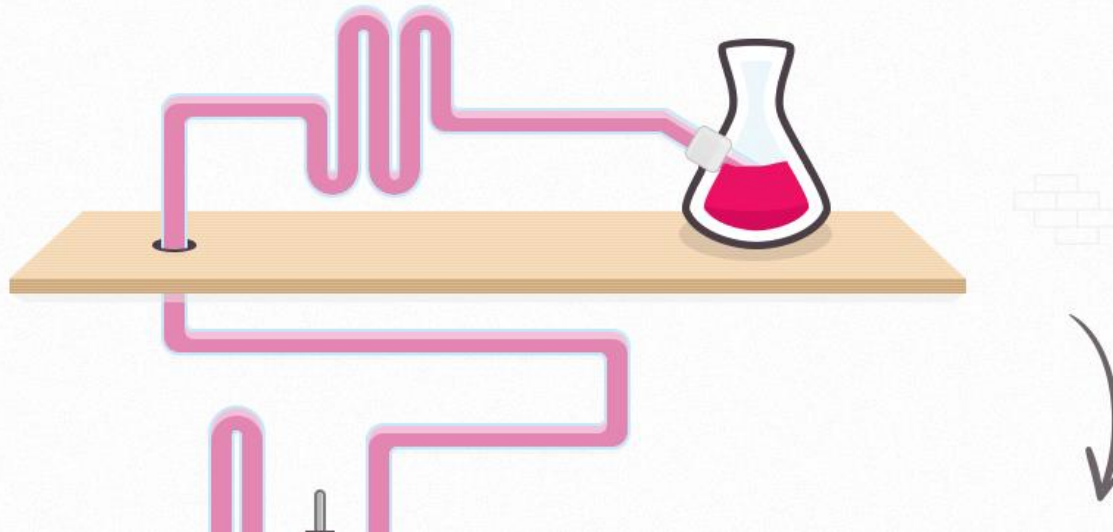
Présentation du langage CSS

Après

[Services](#)[Portfolio](#)[Our Team](#)[Blog](#)[Contact](#)[We're Hiring](#)

We create **digital chemistry**

We're proud of what we do. All our projects are carefully handcrafted, tailored and refined by us from our offices in Southsea. Our clients choose to work with us because this is more than just a job - we live to design and code.





Présentation du langage CSS

Toutes les propriétés du CSS3 ou du CSS2 ne sont pas comprises par les (très) anciens navigateurs (important pour certaines grandes entreprises ou l'Etat, ou pour les pays émergents).

Il en est de même pour les balises HTML5. Exemple de la balise audio :

<http://caniuse.com/#feat=audio>



Mise en place du CSS

Placement du CSS

3 options :

- Dans un fichier .css (il servira au design de tout le site web – il faut éviter les duplicata de code !- : on aura plusieurs fichiers .html mais un seul .css dans ce cours, ou un seul .css compressé pour des sites en ligne afin de gagner de la performance)
- Dans l'en-tête <head> du fichier HTML visé (à éviter)
- Directement dans les balises HTML de la page visée, via l'attribut style (cela arrive en javascript).



Mise en place du CSS

Placement du CSS

Les deux dernières méthodes sont à éviter pour la maintenabilité du site : l'avantage du fichier CSS est qu'il *centralise* l'information relative au design : en une ligne, on peut changer des milliers d'éléments de même type.

Remarque : la méthode avec l'attribut style peut être utile dans certains cas, lorsqu'on veut effectuer une modification locale du design avec JS ou PHP.

Exemple: surligner dans la page un mot recherché par l'auteur: on rajoutera un fond (jaune par exemple) à tous les mots correspondants.



Mise en place du CSS

Le fichier CSS

Il faut qu'il ait l'extension .css. On l'appellera par exemple design.css.

Il faut rajouter dans chaque fichier HTML une balise link dans l'en-tête <head> (ici design.css est dans le même dossier que la page html) :

```
<head>
  <meta charset="utf-8" />
  <link rel="stylesheet" href="design.css" />
  <title>Formation web</title>
</head>
```



Mise en place du CSS

Le fichier CSS

Exercice : créer un fichier design.css, le lier à une page HTML contenant au moins un paragraphe.

Dans le fichier CSS, écrire :

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

Puis ouvrir la page HTML. Contempler le résultat.



Mise en place du CSS

Court exemple sur l'en-tête head (<style> fonctionne aussi ailleurs dans la page)

```
<title>Formation web EPSI</title>
<style>
p
{
    color: red;
}
</style>
```

Le style ne s'appliquera qu'à cette page !



Mise en place du CSS

Directement dans la balise

```
<p style="color: red;"> test </p>
```

Le style ne s'appliquera qu'à cette balise dans cette unique page !

=> Répétition de code, changement difficiles...



Mise en place du CSS

Syntaxe du langage CSS : on simple une ou des balises puis accolades et liste de propriétés-valeurs :

```
balise1
{
    propriete1 : valeur1;
    propriete2 : valeur2;
}
balise2
{
    propriete1: valeur3;
}
```

Exemple avec pour balise <p>, la propriété color et la valeur red:

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



Préparer le HTML pour CSS

Les balises universelles (ou génériques)

Elles vont nous permettre d'appliquer des styles à des éléments n'ayant pas de valeur sémantique particulière (cad n'étant pas entourés de balises classiques comme `<p>`, `<nav>`, `<header>`...)

L'une est de type inline (comme ``, ``... pas de retour ligne généré, pas de largeur réglable en CSS) et s'applique à des petites portions de texte :

`` du texte **``**



Préparer le HTML pour CSS

Les balises universelles (ou génériques)

L'autre est de type block : elle peut englober d'autres balises de type block comme `<p>`, `<nav>`, ``... pratique pour regrouper.

`<div>` `<!-- du code HTML -->` **`</div>`**

Ces balises sont invisibles tant qu'on ne leur a pas appliqué de style CSS !



Préparer le HTML pour CSS

Retour sur la syntaxe CSS

Pour appliquer un même style à plusieurs types de balises, on sépare ces types par des virgules :

```
h1, h2, h3  
{  
    color: blue;  
}
```




Préparer le HTML pour CSS

Retour sur la syntaxe CSS

Les commentaires se font avec la syntaxe suivante:

```
/* un commentaire */
```

Exemple:

Utile pour suspendre
temporairement
certains styles

```
/*  
fichier design.css  
Feuille du site "trello.com"  
Par Isaac Asimov  
*/  
  
p /* apparence des paragraphes */  
{  
    color: #d4deff;  
    font-size: 150%;  
}
```



Préparer le HTML pour CSS

Les class et les id : appliquer un style d'une manière sélective

Ce sont des attributs absolument nécessaire pour les `` et les `<div>`, qui servent à cibler des éléments particuliers, mais parfois aussi pour des balises classiques qui ne doivent pas être toutes concernées par le style en question.

L'attribut **id** permet de cibler une UNIQUE balise (chaque id doit avoir une valeur différente).

```
<p id="paragraphe1"> test1 </p>
```

L'attribut **class** permet de cibler un ensemble de balises.

```
<p class="paragraphebleu"> test2 </p>  
<p class="paragraphebleu"> test3 </p>
```



Préparer le HTML pour CSS

Les class et les id

Dans le CSS, les noms des ID sont précédés d'un dièse et les noms de classe d'un point :

HTML

```
<p id="paragraphe1"> test1 </p>
```

```
<p class="paragraphebleu"> test2 </p>  
<p class="paragraphebleu"> test3 </p>
```

CSS

```
#paragraphe1  
{  
    color: red;  
}  
.paragraphebleu  
{  
    color: blue;  
}
```



Préparer le HTML pour CSS

Exemple d'utilisation des balises universelles et des attributs class et id :

- ✓ Dans une page html avec une image de fond, créer trois paragraphes contenant chacun la phrase « La pratique, c'est quand ça marche mais que l'on ne comprend pas. La théorie, c'est quand on comprend mais que ça ne marche pas. ».
- ✓ Mettre le premier paragraphe en rouge, mettre les deux autres paragraphes en bleu.
- ✓ Dans le premier paragraphe, mettre le mot « pratique » en noir (indice : utiliser la balise `` autour du mot et lui associer une classe ou un id).



Sélectionner les éléments en CSS

Les sélecteurs CSS

Pour appliquer un style à un élément, il faut pouvoir le désigner dans le CSS. On a vu qu'on pouvait écrire le nom de la balise, mais alors toutes les balises seront concernées par le style.

On a vu que les attributs class et id sont plus précis.

Voici quelques autres sélecteurs utiles :



Sélectionner les éléments en CSS

Les sélecteurs CSS

Balise1 balise2 : on sélectionne toutes les balises2 contenues dans une balise1. Exemple:

```
p em /* on sélectionne tous les em contenu dans un paragraphe */  
{  
    font-style: italic;  
}
```

Balise1 + balise2 : on sélectionne toutes les balises2 précédées directement d'une balise1.



Sélectionner les éléments en CSS

Les sélecteurs CSS

Balise1[attribut]: on sélectionne toutes les balises1 qui possèdent l'attribut entre crochets.

Exemple :
colorier en noir tous les liens
ayant un attribut title :

```
a[title]
{
    color: black
}
```

```
<a href="mapage.html" title="ma page">lien vers ma page</a>
```



Sélectionner les éléments en CSS

Les sélecteurs CSS

On peut combiner les sélecteurs entre eux :

```
p a[title], nav #lien1
{
    color: black
}
```

On sélectionne tous les liens contenus dans un paragraphe et possédant un attribut title, et le lien d'id « lien1 » contenu dans la balise <nav>.



Les propriétés CSS de base

Apparence du texte – taille du texte

On utilise la propriété **font-size**.

On peut lui indiquer une taille absolue en pixels (px), centimètres (cm) ou millimètres (mm).

Mais il est préférable de lui indiquer une taille relative en « *rem* » (« *em* » est une propriété semblable mais qui hérite de la taille de police du parent).

Exemples :

```
p
{
    font-size: 150%;
}
p
{
    font-size: 1.5em;
}
```



Les propriétés CSS de base

Apparence du texte – la police

On utilise la propriété **font-family**.

Au cas où l'internaute ne possède pas la police spécifiée, on propose plusieurs polices (séparées par des virgules) : le navigateur prendra la première police qu'il connaît.

```
p
{
    font-family: Impact, arial, "Courier New", Georgia;
}
```



Les propriétés CSS de base

Apparence du texte – la police

```
p  
{  
    font-family: Impact, arial, "Courier New", Georgia;  
}
```

Si le navigateur ne connaît pas impact, il prendra Arial, et sinon l'une des suivantes, et au pire sa police par défaut.

Remarque : si une police a un nom composé, l'entourer de guillemets.

Quelques noms de polices courantes : Arial, Georgia, Impact, Time New Roman, Trebuchet MS, Verdana...



Les propriétés CSS de base

Apparence du texte – la police

Avec CSS3, on peut définir une police personnalisée. Par contre, le visiteur devra télécharger le fichier la police (parfois 1MO).

Avant tout, il faut avoir la police sur le serveur. Deux sites proposent de télécharger des polices libres de droit :

<http://www.dafont.com/theme.php?cat=401>

<http://www.fontsquirrel.com/fonts/list/style/Calligraphic>



Les propriétés CSS de base

Apparence du texte – la police

Vous y trouverez plusieurs formats. Il vous faudra au moins le format .ttf, qui fonctionne sur IE9 et tous les autres navigateurs, et le format .eot, qui fonctionne sur toutes les versions d'IE. Enfin, le format .svg est utile pour les tablettes d'Apple.



Attention ! Certaines polices anglaises ne gèrent pas les accents !



Les propriétés CSS de base

Apparence du texte – la police

Dans le CSS, il faut ensuite définir votre nouvelle police en lui donnant un nom et en entrant ses fichiers sources :

```
@font-face
{
    font-family: 'Gothic';
    src: url('AGothiqueTime.ttf');
    /*si plusieurs formats de police sont
    disponible (.eot, .svg...), rajouter
    des url('cheminPolice') séparées par
    des virgules*/
}
```

Ensuite il suffit d'appeler la police par son nom :

```
p
{
    font-family: 'Gothic', Impact;
}
```

Résultat :

Gothic fiction, sometimes referred to as Gotchi horror, is a genre or mode of literature that



Les propriétés CSS de base

Exercice

Importez une police personnalisée et utilisez-la sur un paragraphe !



Les propriétés CSS de base

Apparence du texte – italique, gras, souligné...

La plupart des navigateurs mettent le texte encadré par `` en italique par défaut, et le texte entouré par `` en gras. Mais ces balises servent à dire que certains mots sont importants et ne doivent pas être utilisées massivement !



Les propriétés CSS de base

Apparence du texte – italique, gras, souligné...

Mettre en italique quelques mots :

```
<p> Gothic fiction, sometimes referred to as <span class="italique">Gotchi horror</span>,</p>
```

```
.italique
{
    font-style: italic;
}
```

Enlever l'effet italique des balises :

```
em
{
    font-style: normal;
}
```



Les propriétés CSS de base

Apparence du texte – italique, gras, souligné...

Mettre en gras :

```
.gras  
{  
    font-weight: bold;  
}
```

Remarque : on peut associer plusieurs classes à une même balise :

```
<span class="italique gras">Gothic horror</span>,
```

Le fragment de texte seront donc en italique ET en gras.



Les propriétés CSS de base

Apparence du texte – italique, gras, souligné...

Soulignement et autres décorations :

propriété **text-decoration**

Valeurs possibles :

- Underline: souligné
- Line-through: barré
- Overline: ligne au-dessus
- None : aucun (par défaut)

Exemple :

```
.souligne  
{  
    text-decoration: underline;  
}
```



Les propriétés CSS de base

Apparence du texte – l'alignement

Définir l'alignement est utile pour un bloc de texte (souvent plusieurs lignes) contenu dans une balise de type block comme `<p>`, `<div>`, `<footer>`, `<h1>`...

Ce n'est pas possible pour les balises de type inline comme ``, ``...

Propriété : **text-align**

Valeurs :

- left
- center
- right
- justify : le texte est justifié, cad qu'il ne laisse pas d'espace blanc en fin de ligne



Les propriétés CSS de base

Apparence du texte – l'alignement

Exercice : créer un footer, y placer du texte, et centrer son texte.



Les propriétés CSS de base

La couleur

On peut appliquer une couleur sur le texte, sur un fond, sur une bordure...

Il existe différentes manières d'indiquer une couleur :

- Par son nom (grey, red, maroon, blue, navy...)
- Par sa notation hexadécimale, par exemple : #a8bade
- Par sa notation RGB (Red Green Blue), qui se base sur les 3 composantes d'une couleur (valeurs de 0 à 255). Exemple : rgb(122,100,255)

Quelques logiciels utiles : Paint, GIMP (sorte de photoshop gratuit) pour trouver une couleur d'une image



Les propriétés CSS de base

La couleur

Propriété pour modifier la couleur d'un texte :
color

Propriété pour modifier la couleur de fond :
background-color

Exemple :

```
body
{
    background-color: rgb(100,200,100);
}
```

Toutes les balises contenues dans body auront le même fond, sauf si on leur définit un autre fond.



Les propriétés CSS de base

Image de fond

Appliquer une image de fond à la page :

```
body
{
    background-image: url("monimage.png");
}
```

Le chemin se donne en relatif par rapport au fichier CSS. Attention aux chemins en absolus : si le site qui héberge votre image ferme, celle-ci ne sera plus affichée !

Vous pouvez appliquer des fonds à d'autres éléments (<header> pour la bannière, éventuellement pour des onglets...) !



Les propriétés CSS de base

Image de fond

Options disponibles sous forme de propriétés supplémentaires:

- **Background-attachment**

- Fixed (l'image ne défile pas quand on descend dans la page)
- Scroll (défaut)

- **Background-repeat**

- *No-repeat* (l'image de fond n'apparaît qu'une seule fois : utile si l'image est grande)
- Repeat-x (fond répété horizontalement)
- Repeat-y (fond répété verticalement)
- Repeat (par défaut, le fond est répété comme une mosaïque)



Les propriétés CSS de base

Image de fond

Options disponibles sous forme de propriétés supplémentaires:

Background-position permet de définir la position du fond (utile si on utilise un no-repeat).

Elle prend comme valeurs la distance en pixels que doit avoir l'image de fond par rapport au coin supérieur gauche de la page:

```
body
{
    background-image: url("monimage.png");
    background-position: 100px 0px; /* fond décalé 100 px à droite */
}
```

On peut aussi lui donner les valeurs *top*, *bottom*, *left*, *center*, *right* à la place des valeurs en pixels !



Les propriétés CSS de base

Image de fond

La super-propriété **background** :

Elle permet de combiner en une ligne les valeurs des propriétés vues précédemment (background-image, background-repeat...).

L'ordre des valeurs n'a pas d'importance, et chaque valeur est facultative.

Exemple :

```
body
{
    background: url("monimage.jpg") no-repeat 100px 0px;
}
```



Les propriétés CSS de base

Image de fond

La propriété CSS **background-size** spécifie la dimension des images d'arrière-plan afin d'adapter l'image à la taille réelle de l'élément :

- cover (agrandit l'image à la taille de son conteneur)
 - contain (permet de faire rentrer l'image dans son conteneur)
 - Des pourcentages ou des px (ex: 50% 60% ou 50px)
-
- Elle ne rentre pas dans la super-propriété background !



Les propriétés CSS de base

Image de fond

Depuis CSS3, on peut utiliser plusieurs images de fond, en séparant les déclarations par des virgules :

```
body
{
    background: url("monimage.png") no-repeat 100px 0px,
               url("uneautreimage.png") fixed top right;
}
```



Les nouvelles propriétés CSS 3

La transparence

Depuis le CSS3, on peut utiliser la transparence sur des éléments.

Propriété **opacity** : elle permet d'indiquer le niveau d'opacité d'un élément (et tout ce qu'il peut contenir : texte, images...). Ce niveau va sur une échelle de 0 (transparence totale) à 1 (opacité totale : c'est le comportement par défaut).

Exemple : rendre les images semi-transparentes.

```
img
{
    opacity : 0.5;
}
```



Les nouvelles propriétés CSS 3

La transparence

On peut également cibler une couleur pour la rendre transparente.

Par exemple, rendre le fond rouge d'un élément semi-transparent, ou le texte d'un élément semi-transparent.

On utilise la propriété `rgba`, qui fonctionne comme `rgb` mais à laquelle on rajoute un dernier paramètre correspondant au niveau de l'opacité.

```
p
{
    color: rgba(255,0,0,1); /* équivaut à un rgb(255,0,0) */
    background: rgba(0,200,0,0.5) /* fond vert semi-transparent */
}
```



Les nouvelles propriétés CSS 3

Les bordures

On peut rajouter des bordures à n'importe quel élément !

On utilise la super-propriété **border**.

On lui spécifie la largeur de la bordure (en pixels), une couleur et un type de bordure :

none : pas de bordure (par défaut)

solid : un trait simple

dotted : pointillés

dashed : tirets

double : bordure double

groove : en relief

ridge : autre effet relief

inset : effet 3D global enfoncé

outset : effet 3D global surélevé.



Les nouvelles propriétés CSS 3

Les bordures

Exemple :

```
footer
{
    border: 2px red dotted;
}
```

On peut également affecter des bordures différentes à chaque côté avec les propriétés :

- Border-top
- Border-bottom
- Border-left
- Border-right

Exercice: ajouter une bordure noire en pointillé de 3px de large en haut des titres <h1>, et une bordure bleue en tirets en bas des titres <h1>.



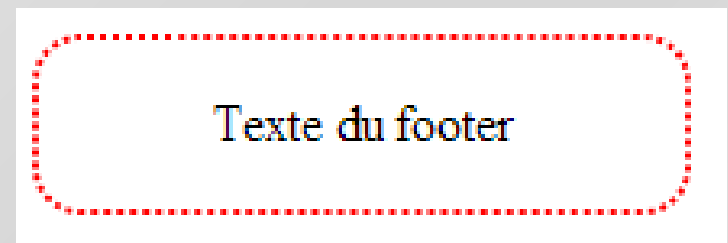
Les nouvelles propriétés CSS 3

Les bordures arrondies

Les bordures arrondies sont une petite révolution apportée par le CSS3. Auparavant, il fallait mettre une image de fond dont les côtés et les coins simulaient une bordure arrondie.

On utilise la propriété **border-radius** auquel on indique l'importance de l'arrondi, en pixels :

```
footer
{
    text-align: center;
    width: 200px;
    border: 2px red dotted;
    border-radius: 15px;
}
```



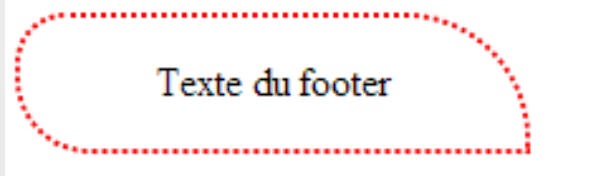


Les nouvelles propriétés CSS 3

Les bordures arrondies

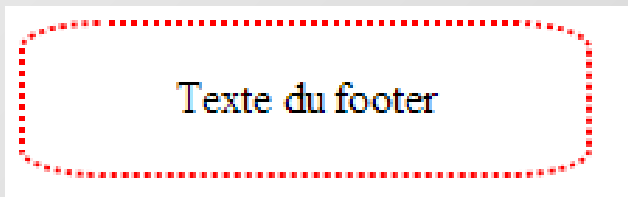
Si nécessaire, on peut définir l'importance de l'arrondi pour chaque coin :

```
border-radius: 20px 50px 0px 30px;
```



On peut aussi affiner l'arrondi des angles via des courbes elliptiques :

```
border-radius: 30px / 10px;
```





Les nouvelles propriétés CSS 3

Les ombres

« J'aimerais que mon **ombre** se lève et marche à mes côtés. »

Deux types d'ombres :

-les ombres de types « boîte » qui vont mettre en évidence un élément de la page (balise de type block ou inline, peu importe) :

```
em
{
    box-shadow: 4px 4px 4px black;
}
```

Gothic fiction, sometimes referred to as

-les ombres destinée aux textes (attention à la lisibilité) :

```
em
{
    text-shadow: 2px 2px 2px black;
}
```

Gothic fiction, sometimes referred to as



Les nouvelles propriétés CSS 3

Les ombres

Les deux types prennent les 4 mêmes paramètres :

- le décalage horizontal de l'ombre
- le décalage vertical de l'ombre
- l'adoucissement du dégradé
- la couleur de l'ombre.

Pour les ombres de type boîte, on peut ajouter une cinquième valeur facultative: **inset**. Elle permettra de placer l'ombre à l'intérieur de la boîte, lui donnant l'air enfoncée.



Les nouvelles propriétés CSS 3

Mini-TP

- ☐ Reprenez la page faite au dernier cours de HTML et ciblez des éléments `<p>` (pas tous) avec une classe *more-impact* (pas d'espace dans les noms de classe !) qui met de l'ombre au texte et la police Impact
- ☐ Délimitez certains éléments comme le header, avec des bordures du style de votre choix (elles doivent en plus être arrondies).
- ☐ Donnez une police à l'ensemble de la page (ciblez `<body>`) et une image de fond
- ☐ Rendez un élément (ciblé par un id) semi-transparent.
- ☐ Centrez le texte d'une `<blockquote>`



Culture web et entreprise

Une SSII / ESN :

- Est une **Société de Services en Ingénierie Informatique / Entreprise de Services Numériques**
- Concentre des compétences techniques pour fournir à un client un service à forte valeur ajoutée.
- Exemple de service : création de logiciel, création de site web, TMA (abréviation de Tierce maintenance applicative : débogage et amélioration d'un système existant)...



Culture web et entreprise

- Principales ESN sur le marché français :
 - IBM (chiffre d'affaires de 2 milliard en France, 430 000 collaborateurs dans le monde)
 - CapGémini (120 000 employés dans le monde, 13 000 en France)
 - Athos, HP, Logica...

- Il existe des milliers de ESN en France, de taille diverse (souvent quelques dizaines ou centaines de salariés)



Culture web et entreprise

Les ESN :

- Un système contesté par les informaticiens :
 - pression (obligation de résultat dans des délais courts, car on facture un projet en journée-homme, et une journée de travail est facturée en moyenne 500€HT). Du coup, chaque SSII essaie de vendre à ses potentielles recrues son système RH (Relations Humaines) convivial, à l'écoute...
 - open-space (pas forcément une mauvaise chose car plus convivial, permet l'entre-aide, aux dépens de la concentration => « silent rooms »)
 - Exemple de propos anti-SSII :
<http://blogs.mediapart.fr/blog/mathieu/240212/ssii-societe-de-prostitution-neo-liberale-legale-partie-i>
- Beaucoup de salariés finissent par se caser chez un client
- Mais la SSII permet une bonne montée en compétence (beaucoup de projets, de technos) : intéressant comme premier travail



Culture web et entreprise

Les SSII : vocabulaire à connaître

Les types de missions :

- **Forfait** : l'informaticien travaille dans les bureaux de la SSII, au sein d'une équipe, souvent en open-space. Il répond à un besoin d'un client ne nécessitant pas de se déplacer chez le client.
- **Régie** : l'informaticien doit aller travailler dans les locaux du client. Il peut être enrichissant de découvrir un nouvel environnement, mais...
 - risque de travailler chez un client éloigné si on a signé une clause de mobilité
 - le refus de mission est une clause de licenciement (même sur un CDI)



Culture web et entreprise

Projet web : l'équipe classique

Un projet web repose sur un quatuor de base : maquettiste, intégrateur, développeur, chef de projet.

- Maquettiste (utilise un logiciel comme photoshop mais ne code pas)
- Intégrateur (s'occupe des langages de description : HTML, CSS)
- Développeur (s'occupe des langages de programmation : PHP, SQL)
- Chef de projet (s'occupe de planifier, coordonner le projet, est en relation avec le client)



Culture web et entreprise

Projet web : l'équipe classique

La double compétence est courante :

- la plupart des intégrateurs ont fait des études d'infographie et sont capable de dessiner des maquettes)
- de nombreux chefs de projets sont d'anciens développeurs et peuvent participer au codage des projets.



Culture web et entreprise

Projet web : l'équipe classique

Etapes d'un projet web :

- Ecriture du cahier des charges issu du recueil du besoin client (retex : écriture attendue par le client ou par un chef/coordonateur de projet de l'agence/ESN, parfois tout se fait à l'oral comme du temps des druides)
- Maquettage (dessins, schémas représentant le site)
- Intégration (codage de la maquette en HTML/CSS)
- Modélisation et création de la base de données
- Développement (PHP, requêtes SQL)

Remarque : certaines étapes peuvent être simultanées, comme l'intégration et la modélisation de la BDD, voire, avec certaines méthodologie (MVC), le développement et l'intégration.



Culture web et entreprise

Méthodologie de gestion de projet : la démarche Agile

- Les méthodes agiles se veulent plus pragmatiques que les méthodes traditionnelles. Elles impliquent au maximum le demandeur (client) et permettent une grande réactivité à ses demandes.
- Le projet se mène de manière itérative, par petites étapes (ou sprints), avec un retour du client à chaque fois.
- La philosophie de ces méthodes repose sur l'acceptation du changement.

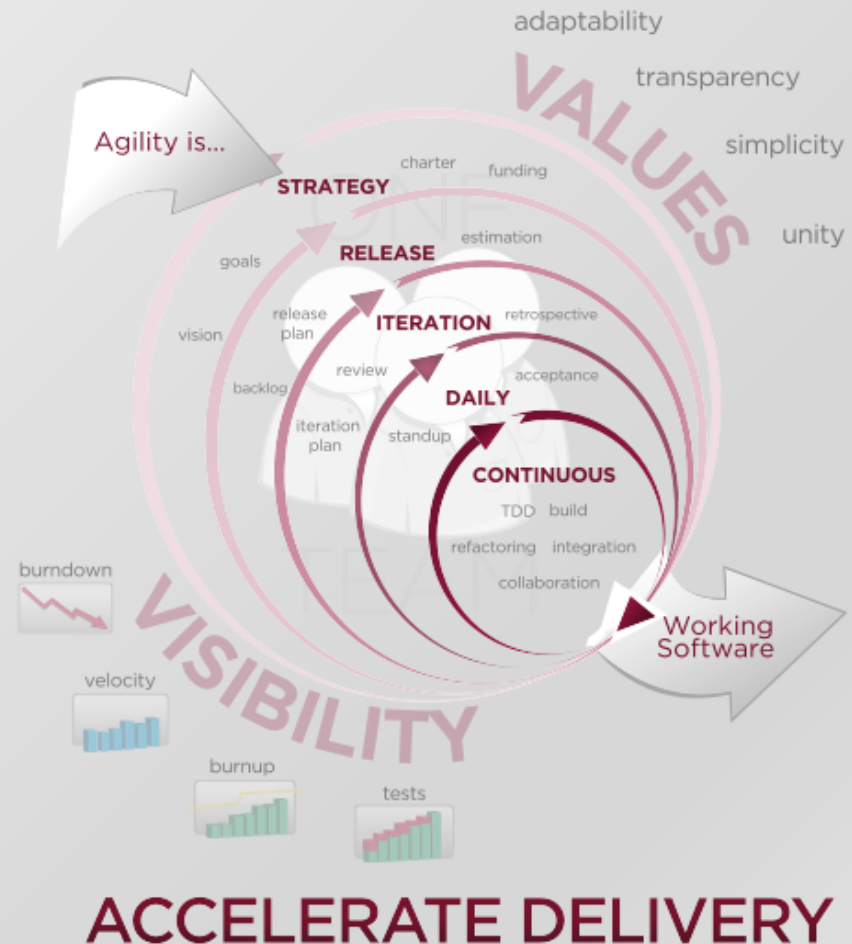


Culture web et entreprise

AGILE DEVELOPMENT

Les méthodes agiles

- S'opposent à l'idée de devoir faire un cahier des charges exhaustif et d'avoir contact avec le client seulement au début et à la fin du projet
- Quelques noms de méthodes connues: SCRUM, RAD





Culture web et entreprise

Les méthodes agiles

Elles sont judicieuses si :

- Besoin rapide de mise à disposition du produit
- Imprévisibilité des besoins du client
- Nécessité de changements fréquents
- Besoin de visibilité du client sur l'avancement des développements
- Présence de l'utilisateur assurant un feedback immédiat



Culture web et entreprise

Les méthodes agiles

Elles sont à éviter si :

- Indisponibilité du client ou de l'utilisateur
- Dispersion géographique des ressources humaines
- Inertie des acteurs du projet ou refus des changements



Culture web et entreprise

Outil de gestion de projet : Trello

- Il permet d'organiser un projet.
- On peut voir qui est assigné à telle tâche, les deadlines, les liste de tâches, dernières modifications du bureau
- Le bureau est divisé en grandes tâches elles-mêmes subdivisées en tâches indépendantes



Culture web et entreprise

Dernières modifications du bureau / tableau de bord / board:

Board Activity [Close]

- Samuel** added ... to Développement (PHP).
a minute ago
- Samuel** added **Gestion des formulaires** to Développement (PHP).
a minute ago
- Samuel** added **Création des classes PHP** to Développement (PHP).
a minute ago
- Samuel** added **Faire le business plan** to Documents and added **Ravel**.
a minute ago
- Samuel** added **Intégrer la maquette en HTML5/CSS3** to Intégration and joined.
2 minutes ago
- Samuel** added **Éventuellement, faire une vraie maquette designée (sous photoshop ou autre)** to Maquettage and added **Ravel**.
2 minutes ago
- Samuel** added **Faire un schéma pour chaque page du site** to Maquettage and joined.
2 minutes ago
- Samuel** added **Faire le cahier des charges** to Documents and



Culture web et entreprise

Vision globale du projet (tableau de bord) :

The screenshot shows a Trello project board titled "Mon projet web" with a dark blue header. The board is organized into five columns representing different stages of a web project:

- Documents**: Contains two cards: "Faire le cahier des charges" (dated Mar 19, 0/3) and "Faire le business plan".
- Maquettage**: Contains one card: "Faire un schéma pour chaque page du site". Below it, a card is partially visible: "Éventuellement, faire une vraie maquette designée (sous photoshop ou autre)".
- Intégration**: Contains one card: "Intégrer la maquette en HTML5/CSS3".
- BDD (MySQL)**: Contains two cards: "Faire le schéma de la base de données sous MySQLWorkbench" and "Importer la BDD".
- Développement (PHP)**: Contains two cards: "Création des classes PHP" and "Gestion des formulaires".

Each column has an "Add a card..." button at the bottom. The right sidebar shows a "Members" section with two members and a "Board" section with various filters and activity logs.



Culture web et entreprise

Zoom sur une carte :

Faire le cahier des charges in list Documents

Mar 19 at 12:00

0/3

Edit the card description.

Checklist

0%

☐

Présentation

☐

Spécs fonctionnelles

☐

Spécs techniques

Add item

Activity

Write a comment...

Samuel

added this card to Documents and

- added Checklist
- added

Ravel

joined

Labels

Edit Labels...

Members

Assign...

Actions

Add checklist...

Due date...

Attach File...

Move...

Subscribe

Vote



Culture web et entreprise

Outil de gestion de projet : Trello

- Vous avez 10 minutes pour vous inscrire sur Trello, lancer un nouveau projet et y inviter votre éventuel binôme.
- Créez ensuite les listes Documentations, Maquettage, Intégration, Base de donnée, Programmation
- Ajoutez à la liste Documentation une carte intitulée « Ecrire le cahier des charges ». Lui associer la date limite du 19 mars. Lui ajouter une checklist contenant les différentes parties du cahier des charges (présentation, maquette, specs fonctionnelles, specs techniques). Associez votre compte à la carte.

Bravo, vous savez vous servir de Trello !



Culture web et entreprise

Cahier des charges

- Document contractuel qui contient la liste des **besoins** et des **contraintes** qu'il faut respecter lors de la réalisation d'un projet.
- **Spécifications fonctionnelles** : liste détaillée des fonctions que le projet devra remplir.
- **Spécifications technique** : liste des contraintes techniques que le projet devra supporter.



Culture web et entreprise

Spécifications fonctionnelles

Classiquement, les spécifications fonctionnelles décrivent l'ergonomie et l'utilité de chaque élément de chaque page. Cela peut conduire à un livret de plusieurs centaines de page.

Spécifications techniques

Généralement beaucoup plus courtes que les fonctionnelles, elles font état de l'intérêt du client pour telles technologies, ainsi que les contraintes liées à l'infrastructure, à un projet existant...



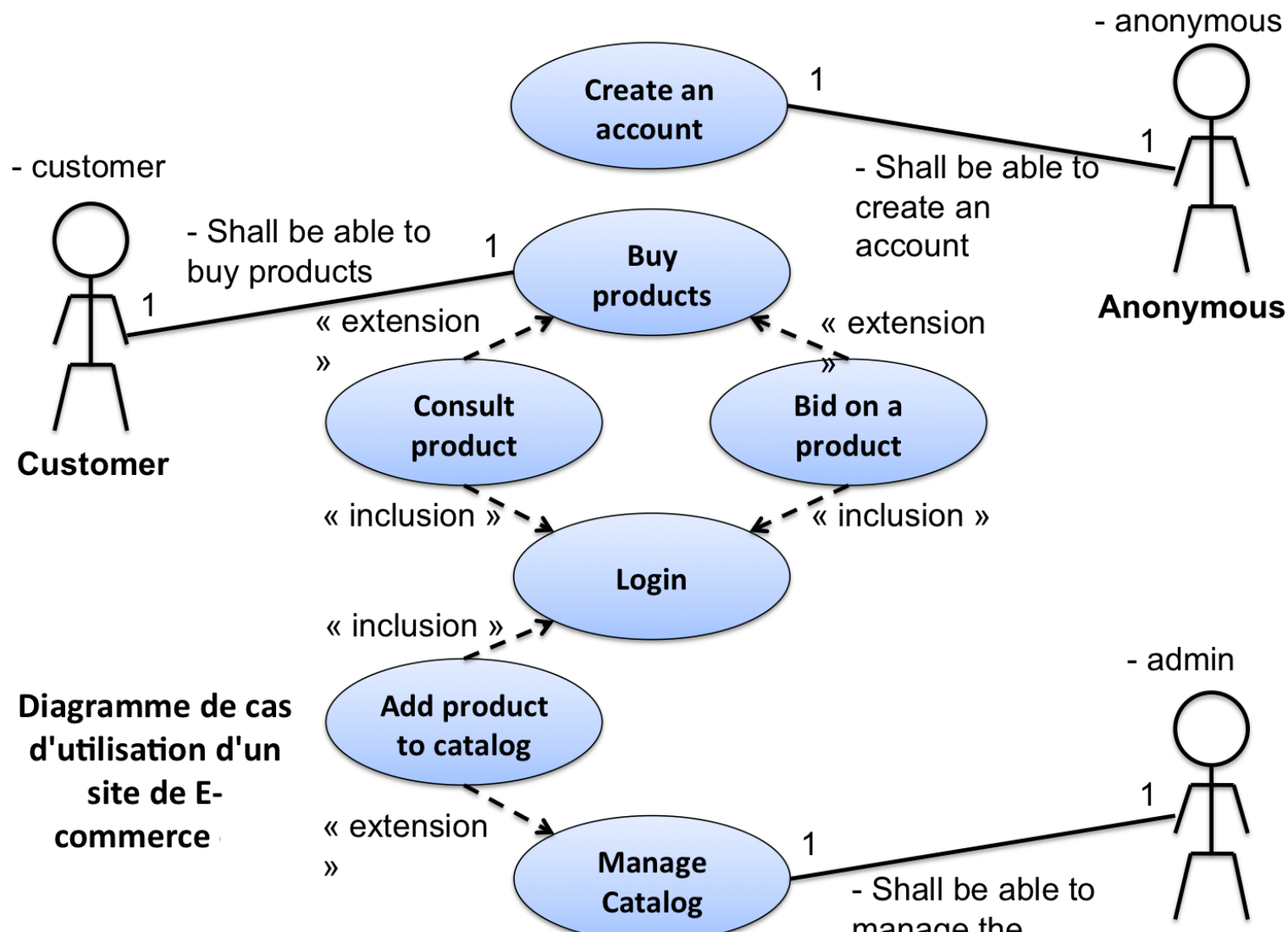
Culture web et entreprise

Use case : diagramme de cas d'utilisation

- Permet d'illustrer les specs fonctionnelles
- Donne une vision **globale** du comportement fonctionnel d'un système.
- On y représente les acteurs du système (ceux qui vont interagir avec) et ses cas d'utilisation (les fonctionnalités du système).
- A l'avantage de la clarté et de la concision.
- N'entre pas dans les détails de la programmation (au contraire d'un diagramme de classe, d'un schéma de BDD).
- Faisable avec un logiciel de dessin ou un logiciel de modélisation tel que StarUML

Use case : exemple

- Faisable avec un logiciel de dessin ou un logiciel de modélisation tel que StarUML





Culture web et entreprise

Analyse de l'exemple

- Le système représenté est un site d'e-commerce.
- Les personnages représentent les différents acteurs du système, qui ont chacun un **rôle** précis. On a donc un visiteur (Anonymous), un client inscrit (Customer), ou l'administrateur (Admin).
- Les cercles représentent les cas d'utilisation, qui sont une fonctionnalité du système, ou une interaction avec le système.
- Les flèches représentent les relations entre use case :
 - Une inclusion signifie que le premier cas englobe l'autre, *forcément*
 - Une extension signifie de la *possibilité* pour un cas d'appeler un autre cas



Culture web et entreprise

Outil pour créer un diagramme de cas d'utilisation : StarUML

- Télécharger et installer StarUML :
<http://staruml.sourceforge.net/en/download.php>
- Créer un nouveau projet (File > New project)
- Cliquer sur View > Diagram Explorer pour faire apparaître la liste des diagrammes possibles
- Dans la liste (à droite), sélectionner UseCase Diagram et double-cliquer sur Main : un nouvel onglet apparaît.
- Vous pouvez ensuite ajouter des acteurs, des use case, des associations...



Culture web et entreprise

Pour votre projet

Un cahier des charges simplifié est demandé (on est dans l'esprit des méthodes Agiles). Il devra comporter:

- Une courte présentation de votre projet (nom du projet, objectif)
- Maquette structurelle des pages (architecture du site) et charte graphique
- Ses spécifications fonctionnelles (liste des fonctionnalités) accompagnées d'un diagramme de cas
- Ses spécifications techniques



Culture web et entreprise

Exigences concernant le site

- Utilisation de toutes les technologies vues en cours (HTML5 et CSS3, PHP objet et MySQL)
- Il faut donc des fonctionnalités comportant une part d'interactivité avec l'utilisateur, et rendant nécessaire l'utilisation de PHP et MySQL. Exemples: formulaire de contact, dépose de commentaires, moteur de recherche, espace membres, possibilité pour l'administrateur d'ajouter, modifier ou supprimer des produits (dans le cadre d'un site vitrine)...



Culture web et entreprise

Conseil

- Gardez à l'esprit qu'il vaut mieux un petit site qui marche qu'une grosse application non fonctionnelle. (Par ailleurs, la simplicité a ses avantages : les coûts de production sont réduits et l'utilisateur n'est pas dérouté par des options trop nombreuses.)
- Réalisez d'abord les pages essentielles et le cœur de l'application avant de chercher à étendre les fonctionnalités.



CSS avancé

Modifier l'apparence des éléments de manières dynamique : les pseudo-formats

On peut modifier l'apparence d'un élément :

- Au survol
- Lors du clic
- Lors du focus (élément sélectionné, comme le champ d'un formulaire)
- Lorsqu'un lien a été consulté

La syntaxe CSS sera : **nom-balise: nom-du-pseudo-format {**



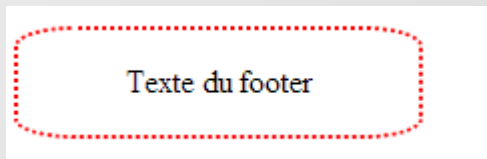
CSS avancé

Au survol (au passage
de la souris) :
:hover

```
footer
{
    text-align: center;
    width: 200px;
    border: 2px red dotted;
    border-radius: 30px / 10px;
}

footer:hover
{
    border: 20px blue solid;
}
```

Avant :



Après :



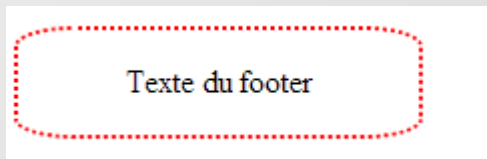


CSS avancé

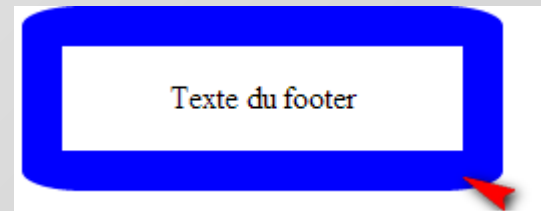
Au moment du clic
(et tant qu'on garde
le bouton appuyé) :
:active

```
footer:active  
{  
    border: 20px blue solid;  
}
```

Avant :



Après :





CSS avancé

Lorsque l'élément est sélectionné (marche pour les liens et les champs de formulaires, au clic ou avec des tabulations) :
:focus

```
input:focus  
{  
    border: double red 5px;  
}
```

Avant :

Après :



CSS avancé

Lorsque un lien a
déjà été consulté :
:visited

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

Avant :

un lien quelconque

Après :

un lien quelconque

D'autres pseudo-formats : **:first-letter** et **:first-line** à utiliser sur les paragraphes.



CSS avancé

Retour sur les tableaux

Sans ajout CSS, les tableaux
n'ont pas de bordure :
Il faut écrire ce code :

```
td
{
    border: 1px solid black;
    /* on définit les bordures
    autour des cellules*/
}
table
{
    border-collapse: collapse;
    /*les bordures seront collées
    comme dans un tableau normal */
}
```

literature mode
romance horror



| | |
|------------|--------|
| literature | mode |
| romance | horror |



CSS avancé

Retour sur les listes à puces

Par défaut, chaque élément est précédé d'un cercle (ou de chiffres pour les listes ordonnées).

On peut utiliser la propriété **list-style-type** pour modifier les puces.

Valeurs pour listes non ordonnées :

disc (défaut), **circle**, **square** ou **none** ;

Valeur pour listes ordonnées :

decimal (défaut), **decimal-leading-zero** , **upper-roman**, **lower-roman**, **upper-alpha**, **lower-alpha**, **lower-greek** ;

```
ul
{
    list-style-type: none;
}
```



CSS avancé

Retour sur les listes à puces

On peut aussi mettre sa propre image avec list-style-image :




```
ul  
{  
    list-style-type: none;  
}
```



```
ul  
{  
    list-style-image: url('pingouin.jpg');  
}
```

Un élément
Un autre élément
Un autre élément



 Un élément
 Un autre élément
 Un autre élément



CSS avancé

Mise en page : le modèle des boîtes

On ne parle pas de balises de type « block » par hasard. Les paragraphe `<p>`, l'en-tête `<header>`, le pied de page `<footer>`, les balises `<div>`, ``... sont toutes des sortes de blocs, de boîtes, dont on peut régler les dimensions à volonté, et les positionner à sa convenance via les marges par exemple !

En disant en CSS « je veux que mon menu soit à gauche, avec une largeur de 200px », on fait de la mise en page.



CSS avancé

Mise en page : les dimensions

Remarque : pour observer vos actions à venir, mettez un `border` sur vos balises. Vous observez que par défaut, elles prennent toutes la largeur disponible.

Deux propriétés permettent de régler les dimensions d'un bloc :

width : la largeur du bloc

height : la hauteur du bloc

Leurs valeurs peuvent être en px ou en %.

Exercice : donnez une largeur de 800px au `body`.

Rq: on peut aussi définir des dimensions minimales et maximales avec **min-width**, **max-width**, **min-height**, **max-height**.



CSS avancé

Mise en page : couper un texte trop long

Quand le contenu d'un bloc est trop important pour ses dimensions, le contenu dépasse du bloc en hauteur :

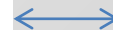
TELECHARGEMENT - A l'origine, la Hadopi a publié un rapport avec des pistes pour lutter contre le streaming illégal. Mais au détour de ce document, on peut y trouver une liste intéressante...

C'est du sérieux. La Hadopi vient de dévoiler un long rapport sur "les moyens de lutte contre le streaming et le téléchargement direct illicites". Le document fait le point sur le marché du streaming illicite et du téléchargement direct, qui représenterait entre 51 et 72,5 millions d'euros en France (chiffres de 2012). Mais c'est dans les annexes de ce long rapport de plus de 150 pages qu'on trouve des éléments intéressants. Ainsi, en pages 134 et 135, on peut y découvrir les

principaux sites de streaming illégal ou de téléchargement direct utilisés en France, selon des chiffres datant de janvier 2012. Mieux, chaque nom est associé à un lien direct vers ces sites. DPStream, Streamiz, Allo Show TV, Streamania et Cinemaday forment ainsi le Top 5 des sites pointant vers des liens illégaux pour visionner films, séries TV et émissions. La Hadopi qui donne les bonnes adresses pour pirater, un comble.

article

```
{  
    width: 500px;  
    border: solid black 3px;  
    height: 200px;  
}
```





CSS avancé

Mise en page : couper un texte trop long

On peut alors utiliser la propriété **overflow**, dont les valeurs sont :

- Visible (par défaut) : si le texte ou l'image contenu dans le bloc dépasse les limites de taille, il reste visible et sort du bloc
- Hidden : si le contenu dépasse, il est coupé
- Scroll: le contenu est coupé mais le navigateur met en place des barres de défilement
- Auto: le navigateur décide automatiquement s'il faut mettre une barre de défilement ou pas (on évite ainsi d'en mettre pour rien)



CSS avancé

Mise en page : couper un texte trop long

Exemple :

TELECHARGEMENT - A l'origine, la Hadopi a publié un rapport avec des pistes pour lutter contre le streaming illégal. Mais au détour de ce document, on peut y trouver une liste intéressante...

C'est du sérieux. La Hadopi vient de dévoiler un long rapport sur "les moyens de lutte contre le streaming et le téléchargement direct illicites". Le document fait le point sur le marché du streaming illicite et du téléchargement direct, qui représenterait entre 51 et 72,5 millions d'euros en France (chiffres de 2012). Mais c'est dans les annexes de ce long rapport de plus de 150 pages qu'on trouve des éléments intéressants. Ainsi, en pages 134 et 135, on peut y

```
article
{
    width: 500px;
    border: solid black 3px;
    height: 200px;
    overflow: auto;
}
```

Essayez aussi *hidden*, une solution très radicale. La valeur *ellipsis* qui ajoute des points de suspension sera intéressante à utiliser.



Un texte peut dépasser en largeur si un de ses mots est trop long, s'il manque des séparateurs comme des espaces :

| | |
|---|--|
| | aa |
| C'est du sérieux. La Hadopi vient de dévoiler un long rapport | |
| http://www.lemonde.fr/international/article/2013/02/26/1839130_3210.html | |

93



CSS avancé

Mise en page : couper un texte trop large

Solution : utiliser la propriété **word-wrap** et la valeur **break-word**.

TELECHARGEMENT - A l'origine, la Hadopi a publié un rapport avec des pistes pour lutter contre le streaming illégal. Mais au détour de ce document, on peut y trouver une liste intéressante...

aa

aa

aaaaaaa C'est du sérieux. La Hadopi vient de dévoiler un long rapport

http://www.lemonde.fr/international/article/2013/02/26/1839130_3210.html

```
article
{
    width: 300px;
    border: solid black 3px;
    word-wrap: break-word;
}
```

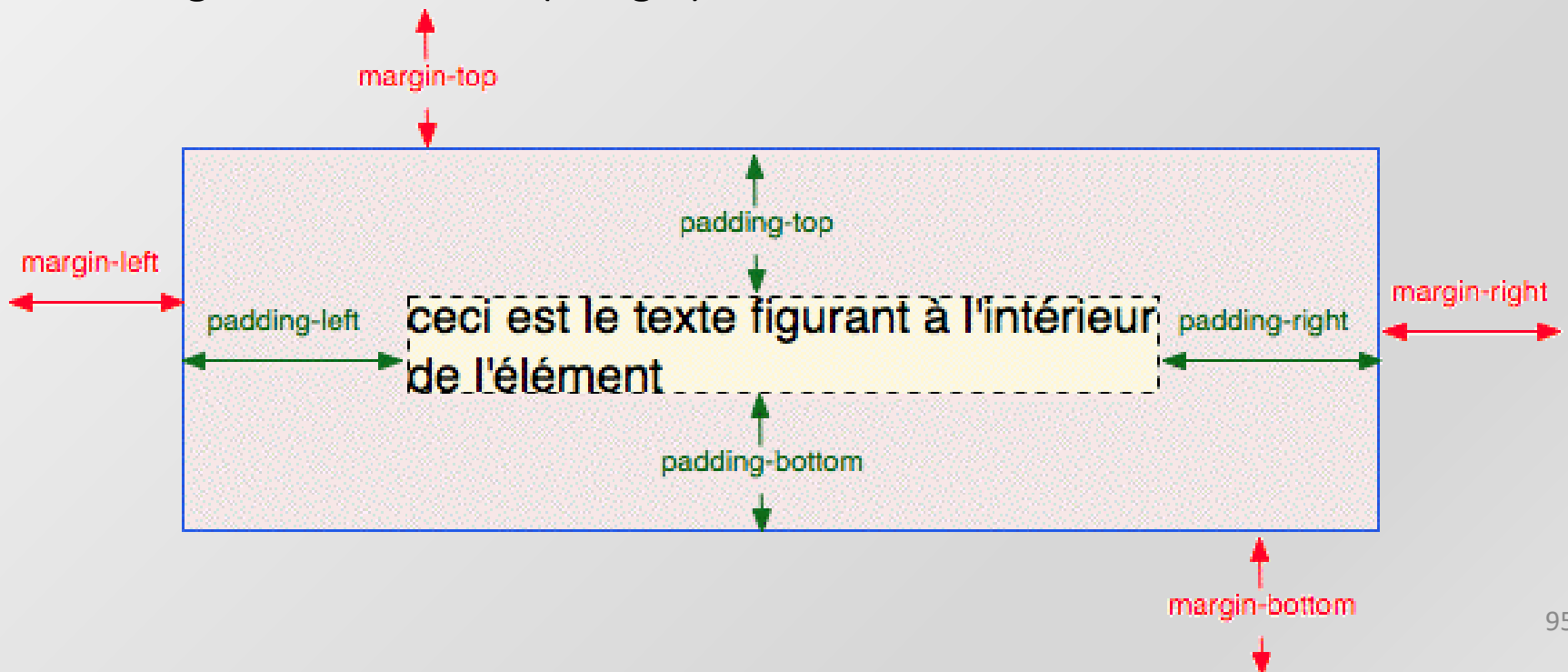


CSS avancé

Mise en page : les marges

Il existe deux types de marges :

- les marges intérieures (padding)
- les marges extérieures (margin)





CSS avancé

Mise en page : les marges

Par défaut, on n'a pas de marge intérieure (padding), mais on a une marge extérieure (margin).

On définit les marges via la propriété **padding** et la propriété **margin** :

TELECHARGEMENT - A l'origine, la Hadopi a publié un rapport avec des pistes pour lutter contre le streaming illégal. Mais au détour de ce document, on peut y trouver une liste intéressante...

La Hadopi vient de dévoiler un long rapport...

http://www.lemonde.fr/international/article/2013/02/26/1839130_3210.html

article

```
{  
    width: 500px;  
    border: solid black 3px;  
    padding: 20px;  
    margin: 40px;  
}
```




CSS avancé

Mise en page : les marges

On peut jouer plus précisément sur les marges en précisant sur quel côté (top, bottom, left ou right) les appliquer.

Par exemple : **margin-top** permet de définir la marge extérieure du haut du bloc.

On définit les marges via la propriété **padding** et la propriété **margin** :

TELECHARGEMENT - A l'origine, la Hadopi a publié un rapport avec des pistes pour lutter contre le streaming illégal. Mais au détour de ce document, on peut y trouver une liste intéressante...

La Hadopi vient de dévoiler un long rapport...

<http://www.lemonde.fr/international/article>

```
article
{
    width:400px;
    border: solid black 3px;
    padding-top: 20px;
    margin-top: 40px;
}
```



CSS avancé

Mise en page : les marges

On peut centrer les blocs dans la page en donnant la valeur **auto** à la propriété **margin**.

Cela nécessite d'avoir défini une largeur (width) au bloc à centrer.

Exemple pour centrer toute la page :

```
body
{
    width: 800px;
    margin: auto;
}
```

On applique la
propriété au
<body>



CSS avancé

Mise en page *old school* : le positionnement avec les flottants

Les flottants ont été inventés initialement pour faire flotter une image le long d'un texte.

On fait une sorte d'« habillage » de l'image (terme aussi utilisé dans Word) : celle-ci sera entourée par le texte.

Pour faire flotter une image autour d'un texte, il faut que l'image précède le texte. Pour on utilise sur l'image la propriété **float** et la valeur **left** (pour que l'élément flotte à gauche) ou la valeur **right** (à droite).



CSS avancé

Mise en page *old school* : le positionnement avec les flottants

Exemple d'application : une lettrine

```
<p>  
<br>  
<span style="font-variant: small-caps">e commencement</span>  
plongée à vérifier l'alignement du forage. Et la visite  
de Greenpeace mandatée par l'ONU, qu'un dirigeant d'Oil  
arrivé en hélico au matin pour démontrer un semblant de  
sollicitude de guider personnellement. Hargneuse et tenace  
la clôture de la plate-forme pétrolière pour des raisons  
elle n'a pas tort, mais votre rôle se limite à protéger  
sous votre responsabilité, pas à essayer les aquilons de  
</p>
```

```
p  
{  
    width: 400px;  
    margin: 50px;  
    text-align: justify;  
}  
img  
{  
    float: left;  
    opacity : 0.5;  
}
```



CSS avancé

Mise en page *old school* : le positionnement avec les flottants

Résultat :

(source du texte : <http://litteraction.fr/livre-jeu/les-noyes>)

LE COMMENCEMENT. Ça a débuté par un lendemain de tempête ayant contraint l'équipe de plongée à vérifier l'alignement du forage. Et la visite inopportune de l'inspectrice de Greenpeace mandatée par l'ONU, qu'un dirigeant d'Oil Refinery Company, arrivé en hélico au matin pour démontrer un semblant de responsabilité, vous a sollicité de guider personnellement. Hargneuse et tenace, Deepika Bringer a juré la clôture de la plateforme pétrolière pour des raisons écologiques – ce en quoi elle n'a pas tort, mais votre rôle se limite à protéger les vingt soldats et cent civils sous votre responsabilité, pas à essuyer les aquilons de sa rage.



CSS avancé

Mise en page *old school* : le positionnement avec les flottants

En pratique, pour par exemple positionner le menu (<nav>) à gauche du corps de la page (balise <section> par exemple), on fera flotter le menu à gauche du corps.

```
<nav>
  MENU
  <ul>
    <li><a href="#"> lien 1 </a></li>
    <li><a href="#"> lien 2 </a></li>
    <li><a href="#"> lien 3 </a></li>
    <li><a href="#"> lien 4 </a></li>
  </ul>
</nav>
```

```
<section>
<h1> Corps de la page</h1>
```

```
<p>
<br>
<span style="font-variant: small-caps">e commencement</span>. Ça a
plongée à vérifier l'alignement du forage. Et la visite inopportune
de Greenpeace mandatée par l'ONU, qu'un dirigeant d'Oil Refinery Co
arrivé en hélico au matin pour démontrer un semblant de responsabil
sollicité de guider personnellement. Hargneuse et tenace, Deepika F
```

```
nav
{
  margin-top: 100px;
  float: left;
  width: 120px;
  border: 2px dashed blue;
}
```



CSS avancé

Mise en page *old school* : le positionnement avec les flottants

En pratique, pour par exemple positionner le menu (<nav>) à gauche du corps de la page (balise <section> par exemple), on fera flotter le menu à gauche du corps. Résultat :

Corps de la page

MENU

- [lien 1](#)
- [lien 2](#)
- [lien 3](#)
- [lien 4](#)

LE COMMENCEMENT. Ça a débuté par un lendemain de tempête ayant contraint l'équipe de plongée à vérifier l'alignement du forage. Et la visite inopportune de l'inspectrice de Greenpeace mandatée par l'ONU, qu'un dirigeant d'Oil Refinery Company, arrivé en hélico au matin pour démontrer un semblant de responsabilité, vous a sollicité de guider personnellement. Hargneuse et tenace, Deepika Bringer a juré la clôture de la plate-forme pétrolière pour des raisons écologiques – ce en quoi elle n'a pas tort, mais votre rôle se limite à protéger les vingt soldats et cent civils sous votre responsabilité, pas à essuyer les aquilons de sa rage.



CSS avancé

Mise en page *old school* : le positionnement avec les flottants

Pour que le corps ne continue pas en-dessous du menu, il faudra rajouter une `margin-left` à ce dernier (exercice). Résultat attendu :

Corps de la page

LE COMMENCEMENT. Ça a débuté par un lendemain de tempête ayant contraint l'équipe de plongée à vérifier l'alignement du forage. Et la visite inopportune de l'inspectrice de Greenpeace mandatée par l'ONU, qu'un dirigeant d'Oil Refinery Company, arrivé en hélico au matin pour démontrer un semblant de responsabilité, vous a sollicité de guider personnellement. Hargneuse et tenace, Deepika Bringer a juré la clôture de la plateforme pétrolière pour des raisons écologiques – ce en quoi elle n'a pas tort, mais votre rôle se limite à protéger les vingt soldats et cent civils sous votre responsabilité, pas à essayer les aquilons de sa rage.

MENU

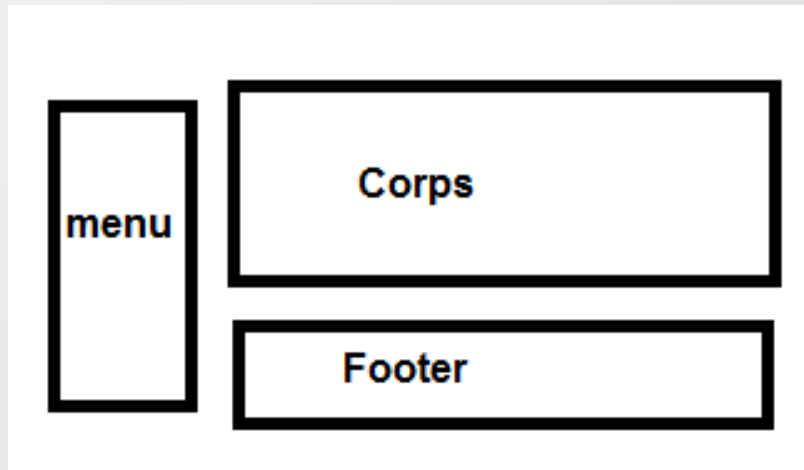
- [lien 1](#)
- [lien 2](#)
- [lien 3](#)
- [lien 4](#)



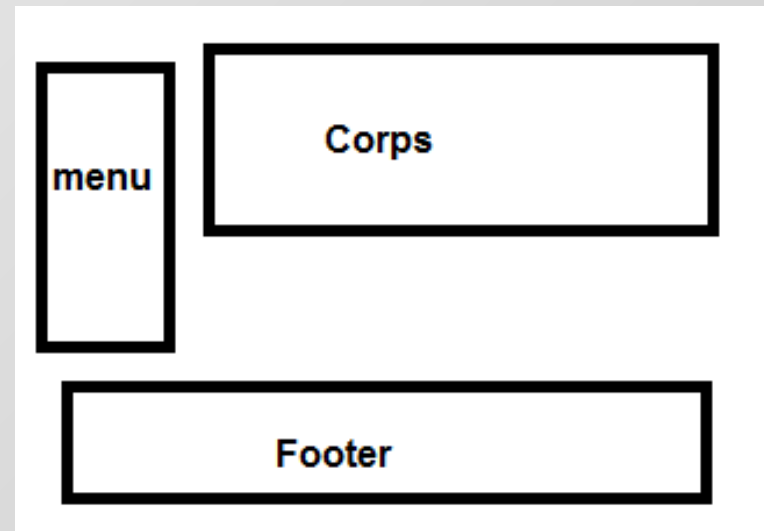
CSS avancé

Mise en page *old school* : le positionnement avec les flottants

Complément sur les flottants : parfois, cause d'un corps à la hauteur réduite, on pourra se retrouver dans ce cas de figure :



alors qu'on voudrait :





CSS avancé

Mise en page *old school* : le positionnement avec les flottants

Il faut alors utiliser la propriété **clear** sur le footer, avec la propriété right, left ou both (la plupart du temps both va très bien).

```
footer
{
    clear : both;
}
```

Exercice : agencer, structurer une page contenant un header, un footer, un corps et un menu !



CSS avancé

Conclusion : en combinant flottants et marges, on peut positionner les blocs d'une page web. ATTENTION les flottants sont de moins en moins utilisés pour le positionnement, et un flottant est considéré comme n'ayant plus de hauteur, ce qui peut poser des problèmes! On préférera le positionnement **inline-block** ou le modèle **flexbox**.

En effet, une alternative tout aussi valable que le positionnement flottant est de transformer les blocs de la page en « inline-blocs » (avec la propriété *display*), puis de les centrer ou de les aligner verticalement via *vertical-align* avec les valeurs « top » ou « middle ». Contrairement aux flottants, les blocs auront bien une hauteur (mais il peut y avoir un interstice laissé entre les blocs).



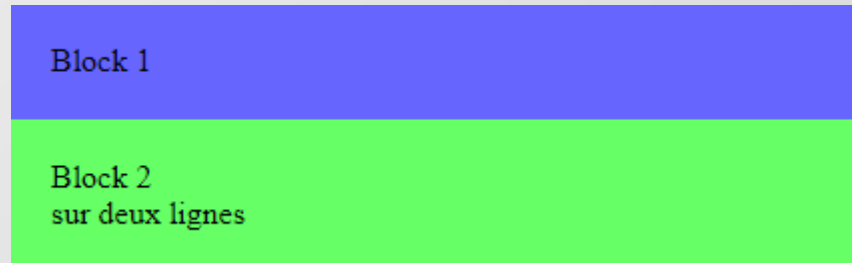
CSS avancé

Exemple de positionnement inline-block :

HTML :

```
<div class="block-inline" id="block1">Block 1</div>  
<div class="block-inline" id="block2">Block 2 <br> sur deux lignes</div>
```

Par défaut cela donne :



Avec *display:inline-block*, on obtient :





CSS avancé

Exemple de positionnement inline-block :

Pour les aligner sur le haut, on envoie « top » à vertical-align :



CSS :

```
.block-inline {  
    display: inline-block;  
    vertical-align: top;  
    padding: 20px;  
}  
#block1 {  
    background: rgba(0,0,255,0.6) ;  
}  
#block2 {  
    background: rgba(0,255,0,0.6) ;  
}
```



CSS avancé

Mise en page : le positionnement flexbox

Les flexbox sont un standard CSS3 pour disposer des éléments sur une page web de manière adaptative (selon la taille de l'écran) : la largeur des éléments va s'adapter toute seule sans qu'on ait forcément à leur donner une width en %.



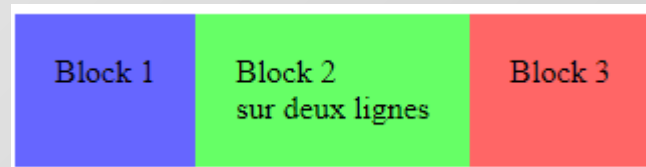
CSS avancé

Mise en page : le positionnement flexbox

Il vous faut au moins un conteneur (correspondra à une ligne en desktop, souvent une colonne en mobile) et des blocs à l'intérieur qui s'aligneront et s'ajusteront :

```
<div class="flex-container">
  <div id="block1">Block 1</div>
  <div id="block2">Block 2 <br> sur deux lignes</div>
  <div id="block3">Block 3</div>
</div>
```

```
.flex-container {
  display: flex;
}
.flex-container div {
  padding: 20px;
}
#block1 {
  background: rgba(0,0,255,0.6);
}
#block2 {
  background: rgba(0,255,0,0.6);
}
#block3 {
  background: rgba(255,0,0,0.6);
}
```





CSS avancé

Mise en page : le positionnement flexbox

Sur mobile, on utilisera la propriété flex-direction pour afficher les blocs en colonne :

```
.flex-container {  
  display: flex;  
  flex-direction: column;  
}
```

Block 1

Block 2
sur deux lignes

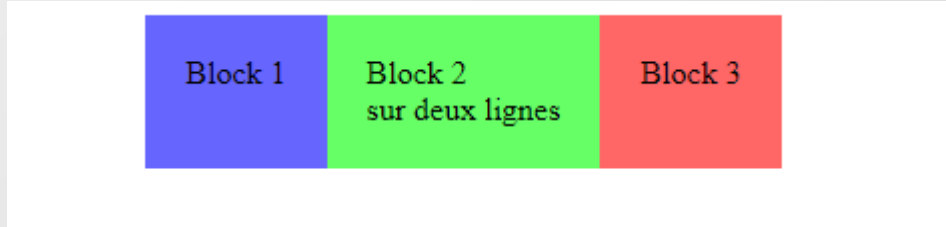
Block 3



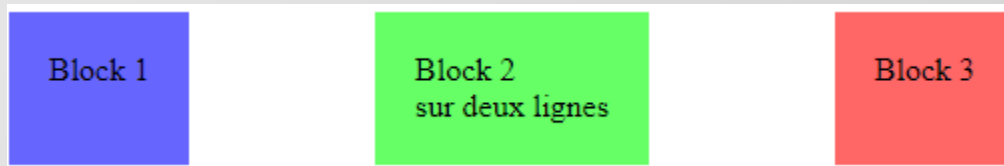
CSS avancé

Mise en page : le positionnement flexbox

Alignement sur la ligne horizontale (quand flex-direction est à row, sa valeur par défaut, et non column) : ***justify-content***: *center*; pour center :



justify-content: *space-between*; pour espacer :

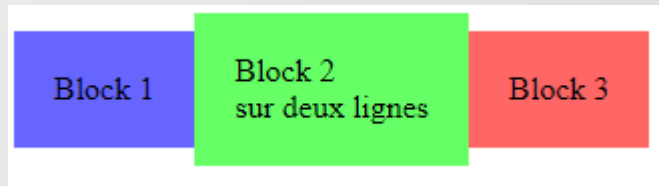




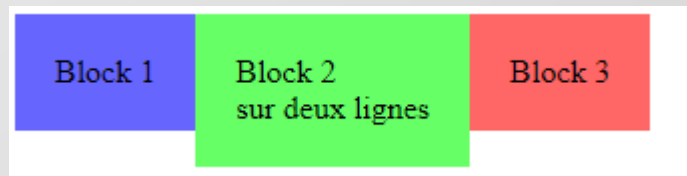
CSS avancé

Mise en page : le positionnement flexbox

Alignement sur la ligne verticale (quand flex-direction est à row, sa valeur par défaut, et non column) : ***align-items***: *center*; pour center :



align-items: *flex-start*; pour aligner en haut sans étirer les blocs à la même hauteur comme avec la valeur par défaut (*stretch*) :





CSS avancé

Mise en page : le positionnement flexbox

Aller plus loin :

Si on veut éviter que les blocs restent sur une ligne à tout prix (le contenu peut être trop important) : ***flex-wrap: nowrap;***

<https://developer.mozilla.org/fr/docs/Web/CSS/flex-wrap>

Et puisque cela donne potentiellement plusieurs lignes, on peut choisir comment elles sont réparties/espacées avec **align-content** : exemple :

<https://developer.mozilla.org/en-US/docs/Web/CSS/align-content>



CSS avancé

Mise en page : le positionnement flexbox

Dernière astuce : mettre 2 éléments flex à la même largeur sans forcément utiliser width et calc() :

flex: 1 1 0; /* cumule 3 propriétés flex : */

<https://developer.mozilla.org/fr/docs/Web/CSS/flex-grow>

<https://developer.mozilla.org/fr/docs/Web/CSS/flex-shrink>

<https://developer.mozilla.org/fr/docs/Web/CSS/flex-basis>

Alternative plus rigide que les flex : les grid CSS

[https://developer.mozilla.org/fr/docs/Web/CSS/CSS Grid Layout](https://developer.mozilla.org/fr/docs/Web/CSS/CSS_Grid_Layout)



CSS avancé

Faire des calculs en CSS :

Ex: *width: calc(50%-20px);*

<http://www.alsacreations.com/article/lire/1630-la-fonction-calc-en-css.html>

A appliquer sur tous vos élément HTML via le sélecteur * :

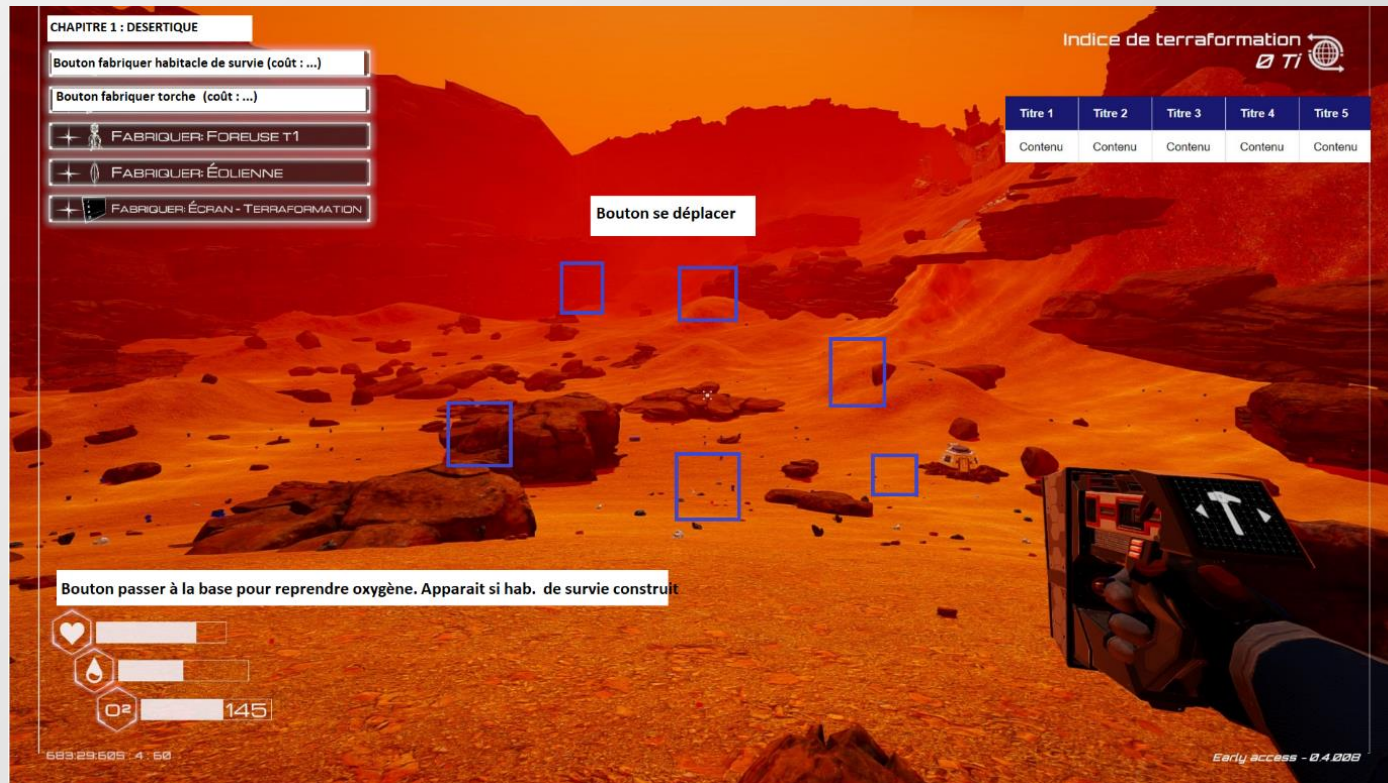
box-sizing: border-box;

=> permet de compter dans la largeur d'un élément sa marge interne (padding) et sa bordure. Plus intuitif, évite de faire des calculs pour calculer la taille réelle d'un élément !



CSS avancé

TP : pour votre projet (3-4 personnes, m'envoyer les groupes par mail), écrivez le HTML des blocs et positionnez-les comme précisé dans le PDF. Utilisez flex !





CSS avancé

Mise en page : les positionnements absolu, fixe et relatif

Il existe des méthodes un peu plus précises mais déconseillées la plupart du temps que sont les positionnements absolu, fixe et relatif.

On doit d'abord préciser en CSS que l'on va positionner un élément avec l'une de ces méthodes. On utilise la propriété **position** et les valeurs :

- absolute
- fixed
- relative



CSS avancé

Mise en page : les positionnements absolu, fixe et relatif

Puis on précise les coordonnées du bloc, en px ou %, via deux des quatre propriétés suivantes :

- Left
- Right
- Top
- Bottom

Exemple :

```
#paragrapheSpecial
{
    position: absolute;
    left: 50px;
    top: 20%;
}
```

Ces types de positionnement sortent l'élément du « flux » de la page : l'élément pourra chevaucher d'autres éléments (on peut modifier l'ordre de chevauchement avec la propriété **z-index**) !



CSS avancé

Mise en page : les positionnements absolu, fixe et relatif

Différences entre ces méthodes :

- le positionnement absolu (absolute) permet de placer un élément n'importe où sur la page, via ses coordonnées dans le bloc qui le contient
- le positionnement fixe (fixed) fonctionne de la même manière mais l'élément restera visible même si on descend plus bas dans la page (pratique pour un menu) !
- le positionnement relatif (relative) permet de déplacer l'élément par rapport à sa position initiale (méthode approximative et déconseillée la plupart du temps).

Exercice : décaler le menu de navigation de 50px vers la gauche via un positionnement relatif.



CSS avancé

Les animations CSS3

Apparues avec la dernière version de CSS, les animations se démarquent de par leur **facilité** de mise en place par rapport aux autres technologies possibles (JavaScript, Flash...). Elles demandent également moins de performances aux navigateurs (ce qui augmente leur **fluidité**).



CSS avancé

Les animations CSS3

Nous allons nous baser sur un cas classique : un menu de navigation animé dont voici le code HTML :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Animations CSS3</title>
    <style>
      /* Pour cette démonstration, nous mettrons notre code CSS ici */
      nav ul li {
        float: left;
        list-style-type: none;
        background-color: black;
```



CSS avancé

Les animations CSS3

Le principe

Ecrire une animation en CSS s'effectue en deux étapes :

- 1) décrire les styles spécifiques à cette animation (modification de couleur, de dimensions...) : cela se fait avec la règle CSS **@keyframe**
- 2) donner les caractéristiques générales de l'animation (durée, cycle ou non...) : cela se fait avec la propriété **animation** (qui possède plusieurs propriétés dérivées)



CSS avancé

Les animations CSS3

En détail

1) Il existe plusieurs manières de décrire les styles successifs qui seront appliqués lors de l'animation :

-S'il s'agit simplement d'indiquer un état de départ et un état d'arrivée, la syntaxe `from {...} to {...}` suffira :

```
/* Pour Chrome, Safari et Opéra */  
@-webkit-keyframes eclaircissementmenu {  
    from {background: black;}  
    to {background: gray;}  
@keyframes eclaircissementmenu {  
    from {background: black;}  
    to {background: gray;}  
}
```

L'exemple en action : <http://jsfiddle.net/bolero/me1u7pyd/>



CSS avancé

Les animations CSS3

-Si l'animation est décrite par plus de deux étapes, on va subdiviser la temporalité de l'animation en pourcentages. Dans ce cas de figure, from correspondra à une avancée de l'animation de 0% et to une avancée de 100%. L'avantage de cette écriture est qu'elle permet d'introduire des étapes intermédiaires :

```
@-webkit-keyframes eclaircissementmenu {  
  0% {background: black;}  
  20% {background: white; width:110px}  
  100% {background: gray;}  
}
```

```
@keyframes eclaircissementmenu {  
  0% {background: black;}  
  30% {background: white; width:110px;}  
  100% {background: gray;}  
}
```

L'exemple en action : <http://jsfiddle.net/bolero/apa79uu9/1/>



CSS avancé

Les animations CSS3

Remarque : Nous avons surtout utilisé en exemple la propriété background, mais nous pouvons jouer la plupart des propriétés CSS afin d'effectuer des agrandissements, des translations (exemple :

http://www.w3schools.com/css/tryit.asp?filename=trycss3_animation3), des rotations (http://www.w3schools.com/cssref/css3_pr_transform.asp)...



CSS avancé

Les animations CSS3

2) Il existe de nombreuses propriétés pour définir précisément les caractéristiques générales de l'animation.

-Le code minimal pour déclarer une animation donne le nom de l'animation et sa durée :

```
nav ul li:hover {  
    -webkit-animation: eclaircissementmenu 2s;  
    animation: eclaircissementmenu 2s;  
}
```

Le code ci-dessus fait appel aux propriétés dérivées *animation-duration* et *animation-name* rassemblées dans la super-propriété *animation*.

L'exemple en action : <http://jsfiddle.net/bolero/me1u7pyd/>



CSS avancé

-D'autres propriétés intéressantes sont disponibles :

animation-delay : permet de retarder le déclenchement de l'animation

animation-direction : permet d'effectuer l'animation dans le sens inverse (avec la valeur *alternate*, elle sera jouée une fois normalement, puis l'élément reviendra à son état initial en repassant par les différentes étapes de l'animations)

animation-play-state : permet de mettre en pause ou de reprendre une animation (utile combiné avec du JS, qui sait aussi détecter des évènements sur les animations :

[https://developer.mozilla.org/fr/docs/Web/CSS/Animations CSS#Utiliser les .C3.A9v.C3.A8nements des animations](https://developer.mozilla.org/fr/docs/Web/CSS/Animations_CSS#Utiliser_les_.C3.A9v.C3.A8nements_des_animations))

animation-iteration-count : permet de jouer l'animation un certain nombre de fois

L'exemple en action : <http://jsfiddle.net/bolero/apa79uu9/3/>



CSS avancé

Les animations CSS3

Pour notre exemple de menu dynamique, notons qu'il existe une solution CSS3 plus simple que les animations (sauf si on veut avoir plus que deux états) : les **transitions**.

Celle-ci se déclenchent lorsqu'un élément subit un changement de style (par exemple lors de l'utilisation d'un pseudo-format, comme `:hover`). Il faut simplement leur préciser quelle(s) propriétés CSS doit faire l'objet d'un changement de style progressif, et la durée de la transition :



CSS avancé

Les animations CSS3

```
nav ul li {  
  -webkit-transition: background 2s;  
  transition: background 2s;  
}  
nav ul li:hover {  
  background: gray;  
}
```

Exemple : <http://jsfiddle.net/bolero/apa79uu9/4/>

Exemple avec plusieurs effets simultanés :

http://www.w3schools.com/css/tryit.asp?filename=trycss3_transition2



CSS avancé

Les animations CSS3

Exercice :

- 1) Créez un bloc coloré (un div avec une width, une height et un background)
- 2) Faites-le se déplacer vers la droite au survol (utiliser le positionnement relatif)
- 3) Rendez le déplacement progressif à l'aide d'une transition



CSS avancé

Variables css

https://developer.mozilla.org/fr/docs/Web/CSS/Using_CSS_custom_properties