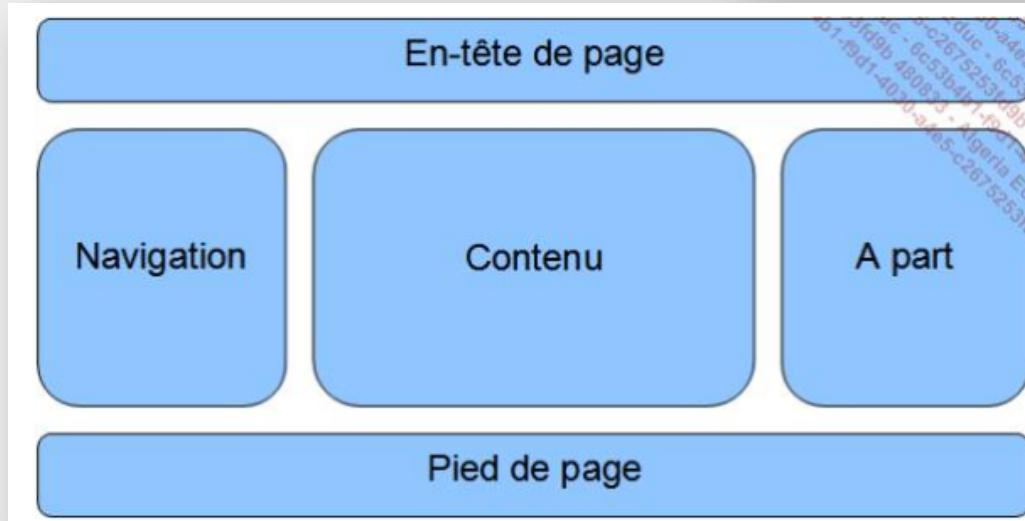
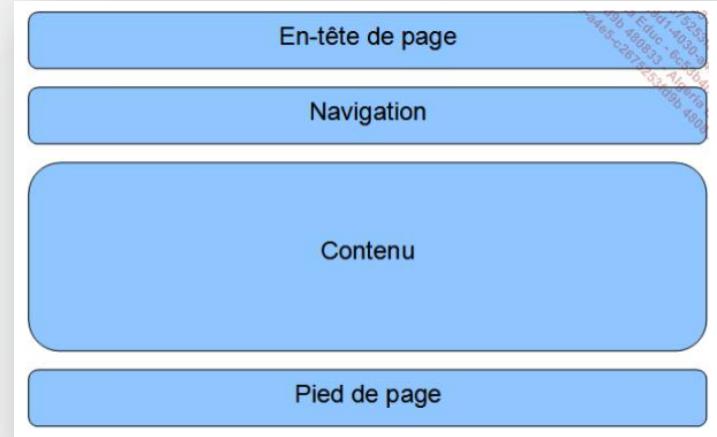
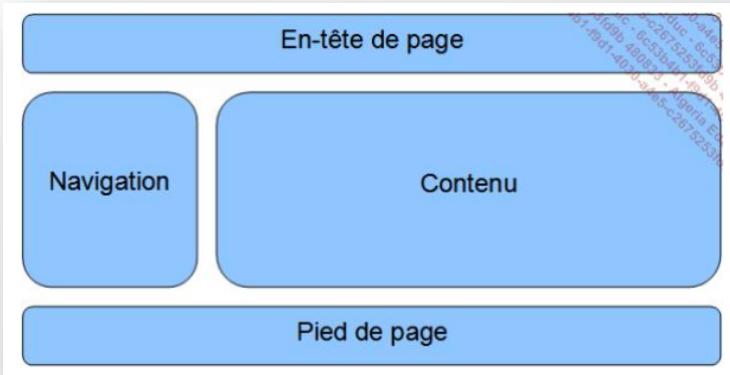


# Exemples de disposition



□ Il existe deux grands groupes de balises :

- **les balises de rendu CSS bloc (*block*)**
  - Généralement, le navigateur insère automatiquement un espace au dessus et en dessous du contenu de la balise
    - La balise `<div> ... </div>` spécialement conçue pour introduire une division. Cette balise est particulièrement utile pour disposer des éléments au sein d'une page.
    - On a également : hx, p, hr, ol, ul, dl, table, form
- **les balises de rendu CSS en ligne (*inline*)**
  - les éléments en ligne se placent toujours l'un à côté de l'autre afin de rester dans le flux du texte.
    - La balise `<span>` pour introduire une division en ligne dans le texte.
    - On a également : a, img, i, b, input, select, textarea

## Les Balises sémantiques

- Si l'on effectue une synthèse des pages existantes sur la toile, on peut constater qu'elles comportent la plupart du temps, en totalité ou en partie, les éléments suivants :
  - Un entête de page avec, par exemple, un logo, une bannière, le nom du site, un slogan ou un champ de recherche.
  - Les outils de navigation, précieux et indispensables pour la consultation des différentes parties du site.
  - Une partie consacrée au contenu.
  - Une zone annexe qui permet d'apporter des éléments accessoires au contenu proprement dit, comme une publicité.
  - Un pied de page avec par exemple la mention d'un copyright, le plan du site, les mentions légales, les règles d'accessibilité, etc.

# Feuilles de Style CSS

Cascading Style Sheets

## Concept des Feuilles de Style

- Le concept de feuilles de style repose sur le principe de la séparation du contenu et de la présentation dans l'élaboration d'applications Html.
- Ainsi un même contenu pourrait être utilisé, selon la feuille de style adoptée, pour un affichage sur des médias aussi divers qu'un écran traditionnel, l'écran d'un ordinateur de poche ou d'un mobile, des feuilles de papier imprimées, une barrette braille, etc.
- C'est ainsi tout le domaine de la présentation qui est pris en charge par les feuilles de style, le rôle du Html se limitant alors à la structure et à l'encodage de l'information brute.

## Bref Historique

- **1991** : Tim Berners-Lee inventa le Web, le langage Html était principalement conçu pour faciliter l'échange d'informations entre scientifiques.
- **1994** : Ajoute au langage HTML, de la mise en forme.
- **1996** : le W3C fait du CSS1, une recommandation officielle.
- **1998** : Sortie de CSS2
- **2006** : Sortie de CSS2.1
- **2009** : Pas de Sortie Officielle de CSS3, en parallèle certains modules sont rajoutés :
  - *CSS Color Module Level 3*
  - *CSS Text Level 3*
  - *CSS Backgrounds and Borders Level 3*
- Les différentes marques de navigateurs portent un intérêt évident pour ces nouvelles CSS3 et les intègrent rapidement.
- Parfois, celles-ci sont même implantées de façon provisoire en attendant la spécification définitive.

# Utilité des Feuilles de Style

- Un passage obligé en Html5
  - Le Html5 va au bout du concept de la séparation du contenu et de la présentation. Presque tous les éléments de présentation des balises ou des attributs présents dans le Html 4.0 ont ainsi disparu.
- Une simplification du code
  - Les feuilles de style ont permis d'alléger considérablement le code source en le rendant plus lisible et plus accessible.
- Une maintenance de site facilitée
  - Centralisation au sein d'un fichier CSS
- Une voie vers l'accessibilité
  - Depuis quelques années, l'accent a été mis sur l'accessibilité des sites Web pour les personnes présentant des déficiences visuelles.

# Compatibilité des Navigateurs

- CSS1 et CSS2 sont bien reprises par les navigateurs présents sur le marché, même s'il subsiste quelques bémols et bugs dans les navigateurs plus anciens
- La situation est loin d'être la même pour les CSS3. Il importe dès à présent d'insister sur le côté nouveau voire expérimental de l'incorporation de ces feuilles de style CSS3 dans les navigateurs modernes.
- Certaines propriétés nécessitent si on souhaite les utiliser sur certaines navigateurs l'utilisation d'un préfixe
  - -moz
  - -webkit

# CSS : Règles syntaxiques

- La déclaration d'un style s'effectue par le binôme
  - propriété: valeur;
    - Exemple : background-color: red;
- La propriété identifie ce qui sera défini dans le style adopté. Ces propriétés sont énumérées dans les spécifications CSS. Il existe de nombreuses propriétés portant, par exemple, sur la police de caractère (*font*), sur le texte (*text*), sur l'arrièreplan (*background*), sur la bordure (*border*), etc.
- La propriété est séparée de sa valeur par un double point.
- Les espaces sont permis. Ainsi certains auteurs ont pris l'habitude de mettre un espace entre le double point et la valeur pour une meilleure lisibilité du code.
- La valeur identifie la nature de l'effet de style souhaité. La valeur s'exprime par un mot clé, un pourcentage, une grandeur en fonction de la propriété à laquelle elle est assignée.
- Une déclaration de style se termine toujours par un point-virgule.
- Il est possible de définir plusieurs déclarations de style pour un sélecteur.
- Exemple :

propriété1:valeur; propriété2:valeur; propriété3:valeur;

# Incorporation de Styles

- Style en ligne :
- La déclaration de style est ajoutée directement à une balise donnée dans le code Html par l'attribut style.

```
<h1 style="text-align: center;"> ... </h1>
```

*Cette façon basique ne respecte pas la règle de la séparation du contenu de la présentation. Son usage doit donc rester exceptionnel.*

- Style interne :
- Les différentes déclarations de style sont regroupées dans l'entête du document soit entre les balises <head> ...</head>.

```
<style type="text/css">
```

**Déclarations de style**

```
</style>
```

□ Intérêts :

- Respect de la séparation du contenu et de la présentation
- Possibilité de l'appliquer à plusieurs fichiers HTML
- Maintenance facilitée

□ Ajouter dans le head :

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

- **link** indique au navigateur que ce qui suit est un lien.
- **rel="stylesheet"** précise que ce lien est relatif à une feuille de style.
- **href="exemple.css"** est l'écriture classique d'un lien en Html.
- Le lien peut être absolu (commençant par <http://...>) ou relatif.
- Rien n'empêche de mettre plusieurs balises `<link>` vers des feuilles de style externes différentes.
- Rien n'empêche non plus d'utiliser également une feuille de style interne.

□ Sélecteur de balise :

sélecteur { déclaration de style }

```
h1 { color: red; }  
h1,h3,h5 {font-style:italic;}
```

□ Sélecteur de classe :

.nom\_de\_la\_classe { déclaration(s) de style }

```
.rouge {background-color: red;}
```

- Utilisation en HTML :

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

.nom\_balise.nom\_classe { déclaration(s) de style }

```
body.rouge {background-color: red;}
```

## □ Les sélecteurs d'identifiant

Le sélecteur id permet donc d'identifier un élément unique dans la page. Cette distinction prend toute son importance quand on fera appel à du JavaScript ou du DHTML.

#nom\_de\_l'identifiant {déclaration(s) de style}

#rouge { background-color: red; }

- Utilisation en HTML :

```
<balise id="nom_de_l'identifiant">  
<div id="rouge"> ... </div>
```

nom\_balise#nom\_de\_l'identifiant {déclaration(s) de style}

div#entete {font-style:italic;}

# Les Unités de Mesure

Les valeurs absolues sont :

<b>Unité</b>	<b>Nom</b>	<b>Description</b>	<b>Valeur</b>	<b>Exemple</b>
pt	point	72 pt = 1 inch	entier	48pt
pc	pica	1 pc = 12 pt	réel	4.5pc
mm	millimètre	1 mm = .24 pc	entier	60mm
cm	centimètre	1 cm = 10 mm	entier	6cm
in	inch	1 in = 2.54 cm	réel	0.1in

Les valeurs relatives sont :

<b>Unité</b>	<b>Description</b>	<b>Valeur</b>	<b>Exemple</b>
em	Unité relative se basant sur la taille de police par défaut de la page.	réel	1.8em
ex	Unité relative à la hauteur de la minuscule de l'élément sélectionné.	réel	1.3ex
px	Le pixel est la plus petite partie d'une image. Dépend de la résolution d'écran.	entier	220px
%	Pourcentage	entier	80%

# Notation des Couleurs

- Les feuilles de style CSS proposent de multiples notations pour déclarer une couleur.
  - en Hexadécimal : #rrggbb
  - rgb(0, 0, 255)
  - rgb(25%,50%,0%)
    - La valeur 0% signifie l'absence de la composante, 100% qu'elle est à son maximum.
  - color : red
    - Elles sont au nombre de 17. green (vert), yellow (jaune), blue (bleu moyen), orange (orange), white (blanc), red (rouge), black (noir), maroon (brun marron), lime (vert citron), aqua (turquoise), teal (cyan foncé), navy (bleu marine), olive (olive), fuchsia (fuchsia), purple (pourpre), silver (gris clair) et gray ou grey (gris foncé).
  - RGBA :
    - La dernière valeur indiquant le degré d'opacité ou de transparence entre 0 et 1.

- font-family : nom de la police ou famille de police
  - Si l'on indique plusieurs polices, le nom de celles ci doit être séparé par une virgule.
  - Le résultat peut varier d'un ordinateur à l'autre en fonction des polices installées
  - Au lieu de spécifier le nom de la police, on peut utiliser des familles de police ou des noms de polices génériques :
    - **serif**. Les polices de type serif ont des terminaisons au bout de leurs traits et un espacement des lettres proportionnel (Times New Roman, Bodoni, Georgia, ...).
    - **sans-serif**. Les polices sans serif ont les bouts des traits pleins, sans évasement et un espacement des lettres proportionnel (Arial, Verdana, Helvetica, Trebuchet...).
    - **cursive**. Famille de polices dont le résultat est proche d'une écriture manuscrite (Script, Adobe Poetica...).
    - **fantasy**. Les polices fantaisies ou décoratives (Wingdings...).
    - **monospace**. Toutes les lettres ont les mêmes dimensions.

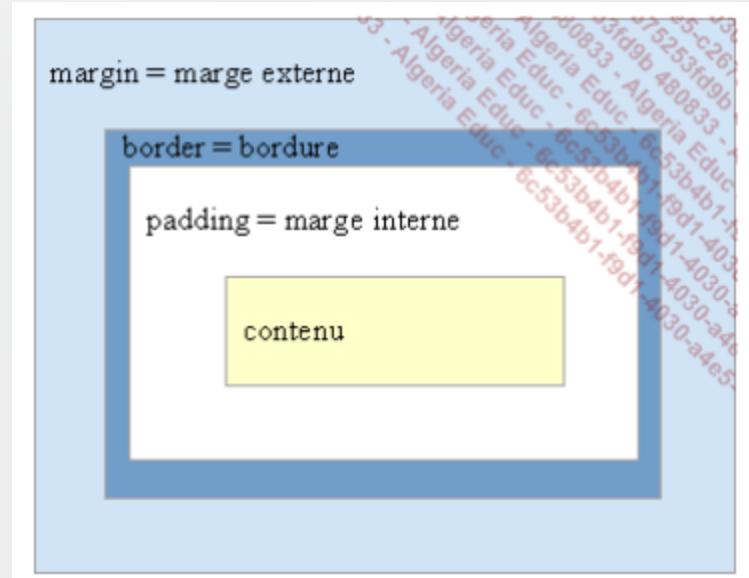
- font-style
- font-variant
- font-weight
- font-size
- font: italic bold small-caps 24pt  
          Arial, sans-serif;
- Cette seule déclaration correspond  
à :
- color
- text-decoration
- text-transform
- letter-spacing
- word-spacing
- line-height
- white-space
- text-align
- vertical-align
- 

```
font-style: italic;  
font-weight: bold;  
font-variant: small-caps;  
font-size: 24pt;  
font-family: Arial, sans-serif;
```

- width|height
- Puces
  - list-style-type
  - list-style-image: url(puce.gif);
  - list-style-position
  - Ecriture Raccourcie :  
list-style: url(puce.gif) circle inside;
- Tableaux
  - border-spacing
  - border-collapse
  - empty-cells
  - caption-side
- Arrière-plan
  - background-color
  - background-image
  - background-repeat
  - background-position
  - background-attachment
  - Ecriture Raccourcie :  
background: url(format\_image) white  
no-repeat right top fixed;

# Concept du modèle de boîte

- Le W3C définit un élément boîte comme une zone rectangulaire constituée :
  - d'un contenu,
  - d'une marge intérieure (*padding*),
  - d'une bordure (*border*),
  - d'une marge extérieure (*margin*).
- La dimension totale de l'élément boîte se calcule donc ainsi :  
contenu + marges intérieures + bordures + marges extérieures.



# Élément bloc ou en ligne

- display : block|inline
- width : auto|valeur|%
- height : auto|valeur|%
  - border-right-color
  - border-bottom-color
  - border-left-color
- Marges
  - margin
  - margin-top
  - margin-bottom
  - margin-right
  - margin-left
- Marges Internes
  - padding
  - padding-top
  - padding-bottom
  - padding-right
  - padding-left
- Bordures
  - border-color
  - border-top-color
- Largeur de Bordures
  - border-width
  - border-top-width
  - border-right-width
  - border-bottom-width
  - border-left-width
- Style de Bordures
  - border-style
  - border-top-style
  - border-right-style
  - border-bottom-style
  - border-left-style
- Ecriture Raccourcie :  
**border: red double 5px;**

# Pseudo-Classes

## □ Syntaxes

sélecteur:pseudo-classe  
{ déclaration(s) de style }  
sélecteur.classe:pseudo-classe  
{ déclaration(s) de style }

## □ Insérer un contenu

- :before
- :after
- 

## □ Pseudo-Classes de Liens

- a:link
- a:visited
- a:hover
- a:active

## □ Pseudo-Classes de Paragraphes

- :first-letter
- first-line

# Modes de Positionnement

- Statique : Positionnement normal de l'élément selon la façon habituelle de procéder du navigateur

`position:static;`

- Relatif : C'est le positionnement d'un élément par rapport à sa position normale ou statique

`position: relative;  
left: valeur ou %;  
top: valeur ou %;  
right: valeur ou %;  
bottom: valeur ou %;`

- Absolu : Le positionnement absolu crée un élément indépendant du reste du document. Les éléments définis en position absolue sont retirés du flux normal.

`position: relative;  
left: valeur ou %;  
top: valeur ou %;  
right: valeur ou %;  
bottom: valeur ou %;`

- Fixe : crée aussi un élément indépendant dont on peut définir la position. Ici l'élément reste fixe même lorsque l'on fait défiler le document.

■ `position: fixed;`

# Propriétés d'affichage

## □ Flottement

- float : retire un élément boîte du flux normal pour la placer le plus à droite ou le plus gauche possible dans son élément parent, soit son conteneur.
  - float : left|right|none

## □ Dégagement

- clear : permet d'annuler le flottement introduit par la propriété float.
  - clear : right|left|both|none

## □ Superposition

- z-index ajoute un axe en profondeur permettant de positionner des éléments au-dessus ou en dessous d'un autre élément.

## □ Dépassemment

- overflow détermine ce que le navigateur doit faire lorsqu'un élément est plus grand que l'élément parent qui le contient.
  - overflow: hidden|scroll|visible|auto;

# Propriétés d'affichage (suite)

- Découpage
  - **clip** : détermine la partie visible de l'élément, généralement une image. Ainsi, l'image entière sera bien présente dans le document mais seulement une partie (rectangulaire) de celle-ci sera visible.
    - clip: rect(sup\_gauche sup\_droit inf\_droit inf\_gauche);
    - clip: rect(10px 270px 190px 50px);
- Affichage
  - **display** : permet de contrôler l'affichage des éléments dans la page
    - display: block|inline|none;
- Visibilité
  - **visibility** : détermine si un élément est visible ou caché
    - visibility: visible|hidden;
- Curseur de la Souris
  - cursor: pointer|move|wait|crosshair ....
- Numérotation automatique
  - **counter-reset** : crée ou réinitialise le compteur désigné.
  - **counter-increment** incrémente le compteur désigné.

**Steeve ASSOUS ©**

# **Feuilles de Style CSS3**

- Les feuilles de style permettent des représentations différentes selon le media de sortie.
- Si nous sommes des habitués du Web à partir de l'écran, d'autres internautes le sont à partir d'un ordinateur de poche, d'un smartphone, d'une interface vocale, d'une barrette Braille ou d'un document imprimé.

media="all"	Les styles définis s'appliquent à tous les types de médias (valeur par défaut).
media="screen"	Les styles s'appliquent à l'affichage sur un écran classique.
media="print"	Les styles s'appliquent à l'impression sur papier. Les navigateurs utiliseront ces spécifications lorsque l'utilisateur demande d'imprimer la page Web.
media="projection"	Les styles s'appliquent à la projection de données avec des projecteurs.
media="handheld"	Les styles s'appliquent à l'affichage sur des ordinateurs de poche ou des smartphones permettant la navigation sur le Web.
media="aural"	Les styles s'appliquent à des systèmes de restitution vocale assistée par ordinateur.
media="braille"	Les styles s'appliquent à des barrettes Braille conçues pour les non-voyants.
media="embossed"	Les styles s'appliquent à des imprimantes à perforation Braille conçues pour les non-voyants.

## □ Hiérarchiques

A B	Descendants	Sélectionne tout élément B qui est descendant d'un élément A.
A > B	Enfants	Sélectionne tout élément B qui est enfant direct d'un élément A.
A + B	Frère adjacent	Sélectionne tout élément B immédiatement précédé d'un élément A.

# CSS3 : Pseudo-Classes de Sélection

:root	Représente l'élément racine ou l'élément le plus haut d'un document. Par exemple, en Html5, la balise <html>.
:empty	Correspond aux éléments vides et qui n'ont donc pas d'enfants.
:only-child	Reprend l'enfant unique. Celui-ci n'a donc pas d'élément frère.
:first-child	Sélectionne le premier élément enfant.
:last-child	Sélectionne le dernier élément enfant.
:nth-child(n)	Désigne le nième élément enfant où n est un nombre ou les mots-clés even (pairs) ou odd (impairs).
:nth-last-child(n)	Sélectionne le nième enfant en partant du dernier élément.
:only-of-type	Reprend l'unique élément de ce type.
:first-of-type	Représente le premier élément de ce type.
:last-of-type	Représente le dernier élément de ce type.
:nth-of-type(n)	Désigne le nième élément de ce type où n est un nombre ou les mots-clés even (pairs) ou odd (impairs).
:nth-last-of-type(n)	Sélectionne le nième élément de ce type en partant du dernier élément.

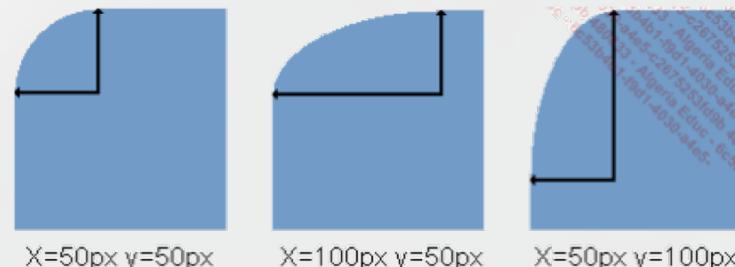
# CSS3 : Sélecteurs d'Attributs

[attr]	Désigne un élément qui comporte l'attribut attr indiqué.
[attr="valeur"]	Désigne un élément qui comporte l'attribut attr fourni par la valeur indiquée.
[attr~="valeur"]	Correspond à tout élément dont l'attribut attr contient une liste de valeurs séparées par des espaces et dont l'une d'elles est valeur.
[attr = "valeur"]	Correspond à tout élément dont l'attribut attr contient une liste de valeurs séparées par des tirets débutant par valeur.
[attr^="valeur"]	Représente un élément dont l'attribut attr commence avec le préfixe fourni par la valeur indiquée.
[attr\$="valeur"]	Représente un élément dont l'attribut attr se termine avec le suffixe fourni par la valeur indiquée.
[attr*="valeur"]	Représente un élément dont l'attribut attr contient une instance de la valeur indiquée.

# Coins arrondis

- ❑ Avec le CSS3, ces bords arrondis sont réalisés directement avec quelques lignes de propriétés de style.

```
border-radius : x, y ;  
border-top-right-radius : x,y  
border-bottom-right-radius : x,y  
border-bottom-left-radius : x,y  
border-top-left-radius : x,y
```



- ❑ Firefox :  
-moz-border-radius
- ❑ Safari / Chrome :  
-webkit-border-radius

```
#round1 { width: 70px; height: 70px;  
background-color: rgb(185,205,225);  
border: 1px solid gray;  
border-radius: 35px;  
-moz-border-radius: 35px;  
-webkit-border-radius: 35px; }
```

□ Au niveau du texte :

**text-shadow: x y z couleur**

x est le déport de l'ombre vers la droite.

y est le déport de l'ombre vers le bas.

z est l'intensité du dégradé ou du flou (facultatif, par défaut 0).

couleur est la couleur de l'ombre.

□ Au niveau des boîtes :

**box-shadow x y z couleur**

x est le déport de l'ombre vers la droite.

y est le déport de l'ombre vers le bas.

z est l'intensité du dégradé ou du flou (facultatif, par défaut 0).

couleur est la couleur de l'ombre.

# Polices Personnalisées

- Jusqu'à présent, au sein des pages Web, les polices dépendaient de celles installées sur l'ordinateur de l'internaute.
- Le CSS3 permettent d'importer et d'utiliser une police personnalisée même si celle-ci n'est pas installée chez l'utilisateur.
- Il faut d'abord télécharger la police de caractères pour l'incorporer dans la page.

```
@font-face { font-family: "nom_police";  
src: url("nom_police.extension_police");}
```

Exemple :

```
@font-face { font-family: "mapolice";  
src: url('fantaisie.ttf');}
```

```
.monTitre { font: 100px mapolice, fantasy;}
```

# Polices Personnalisées : Alternative 2

## □ Google Font Api - <https://fonts.googleapis.com/>

Les polices proposées par ce service sont parfaitement adaptées à l'affichage dans un navigateur et d'un poids réduit.

Il suffit simplement d'ajouter une feuille de style à votre page Html afin d'obtenir la nouvelle police.

Illustration :

```
<meta charset="utf-8">
  <link href="https://fonts.googleapis.com/css?family=Shadows+Into+Light" rel="stylesheet">
<style>
  h1 { font-family: 'Shadows Into Light', cursive; }
</style>

<body>
<h1>Making the Web Beautiful !</h1>
```

Making the Web Beautiful !

# Arrière-plans Multiples

- Le CSS3 rendent possible l'affichage de plusieurs images dans un même arrière-plan en permettant de cumuler les valeurs au sein des propriétés background-image, background-position et background-repeat.

```
#box
{
    width:300px; height: 200px;
    border: 1px solid gray;
    background: url(coinSG.png) top left no-repeat,
                url(coinSD.png) top right no-repeat,
                url(coinID.png) bottom right no-repeat,
                url(coinIG.png) bottom left no-repeat;
}
```

## L'opacité ou la transparence

- Avec les CSS3, il est possible de faire varier l'opacité ou la transparence d'un élément.
- Cette propriété a été adoptée très tôt par l'ensemble des navigateurs.

opacity: Valeur comprise entre 1 et 0. Avec la valeur 1, l'opacité est complète et la transparence nulle. Avec 0, l'opacité est nulle et l'élément complètement transparent

# Colonnes Multiples

- ❑ Permet mettre en place plusieurs colonnes :

**column-count:** Entier qui détermine le nombre de colonnes dans lesquelles sera affiché le contenu de l'élément.

**column-width:** Valeur qui décrit la largeur optimale de chaque colonne (facultatif).

**column-gap:** Valeur qui détermine l'espace ou le padding entre les colonnes.

**column-rule:** Définit une bordure entre les colonnes.

```
#box
{
    width:420px;
    height: 240px;
    text-align: justify;
    column-count: 2;
    column-gap: 2em;
    column-rule: 1px solid silver;
    -moz-column-count: 2;
    -moz-column-gap: 2em;
    -moz-column-rule: 1px solid silver;
    -webkit-column-count: 2;
    -webkit-column-gap: 2em;
    -webkit-column-rule: 1px solid silver;
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud

exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

# **Exercice Gabarit (div)**

**Logo**

**ACCUEIL**

**Sujet de la page**

Accueil  
Item 1  
Item 2  
Item 3  
Contact

Article 1

Article 2

Article 3

Archives

- Août 201x
- Juillet 201x
- Juin 201x
- Mai 201x

[Accueil](#) [Plan](#) [Contact](#) [CGU](#)

Gabarit - Copyright © 201x

# Nouvelles Balises sémantiques

- ❑ Ces balises d'organisation sont plutôt destinées au concepteur de la page afin qu'il puisse distinguer plus aisément des portions de code.
  - header
  - nav
  - main
  - aside
  - footer
- ❑ Le contenu principal peut être organisé autour de 2 balises
  - section
    - détermine une partie du contenu de la page se rapportant à un thème déterminé.
  - article
    - définit un contenu indépendant du document qui possède une identité à part entière dans la page comme l'article d'un blog, un post dans un forum ou un produit dans un site commercial

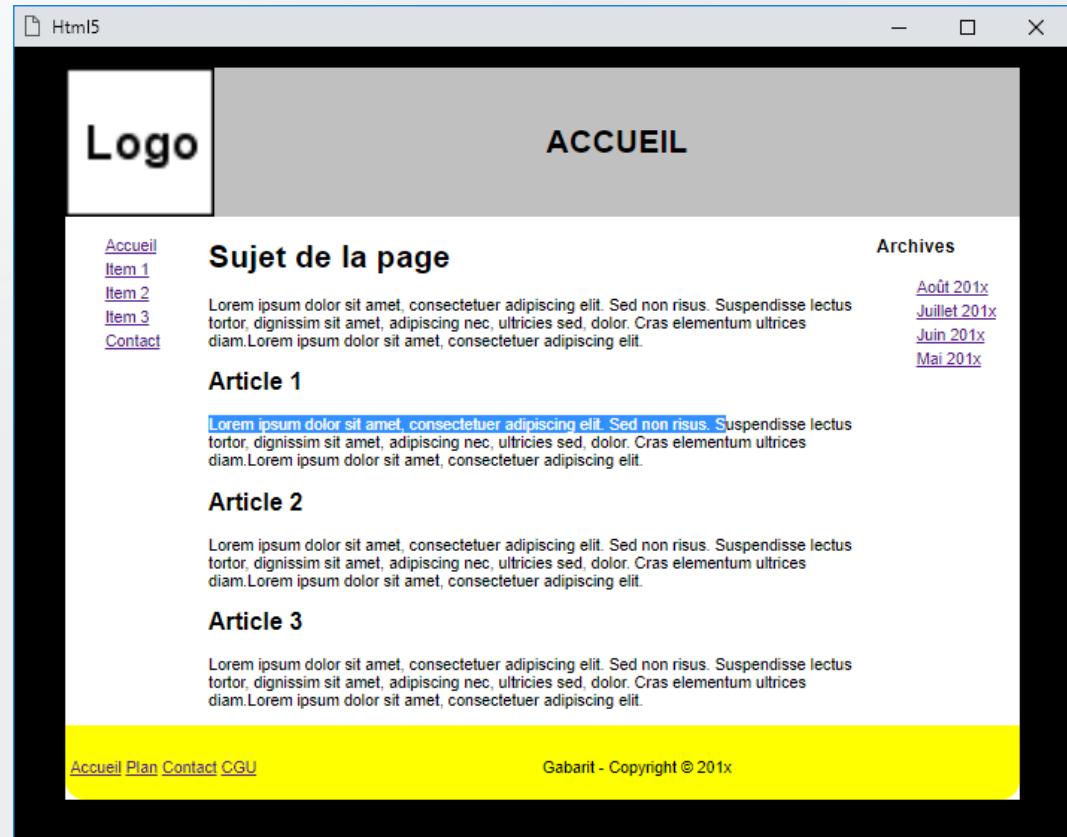
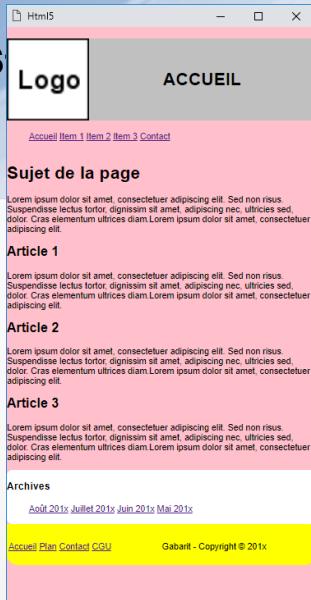
# Exercice Gabarit (Balises Sémantiques)

The screenshot displays a website template with the following structure:

- Header:** A white box containing the word "Logo". To its right, the word "ACCUEIL" is centered.
- Left Sidebar:** A vertical column with a light gray background. It contains a navigation menu:
  - [Accueil](#)
  - [Item 1](#)
  - [Item 2](#)
  - [Item 3](#)
  - [Contact](#)
- Main Content Area:** A large central area with a light gray background.
  - Sujet de la page:** A section title.
  - Article 1:** A heading followed by a paragraph of placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Lorem ipsum dolor sit amet, consectetur adipiscing elit."
  - Article 2:** A heading followed by a paragraph of placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Lorem ipsum dolor sit amet, consectetur adipiscing elit."
  - Article 3:** A heading followed by a paragraph of placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Lorem ipsum dolor sit amet, consectetur adipiscing elit."
- Right Sidebar:** A vertical column with a light gray background titled "Archives". It lists four dates:
  - [Août 201x](#)
  - [Juillet 201x](#)
  - [Juin 201x](#)
  - [Mai 201x](#)
- Footer:** A yellow horizontal bar at the bottom containing the text "Accueil Plan Contact CGU" on the left and "Gabarit - Copyright © 201x" on the right.

# Media Queries

S



- De nos jours, les individus consultent les sites sur des supports de plus en plus diverses.
- Par conséquent, il convient de s'assurer que le site consulté soit simple à utiliser et esthétique quelque soit le support utilisé en respectant les exigences du canal utilisé.
- Il s'agit aujourd'hui non plus de concevoir un simple site mais de réfléchir dès sa conception à le rendre utilisable sur un quelconque support. (Smartphones, Tablettes, Ordinateurs, Téléviseurs quelque soit la taille de l'écran).
- Ainsi, votre site sera dit : "**RESPONSIVE**"
- Afin de rendre cela possible, une alternative serait l'utilisation des : "**Media Queries**"

- Les Media Queries nous permettent de créer des règles CSS basées sur différents paramètres (taille, type de support, ...) afin d'y disposer convenablement nos éléments.

	Petit écran (smartphone)	Ecran réduit (tablette)	Ecran moyen (desktop)	Grand écran (desktop)
Conteneur maximum	Automatique	750 px	970 px	1170 px
Valeur de référence	< 768 px	= 768 px	= 992 px	= 1200 px

- Exemples :
  - @media TV and (min-width: 1200px) {...}
  - @media screen and (max-width: 800px) {...}

# Utilisation des Media Queries

```
/**  
 * GESTION MEDIA QUERY Start  
 */  
@media only screen and (max-width:1200px) {  
    body {  
        background-color:black;  
    }  
    #global{  
        width: 960px;  
        background-color:white;  
  
    }  
}  
@media only screen and (max-width:990px) {  
    body {  
        background-color:green;  
    }  
    #global{  
        width: 768px;  
        background-color:green;  
    }  
    aside, #menuD, #centre  
    {  
        width: 100%;  
    }  
}  
}
```

```
/* GESTION MEDIA QUERY Start */  
@media only screen and (max-width:768px) {  
    body {  
        background-color:pink;  
    }  
    #global{  
        width: 100%;  
        background-color:pink;  
    }  
  
    aside, #menuD, #centre  
    {  
        width: 100%;  
    }  
  
    aside ul li  
    {  
        display : inline-block;  
    }  
  
}  
/* GESTION MEDIA QUERY End */
```

- Il s'agit d'un jeu de règles CSS intégré dans un package, qui peut aider à résoudre certaines problématiques en terme de disposition des éléments.
- Permet de faciliter les réalisations au quotidien
- Lorsque nous utilisons un framework, nous adaptons le code du framework en vue de créer ses propres fonctionnalités
- Ces Frameworks nous aident à mettre en forme les pages web : organisation, aspect, animation... Ils sont devenus à la mode et il en existe un certain nombre :
- Quelques exemples de Frameworks CSS :
  - Skeleton
  - Foundation
  - Bootstrap

# **Bootstrap : Framework CSS**

# Avantages / Inconvénients

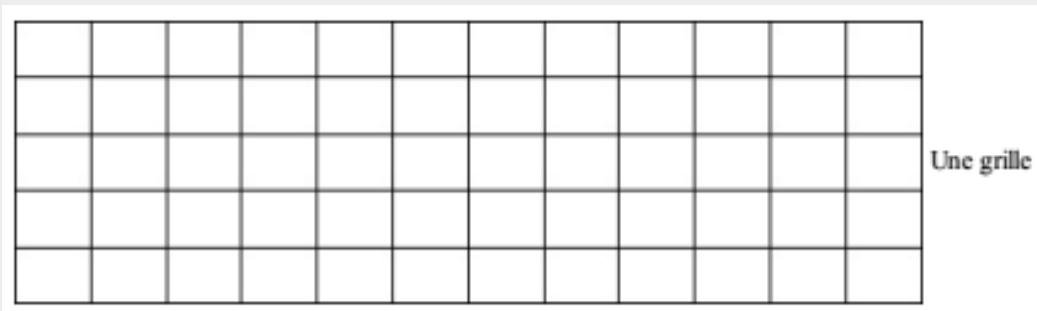
## □ AVANTAGES

- Les frameworks sont cross-browser, c'est à dire que la présentation est similaire quel que soit le navigateur utilisé et d'une parfaite compatibilité ;
- Les frameworks CSS font gagner du temps de développement parce qu'ils nous proposent les fondations de la présentation ;
- Les frameworks CSS normalisent la présentation en proposant un ensemble homogène de styles ;
- Les frameworks CSS proposent en général une grille pour faciliter le positionnement des éléments ;
- Les frameworks CSS offrent souvent des éléments complémentaires : boutons esthétiques, barres de navigation, etc. ;
- La grande diffusion de nouveaux moyens de visualisation du web (smartphones, tablettes...) impose désormais la prise en compte de tailles d'écran très variées ;

## □ INCONVÉNIENTS

- Pour utiliser efficacement un framework il faut bien le connaître, ce qui implique un temps d'apprentissage ;
- La normalisation de la présentation peut devenir lassante en lissant les effets visuels.

- Ces Frameworks utilisent un système de grille qui réorganise automatiquement les éléments en vue de l'adapter aux différents supports.
- Le but est de diviser une page en colonnes dans lesquelles nous pouvons positionner nos éléments.
- Bootstrap, notamment, utilise un système de grille de 12 colonnes qui facilite la mise en œuvre de sites Web Responsives.



## Notion De Container

- La grille de Bootstrap doit être placée dans un conteneur. Bootstrap propose les classes container et container-fluid. Leur utilisation était auparavant optionnelle.
- Il est désormais clairement indiqué dans la documentation qu'il faut la mettre en œuvre systématiquement si on veut obtenir des alignements et des espacements corrects.
- La classe container contient et centre la grille sur une largeur fixe, qui s'adapte en fonction de la largeur de l'écran.

# Tableau des Ecrans

	Petit écran (smartphone)	Écran réduit (tablette)	Écran moyen (desktop)	Grand écran (desktop)
Comportement	Redimensionnement	Redimensionnement	Redimensionnement	Empilage puis redimensionnement
Classe	col-xs-*	col-sm-*	col-md-*	col-lg-*
Valeur de référence	< 768 px	> 768 px	> 992 px	> 1200 px

- La première classe à connaître est row, qui représente une rangée. Il faut ensuite définir le nombre de colonnes pour chaque élément en sachant qu'il y en a au maximum 12.
- Pour définir le nombre de colonnes utilisées pour chaque élément, on dispose de quatre batteries de 12 classes :
- - col-xs-1 ou col-sm-1 ou col-md-1 ou col-lg-1
  - col-xs-2 ou col-sm-2 ou col-md-2 ou col-lg-2
  - ...
  - col-xs-12 ou col-sm-12 ou col-md-12 ou col-lg-12

```
<head>
  <meta charset="UTF-8" />
  <title>Votre Titre</title>

  <!-- Latest compiled and minified CSS -->
  <link rel="stylesheet" href="./bootstrap-4.1.1/dist/css/bootstrap.min.css">

  <!-- Latest compiled and minified JavaScript -->
  <script src="./bootstrap-4.1.1/dist/js/bootstrap.min.js"></script>
  <style type="text/css">

    img { width :100%; }

    body
    {
      background: #0CF;
    }
    ...
  </style>
</head>
<body>
  <div class="container">
    <header class="row">
      <div class="col-xs-12 col-sm-6 col-lg-6">
        
      </div>
      <div class="col-xs-12 col-sm-6 col-lg-6">
        <h1>Accueil</h1>
      </div>
    </header>
    ...
  </div>
</body>
```

**Steeve ASSOUS ©**