Développement web

Caroline DEVRED

Université d'Angers

Plan

Introduction

En savoir plus

- Ce cours est basé sur «Réussir son site web avec XHTML et CSS » de Mathieu Nebra chez Eyrolles.
- Les balises XHTML sur : http ://www.siteduzero.com/tuto-3-4874-1-liste-des-balises-xhtml.html
- Les propriétés de CSS sur : http ://www.siteduzero.com/tuto-3-1938-1-liste-des-proprietes-css.html
- Les transparents du cours : http ://www.info.univ-angers.fr/pub/devred/coursDV.pdf

Langage

- Différence entre ce que l'on voit (le site web) et ce que le webmaster à écrit (le code source).
- Il faut utiliser un langage compréhensible par la machine.
- En fait plusieurs langages.
- Dans ce cours, nous allons séparer : le fond du site et la forme du site. Nous commencerons donc par deux langages en parallèle.

XHTML et CSS

- Le fond :
  - XHTML (eXtensible Hyper Text Markup Langage).
  - Langage de base du web.
- La forme :
  - CSS (Cascade Style Sheets).
  - Rôle «décorer » le site web. Mise en page, police, taille du texte, couleur etc.
  - Un petit tour sur http ://csszengarden.com/

Un site web

- Sert à communiquer.
- Page «intra » (pour une communauté)
- Page «inter » (vendre, faire de la pub, se faire connaître)
- D'où l'importance du visuel (information facile à trouver etc.).
- Quel est le but d'un site web ?
- Il faut choisir ses balises en fonction du but du site.

Le navigateur

- Différence entre ce que l'on voit (le site web) et ce que le webmaster à écrit (le code source).
- Pour voir le site web. Pour surfer sur le net.
- Il va transformer le code source en document plus lisible pour l'être humain.
- C'est comme un traducteur. On dit que c'est un interpréteur.
- Le HIC. Ils ne traduisent pas forcement tous pareil...
- Il faut donc tester son site sur plusieurs navigateurs.
- Exemple de navigateurs : Internet Explorer, Mozilla Firefox, Opera, Safari etc.

Et pour créer ?

- 2 catégories :
  - Les WYSIWYG.
  - Les éditeurs de texte .

Les éditeurs de texte

- Traitement de texte $\neq$ éditeur de texte.
- Traitement de texte conçus pour mettre en forme un texte.
- Éditeur de texte va permettre de travailler le fond.
- Exemple d'éditeurs de texte : NotePad++, JEdit, Quanta etc.

Le début de XHTML

Les pages web

- Une page web peut contenir : du texte, des images, des tableaux, des formulaires, etc.
- Le code source est basé sur des éléments repérés par des balises.

Les balises

- Les balises sont des informations spéciales destinées au navigateur, elles n'apparaissent pas à l'écran (sauf lorsqu'on affiche le code source).
- En XHTML, les balises s'écrivent en minuscules , sans espace, ni accent : <mabalise >
- Les balises ont un sens (titre, paragraphe etc.).
- On choisit les balises en fonction de leur sens , on pourra en modifier la forme avec CSS.

Les balises (suite)

- Les balises paires.
  - <mabalise > mon texte </ma balise >
  - Elles délimitent un élément, une partie du texte.
- Les balises autofermantes (ou monoatomiques).
  - <mabalise/ >
  - Elle serve à inserer une information à un endroit précis.
- Elles servent à structurer le document.
- Elles introduisent une linéarité.

Les attributs

- Ils apportent un complément d'information à la balise.
- En XHTML, les attributs s'écrivent en minuscules , sans espace, ni accent.
- Directement suivis de = et de guillemets.
- Les guillemets encadrent la valeur de l'attribut (pas de rêgle pour la valeur).
- <mabalise monattribut="valeur">

Ma première page

- On enregistre sa page sous l'extension .html (ou .htm). maPage.html
- D'autres extensions existent.

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>mon titre</title>
</head>

<body>
Ma page.
<!-- Mon commentaire -->
</body>
</html>
```

Le paragraphe

- C’est la balise de base.
- Mon paragraphe
- Sert à délimiter un paragraphe.
- Se met, comme tout ce que nous allons voir par la suite, dans <body > </body>

Le paragraphe (suite)

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Sur les paragraphes</title>
</head>

<body>
<p> Bonjour !</p>
<p> Mes premiers paragraphes</p>
<!-- Mon commentaire -->
</body>
</html>
```

Essayons d'aller à la ligne

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Sur les paragraphes</title>
</head>

<body>
<p> Bonjour !</p>
<p> Ma premi&egrave;re ligne.
Ma seconde ligne.
</p>
<!-- Mon commentaire -->
</body>
</html>
```

Aller à la ligne

- Un saut de ligne :
- Se met à l'interieur d'un paragraphe, sinon la page est invalide (on dit encore malformée).

Aller à la ligne

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Aller &agrave; la ligne (le retour)</title>
</head>

<body>
<p> Bonjour !</p>
<p> Ma premi&egrave;re ligne.<br/>
Ma seconde ligne.
</p>
<!-- Mon commentaire -->
</body>
</html>
```

Une parenthèse sur les caractères spéciaux

- é : &eacute ;
- É : &Eacute ;
- è : &egrave ;
- È : &Egrave ;
- à : &agrave ;
- À : &Agrave ;
- ê : &ecirc ;
- ë : &euml ;
- ç : &ccedil ;
- < : &lt ;
- > : &gt ;
- & : &amp ;
- ” : &quot ;

Des titres dans la page

- À ne pas confondre avec le titre de la page.
- Jusqu'à 6 niveaux de titre.
- Le titre se place à l'extérieur du paragraphe (sinon la page est invalide).
- On choisit son titre en fonction de son niveau et pas de la grosseur d'écriture. On peut changer celle-ci avec CSS.
- Les titres introduisent une hiérarchie.

Des titres dans la page (suite)

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Des titres dans la page</title>
</head>

<body>
<h1> Titre niveau 1</h1>
<h2> Titre niveau 2</h2>
<h3> Titre niveau 3</h3>
<h4> Titre niveau 4</h4>
<h5> Titre niveau 5</h5>
<h6> Titre niveau 6</h6>
<h1> Second titre de niveau 1</h1>
<h2> Seconde titre de niveau 2 </h2>
</body>
</html>
```

Mettre son texte en valeur

- Important : mon texte
- Très important : mon texte
- Se place à l'intérieur d'un paragraphe.
- Interprétation diffère suivant les navigateurs.

Mettre son texte en valeur

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Mettre du texte en valeur</title>
</head>

<body>
<p> Le <em>cours</em> est important, la <strong>pratique</strong> encore plus !</p>
</body>
</html>
```

Les citations courtes

- Juste quelques mots.
- À l'intérieur d'un paragraphe.
- <q >ma citation </q >

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Citation courte</title>
</head>

<body>
<p> <q>To be or not to be</q> une page invalide.</p>
</body>
</html>
```

Les citations longues

- Quelques paragraphes.
- À l'extérieur d'un paragraphe, contient des paragraphes.
- <blockquote >ma citation </blockquote >

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Citation longue</title>
</head>

<body>
<p> Extrait d'un po&egrave;me de Paul Verlaine : </p>
<blockquote>
<p>Les sanglots longs<br />
Des violons de l'automne<br />
Blessent mon coeur<br />
D'une langueur monotone.
</p>
 </blockquote>
</body>
</html>
```

Indice,exposant

- <sub >en indice </sub >
- <sup >en exposant </sup >

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Indice et exposant</title>
</head>

<body>
<p> l<sup>&egrave;re</sup> mol&eacute;cule : H<sub>2</sub>0.</p>
</body>
</html>
```

Abréviation et title

- <abbr title=”la signification de l’abréviation”>l’abréviation</abbr >

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Abr&eacute;viation</title>
</head>

<body>
<p> La mol&eacute;cule : <abbr title="l'eau">H<sub>2</sub>O</abbr>.</p>
</body>
</html>
```

Les liens

- Les liens hypertextes , une invention à la base du Web.
- Ils permettent de naviguer d'un endroit à un autre.
- Casse le déroulement linéaire.
- 2 types de liens :
  - Lien vers une autre page.
  - Lien vers un autre endroit de la page.
- Ils permettent aussi de déclancher des actions

Un lien vers une autre page de son site

- Balise <a >mon lien crée mon lien.
- Mais, où va-t'il ?
- Il faut rajouter l'attribut href="nom_page_cible.html"
- <a href="nom_page_cible.html">mon lien
- On peut mettre une info-bulle avec l'attribut title="info bulle"

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Lien vers une autre page de mon site</title>
</head>

<body>
<p> Cliquez <a href="005titre.html" title="mon premier lien"> ici </a> pour aller vers la page qui
pr&eacute;sente les titres.
</p>
</body>
</html>
```

Un lien pour envoyer un mail

- Balise <a href=”mailto :adresse_mail” >mon lien
- On peut même imposer le sujet : <a href=”mailto :adresse_mail ?subject=mon sujet” >mon lien

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Lien pour envoyer un mail</title>
</head>

<body>
<p> Cliquez <a href="mailto:devred@info.univ-angers.fr?subject= J'ai rien compris"> ici </a> pour
m'envoyer un mail. </p>
</body>
</html>
```

Un lien pour sauvegarder un fichier

- Il suffit de remplacer nom_page_cible.html par le nom (extension comprise) du fichier à télécharger (ou autre suivant la configuration).

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Lien pour sauvegarder un fichier</title>
</head>

<body>
<p> Cliquez <a href="lien1.eps"> ici </a> pour t&eacute;l&eacute;charger le fichier lien1.eps. </p>
</body>
</html>
```

Ancre, un lien particulier

- Lien vers un autre endroit de la page.
- On va créer un repère invisible dans la page.
- Ce repère s'appelle une ancre .
- Et on va créer un lien vers cette ancre.

Ancre (suite)

- Pour créer l'ancre, on ajoute l'attribut id="nom_de_l_ancre" dans à peu près n'importe quelle balise.
- Chaque nom d'ancre doit être unique.
- On crée un lien vers l'ancre nom_de_l_ancre :
  <a href="#nom_de_l_ancre">mon lien

Ancre (suite)

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Ancre</title>
</head>

<body>
<p> Cliquez <a href="#fin"> ici </a> pour aller &agrave; la fin de la page. </p>
<p> Une chanson de Cali (Comme j'&eacute;tais en vie -- L'espoir)</p>
<p>
J'&eacute;tais dingue de toi<br />
```

...

```
<p id="fin"> Au revoir ...</p>
</body>
</html>
```

Ancre dans une autre page

- Pour créer l'ancre, on ajoute l'attribut id="nom_de_l_ancre" dans à peu près n'importe quelle balise de la page cible (par exemple cible.html).
- On crée un lien vers l'ancre nom_de_l_ancre dans notre page :
  <a href="cible.html#nom_de_l_ancre">mon lien

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Ancre dans une autre page</title>
</head>

<body>
<p> Cliquer <a href="014ancre.html#fin"> ici</a> pour se retrouver &agrave; la fin de la chanson de
Cali.</p>
</body>
</html>
```

Liens relatifs

- Dépend de la page contenant le lien.
- Pour les pages internes du site.
- On indique le chemin depuis la page contenant le lien.

Liens relatifs (suite)

- Si mes pages sont dans le dossier page.
- <a href="t.html">signifie qu'on va chercher le fichier t dans page.
- Si page contient un dossier truc contenant le fichier toto.html, pour y accéder il faut taper : <a href="truc/toto.html">
- Si toto.html est contenu dans un dossier contenant page, on tape <a href="../toto.html">
- / même sous windows.
- Qu'est-ce qu'on structure, quel nom on leur donne (img, css etc.) ?

Liens absolus

- Il faut connaître l'adresse complète de la page cible.
- Pour les pages extérieurs à notre site.
- url : adresse du site, dossier et sous-dossiers. Peut avoir des paramêtres.

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Ancre dans une autre page</title>
</head>

<body>
<p> Cliquer <a href="http://www.google.fr"> ici</a> pour se retrouver sur google.</p>
</body>
</html>
```

Les images

- Plusieurs formats. Oubliez bmp, préférez jpg, jpeg, png, gif.
- Balise <img / >.
- 2 attributs obligatoires :
  - src la source de l'image (extension comprise) en chemin absolu ou relatif.
  - alt le texte alternatif à l'image en cas de non affichage. Il est de taille limitée et peut être vide : ""
-
- On peut mettre une info bulle avec l'attribut title.
-

Les images (suite)

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Image</title>
</head>

<body>
<p> Souvenir de week-end :<br />
<img src="img/img1.jpg" alt="Clocher de l'&eacute;glise du Viel Baug&eacute;" title="Torticolis ?"/>
</p>
</body>
</html>
```

Les images liens

- On peut faire d'une image un lien.
- Il suffit d'imbriquer la balise <a >et la balise <img/ >
- <a href="mon_lien">

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Image lien</title>
</head>

<body>
<p> Cliquez
<a href="http://www.univ-angers.fr">
<img src="img/logo.jpg" alt="logo de l'Universit&eacute; d'Angers"/>
</a>
</p>
</body>
</html>
```

Introduction à CSS

CSS

- CSS accés aux propriétés des éléments (couleur, police etc.).
- Propriété plus ou moins accessible par leur attribut.
- Propriété accessible par CSS via leur nom et leur valeur (valeur fixe ou qu'on paramêtre).

CSS (suite)

- Pour faire des sites *haut en couleurs*.
- La meilleur solution, mettre le code CSS dans un fichier à part. Par exemple : mon_style.css
- Pour indiquer qu'on utilise un style CSS dans une page web, dans le fichier html entre les balise <head > et </head >on indique :
<link rel="stylesheet" type="text/css" title="Mon style" href="mon_style.css"/>

Notre première page de test

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <link rel="stylesheet" type="text/css" media="screen" title="Mon style"
href="css/style1.css"/>
        <title>Test CSS 1</title>
</head>

<body>
<h1> Bienvenue sur la page test du CSS</h1>
<h2> Pr&eacute;sentation</h2>
<p> Nous allons tester nos premiers fichiers CSS sur cette page.</p>

<p> Pour ce faire un petit paragraphe et un petit blockquote qui donne une citation de Paul
Verlaine.</p>
<blockquote>
<p>Les sanglots longs<br />
Des violons de l'automne<br />
Blessent mon coeur<br />
D'une langueur monotone.
</p>
</blockquote>
</body>
</html>
```

Notre premier fichier .CSS

- /* Ceci est un commentaire */
  * /* concerne toutes les balises */
  {
  text-align :center ; /* propriétés du CSS, ici il n'y en a qu'une. Elles s'écrivent toujours propriété :valeur ; */
  }
- Centre TOUT.

Un style sur un type d'élement

- À la place de *, on met le nom de la balise définisant l'élément.
- Par exemple, sur les titres de niveau 1.
- h1
  {
  text-align :center ;
  }

Un style sur plusieurs types d'élément

- À la place de *, on met le nom de les noms des balises, sépararés par des virgules .
- Par exemple, sur les titres de niveau 1 et les titres de niveau 2.
- h1,h2
  {
  text-align :center ;
  }

Un style sur des éléments imbriqués

- À la place de *, on met le nom des balises, de la plus externe jusqu'à la balise imbriquée souhaitée.
- On sépare les balises par des espaces .
- Par exemple, sur les paragraphes contenus dans une citation.
- blockquote p
  {
  text-align :center ;
  }

Un style sur un élément particulier

- On nomme la balise concernée à l'aide de l'attribut id (comme pour les ancres)
- Concerne 1 balise, car l'id contient un nom unique à chaque fois.

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style5.css"/>
        <title>Test CSS 5</title>
</head>

<body>
<h1> Bienvenue sur la page test du CSS</h1>
<h2> Pr&eacute;sentation</h2>
<p id="prem"> Nous allons tester nos premiers fichiers CSS sur cette page.</p>

<p> Pour ce faire un petit paragraphe et un petit blockquote qui donne une citation de Paul
Verlaine.</p>
<blockquote>
<p>Les sanglots longs<br />
```

Un style sur un élément particulier (suite)

- À la place de *, on met le nom contenu dans id précédé d'un #
- Par exemple, sur le premier paragraphe.
- #prem
  {
  text-align :center ;
  }

Un style sur plusieurs éléments

- Sur des éléments de même nature ou de nature différente.
- On nomme les balises concerné à l'aide de l'attribut class
- C'est comme id sauf que le nom n'est pas unique à chaque fois.

```
h1
{
font-weight:normal;
text-decoration:blink;
}

blockquote p
{
text-transform:capitalize;
font-variant:small-caps;
}

h2
{
font-style:oblique;
}
```

Un style sur plusieurs éléments (suite)

- À la place de *, on met le nom contenu dans class précédé d'un .
- Par exemple, sur le titre de niveau 1 et le premier paragraphe.
- .deus
  {
  text-align :center ;
  }

L'alignement

- text-align :valeur ;
- valeur :
  - left : alignement à gauche.
  - right : alignement à droite.
  - center : centré.
  - justify : justifié.
- nom_balise
  {
  text-align :left ;
  }

Alinéa

- text-indent :valeur ;
- valeur : la longueur de la marge à créer
- en pixel :

```
nom_balise
{
text-indent :30px ;
}
```

La police

- font-family :valeur ;
- Valeur la police désirée.
- Attention la police doit être installée sur le navigateur.
- Celles dont on est sûr qu'elles sont installée : sans-serif, serif.
- Les plus standard : Arial, Arial Black, Comic Sans MS, Courier New, Georgia, Impact, Times New Roman, Trebuchet MS, Verdana.
- En général, on met plusieurs polices. Si la première ne marche pas ce sera la deuxième, sinon la troisième etc.
- Éviter de mettre trop de polices.

La police

- nom_balise
  {
  font-family :Elephant, “Comic Sans MS”, sans-serif ;
  }

La taille du texte

- font-size :valeur ;
- Valeur :
  - En pixel : nom_balise { font-size :18px ; }
  - En valeur relative (1em = taille normale, relative au m) : nom_balise { font-size :1.5em ; }
  - En valeur relative à la hauteur de la lettre x : nom_balise { font-size :1.5ex ; }
  - En pocentage par rapport à la taille normale : nom_balise { font-size :150% ; }

La taille du texte (suite)

- Valeur :
  - Avec un nom (du plus petit au plus grand) :
    - xx-small
    - x-small
    - small
    - medium
    - large
    - x-large
    - xx-large
  - nom_balise { font-size :large ; }

Soufflons un peu : un exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style7.css"/>
        <title>Test CSS 7</title>
</head>

<body>
<h1> Bienvenue sur la page test du CSS</h1>
<h2> Pr&eacute;sentation</h2>
<p id="prem"> Nous allons tester nos premiers fichiers CSS sur cette page.</p>

<p> Pour ce faire un petit paragraphe et un petit blockquote qui donne une citation de Paul
Verlaine.</p>
<blockquote>
<p>Les sanglots longs<br />
Des violons de l'automne<br />
Blessent mon coeur<br />
D'une langueur monotone.
</p>
</blockquote>
</body>
</html>
```

Soufflons un peu : un exemple (suite)

style7.css

```
1  h1
2  {
3  text-align:center;
4  font-family:Elephant,"Comic Sans MS", sans-serif;
5  }
6
7  #prem, blockquote p
8  {
9  text-align:right;
10 font-size:small;
11 }
12
13 h2
14 {
15 text-indent:30px;
16 }
```

Mise en gras

- font-weight :valeur ;
- Valeur :
  - bold : en gras.
  - normal : normal (valeur par défaut).
  - nom_balise
    {
    font-weight :blod ;
    }

Mise en italique

- font-style :valeur ;
- Valeur :
  - italic : en italique.
  - oblique : en italique.
  - normal : normal (valeur par défaut).
  - nom_balise
    {
    font-style :italic ;
    }

Mise en majuscules, mise en minuscule

- text-transform :valeur ;
- Valeur :
  - uppercase : texte en majuscule.
  - lowercase : texte en minuscule.
  - capitalize : première lettre de chaque mot en majuscule.
  - none : pas de changement (valeur par défaut).
- nom_balise
  {
  text-transform :lowercase ;
  }

Lettre en petites capitales

- font-variant :valeur ;
- Valeur :
  - small-caps : en petite capitale.
  - normal : normale (par défaut).
- nom_balise
  {
  font-variant :small-caps ;
  }

Souligné ou barré ou clignotant

- text-decoration :valeur ;
- Valeur :
  - underline : souligné.
  - line-through : barré.
  - blink : clignotant. Attention, ne fonctionne pas avec internet explorer !
  - none : normal. Valeur par défaut.

Encore un exemple

```
h1
{
font-weight:normal;
text-decoration:blink;
}

blockquote p
{
text-transform:capitalize;
font-variant:small-caps;
}

h2
{
font-style:oblique;
}
```

Couleur du texte

- color :ma_couleur
- 3 façons de déterminer ma_couleur :
  - Avec des mots. Les standards : white, silver, gray, black, red, marron, lime, green, orange, yellow, olive, aqua, blue, navy, fuschia, purple, teal .
    nom_balise { color : teal ; }
  - En hexadécimal.
  - En RGB (red, green, blue).
    nom_balise{color :rgb(x,y,z) ;}
    avec 0 (rien) $\geq$ x, y, z $\geq$ 255 (le max).
    Par exemple rgb(255,0,0) = rouge.
- S'applique pour toutes les balises.

Couleur de fond

- background-color :ma_couleur
- On détermine ma_couleur comme pour la couleur du texte.
- S'applique pour toutes les balises :
  - pour body (couleur de fond) ;
  - mais aussi pour les autres, on obtient alors du remplissage.

Les couleurs, un exemple

```
1  *
2  {
3  color:rgb(255,0,0);
4  }
5
6  body
7  {
8  background-color : teal;
9  }
10
11 h1
12 {
13 background-color : navy;
14 }
15
```

L'image de fond

- background-image :url("image_de_fond")
- Avec l'extension.
- Le chemin se fait depuis l'endroit où est placé le ficier CSS.
- Comme pour la couleur de fond s'applique à toutes les balises !
- body
  {
  background-image :url("../img/candy.gif") ;
  }
- Remarque : par défaut, l'image bouge avec le texte.

Bloquer l'image de fond

- background-attachment :valeur ;
- Valeur :
  - fixed : fixée.
  - scroll : le fond défile avec le texte. Valeur par défaut.
- body
  {
  background-image :url("../img/candy.gif") ;
  background-attachment :fixed ;
  }

Tu vois, elle se multiplie à l'infini

- Par défaut, l'image se multiplie à l'infini. C'est modifiable.
- background-repeat :valeur ;
- Valeur :
  - no-repeat : l'image ne se répètte pas.
  - repeat-x : se répète horizontalement sur la première ligne.
  - repeat-y : se répète verticalement sur la première colonne.
  - repeat : se répète à l'infini. Valeur par défaut.
- body
  {
  background-image :url("../img/candy.gif") ;
  background-repeat :no-repeat ;
  }

Placer l'image de fond

- background-position :valeur ;
  - Valeur :
    - Par rapport à la gauche de l'écran à g pixel et par rapport au haut de l'écran à h pixel.
      body { background-image :url("../img/candy.gif") ;
      background-position : gpx hpx ; }
  - Avec des mots :
    - top (en haut), middle (centrée verticalement), bottom (en bas)
    - left (à gauche), center (centrée horizotalement), right (à droite).
      Par exemple en bas à droite : body {
      background-image :url("../img/candy.gif") ;
      background-position : bottom right ; }

Que de background...

- Heureusement, on peut tout compiler en un.
- c'est une super propriété
- background :valeur ...
- Les 4, ou parmis les 4.
- Pas d'ordre.
- body
  {
  background :url("../img/candy.gif") bottom right fixed
  no-repeat ;
  }

Que ça bouge : les pseudo-formats

- Tout ce qu'on a vu comme propriété sur CSS s'applique tout le temps. Maintenant, on peut vouloir que ce ne soit pas toujours le cas.
- Marche mal avec internet explorer.
- nom_balise : pseudo-format
  {
  }
- pseudo-format :
  - hover : quand la souris passe dessus.
  - active : quand on clique dessus avec la souris.
  - visited : quand on a déjà cliqué dessus.
  - link : pour les liens non visités.

Que ça bouge : les pseudo-formats (suite)

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style14.css"/>
        <title>Lien vers une autre page de mon site (le retour)</title>
</head>

<body>
<p> Cliquez <a href="005titre.html" title="mon premier lien"> ici </a> pour aller vers la page qui
pr&eacute;sente les titres.
</p>
</body>
</html>
```

Que ça bouge : les pseudo-formats (fin)

style14.css

```
1  a
2  {
3  color:green;
4  }
5
6  a:hover
7  {
8  color:navy;
9  background-color:yellow;
10 }
```

Un peu plus structurant

Les listes à puces

- Pour lister, mais aussi pour faire des menus.
- 3 sortes de listes :
  - Les listes non-ordonnées (non numérotées).
  - Les listes ordonnées (numérotées).
  - Les listes de définitions (lexique etc.).

Liste non-ordonnée

- À l'extérieur des paragraphes.
- Une liste non-ordonnée se place entre les balises et .
- Une élément de la liste de place entre les balises et .
-
  élément de la liste
  ...
  .

Liste non-ordonnée : exemple

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Liste non ordonn&eacute;e</title>
</head>

<body>
<p>Le d&eacute;but de la liste des courses :</p>
<ul>
        <li>carottes</li>
        <li>yaourts</li>
        <li>jus d'orange</li>
</ul>

<p> Fin de la liste des courses.</p>
</body>
</html>
```

Liste ordonnée

- À l'extérieur des paragraphes.
- Une liste ordonnée se place entre les balises et .
- Une élément de la liste de place entre les balises et .
-
  élément de la liste
  ...
  .

Liste ordonnée : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Liste ordonn&eacute;e</title>
</head>

<body>
<p>Le d&eacute;but de la liste des courses :</p>
<ol>
        <li>carottes</li>
        <li>yaourts</li>
        <li>jus d'orange</li>
</ol>

<p> Fin de la liste des courses.</p>
</body>
</html>
```

Imbrication

- On peut imbriquer des listes :
-
  élément de la liste

  élément de la sous-liste
  ...
  ...

Listes imbriquées : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
.       <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
.       <title>Listes imbriqu&eacute;es</title>
</head>

<body>
<p>Le d&eacute;but de la liste des courses :</p>
<ol>
.       <li>carottes</li>
.       <li>yaourts<br/>
.       <ol>
.       .       <li> natures </li>
.       .       <li> aux fruits</li>
.       </ol>
.       </li>
.       <li>jus d'orange</li>
</ol>

<p> Fin de la liste des courses.</p>
</body>
</html>
```

Liste de définitions

- À l'exterieur des paragraphes.
- Une liste de définition se place entre les balises <dl > et </dl >.
- Une élément de la liste est composé de 2 parties :
  - Entre <dt > et </dt >pour le mot à définir.
  - Entre <dd > et </dd >pour la définition.
- <dl >
  <dt >Mot à définir </dt >
  <dd >Définition </dd >
  ...
  </dl >.

Liste de définition : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title>Liste de d&eacute;finition</title>
</head>

<body>
<p>Le d&eacute;but de la liste de d&eacute;fintion :</p>
<dl>
        <dt>Fromage</dt>
        <dd>Lait ferment&eacute;</dd>

        <dt> Yaourt</dt>
        <dd> Aussi du lait ferment&eacute;</dd>
</dl>

<p> Fin de la liste de d&eacute;finiton.</p>
</body>
</html>
```

Le retour de CSS

- On peut utiliser tous les éléments de CSS que nous avons vu précédement.
- Mais, aussi :

Décalage du texte

- list-style-position :valeur ;
- Valeur :
  - inside : puce dans le bloc du texte.
  - outside : puce en retrait du texte. Valeur par défaut.
- nom-balise-liste
  {
  list-style-position :inside ;
  }

Décalage du texte :exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style15.css"/>
        <title>Listes d&eacute;calage du textes</title>
</head>

<body>
<ol id="prem">
        <li>yaourts <br />
         non en retrait </li>
</ol>

<ol id="deus">
        <li> yaourts<br />
        en retrait</li>
</ol>

</body>
</html>
```

Décalage du texte :exemple (suite)

```
1  #prem
2  {
3  list-style-position : inside;
4  }
5
6  #deus
7  {
8  list-style-position : outside;
9  }
```

Présentation de la puce

- list-style-type :valeur .

Présentation de la puce (suite)

- Avec valeur pour ul :
  - disc : un disque noir. Valeur par défaut.
  - circle : un cercle.
  - square : un carré.
  - none : pas de puce.
- ul
  {
  list-style-type : square ;
  }

Présentation de la puce (suite)

- Avec valeur pour ol :
  - decimal : 1, 2, 3, 4 etc. Valeur par défaut.
  - decimal-leading : 01, 02, 03, 04 etc.
  - upper-roman : I, II, III, IV etc.
  - lower-roman : i, ii, iii, iv, etc.
  - uper-alpha : A, B, C, D, etc.
  - lower-alpha : a, b, c, d, etc.
  - lower-greek : $\alpha, \beta, \gamma, \delta,$ etc.
- ol
  {
  list-style-type : lower-roman ;
  }

Présentation de la puce : exemple

```
1  ol
2  {
3  list-style-type:upper-roman;
4  }
5
6  ul
7  {
8  list-style-type:square;
9  }
```

Présentation de la puce : exemple (suite)

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style16.css"/>
        <title>Listes imbriqu&eacute;es (le retour)</title>
</head>

<body>
<p>Le d&eacute;but de la liste des courses :</p>
<ul>
        <li>carottes</li>
        <li>yaourts</li>
        <ol>
                <li> natures </li>
                <li> aux fruits</li>
        </ol>
        <li>jus d'orange</li>
</ul>

<p> Fin de la liste des courses.</p>
</body>
</html>
```

Changer la puce pour une image

- list-style-image :url(“mon_image”)
- Avec l’extension.
- Chemin relatif depuis le repertoire contenant le fichier .css.
- ul
  {
  liste-style-image :url(“../img/puce.gif”) ;
  }

Changer la puce pour une image : exemple

```
1  ul
2  {
3  list-style-image:url("../img/puce.gif");
4  }
5
```

Changer la puce pour une image : exemple (suite)

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style17.css"/>
        <title>Listes imbriqu&eacute;es (le retour du come back)</title>
</head>

<body>
<p>Le d&eacute;but de la liste des courses :</p>
<ul>
        <li>carottes</li>
        <li>yaourts</li>
        <ol>
                <li> natures </li>
                <li> aux fruits</li>
        </ol>
        <li>jus d'orange</li>
</ul>

<p> Fin de la liste des courses.</p>
</body>
</html>
```

Les tableaux

- À l'extérieur d'un paragraphe.
- Tout ce qui concerne le tableau est entre les balises <table > et </table >.
- Tout ce qui concerne une lignes du tableau est entre les balises <tr > et </tr >.
- Tout ce qui concerne une cellule d'une ligne du tableau est entre les balises <td > et </td >.

Premier tableau

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <title> Premier tableau </title>
</head>

<body>
<table>
        <tr><td> ligne 1, colonne 1</td> <td>ligne 1, colonne 2</td></tr>
        <tr><td> ligne 2, colonne 1</td> <td>ligne 2, colonne 2</td></tr>
        <tr><td> ligne 3, colonne 1</td> <td>ligne 3, colonne 2</td></tr>
</table>
</body>
</html>
```

La bordure

- Et oui, la bordure, c'est de la décoration !
- C'est donc dans le fichier .css
- border-style :valeur ; (une bordure, mais laquelle ?)
  - solid : trait plein.
  - double : trait double.
  - dashed : trait en tirets.
  - dotted : trait en pointillés.
  - inset : effet 3D enfoncé.
  - outset : effet 3D surélevé.
  - ridge : encore un effet 2D.
  - none : pas de bordure. Valeur par défaut.

La bordure (suite)

- border-width :valeur ; (une bordure, mais de quelle épaisseur ?)
- valeur en pixel.
- border-color :ma_couleur ; (une bordure, mais de quelle couleur ?)
- ma_couleur comme vu précédement.

La bordure : exemple

```
1  table
2  {
3  border-style:solid;
4  border-width:3px;
5  border-color:navy;
6  }
7
```

La bordure : exemple (suite)

```
1  table, td
2  {
3  border-style:solid;
4  border-width:3px;
5  border-color:navy;
6  }
7
```

La bordure : exemple (fin)

- border-collapse : collapse ;

```
1  table, td
2  {
3  border-style:solid;
4  border-width:3px;
5  border-color:navy;
6  border-collapse:collapse;
7  }
```

super propriété : 3 en 1

- border-style, border-width, border-color réunies dans border : valeur1 valeur2 ma_coleur ;
- Pas d'ordre.
- Les trois ou pas.
- table, td
  {
  border :solid 3px navy ;
  border-collapse :collapse ;
  }
- Existe border-left , border-right , border-top , border-bottom .

Tout CSS aux tableaux

- Tout ce que nous avons vu dans CSS s'applique aux tableaux.
- ET les bordures s'appliquent à toutes les balises.

- On peut mettre ce que l'on veut dans un tableau.

Les entêtes de tableau

- Si on remplace td par th , on obtient les entêtes de tableau.
- Se met en général sur la première ligne.

Les entêtes de tableau : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style21.css"/>
        <title> Les ent\&ecirc;tes </title>
</head>

<body>
<table>
        <tr><th> ent&ecirc;te 1</th> <th> ent&ecirc;te 2</th></tr>
        <tr><td> ligne 1, colonne 1</td> <td>ligne 1, colonne 2</td></tr>
        <tr><td> ligne 2, colonne 1</td> <td>ligne 2, colonne 2</td></tr>
        <tr><td> ligne 3, colonne 1</td> <td>ligne 3, colonne 2</td></tr>
</table>
</body>
</html>
```

Les entêtes de tableau : exemple (suite)

```
1  table, td, th
2  {
3  border-style:solid;
4  border-width:3px;
5  border-color:navy;
6  border-collapse:collapse;
7  }
```

Un titre au tableau

- Les balises <caption > et</caption >placé juste après la balise <table > permet de mettre un titre au tableau.
- La propriété CSS caption-side :valeur , appliquée au nom de balise caption , permet de le placer par rapport au tableau.
  - left : à gauche du tableau.
  - rigth : à droite du tableau.
  - top : en haut du tableau. Valeur par défaut.
  - bottom : en bas du tableau.

Le titre du tableau : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style22.css"/>
        <title> Les titres des tableaux </title>
</head>

<body>
<table>
<caption>Mon premier tableau</caption>
        <tr><th> ent&ecirc;te 1</th> <th> ent&ecirc;te 2</th></tr>
        <tr><td> ligne 1, colonne 1</td> <td>ligne 1, colonne 2</td></tr>
        <tr><td> ligne 2, colonne 1</td> <td>ligne 2, colonne 2</td></tr>
        <tr><td> ligne 3, colonne 1</td> <td>ligne 3, colonne 2</td></tr>
</table>
</body>
</html>
```

Le titre du tableau : exemple (suite)

```
1  table, td, th
2  {
3  border-style:solid;
4  border-width:3px;
5  border-color:navy;
6  border-collapse:collapse;
7  }
8
9  caption
10 {
11 caption-side: bottom;
12 }
13
```

Pour le fun : diviser le tableau en 3 parties

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style22.css"/>
        <title> Les titres des tableaux </title>
</head>

<body>
<table>
<caption>Mon premier tableau</caption>
        <thead>
        <tr><th> ent&ecirc;te 1</th> <th> ent&ecirc;te 2</th></tr>
        </thead>
        <tfoot>
        <!-- IMPORTANT DANS CET ORDRE -->
        <tr><th> pied 1</th> <th> pied 2</th></tr>
        </tfoot>
        <tbody>
        <tr><td> ligne 1, colonne 1</td> <td>ligne 1, colonne 2</td></tr>
        <tr><td> ligne 2, colonne 1</td> <td>ligne 2, colonne 2</td></tr>
        <tr><td> ligne 3, colonne 1</td> <td>ligne 3, colonne 2</td></tr>
        </tbody>
</table>
</body>
</html>
```

Fusionner des cellules d'une même ligne

- On peut fusionner 2 (ou plus) cellules d'une même ligne.
- Pour ce faire on remplace les définitions de cellules à fusionner par <td colspan="x" >blabla </td >.
- Avec x le nombre de cellules que l'on fusionne.

Fusionner des cellules d'une même ligne : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style22.css"/>
        <title> Fusion sur une même ligne </title>
</head>

<body>
<table>
<caption>Mon premier tableau</caption>
        <tr><th> ent&ecirc;te 1</th> <th> ent&ecirc;te 2</th></tr>
        <tr><td colspan="2"> ligne 1 </td></tr>
        <tr><td> ligne 2, colonne 1</td> <td>ligne 2, colonne 2</td></tr>
        <tr><td> ligne 3, colonne 1</td> <td>ligne 3, colonne 2</td></tr>
</table>
</body>
</html>
```

Fusionner des cellules d'une même colonne

- On peut fusionner 2 (ou plus) cellules d'une même colonne.
- Pour ce faire on remplace une des définitions de cellule à fusionner par <td rowspan="x" >blabla </td >, et on enlève les autres.
- Avec x le nombre de cellules que l'on fusionne.

Fusionner des cellules d'une même colonne : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style22.css"/>
        <title> Les titres des tableaux </title>
</head>

<body>
<table>
<caption>Mon premier tableau</caption>
        <tr><th> ent&ecirc;te 1</th> <th> ent&ecirc;te 2</th></tr>
        <tr><td> ligne 1, colonne 1</td> <td>ligne 1, colonne 2</td></tr>
        <tr><td rowspan="2"> une partie de la colonne 1</td> <td>ligne 2, colonne 2</td></tr>
        <tr><td>ligne 3, colonne 2</td></tr>
</table>
</body>
</html>
```

La totale

C’est tout moche :’(

- Pour le moment tout s’affiche l’un à la suite de l’autre.
- Nous allons apprendre à changer ça.
- Pour ce faire, nous allons utiliser les techniques de positionnement CSS.

2 types de balises

- Les balises bloc : créent des blocs de textes qui s'affichent les uns en dessous des autres.
- Par exemple : <p >, <table >, <h1 >, <h2 >, etc.
- Les balises bloc se placent les unes en dessous des autres.
- Elles prennent toute la largeur de l'écran .
- Les balises en ligne : se trouvent toujours dans une balise bloc. Se place au fil du texte sur la même ligne .
- Par exemple : <img/ >, <a >, , , <q >, etc.

Comportement des blocs : exemple

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style23.css"/>
        <title> Les balises de type bloc </title>
</head>

<body>
<h1>Le Seigneur des Anneaux</h1>

<p> Article de Wikip&eacute;dia l'encyclop&eacute;die libre</p>

<p>Le Seigneur des Anneaux (The Lord of the Rings) est un roman en trois volumes de <strong>John
Ronald Reuel Tolkien</strong> paru en 1954,1955.</p>

<p>Il est la suite de Bilbo le Hobbit, m&ecirc;me si les deux histoires peuvent &ecirc;tre lues de
fa&ccedil;on ind&eacute;pendante ; suite que son &eacute;diteur avait demand&eacute;e &agrave;
Tolkien. Durant les douze ann&eacute;es de sa r&eacute;daction, il s'attache &agrave; faire vivre le
monde dont il est le cr&eacute;ateur, la Terre du Milieu, en truffant sa nouvelle oeuvre de
r&eacute;f&eacute;rences et d'allusions qui la relient au monde du Silmarillion sur lequel il
travaille depuis 1917 et dans lequel Bilbo le Hobbit  a &eacute;t&eacute; attir&eacute; "contre
l'intention premi&egrave;re" de son auteur. </p>
```

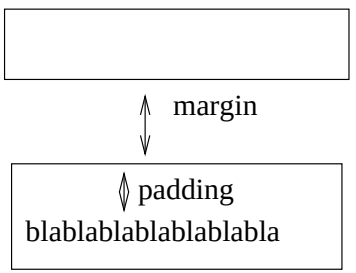
Comportement des blocs : exemple (suite)

```
h1
{
border-style:solid;
}

p
{
border-style:solid;
}
```

Le principe des boites

- En XHTML, chaque élément (bloc ou en ligne) est considéré comme une boite.
- Il possède donc les propriétés CSS suivantes (valeur par défaut, ou celles que vous leur donnez) :
  - Un contenu (votre blabla).
  - Une marge interne (padding).
  - Une marge externe (margin).
  - Une bordure (border , que nous avons déjà vu).
- Les margin ne s'additionnent pas 10 + 20 = 20.



Modifier les dimensions d'un bloc

- On ne peut pas modifier les dimensions d'un élément en ligne.
- On va utiliser les propriétés CSS :
  - width :valeur ; la largeur.
  - height :valeur ; la hauteur.
- Avec valeur en px, %, em, ex.
- Pour la largeur, il est préférable d'utiliser les % c'est ainsi que votre page s'adapte à toutes les configurations.
- Attention le % ne marche pas avec la hauteur (dommage :'( ).

Modifier les dimensions d'un bloc (exemple)

```
h1
{
border-style:solid;
height:50px;
}

p
{
border-style:solid;
width:50%;
}
```

Modifier les marges d'un bloc ou d'un élément en ligne

- Propriétés CSS :
  - padding :valeur ; marges internes (ou padding-top, padding-bottom, padding-right, padding-left).
  - margin :valeur ; marges externes (ou margin-top, margin-bottom, margin-right, margin-left).
- avec valeur en px, %, em, ex.

Modifier les marges d'un bloc ou d'un élément en ligne

```
h1
{
border-style:solid;
height:50px;
padding: 15px;
}

p
{
border-style:solid;
width:50%;
padding:10px;
margin:30px;
}
```

Centrer horizontalement un bloc

- D'abord donner une largeur au bloc, avec la propriété CSS width .
- Ensuite utiliser les propriétés CSS :
  - margin-left :auto ;
  - margin-right :auto ;

Centrer horizontalement un bloc : exemple

```
h1
{
border-style:solid;
height:50px;
padding: 15px;
}

p
{
border-style:solid;
width:50%;
padding:10px;
margin:30px;
margin-left:auto;
margin-right:auto;
}
```

Positionner les éléments sur la page

- 4 types de positionnements :
  - Positionnemement flottant.
  - Positionnement absolu.
  - Positionnement fixe.
  - Positionnement relatif.
- Fonctionnent avec les éléments de type bloc et les élements de type en ligne.

Positionnement flottant

- L'élement flotte et le reste l'entoure.
- Propriété CSS : float :valeur ;
- Avec valeur :
  - left flottement à gauche (l'élément flottant se trouve à gauche).
  - right flottement à droite.
  - none pas de flottement.
- On peut arrêter un flottement avec la propriété CSS clear :valeur ;
- Avec valeur :
  - left arrête le flottement à gauche.
  - right arrête le flottement à droite.
  - both arrête le flottement à droite et à gauche.
  - none ne fait rien.
- L'élément contenant clear se replace en dessous du dernier élément.

Positionnement flottant : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
        <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style27.css"/>
        <title> Flottement &agrave; gauche </title>
</head>

<body>
<h1>Le Seigneur des Anneaux</h1>

<p> Article de Wikip&eacute;dia l'encyclop&eacute;die libre</p>

<p>Le Seigneur des Anneaux (The Lord of the Rings) est un roman en trois volumes de <strong>John
Ronald Reuel Tolkien</strong> paru en 1954,1955.</p>

<p><img src="img/aragorn.gif" alt="Le beau Aragorn" id="ara"/>Il est la suite de Bilbo le Hobbit,
m&ecirc;me si les deux histoires peuvent &ecirc;tre lues de fa&ccedil;on ind&eacute;pendante ; suite
que son &eacute;diteur avait demand&eacute;e &agrave; Tolkien. Durant les douze ann&eacute;es de sa
r&eacute;daction, il s'attache &agrave; faire vivre le monde dont il est le cr&eacute;ateur, la Terre
du Milieu, en truffant sa nouvelle oeuvre de r&eacute;f&eacute;rences et d'allusions qui la relient
au monde du Silmarillion sur lequel il travaille depuis 1917 et dans lequel Bilbo le Hobbit  a
&eacute;t&eacute; attir&eacute; "contre l'intention premi&egrave;re" de son auteur. </p>
```

Positionnement flottant : exemple (suite)

```
h1
{
border-style:solid;
height:50px;
padding: 15px;
}

p
{
width:50%;
padding:10px;
margin:30px;
}

#ara
{
float:left;
margin:2px;
}
```

Positionnement absolu

- Placer l'objet au pixel prés.
- Ne pas en abuser (internet explorer bogue).
- Propriété CSS.
- nom_balise
  {
  position :absolute ;
  left :valeur ; /* optionnel */
  right :valeur ; /* optionnel */
  top :valeur ; /* optionnel */
  bottom :valeur ; /* optionnel */
  }
- Avec valeur en px.
- En théorie 2 optionnels bien choisis suffisent.
- Le (0,0) et en HAUT à gauche.

Positionnement absolu : exemple

```
h1
{
border-style:solid;
height:50px;
padding: 15px;
}

p
{
width:50%;
padding:10px;
margin:30px;
}

#ara
{
margin:2px;
position:absolute;
left:100px;
top:200px;
}
```

Positionnement fixe

- Le même principe que l'absolu, sauf que l'objet reste à sa position même si on descend dans la fenêtre.
- Propriété CSS.
- nom_balise

```
{
position :fixed ;
left :valeur ; /* optionnel */
right :valeur ; /* optionnel */
top :valeur ; /* optionnel */
bottom :valeur ; /* optionnel */
}
```

- Avec valeur en px.
- En théorie 2 optionnels bien choisis suffisent.
- Le (0,0) et en HAUT à gauche de celui qui le contient positionné.

Positionnement fixe : exemple

```
h1
{
border-style:solid;
height:50px;
padding: 15px;
}

p
{
width:50%;
padding:10px;
margin:30px;
}

#ara
{
margin:2px;
position:fixed;
left:100px;
top:200px;
}
```

Positionnement relatif

- Position par rapport à l'endroit où il devrait être.
- Propriété CSS.
- nom_balise

```
{
position :relative ;
left :valeur ; /* optionnel */
right :valeur ; /* optionnel */
top :valeur ; /* optionnel */
bottom :valeur ; /* optionnel */
}
```

- Avec valeur en px.

Positionnement fixe : exemple

```
h1
{
border-style:solid;
height:50px;
padding: 15px;
}

p
{
width:50%;
padding:10px;
margin:30px;
}

#ara
{
margin:2px;
}

strong
{
position:relative;
left: 100px;
top:50px;
}
```

Conception

Faire le site

- Étape de réalisation :
  - Dessiner la maquette (plan des pages, hierarchie des pages etc.).
  - Coder en XHTML (la page principale doit s'appeler index.html).
  - Faire le CSS (en testant à chaque étape).
  - Vérifier la validité du code :
    - XHTML : http ://validator.w3.org
    - CSS : http ://jigsaw.w3.org/css-validator

Oui, mais on fait comment pour tout agencer ?

- Nous allons utiliser la balise générique des blocs .
- Nous allons créer des blocs dans le code XHTML. Un bloc par élément de la maquette.
- Nous mettrons le contenu de chaque bloc entre et et nous nommerons ces div à l'aide de id ou class .
- Donc entre et il y aura des éléments de type bloc (<h1 >, <p >, etc.).
- Puis nous placerons chaque bloc grâce au fichier CSS.
- Ne fuyez pas, j'ai un exemple...

Les formulaires

Les formulaires

- Recueillir les informations des internautes.
- Pour un site interactif.
- Formulaire permet donc de récolter de l'information.
- MAIS, pour traiter l'information il faut un langage script comme le PHP ou l'ASP.
- Pour le moment, on ne va s'intéresser qu'à la récolte via XHTML et CSS.

Les formulaires (suite)

- Formulaire placé entre les balises <form> et </form> à l'extérieur des paragraphes.
- Peut contenir des paragraphes.

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Mon premier formulaire </title>
</head>

<body>
<p> Avant le formulaire.</p>
<form>
<p> J'ai quand m&ecirc;me envie de raconter ma vie.</p>
</form>
<p> Apr&egrave;s le formulaire, j'ai encore des choses &agrave; dire.</p>

</body>
</html>
```

2 attributs XHTML obligatoires

- method=”valeur”
- Avec valeur :
  - get : On voit ce qui est envoyé dans l’url. On ne peut envoyer que 255 caractères au maximum.
  - post : On ne voit pas ce qui est envoyé.
- Et action=”url”
- Avec url l’adresse de la page exécutée par le formulaire.
- C’est le plus souvant une page de scrip (par ex. php ou asp).
- La page qui est exécutée par le formulaire est celle qui va traiter l’information.

2 attributs XHTML obligatoires : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Mon second formulaire </title>
</head>

<body>
<p> Avant le formulaire.</p>
<form method="post" action="page.php">
<p> J'ai quand m&ecirc;me envie de raconter ma vie.</p>
</form>
<p> Apr&egrave;s le formulaire, j'ai encore des choses &agrave; dire.</p>

</body>
</html>
```

Les zones de saisie

- Zone que l'utilisateur peut remplir.
- 2 sortes :
  - Zone de texte monoligne (1 ligne et une seule).
  - Zone de texte multilignes (plusieurs lignes).

Zone de texte monoligne

- <input type="text" name="valeur" id="valeur2"/ >
- Avec valeur, le nom du texte qui va être envoyé.
- Avec valeur 2, le nom du champ (fonctionne comme les id pour CSS).
- En rêgle général valeur = valeur2.
- On peut remplacer text par password si on veut que le texte ne s'affiche pas.

Attributs optionnels

- value=”valeur” avec valeur, le texte par défaut.
- size=”valeur” avec valeur, la taille du champ.
- maxlength=”valeur” avec valeur, la taille maximal du texte tapé.

Zone de texte monoligne : exemple 1

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Zone de texte monoligne </title>
</head>

<body>

<form method="post" action="page.php">
<p>
<input type="text" name="zone" id="zone" size="12"/>
</p>
</form>

</body>
</html>
```

Zone de texte monoligne : exemple 2

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Zone de texte monoligne secr&ecirc;t </title>
</head>

<body>

<form method="post" action="page.php">
<p>
<input type="password" name="zone" id="zone"/>
</p>
</form>
</body>
</html>
```

Les libellés

- Il vaut mieux attacher un libellé aux zones remplies par l'utilisateur.
- <label for="valeur" >mon libellé </form >
- Avec valeur, la valeur de l'id du champ.

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Zone de texte monoligne secrêt </title>
</head>

<body>

<form method="post" action="page.php">
<p>
<label for="zone"> Donnez votre nom </label>
<input type="text" name="zone" id="zone" value="votre nom"/>
</p>
</form>
```

Zone de texte multilignes

- <textarea name="valeur" id="valeur2" >texte par défaut (peut être vide) </textarea >
- Avec valeur le nom du texte.
- Avec valeur2 l'identifiant du champ.
- En général, valeur = valeur2.
- Attributs optionnels :
  - rows="valeur" où valeur est égal au nombre de lignes.
  - cols="valeur" où valeur est égal au nombre de colonnes.

Zone de texte multilignes : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Zone de texte multilignes </title>
</head>

<body>

<form method="post" action="page.php">
<p>
<label for="zone"> Donnez votre appr&eacute;ciation </label><br/>
<textarea name="zone" id="zone" rows="3" cols="20"> Extra ! </textarea>
</p>
</form>

</body>
</html>
```

Les options

- Éléments permettant de faire un choix.
- 3 sortes :
  - Les cases à cocher.
  - Les zones d'option.
  - Les listes déroulantes.

Les cases à cocher

- On peut choisir une ou plusieurs cases.
- 1 case : <input type="checkbox" name="valeur1[]" id="valeur2" value="valeur3" /> .
- Avec valeur1, le nom de la case. Un valeur1[] par groupe.
- Avec valeur2, l'identifiant de la case.
- Avec valeur3, la valeur de la case.
- Si on rajoute l'attribut checked="checked" , la case est la valeur par défaut.

Les cases à cocher : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Case &agrave; cocher </title>
</head>

<body>
<form method="post" action="page.php">
<p> Quel dessert voulez-vous ?</p>
<p><input type="checkbox" name="dessert[]" id="dessert1" value="glace"/>  <label
for="dessert1"> De la glace </label><br/>
<input type="checkbox" name="dessert[]" id="dessert2" value="gateau" checked="checked"/>
 <label for="dessert2"> Du gateau au chocolat </label><br/>
<input type="checkbox" name="dessert[]" id="dessert3" value="tarte"/>  <label
for="dessert3"> De la tarte tatin </label><br/></p>
</form>
</body>
</html>
```

Les zones d'option

- Appelé aussi bouton radio
- On peut choisir une seule cases par groupe.
- 1 bouton radio : <input type="radio" name="valeur1" id="valeur2" value="valeur3" />.
- Un même name pour tous les boutons radios du même groupe.
- Avec valeur2, l'identifiant du bouton radio.
- Avec valeur3, pour donner une valeur au bouton qui va être coché.
- Si on rajoute l'attribut checked="checked" la case est la valeur par défaut.

Les zones d'option : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Zone d'option </title>
</head>

<body>

<form method="post" action="page.php">
<p> De quelle couleur sont vos cheveux ?</p>
<p><input type="radio" name="cheveux" id="c1" value="blond" />  <label for="c1"> Blond comme les
bl&eacute;s </label><br/>
<input type="radio" name="cheveux" id="c2" value="brun" checked="checked"/>  <label for="c2"> Brun
comme le chocolat </label><br/>
<input type="radio" name="cheveux" id="c3" value="roux"/>  <label for="c3"> Roux comme Poil de
Carotte </label><br/>
</p>
<p> Vous &ecirc;tes ?</p>
<p><input type="radio" name="genre" id="g1" value="feminin" />  <label for="g1"> Une
femme</label><br/>
<input type="radio" name="genre" id="g2" value="masculin"/>  <label for="g2"> Un homme
</label><br/></p>
</form>
</body>
</html>
```

La liste déroulante

- Même principe que pour les boutons radios : un choix et un seul.
- <select name="valeur" id="valeur2" >ma liste </selected >
- Avec valeur, le nom de la liste déroulante.
- Avec valeur2, l'identifiant de la liste déroulante.
- Un élément de la liste : <option value="valeur3" >mon choix </option >.
- Avec valeur3, la valeur du choix.
- On rajoute l'attribut selected="selected" dans option pour que ce soit le choix par défaut. Si selected est omis, c'est le premier choix, le choix par défaut.

La liste déroulante : exemple

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> liste d&eacute;roulante </title>
</head>

<body>

<form method="post" action="page.php">
<p> De quelle couleur sont vos cheveux ?</p>
<p><select id="cheveux" name="cheveux">
<option value="blond"> Blond comme les bl&eacute;s </option>
<option value="brun" selected="selected">  Brun comme le chocolat </option>
<option value="roux"> Roux comme Poil de Carotte </option>
</select>
</p>

</form>

</body>
</html>
```

Grouper les choix des listes déroulantes

- <otpgroup label=”nom du groupe”>les choix à l’aide de la balise option </optgroup >.
- Le nom du groupe apparaît à l’utilisateur.

Grouper : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> liste d&eacute;roulante -- groupe </title>
</head>

<body>

<form method="post" action="page.php">
<p> De quelle couleur sont vos cheveux ?</p>
<p><select id="cheveux" name="cheveux">
<optgroup label="clair">
<option value="blond"> Blond comme les bl&eacute;s </option>
<option value="roux"> Roux comme Poil de Carotte </option>
</optgroup>
<optgroup label="fonc&eacute;">
<option value="brun" selected="selected">  Brun comme le chocolat </option>
</optgroup>
</select>
</p>

</form>

</body>
```

Les boutons

- 2 sortes :
  - Les boutons d'envoi. Ils vont permettrent l'envoi des données à la page «action ». La balise : <input type="submit" value="valeur" />, avec valeur ce qui apparaît sur le bouton.
  - Le bouton de remise à zéro. Il remet les valeurs par défaut. La balise : <input type="reset" value="valeur" />, avec valeur ce qui apparaît sur le bouton.

Les boutons : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Boutons </title>
</head>

<body>

<form method="post" action="page.php">
<p> De quelle couleur sont vos cheveux ?</p>
<p><select id="cheveux" name="cheveux">
<optgroup label="clair">
<option value="blond"> Blond comme les bl&eacute;s </option>
<option value="roux"> Roux comme Poil de Carotte </option>
</optgroup>
<optgroup label="fonc&eacute;">
<option value="brun" selected="selected">  Brun comme le chocolat </option>
</optgroup>
</select>
<input type="submit" value="go !" />
<input type="reset" value="z&eacute;ro" /></p>

</form>

</body>
</html>
```

Le bouton envoi en image

- Le bouton d'envoi peut être remplacé par une image.
- <input type="image" src="valeur" />
- Avec valeur le chemin de l'image.

Les boutons images : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Boutons </title>
</head>

<body>

<form method="post" action="page.php">
<p> De quelle couleur sont vos cheveux ?</p>
<p><select id="cheveux" name="cheveux">
<optgroup label="clair">
<option value="blond"> Blond comme les bl&eacute;s </option>
<option value="roux"> Roux comme Poil de Carotte </option>
</optgroup>
<optgroup label="fonc&eacute;">
<option value="brun" selected="selected">  Brun comme le chocolat </option>
</optgroup>
</select>
<input type="image" src="img/puce.gif" />
<input type="reset" value="z&eacute;ro" /></p>


</form>

</body>
</html>
```

Organiser son formulaire

- Tout ce qui est envoyé à une page est dans un seul formulaire.
- On peut organiser son formulaire en plusieurs zones.
- <fieldset ><legend >nom de la zone </legend >Une partie de votre formulaire. </fieldset >
- nom de la zone est visible par l'utilisateur.

- Tout ce que nous venons de voir sur les formulaires est modifiable grace au CSS, of course !

Organiser son formulaire : exemple

```
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Zone de formulaire </title>
</head>

<body>

<form method="post" action="page.php">

<fieldset><legend> Vos cheveux </legend>
<p> De quelle couleur sont vos cheveux ?</p>
<input type="radio" name="cheveux" id="c1" value="blond" />  <label for="c1"> Blond comme les
bl&eacute;s </label><br/>
<input type="radio" name="cheveux" id="c2" value="brun" checked="checked"/>  <label for="c2"> Brun
comme le chocolat </label><br/>
<input type="radio" name="cheveux" id="c3" value="roux"/>  <label for="c3"> Roux comme Poil de
Carotte </label><br/>
<label for="cheveux2"> D&eacute;crivez votre coiffure</label> <br/>
<textarea id="cheveux2" name="cheveux2" rows="5" cols="40"></textarea>
</fieldset>

<fieldset><legend> Homme ou femme ?</legend>
<input type="radio" name="genre" id="g1" value="feminin" />  <label for="g1"> Une femme</label><br/>
<input type="radio" name="genre" id="g2" value="masculin"/>  <label for="g2"> Un homme </label><br/>
</fieldset>
<p>
<input type="submit" value="go !" />
<input type="reset" value="z&eacute;ro" />
</p>
</form>
</body>
</html>
```

Javascript

Introduction

Introduction

- Ce cours est basé sur le tutorial de Gilles Hunault : http ://forge.info.univ-angers.fr/ gh/tuteurs/tutjs.htm et sur le tutorial de Hugo Etievant : http ://cyberzoide.developpez.com/html/js.php3/
- Javascript est un petit langage langage objet pour le web s'exécutant sur la machine de l'utilisateur.
- Il se base sur les prototypes .
- Il ne faut pas confondre javascript et java.
- Attention, problème de compatibilité.
- Attention, peut être désacitvé.

La base

Comment l'intégrer dans les pages .html ?

- 2 possibilités :
  - <script type="text/javascript" > mon programme </script >
  - <script type="text/javascript" src="prog.js" />
- Dans la dernière solution, le programme est contenu dans le fichier prog.js qui est dans le même répertoire que notre page XHTML. Vous pouvez bien sur mettre le chemin que vous voulez.
- On peut ouvrir et fermer plusieurs fois javascript dans un programme.

Mon premier programmme

- window.document.write(“blabla”) ;pour écrire dans la page web.
- Ici on appelle la méthode write avec le paramêtre blabla sur l’objet window.document (document étant la page HTML,windows la fenêtre du navigateur)

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Mon premier programe javascrpit </title>
</head>

<body>
<h1>
<script type="text/javascript">
window.document.write("Hello Everybody");
</script>
</h1>
<p> Comment allez-vous ?</p>
</body>
</html>
```

Les variables

- On déclare les variables avec var (mais c'est optionnel).
- On affecte les variables avec = .
- javascript respecte la case. La variable n est différente de la variable N ;

Les variables : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Mon premier programe javascrpit </title>
</head>

<body>
<h1>
<script type="text/javascript">
var texte;
texte = "Hello Everybody !"
window.document.write(texte);
</script>
</h1>
<p> Comment allez-vous ?</p>
</body>
</html>
```

Les fonctions

- Une fonction est une portion de code qui pourra être utilisée à n'importe quel moment par simple appel à son nom. Classiquement, elle se place dans head (de façon à être disponible).

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Les fonctions javascrpit </title>
<script type="text/javascript">
function bonjour(){
var texte;
texte = "Hello Everybody !"
window.document.write(texte);
}
</script>

</head>

<body>

<h1> Nous allons vous saluer.</h1>
<p>
<script type="text/javascript">
bonjour();
</script>
</p>
```

Structure conditionnelle : if

- if (condition) {action1} else {action2} ;
- La partie else est facultative.
- Si condition est vraie, alors on exécute le code action1.
- Si condition est fausse, alors on exécute le code action2.

Structure conditionnelle : if – exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> if </title>
<script type="text/javascript">
function absolu(i){
i= parseInt(i);
if (i >= 0){ window.document.write("la valeur absolu de ",i," est ", i,"\n")}
        else { window.document.write("la valeur absolu de ", i ," est ", -i,"\n")}
}
</script>
</head>

<body>


<p>
<script type="text/javascript">
absolu(-10);
absolu(5);
</script>
</p>
</body>
</html>
```

Structure itérative : while

- while (condition) {action} ;
- Tant que condition est vraie on excécute le code action.
- Ne pas oublier de faire varier condition dans action.

Structure itérative : while – exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->

<script type="text/javascript">
function cpt(i){
i=parseInt(i);
j=i;
while (j+10>=i)
{ window.document.write(" ",i, " ");
i=i+1;}
}
</script>

        <title> while </title>
</head>

<body>


<p>
<script type="text/javascript">
cpt(10);
</script>
</p>
</body>
</html>
```

Structure itérative : for

- for (variable avec valeur de départ ;condition ; incrémentation) {action} ;
- On démare la boucle avec variable valant valeur de départ.
- Tant que condition est vraie on excécute le code action.
- À chaque fin de tour de boucle, incrémentation est exécutée.

Structure conditionnelle : for – exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> for </title>
<script type="text/javascript">
function cpt(i){
i=parseInt(i);
for (j=i; j+10>=i; i++)
{ window.document.write(" ", i ," ");
}
}
</script>
</head>

<body>


<p>
<script type="text/javascript">
cpt(10);
</script>
</p>
</body>
</html>
```

Date

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Date </title>
</head>

<body>
<p>
<script type="text/javascript" language="javascript">
jour = new Date();
window.document.write("Nous sommes le ",jour.getDate(),"/",jour.getMonth()+1,"/",jour.getYear(),"<br
/>");
window.document.write("Il est ",jour.getHours(),":",jour.getMinutes(),":",jour.getSeconds(),"<br
/>");
</script>
</p>
</body>
</html>
```

Date (suite)

- jour est une variable.
- On dit que c'est une instance de la classe Date .
- Puis on appele sur jour, les fonctions de jour, mois, année, heure, minute, seconde.
- Sur Linux/Unix et Firefox, il manque 1900 à l'année.

Date (suite)

- On peut même connaître le jour de la semaine.

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Jour </title>
</head>

<body>
<p>
<script type="text/javascript" language="javascript">
// Tableau contenant les jours de la semaine
   var Semaine = new Array('Dimanche','Lundi','Mardi','Mercredi','Jeudi','Vendredi','Samedi');
// attention : 0 correspond à Dimanche...

AujourdHui = new Date()

// On l'affiche au format jour/mois/année

window.document.write(AujourdHui.getDate(),'/',
AujourdHui.getMonth()+1,'/',AujourdHui.getYear(),'.');
window.document.write("C\'est un ");
window.document.writeln(Semaine[AujourdHui.getDay()]);
</script>
</p>
</body>
</html>
```

Date (fin)

- tableau = newarray() ; permet de déclarer un tableau à une dimension de nom tableau.
- tableau[i] est la ième case du tableau, elle se manipule comme une variable normale.
- getDay() renvoie le jour de la semaine en chiffre avec 0 = dimanche, 1 = lundi etc.

La fenêtre

- window.open(“url”,”nom”, “paramètre”) ; ouvre une nouvelle fenêtre.
- Avec url l’url de la nouvelle fenêtre (si elle existe).
- Avec nom le nom de la nouvelle fenêtre.
- Avec paramêtre les paramêtres de la fenêtre.
- nom.focus() ; place la fenêtre de nom nom au premier plan.
- nom.close() ; ferme la fenêtre de nom nom.

La fenêtre : exemple 1

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Jour </title>
</head>

<body>
<p>
<script type="text/javascript">
window.open("005titre.html","titre", "width=300 height=100");
</script>
</p>
</body>
</html>
```

La fenêtre (suite)

- On peut remplir la fenêtre à la volée :
- nom.document.open() ouvre le document de la fenêtre de nom nom.
- Après on écrit dedans comme on la vue pour la fenêtre principale.
- On oublie pas de refermer le document : nom.document.close() .

La fenêtre : exemple 2

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Fen&ecirc; &agrave; la vol&eacute;e  </title>
</head>

<body>
<p>
<script type="text/javascript">
fen=window.open("","fen", "width=300 height=100");
fen.document.open();
fen.document.write("Hello everybody !");
fen.document.close();
</script>
</p>
</body>
</html>
```

La fenêtre : appui sur un bouton

- Quand on clique sur un bouton c'est mieux.
- <input type="button" value="valeur" onClick="fctjs()">
- Avec valeur, ce qui est affiché sur le bouton.
- Avec fctjs() une fonction javascript que vous avez définie (qui peut ouvrir une fenêtre comme dans notre exemple ou faire n'importe quoi d'autre).

La fenêtre : exemple 3

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Fen&ecirc;tre -- bouton </title>

<script type="text/javascript">
function hello(){
fen = window.open("","titre", "width=300 height=100");
fen.document.open();
fen.document.write("Hello everybody !");
fen.document.close();
}
</script>

</head>

<body>
<p>
<input type="button" value="afficher" onclick="hello();"/>

</p>
</body>
</html>
```

Boites de dialogue

- Fenêtre d'alerte : alert("blabla") ;
- Fenêtre de confirmation : confirm("blabla") ;
- Boite de dialogue : prompt("blabla","C\'est ici que parle l'utilisateur") ;

- Nous les verrons dans les exemples qui suivent.

Clic sur un lien

- On peut réagir au click d'un bouton, mais aussi :
- au clic sur un lien.
- mon lien
- Avec fctjs() une fonction javascript que vous avez définie (qui peut ouvrir une alerte comme dans notre exemple ou faire n'importe quoi d'autre).

Clic sur un lien : exemple

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Clic </title>

<script type="text/javascript">
function clic(){
alert("Vous avez fait \"clic\"");
}
</script>

</head>

<body>
<p>
<a href="javascript:clic();"> Cliquez ici</a>

</p>
</body>
</html>
```

Passage sur un élément

- blabla .
- Avec fctjs1 et fctjs2 deux fonctions que vous aurez définies.

Passage sur un élément – Rollover

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> rollover </title>

<script type="text/javascript">
\\ On charge la deuxieme image
mon_image = new Image();
mon_image.src = "img/aragorn.gif";
}
</script>
</head>

<body>
<p>
<a href="#" onmouseover="getElementById('mon_image').src='img/aragorn.gif';"
   onmouseout="getElementById('mon_image').src='img/candy2.gif';">
<img src="img/candy2.gif" alt="Candy et des canards" id="mon_image"/>
</a>
</p>
</body>
</html>
```

Les formulaires

On peut aussi interagir avec les formulaires

- Pour ce faire on ajoute un attribut dans a balise d'ouveture form.
- onsubmit="fctjs() ;".
- onsubmit="fctjs() ;" signifie lorsqu'on a soumis le formulaire, on effectue la fonction javascript fctjs. Tout va dépendre de ce que l'on va mettre entre les parenthèses.
- Nous mettrons action="#" car ne nous nous redirigerons pas vers un script.

Les zones monolignes

- Par exemple, <input type="text" name="zone" id="zone" value="votre nom" />dans le formulaire de nom mon_form.
- Pour récuper la valeur qu'a tapé l'utilisateur, on va utiliser la commande : document.getElementById('zone').value
- Attention value ne retournera pas votre nom, mais la valeur contenue dans le champ après soumission.
- Attention value contiendra une chaîne de caractère. Si vous voulez la considérer comme un nombre, il faudra utiliser parseInt(maValue).

Les zones monolignes : exemple

- Nouveauté : le + permet de concaquéner deux chaines de caractère.

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Zone de texte monoligne et javascript</title>
</head>

<body>
<form method="get"  action="#" onsubmit="confirm('Vous vous appelez '+
document.getElementById('zone').value);">
<p>
<label for="zone"> Donnez votre nom </label>
<input type="text" name="zone" id="zone" value="votre nom"/>
<input type="submit" value="ok"/>
</p>
</form>
</body>
</html>
```

Les zones multilignes

- Par exemple, <textarea name="zone" id="zone">dans le formulaire de nom mon_form.
- pour récuper le valeur qu'a tapé l'utilisateur, on va utiliser la commande : document.getElementById('zone').value

Les zones multilignes : exemple

- Nouveauté : Ici, on réécrit dans la même fenêtre.

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Zone de texte multilignes et javascript </title>
<script type="text/javascript">

function afficher(x)
{
window.document.close();
window.document.open();
window.document.write('<html> <head><metahttp-equiv="Content-Type" content="text/html;
charset=iso-8859-1"/><title> Multiligne </title></head><body>');
window.document.write("<h1>Vous avez tap&eacute; :</h1>");
window.document.write("<p>" +x +"</p>");
window.document.write("</body></html>");
}
</script>
</head>
<body>
<form action="#" method="get" onsubmit="afficher(document.getElementById('zone').value);">
<p><label for="zone"> Donnez votre appr&eacute;ciation </label><br/>
<textarea name="zone" id="zone" rows="3" cols="20"> Extra ! </textarea>
<input type="submit" value="ok"/></p>
</form>
</body>
</html>
```

Les cases à cocher

- Notre formulaire à form1 comme id.
- Chaque élément d'un formulaire est rangé dans une case d'un tableau, getElementById('form1').elements, de longueur getElementById('form1').length
- Pour retrouver la case à cocher, on cherche l'élément de nom, le nom de la case à cocher (par ex toto[ ]). getElementById('form1').elements[i].name == 'toto[ ]' pour vérifier que le ième élément du formulaire est la case à cocher de nom toto[ ].
- Si le ième élément du formulaire est une case à cocher, getElementById('form1').elements[i].checked nous retourne true si la case est coché, false sinon.
- Si la case est cochée, getElementById('form1').elements[i].value nous en retourne la valeur.

Les cases à cocher : exemple

```
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Case &agrave; cocher et javascript</title>
<script type="text/javascript">
function afficher(f){
var r="";
for (var i=0; f.length>i;i++) {
  if (f.elements[i].name=="dessert[]"){
      if (f.elements[i].checked) {
         r = r + f.elements[i].value + " " ;
      }
   }
}
confirm("Vous avez choisi :" + r);
}

</script>
</head>
<body>
<form id="form1" onsubmit="afficher(document.getElementById('form1'));" action="#">
<p> Quel dessert voulez-vous ?</p>
<p><input type="checkbox" name="dessert[]" id="dessert1" value="glace"/>  <label for="dessert1"> De
la glace </label><br/>
<input type="checkbox" name="dessert[]" id="dessert2" value="gateau" checked="checked"/>  <label
for="dessert2"> Du gateau au chocolat </label><br/>
<input type="checkbox" name="dessert[]" id="dessert3" value="tarte"/>  <label for="dessert3"> De la
tarte tatin </label><br/>
<input type="submit" value="Go !"/></p>
</form>
</body>
</html>
```

Les cases à cocher : exemple 2

- Nouveauté : on peut savoir si on confirme ou non.

```
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Case &agrave; cocher et javascript</title>
<script type="text/javascript">
function gato(){
if (document.getElementById('dessert2').checked)
{
confirm("Vous avez choisi du gateau au chocolat ") ;
}
else {x = confirm("Vous n'aimez pas le gateau au chocolat ?");
        if (x) alert ("Dommage, c'est bon")
                else alert ("Ouf");
}}
</script>
</head>
<body>
<form id="form1" method="get" action="#" onsubmit="gato();">
<p> Quel dessert voulez-vous ?</p>
<p><input type="checkbox" name="dessert1" id="dessert1" value="glace"/>  <label for="dessert1"> De la
glace </label><br/>
<input type="checkbox" name="dessert2" id="dessert2" value="gateau" checked="checked"/>  <label
for="dessert2"> Du gateau au chocolat </label><br/>
<input type="checkbox" name="dessert3" id="dessert3" value="tarte"/>  <label for="dessert3"> De la
tarte tatin </label><br/>
<input type="submit" value="Go !"/></p>
</form>
</body>
</html>
```

Les zones d'option

- Pour retrouver le bouton radio, on cherche l'élément de nom, le nom des boutons radio (par ex toto). getElementById('form1').elements[i].name == 'toto' pour vérifier que le ième élément du formulaire est le bouton radio de nom toto.
- Si le ième élément du formulaire est le bouton radio, getElementById('form1').elements[i].checked nous retourne true si la case est coché, false sinon.
- Si la case est cochée, getElementById('form1').elements[i].value nous en retourne la valeur.

Les zones d'option : exemple

- Nouveauté : prompt .

```
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Zone d'option </title>
<script type="text/javascript">
function afficher(f){
var r="";
for (var i=0; f.length>i;i++) {
if (f.elements[i].name=="cheveux") {
 if (f.elements[i].checked) { r = r + f.elements[i].value; }
}
}
 x=prompt("Vous avez choisi :" + r,"oui ou non?");
alert(x);
}
</script>
</head>

<body>
<form id="form1" method="get" action="#" onsubmit="afficher(document.getElementById('form1'));">
<p> De quelle couleur sont vos cheveux ?</p>
<p><input type="radio" name="cheveux" id="c1" value="blond" />  <label for="c1"> Blond comme les
bl&eacute;s </label><br/>
<input type="radio" name="cheveux" id="c2" value="brun" checked="checked"/>  <label for="c2"> Brun
comme le chocolat </label><br/>
<input type="radio" name="cheveux" id="c3" value="roux"/>  <label for="c3"> Roux comme Poil de
Carotte </label><br/></p>
<p> Vous &ecirc;tes ?</p>
<p><input type="radio" name="genre" id="g1" value="feminin" />  <label for="g1"> Une
femme</label><br/>
<input type="radio" name="genre" id="g2" value="masculin"/>  <label for="g2"> Un homme </label><br/>
<input type="submit" value="go !" /></p>
</form>

</body>
```

Les listes déroulantes

- Si l'id de votre déroulante est liste, alors :
- document.getElementById('liste') désigne la liste en question.
- document.getElementById('liste').options[i] désigne l'élement i de la liste.
- document.getElementById('liste').selectedIndex donne le numéro de l'élément sélectionné.
- document.getElementById('liste').options[document.getElementById('liste').selectedIndex].value donne la valeur de l'élément selectionné.

Les listes déroulantes

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> liste d&eacute;roulante </title>
</head>

<body>
<form method="get" action="#"
onsubmit="alert(document.getElementById('cheveux').options[document.getElementById('cheveux').selected
Index].value);">
<p> De quelle couleur sont vos cheveux ?</p>
<p><select id="cheveux" name="cheveux">
<option value="blond"> Blond comme les bl&eacute;s </option>
<option value="brun" selected="selected">  Brun comme le chocolat </option>
<option value="roux"> Roux comme Poil de Carotte </option>
</select>
<input type="submit" value="ok"/></p>
</form>
</body>
</html>
```

Quelques évènements javascript

onchange

- Lorsqu'il y a un changement dans un champ ou une selection (bouton radio etc.).
- onchange="fctjs() ;"
- Avec fctjs une fonction javascript.

onchange : exemple sur un champ

- Met tu ne m'auras pas lorsqu'on écrit quelque chose.

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Zone de texte monoligne et javascript</title>
</head>

<body>

<form method="get" action="#" >
<p>
<label for="zone"> Donnez votre nom </label>
<input type="text" name="zone" id="zone" value="votre nom"
onchange="document.getElementById('zone').value='Tu ne m\'aura pas';"/>
<input type="submit" value="ok"/>
</p>
</form>

</body>
</html>
```

onclick

- Lorsqu'il y a clic dans un champ, sur une selection, sur un bouton, un lien, etc.
- onclick="fctjs() ;"
- Avec fctjs une fonction javascript.

onclick : exemple sur un champ

- Met tu ne m'auras pas lorsqu'on clique sur le champ.

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Zone de texte monoligne et javascript</title>
</head>

<body>

<form method="get" action="#">
<p>
<label for="zone"> Donnez votre nom </label>
<input type="text" name="zone" id="zone" value="votre nom"
onclick="document.getElementById('zone').value='Tu ne m\'aura pas';"/>
<input type="submit" value="ok"/>
</p>
</form>

</body>
</html>
```

ondblclick

- Lorsqu'il y a clic dans un champ, sur une selection, sur un bouton, un lien etc.
- ondblclick="fctjs() ;"
- Avec fctjs une fonction javascript.

ondblclick : exemple sur un champ

- Met tu ne m'auras pas lorsqu'on double-clique sur le champ.

```
<!DOCTYPE html
    PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Zone de texte monoligne et javascript</title>
</head>

<body>

<form method="get" action="#">
<p>
<label for="zone"> Donnez votre nom </label>
<input type="text" name="zone" id="zone" value="votre nom"
ondblclick="document.getElementById('zone').value='Tu ne m\'aura pas';"/>
<input type="submit" value="ok"/></p>
</form>

</body>
</html>
```

onfocus/onblur

- onfocus :
  - Lorsqu'il y a focus sur un champ, sur un bouton ou sur un lien .
  - onfocus="fctjs() ;"
  - Avec fctjs une fonction javascript.
- onblur :
  - Lorsqu'il n'y a plus le focus sur un champ, sur un bouton ou sur un lien .
  - onblur="fctjs() ;"
  - Avec fctjs une fonction javascript.

onblur : exemple sur un champ

- Met tu ne m'auras pas lorsqu'on quitte le champ.

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Zone de texte monoligne et javascript</title>
</head>

<body>

<form method="get" action="#">
<p>
<label for="zone"> Donnez votre nom </label>
<input type="text" name="zone" id="zone" value="votre nom"
onblur="document.getElementById('zone').value='Tu ne m\'aura pas';"/>
<input type="submit" value="ok"/>
</p>
</form>

</body>
</html>
```

onmouseover/onmouseout

- onmouseover
  - Lorsque la souris est sur un champ, sur un lien, sur un bouton, etc.
  - onmouseover=”fctjs() ;”
  - Avec fctjs une fonction javascript.
- onmouseout
  - Lorsque la souris quitte un champ, sur un lien, un bouton, etc.
  - onmouseout=”fctjs() ;”
  - Avec fctjs une fonction javascript.

onmouseover : exemple sur un champ

- Met tu ne m'auras pas lorsque la souris est sur le champ.

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Zone de texte monoligne et javascript</title>
</head>

<body>

<form method="get" action="#">
<p>
<label for="zone"> Donnez votre nom </label>
<input type="text" name="zone" id="zone" value="votre nom"
onmouseover="document.getElementById('zone').value='Tu ne m\'aura pas';"/>
<input type="submit" value="ok"/></p>
</form>

</body>
</html>
```

onkeypress/onkeydown

- onkeypress
  - Lorsque une touche du clavier est relachée sur un champ, sur un lien ou sur un bouton.
  - onkeypress=”fctjs() ;”
  - Avec fctjs une fonction javascript.
- onkeydown
  - Lorsque une touche du clavier est appuyée sur un champ, sur un lien ou un bouton.
  - onkeydown=”fctjs() ;”
  - Avec fctjs une fonction javascript.

onkeypress : exemple sur un champ

- Met tu ne m'auras pas relache une touche dans le champ.

```
<!DOCTYPE html
   PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
   "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" lang="fr" xml:lang="fr">

<head>
        <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<!--    <link rel="stylesheet" type="text/css" title="Mon style" media="screen"
href="css/style_final.css"/> -->
        <title> Zone de texte monoligne et javascript</title>
</head>

<body>

<form method="get" action="#">
<p>
<label for="zone"> Donnez votre nom </label>
<input type="text" name="zone" id="zone" value="votre nom"
onkeypress="document.getElementById('zone').value='Tu ne m\'aura pas';"/>
<input type="submit" value="ok"/>
</p>
</form>

</body>
</html>
```