

# Technologies Web

2019/2020

Préparé par : Pr . Jai-Andaloussi Said

1

## Plan du cours

- ◆ Introduction
- ◆ Serveur Web et protocole HTTP
- ◆ HTML / XHTML / CSS
- ◆ Javascript
- ◆ HTML 5

2

## Plan du cours

- ◆ Introduction
- ◆ Serveur Web et protocole HTTP
- ◆ HTML / XHTML / CSS
- ◆ Javascript
- ◆ HTML 5

3

## Introduction

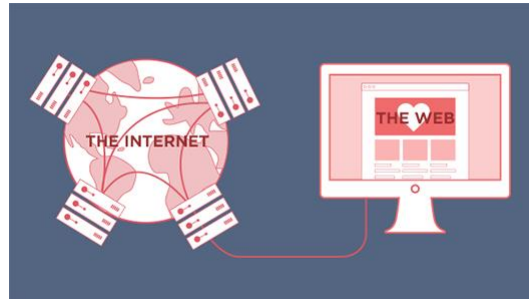
# Web ???



4

## Introduction

# Internet vs Web ???



5

## Introduction

### Historique : Projet Arpanet

- ◆ 1959-68 : Le programme Arpa naît pendant la guerre froide  
**La peur d'une guerre nucléaire**
- Faiblesse du système centralisé distribué
- Proposition d'un maillage d'ordinateurs (1964, P. Baran)
- 1<sup>ère</sup> communication téléphonique entre 2 machines en 1965



6

## Introduction

### Historique : Projet Arpanet

- ◆ **1959-68** : Le programme Arpa naît pendant la guerre froide  
**La peur d'une guerre nucléaire**
  - Faiblesse du système centralisé distribué
  - Proposition d'un maillage d'ordinateurs (1964, P. Baran)
  - 1<sup>ère</sup> communication téléphonique entre 2 machines en 1965
- ◆ **1969** : Arpanet (Advanced Research Projects Agency Network )
- ◆ **1969** : 4 noeuds, 1971 : 15 noeuds, 1972 : 37 noeuds
- ◆ **1970-82** : Ouverture sur le monde
  - Apparition du courrier électronique
  - Communications internationales (Angleterre, Norvège)
  - Apparition de TCP/IP (1974) plus puissant que NCP
- ◆ **1983** : TCP/IP adopté comme standard Arpanet => Internet

7

## Introduction

### Historique : Internet World Wide Web

- ◆ **1983-89** : Expansion du réseau : autoroutes de l'information
  - La NSF (National Science Foundation) effectue des progrès importants (réseau NFSNET)
  - Utilisation importante par les scientifiques
  - Réseaux hétérogènes (NCP et TCP/IP)
  - Fin officielle de Arpanet en 1989 (TCP/IP)
- ◆ **1990-97** : L'explosion d'internet
  - 1990, le physicien Tim Berners Lee (CERN) étend le concept de lien hypertexte à Internet
  - HyperText Markup Language et HyperText Transfer Protocol
  - 1<sup>er</sup> navigateur : NCSA Mosaic
  - 1995 ouverture au grand public (Netscape et Internet Explorer)
  - 1997 des dizaines de milliers de noeuds dans plus de 42 pays

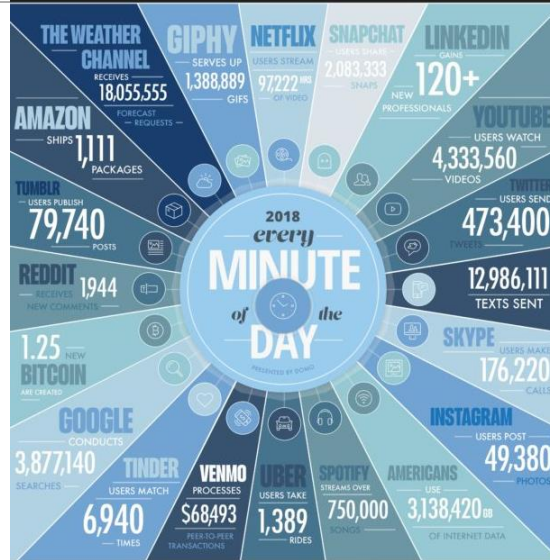
8



## Introduction

Statistiques d'usage d'Internet

Chaque minute sur Internet



## Introduction

Statistiques d'usage d'Internet

### Top 10 des sites les plus visités dans le monde

1. Google.com
2. YouTube
3. Facebook
4. Baidu
5. Wikipédia
6. Yahoo
7. QQ
8. Taobao
9. Amazon
10. Twitter

Source : [Alexa](#) (juillet 2018)

## Plan du cours

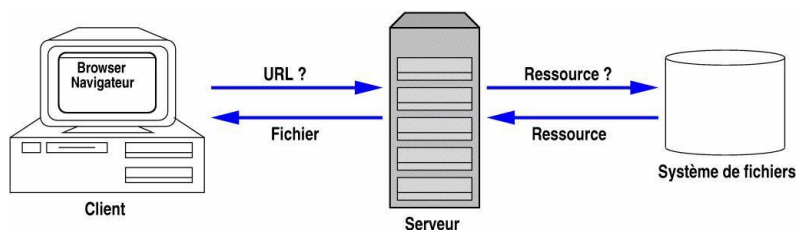
- ◆ Introduction
- ◆ **Serveur Web et protocole HTTP**
- ◆ HTML / XHTML / CSS
- ◆ Javascript
- ◆ HTML 5

13

## Serveur Web et protocole HTTP

Principes de fonctionnement : la base du web

### Architecture Client/Serveur



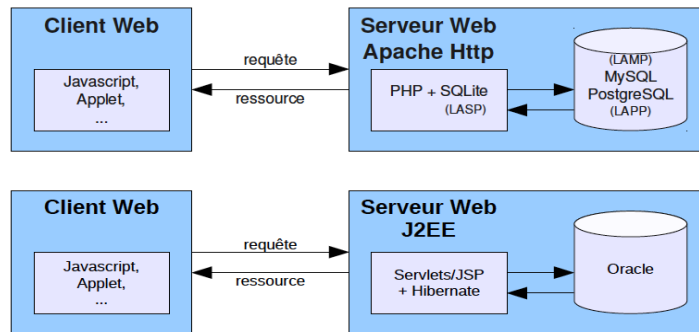
Nécessité d'un protocole de communication : Http

14

## Serveur Web et protocole HTTP

### Principes de fonctionnement : Différents types de ressources

- ◆ Ressources statiques : HTML, images, son, vidéos
- ◆ Ressources dynamiques
  - Côté client : applet (Java), Javascript, Plugin, ActiveX
  - Côté serveur : CGI, SSI, servlets/JSP, scripts serveur (php)



15

## Serveur Web et protocole HTTP

### Principes de fonctionnement : URL, URN et URI

- ◆ URL : Uniform Resource Locator
  - Spécification de la localisation d'une ressource de manière unique
- ◆ URN : Uniform Resource Name
  - Mécanisme de nommage des ressources
  - urn:<Namespace>:<SpecificString>
  - Namespace : identificateur de nommage (ex : isbn (International Standard Book Number))
  - SpecificString : chaîne de caractères spécifique désignant la ressource de manière unique
- ◆ URI : Uniform Resource Identifier
  - URI = URL + URN
  - En pratique, la forme d'URI la plus utilisée est l'URL

16



## Serveur Web et protocole HTTP

Principes de fonctionnement : URL (Uniform Ressource Locator)

Format

`<protocole>://<serveur>:<port>/<chemin>/<ressource>`

Exemples :

<http://www.linux-mandrake.com:80/fr/index.html>

<http://perso.telecom-bretagne.eu/saidandaloussi>

<ftp://ftp.debian.fr.org/>

<file://home/cours/>

<mailto:said.andaloussi@fsac.ac.ma>

<telnet://user:password@host:port>

17

## Serveur Web et protocole HTTP

Le protocole HTTP : Présentation de HTTP

- ◆ Le protocole HTTP (Hypertext Transfer Protocol) est un protocole de communication qui a été développé pour le web
- ◆ <http://www.w3c.org/>
- ◆ Non sécurisé (par opposition à HTTPS)
- ◆ Différents logiciels utilisent HTTP :
  - différents logiciels clients comme les navigateurs web, les robots d'indexation, les aspirateurs de site. . .
  - les logiciels serveurs peuvent être Apache, IIS. . .
- ◆ La liaison entre le client et le serveur peut être direct ou via un serveur mandataire (proxy), . . .

18

## Serveur Web et protocole HTTP

### Le protocole HTTP : Présentation de HTTP

- ◆ La première version du protocole était très simple : le client se connecte et demande une ressource (GET), le serveur lui répond, la connexion est fermée.
- ◆ Aucune autre donnée n'est envoyée ou reçue.
- ◆ HTTP 1.0 (puis HTTP 1.1) ont été créés pour enrichir les possibilités de ce protocole.
- ◆ Plusieurs requêtes sont possibles, des données complémentaires peuvent être envoyées par le client ou par le serveur.

19

## Serveur Web et protocole HTTP

### Le protocole HTTP : Présentation de HTTP

#### Les méthodes

- ◆ OPTIONS : demande les méthodes utilisables sur l'URI
- ◆ GET : demande les informations et les données de l'URI
- ◆ HEAD : demande uniquement les informations sur l'URI
- ◆ POST : envoie de données (ex : formulaire) traitées par l'URI
- ◆ PUT : enregistre le corps de la requête à l'URI
- ◆ DELETE : supprime les données pointées par l'URI
- ◆ TRACE : retourne ce qui a été envoyé par le client (echo)

20

## Serveur Web et protocole HTTP

### Le protocole HTTP : Présentation de HTTP

- ◆ Les deux versions sont très proches dans la syntaxe.
  - Commande URL version (HTTP /1.0 ou HTTP /1.1)
  - En -tête de requête
  - Corps de requête
- ◆ Les commandes les plus utilisées sont : GET et POST.
- ◆ L'entête de requête contient des informations sous la forme Nom : valeur.
- ◆ HTTP 1.1 ajoute le maintien de la connexion entre les requêtes (permet d'envoyer plusieurs requêtes sans attendre la réponse, diminution de la charge du réseau, . . . ).

21

## Serveur Web et protocole HTTP

### Le protocole HTTP : Limites

**Le protocole Http 1.1 ne gère pas les sessions**

=>

**Alternative utilisée par les navigateurs : cookies**

- Fichiers textes stockés sur le disque dur du client
- Durée de vie limitée, fixée par le site visité
- Problème avec les vieux navigateurs : il est possible pour un serveur de récupérer des cookies d'un client dont il n'est pas à l'origine

22

## Serveur Web et protocole HTTP

### Le protocole HTTP : Limites

Le protocole Http n'est pas sécurisé

=>

Alternative à Http : Https

- 's' pour secured
- Combinaison de Http avec SSL ou TLS
- Vérification de l'identité d'un site par un certificat d'authentification.

Un certificat électronique ( aussi appelé certificat numérique ou certificat de clé publique) peut-être vu comme une carte d'identité numérique. Il est utilisé principalement pour identifier une entité physique. Il est signé par un tiers de confiance qui atteste du lien entre l'identité physique et l'entité numérique (Virtuel).

23

## Serveur Web et protocole HTTP

### Le protocole HTTP : Limites

Le protocole Http n'est pas sécurisé

=>

Alternative à Http : Https

- 's' pour secured
- Combinaison de Http avec SSL ou TLS
- Vérification de l'identité d'un site par un certificat d'authentification
- Garantie confidentialité et intégrité des données envoyées par l'utilisateur (ex : formulaires)
- Port par défaut : 443.

24

## Plan du cours

- ◆ Introduction
- ◆ Serveur Web et protocole HTTP
- ◆ **HTML / XHTML / CSS**
- ◆ Javascript
- ◆ HTML 5

25

## HTML / XHTML / CSS

### Introduction : Historique

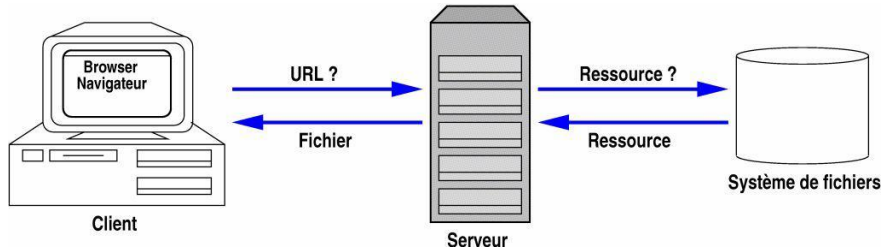
- Années 1990 : HTML est créé par Tim Berner-Lee au Centre Européen de Recherche Nucléaire (CERN)
- 1995 : HTML 2.0 normalisation par l'IETF (Internet Engineering Task Force)
- 1996 : HTML 3.2 ajout des tables, des applets (Java), etc.
- 1998 : HTML 4.01 ajout des feuilles de styles, des frames, etc.
- 2000 : XHTML 1.0 reformulation de HTML 4 en XML 1.0
- 2002 : XHTML 2.0 ?

Enfin le 2 juillet 2009 le W3C a officiellement annoncé l'abandon du développement du XHTML 2 afin de se consacrer entièrement à son nouveau successeur : le HTML 5.

26

## HTML / XHTML / CSS

### Introduction : Principe de fonctionnement



- Le navigateur effectue une requête spécifiée à travers l'URL
- Le serveur retourne un flot typé de données
- Le navigateur interprète le flot de données et l'affiche

27

## HTML / XHTML / CSS

### Introduction : HTML 4 / XHTML 1.0

**XHTML 1.0 n'est qu'une reformulation de  
HTML 4 en tant qu'application XML 1.0**

Principales différences XHTML 1.0 / HTML 4 :

- les noms d'éléments et d'attributs doivent être en minuscule
- les valeurs d'attributs sont entre guillemets ou apostrophes
- un attribut a toujours une valeur
- tout tag ouvert doit être fermé
- tous les éléments doivent être imbriqués correctement : pas de chevauchement de tags

28

## HTML / XHTML / CSS

Que choisir pour écrire de l' HTML?

### A la main, avec un éditeur de texte

Simple Text, Bloc-note, Emacs, WordPad

### Avec un logiciel « assistant » au code HTML

Dreamweaver, Netscape composer, FrontPage, Claris Homepage ...

29

## HTML / XHTML / CSS

Que choisir pour lire HTML?

**Le client doit pouvoir interpréter HTML et afficher le résultat.  
Ils sont divers, tournant sur des systèmes différents:**

– Chrome, Mozilla, Netscape Navigator, Internet explorer,  
Safari, Firefox, Opéra, etc ...

30

## HTML / XHTML / CSS

Que peut-on trouver dans un document HTML ?

type d'information	extension de fichier
texte ascii	.txt
document HTML	.html ou .htm
image GIF	.gif
image TIFF	.tiff
image bitmap XBM	.xbm
Image JPEG	.jpg ou .jpeg
son AIFF	.aiff
son AU	.au
vidéo QuickTime	.mov
vidéo MPEG	.mpeg ou .mpg
fichier PostScript	.ps

31

## HTML / XHTML / CSS

Introduction : Langages à balises

- Un fichier HTML/XHTML est un fichier texte (cf. protocole http) contenant des balises appelant des commandes, dont l'action est limitée au texte contenu entre la balise de début et la balise de fin.
- Extension HTML : .htm ou .html ; XHTML : .xhtml
- Balise de début : <commande[ attribut1=valeur1[ attribut2=valeur2 ...]]>
- Balise de fin : </commande>
- Commentaires : <!--Ceci est un commentaire-->

**Remarque** : retours chariot, succession d'espaces et/ou de tabulations ne sont pas pris en compte.

32



## HTML / XHTML / CSS

### Introduction : Langages à balises

Ces balises peuvent être insérées n'importe où dans le texte, entre 2 phrases, mots, lettres ...

#### Exemple :

```
<gras>Le <italique> cours </italique> HTML</gras>
```



**Le *cours* HTML**

33

## HTML / XHTML / CSS

### Introduction : Langages à balises

## Il faut respecter une logique d'imbrication:

#### Exemple :

##### Bon:

```
<gras><italique> Le cours HTML </italique> </gras>
```

Résultat : **Le *cours* HTML**

##### Mauvais:

```
<gras><italique> Le cours HTML</gras></italique>
```

34

## HTML / XHTML / CSS

### Introduction : Langages à balises

Le langage HTML est insensible à la casse, mais comme XHTML l'est, toujours écrire en minuscules.

Exemple :

Bon:

```
<gras><italique> Le cours HTML </italique> </gras>
```

Mauvais:

```
<GRAS><italique> Le cours HTML </italique> </GRAS>
```

```
<Gras><ITALIQUE> Le cours HTML </ITALIQUE> </Gras>
```

```
<GRAS><ITALIQUE> Le cours HTML </italique> </GRAS>
```

35

## HTML / XHTML / CSS

### Un fichier HTML

Un fichier HTML doit comporter au minimum ces 4 balises:

```
<html>
<head>
<title>
<body>
```

Expert : un fichier XHTML doit comporter ces 4 balises + les déclarations XML et DTD

36

## HTML / XHTML / CSS

### Analyse des balises

`<html> .. </html>` Délimite le début et la fin du document

`<head> .. </head>` Entête du document

`<title> .. </title>` Titre du document

`<body> .. </body>` Corps du document

37

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Titres (éléments de bloc)

Il y a 6 niveaux de titre :

`<h1> ... </h1>`

`<h2> ... </h2>`

`<h3> ... </h3>`

...

`<h6> ... </h6>`

**Ceci est mon titre utilisant la balise h1**

**Ceci est mon titre utilisant la balise h2**

Ceci est mon titre utilisant la balise h3

Ceci est mon titre utilisant la balise h4

Ceci est mon titre utilisant la balise h5

Ceci est mon titre utilisant la balise h6

38

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Listes (éléments de bloc)

Chaque item d'une liste est déclaré par `<li> ... </li>`.

Liste simple :

```
<ul>
  <li> item </li>
  <li> item </li>
  <li> item </li>
</ul>
```

Liste numérotée :

```
<ol>
  <li> item </li>
  <li> item </li>
  <li> item </li>
</ol>
```

Il existe aussi des listes de définitions (`<dl> ... </dl>`).

39

## HTML / XHTML / CSS

### Exemple 1 :

Source	Résultat
<pre>&lt;ol&gt; &lt;li&gt;&lt;em&gt;Exemple de liste ordonnée&lt;/em&gt;&lt;/li&gt; &lt;li&gt;Pique&lt;/li&gt; &lt;li&gt;Coeur&lt;/li&gt; &lt;li&gt;Carreau&lt;/li&gt; &lt;li&gt;Trèfle&lt;/li&gt; &lt;/ol&gt;</pre>	<p><i>Exemple de liste ordonnée</i></p> <ol style="list-style-type: none"> <li>1. Pique</li> <li>2. Coeur</li> <li>3. Carreau</li> <li>4. Trèfle</li> </ol>

### Exemple 2:

Source	Résultat
<pre>&lt;ol&gt; &lt;li&gt;Fleurs &lt;ol&gt; &lt;li&gt;Rose&lt;/li&gt; &lt;li&gt;Tulipe&lt;/li&gt; &lt;li&gt;Iris&lt;/li&gt; &lt;/ol&gt;&lt;/li&gt; &lt;li&gt;Plantes &lt;ol&gt; &lt;li&gt;Caroube&lt;/li&gt; &lt;li&gt;Combava&lt;/li&gt; &lt;li&gt;Surelle&lt;/li&gt; &lt;/ol&gt;&lt;/li&gt; &lt;/ol&gt;</pre>	<ol style="list-style-type: none"> <li>1. Fleurs <ol style="list-style-type: none"> <li>1. Rose</li> <li>2. Tulipe</li> <li>3. Iris</li> </ol> </li> <li>2. Plantes <ol style="list-style-type: none"> <li>1. Caroube</li> <li>2. Combava</li> <li>3. Surelle</li> </ol> </li> </ol>

40

## HTML / XHTML / CSS

### Exemple 1 :

Source	Résultat
<pre> &lt;ol type="I"&gt;   &lt;li&gt;Fleurs     &lt;ol type="A"&gt;       &lt;li&gt;Rose&lt;/li&gt;       &lt;li&gt;Tulipe&lt;/li&gt;       &lt;li&gt;Iris&lt;/li&gt;     &lt;/ol&gt;&lt;/li&gt;   &lt;li&gt;Plantes     &lt;ol type="A"&gt;       &lt;li&gt;Caroube&lt;/li&gt;       &lt;li&gt;Combava&lt;/li&gt;       &lt;li&gt;Surelle&lt;/li&gt;     &lt;/ol&gt;&lt;/li&gt; &lt;/ol&gt; </pre>	<pre> I. Fleurs   A. Rose   B. Tulipe   C. Iris II. Plantes   A. Caroube   B. Combava   C. Surelle </pre>

41

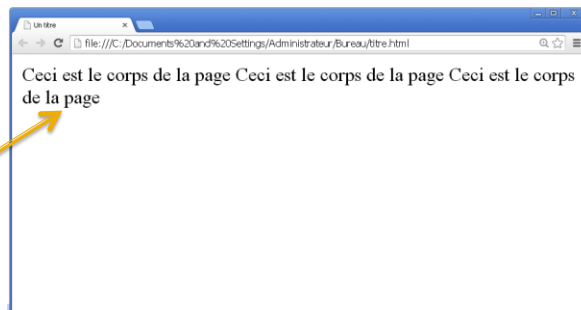
## HTML / XHTML / CSS

### Structure d'une page

```

<HTML>
<HEAD>
<TITLE>Un titre</TITLE>
</HEAD>
<BODY>
Ceci est le corps de la page
Ceci est le corps de la page
Ceci est le corps de la page
</BODY>
</HTML>

```



42

## HTML / XHTML / CSS

### Caractères accentués

Paragraphe normal :


  
`<P [ALIGN=position]>texte du paragraphe</P>`

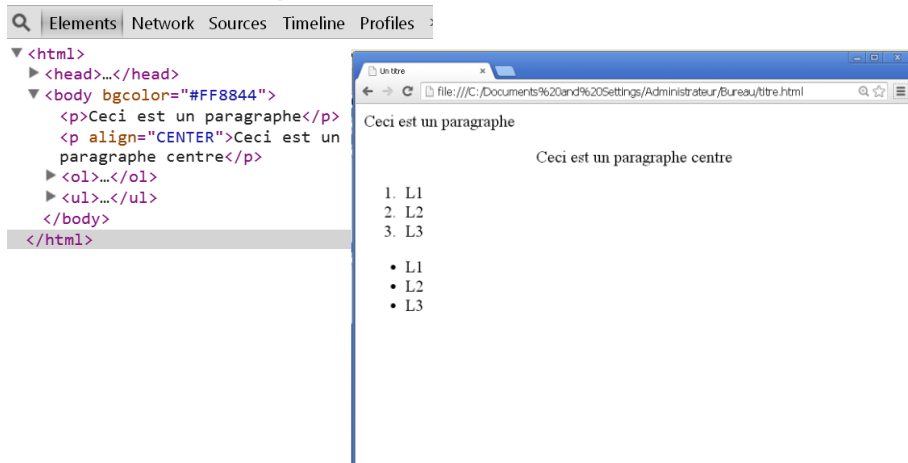
Paragraphe préformaté :

`<PRE [WIDTH=entier]>texte du paragraphe</PRE>`

43

## HTML / XHTML / CSS

### Balises de paragraphe : exemple (1)



44

## HTML / XHTML / CSS

### Balises de formats de caractères

`<b>`séquence de caractères`</b>`

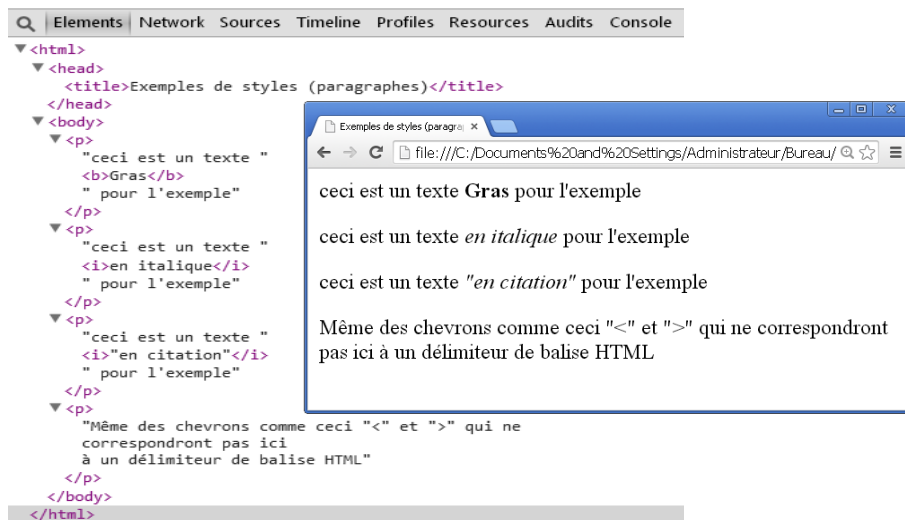
`<i>`séquence de caractères`</i>`

`<u>`séquence de caractères`</u>`

**gras**  
***italique***  
**souligné**

45

## HTML / XHTML / CSS



46

## HTML / XHTML / CSS

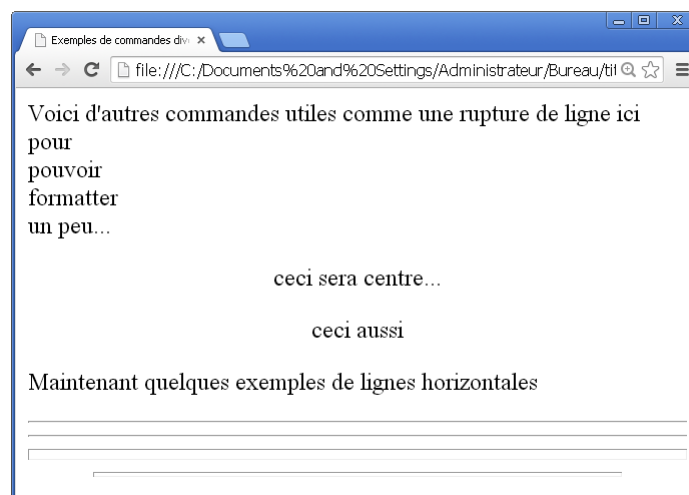
### Quelques commandes utiles :

```

Q Elements Network Sources Timeline Profiles Resources Audits Console
▼ <html>
  ▼ <head>
    <title>Exemples de commandes diverses</title>
  </head>
  ▼ <body>
    ▼ <p>
      "Voici d'autres commandes utiles comme une rupture de ligne
      ici"
      <br>
      "
      pour"
      <br>
      "pouvoir"
      <br>
      "formater"
      <br>
      " un peu..."
    </p>
    ► <center>...</center>
    <p>Maintenant quelques exemples de lignes horizontales</p>
    <hr>
    <hr align="CENTER">
    <hr size="8">
    <hr size="4" width="80%">
  </body>
</html>

```

## HTML / XHTML / CSS





## HTML / XHTML / CSS

### Eléments HTML/XHTML : Les exposants et indices

Les exposants et indices sont rarement utilisés mais peuvent s'avérer pratiques si vous avez l'intention d'écrire des formules mathématiques ou des abréviations.

- `<sup></sup>` : mise en exposant (caractères surélevés).
- `<sub></sub>` : mise en indice (caractères abaissés).

On peut utiliser la mise en exposant pour indiquer une date par exemple :

```
<p>Bienvenue dans le XXI<sup>ème</sup> siècle !</p>
<p>L'eau est notée H<sub>2</sub>O en chimie.</p>
```

Le résultat de ce code source apparaît ci dessous.

Bienvenue dans le XXI<sup>ème</sup> siècle !

L'eau est notée H<sub>2</sub>O en chimie.

49

## HTML / XHTML / CSS

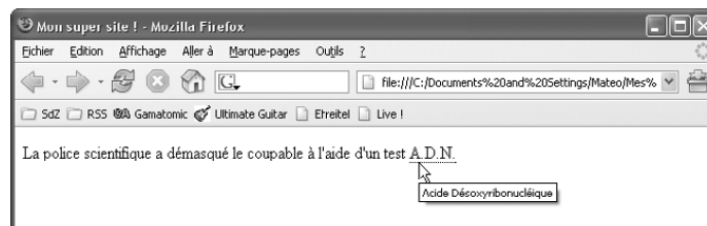
### Eléments HTML/XHTML : Les abréviations

La balise `<abbr>` permet de regrouper les acronymes, abréviations et sigles.

Prenons par exemple le terme ADN, contraction de « **a**cide **d**éoxyribo**n**ucléique » :

```
<p>La police scientifique a démasqué le coupable à l'aide d'un
test <abbr title="Acide Déoxyribonucléique">ADN</abbr></p>
```

L'attribut **title** sert à afficher une infobulle. Si vous survolez ADN avec la souris, une infobulle apparaîtra pour vous indiquer la signification du terme :



50

## HTML / XHTML / CSS

### Eléments HTML/XHTML : liens hypertextes

Les liens hypertextes sont l'invention géniale qui est à la base du Web. Aujourd'hui, tout le monde y est familier : quand on voit que le pointeur de sa souris se transforme en main, on sait qu'il s'agit d'un lien qui va nous amener sur une autre page

Rendez-vous sur mon site !

Il faut savoir qu'on a la possibilité de faire deux types de liens :

- Un lien vers une autre page : c'est le cas le plus courant, que nous verrons en premier.
- Un lien vers un autre endroit sur la même page : cela permet de se déplacer rapidement à différents endroits d'une même page, évitant ainsi à votre visiteur d'avoir à trop utiliser la molette de sa souris ou l'ascenseur.

51

## HTML / XHTML / CSS

### Eléments HTML/XHTML : liens hypertextes

Pour faire un lien, nous allons utiliser la balise XHTML suivante :

`<a></a>`. Vous devez mettre à l'intérieur le texte qui va servir de lien. Par exemple :

**`<a>Cliquez ici</a>` pour aller sur la page 2 !**

Toutefois, il manque encore quelque chose pour que notre lien soit correct : il faut indiquer le nom de la page vers laquelle amène le lien !

Nous allons rajouter un attribut à la balise `<a>` : l'attribut **href**. Cet attribut permet d'indiquer le nom de la page qui sera appelée lorsqu'on cliquera sur ce lien.

**`<a href="page2.html">Cliquez ici</a>` pour aller sur la page 2 !**



52

## HTML / XHTML / CSS

### Eléments HTML/XHTML : liens hypertextes

#### Ajout d'une infobulle

Il est possible d'ajouter une infobulle (ou bulle d'aide) sur votre lien. Cette bulle est un petit message qui s'affiche lorsque vous pointez avec la souris sur le lien.

```
<a href="livreor.html" title="Signez mon livre !">Livre d'or</a>
```



#### Un lien pour envoyer un e-mail

Ce type de lien se crée exactement de la même façon qu'un lien vers une page web, à la différence près que le contenu de l'attribut **href** doit cette fois commencer par le préfixe **mailto:** suivi de votre adresse e-mail.

```
<a href="mailto:moi@vivemoi.com">Envoyez-moi un message !</a>
```

53

## HTML / XHTML / CSS

### Eléments HTML/XHTML : liens hypertextes

#### Insérer un lien vers un autre endroit de la page

un lien en haut de la page qui indiquera au navigateur : « Si le visiteur clique sur ce lien, alors amène-le à l'endroit de la page où se trouve la partie désirée ».

Pour créer ce lien, il suffit de rajouter un attribut **id** à n'importe quelle balise. Par exemple, on peut rajouter une ancre à un titre

```
<h2 id="recettes">la partie désirée</h2>
```

Pour créer ce lien. Vous devrez utiliser la balise **<a>** mais vous indiquez l'attribut **href="#recettes"**, c'est-à-dire le nom de l'ancre précédé d'un dièse (#).

```
<a href="#recettes">Aller voir la partie désirée</a>
```

54

## HTML / XHTML / CSS

### Eléments HTML/XHTML : liens hypertextes

#### Les liens relatifs et absolus

**Les liens relatifs** : ils décrivent la position d'un fichier cible par rapport au fichier actuel. Ces liens s'adaptent très bien aux changements d'adresse précisément car ils ne contiennent pas l'adresse du site. Il est donc conseillé d'utiliser des liens relatifs lorsque vous faites des liens internes, c'est-à-dire des liens pointant vers d'autres pages de votre site.

```
<a href="page.html">Lien vers une page de mon site</a>
```

**Les liens absolus** : ils décrivent la position d'un fichier cible de manière absolue. Cela signifie qu'on peut faire un lien vers un autre site que le sien. Leur défaut vient du fait qu'il faille connaître l'adresse complète du site en question

```
<a  
href="http://www.autresite.com/page.html">Lien  
vers une page d'un autre site</a>
```

55

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Insérer une image dans une page

#### La balise d'image

#### Comment insérer une image dans une page web ?

Grâce à la balise `<img />`, qui est auto-fermante comme vous pouvez le voir. Cette balise doit être insérée dans un paragraphe (`<p></p>`), elle ne peut pas être placée à l'extérieur.

Les attributs suivants sont obligatoires :

**src** : l'URI où se situe l'image

**alt** : courte description de l'image

```

```

cette balise doit être placée à l'intérieur d'un paragraphe. Voici un code plus complet respectant cette règle.

```
<p>Voici une photo de mes dernières vacances en Espagne à Moraira :</p>  
<p></p>
```

56

## HTML / XHTML / CSS

Eléments HTML/XHTML : Insérer une image dans une page

Ajouter une infobulle

L'attribut à utiliser est en fait le même (et il fonctionne d'ailleurs sur un grand nombre de balises) : il s'agit de **title**. Voici un exemple d'utilisation :

```
<p></p>
```



57

## HTML / XHTML / CSS

Eléments HTML/XHTML : Insérer une image dans une page

Créer une image cliquable

Comment faire ?

C'est extrêmement simple, il suffit de combiner les balises **<a>** et **<img />** !

```
<p>
<a href="http://www.m.org">

</a></p>
```



58

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Les tableaux

Les balises d'un tableau doivent être imbriquées dans le bon ordre. Si vous faites une erreur, il est probable que votre tableau n'apparaisse pas du tout comme vous l'aviez imaginé !

Tout d'abord, il faut savoir qu'un tableau se place à l'intérieur des balises `<table></table>`.

les tableaux doivent être à l'extérieur des balises de paragraphe `<p></p>` et non à l'intérieur, car un tableau n'est pas un paragraphe :

```
<p>Paragraphe avant le tableau.</p>
<table></table>
<p>Paragraphe après le tableau.</p>
```

Un tableau se crée ligne par ligne. Chaque ligne est représentée par une paire de balises `<tr></tr>`. Si on veut créer un tableau de 3 lignes, on doit donc écrire :

```
<table>
  <tr></tr>
  <tr></tr>
  <tr></tr>
</table>
```

59

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Les tableaux

Maintenant, il faut découper chaque ligne en plusieurs cellules. On crée une cellule à l'aide de la balise `<td></td>`.

```
<table>
  <tr>
    <td>Première cellule</td>
    <td>Deuxième cellule</td>
  </tr>
  <tr>
    <td>Troisième cellule</td>
    <td>Quatrième cellule</td>
  </tr>
  <tr>
    <td>Cinquième cellule</td>
    <td>Sixième cellule</td>
  </tr>
</table>
```

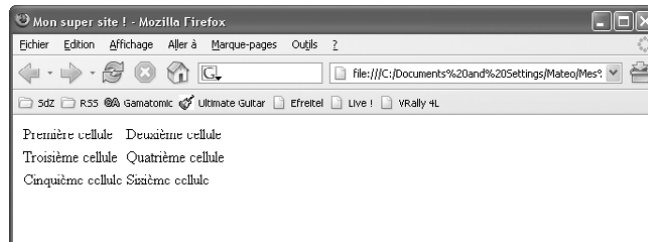
Première cellule	Deuxième cellule
Troisième cellule	Quatrième cellule
Cinquième cellule	Sixième cellule

60

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Les tableaux

Testons dans le navigateur pour voir le résultat.



En effet, le tableau n'a aucune bordure par défaut. Il va falloir utiliser le CSS pour en dessiner une.

61

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Les en-têtes de tableaux

En plus de la balise `<td>`, il existe une balise `<th>` qui permet de créer des cellules servant d'en-tête au tableau. En général, on place ces balises sur la première ligne.

```
<table>
  <tr>
    <th>Pays</th>
    <th>Capitale</th>
  </tr>
  <tr>
    <td>France</td>
    <td>Paris</td>
  </tr>
  <tr>
    <td>Espagne</td>
    <td>Madrid</td>
  </tr>
  <tr>
    <td>Royaume-Uni</td>
    <td>Londres</td>
  </tr>
</table>
```



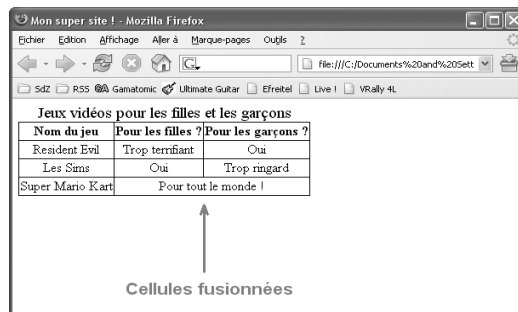
62

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Fusionner des cellules

Pour qu'une cellule fusionne avec d'autres colonnes, on utilise l'attribut colspan de la balise <td>.

```
<table>
<caption>Jeux vidéos pour les filles et les garçons</caption>
<tr>
<th>Nom du jeu</th>
<th>Pour les filles ?</th>
<th>Pour les garçons ?</th>
</tr>
<tr>
<td>Resident Evil</td>
<td>Trop terrifiant</td>
<td>Oui</td>
</tr>
<tr>
<td>Les Sims</td>
<td>Oui</td>
<td>Trop ringard</td>
</tr>
<tr>
<td>Super Mario Kart</td>
<td colspan="2">Pour tout le monde !</td>
</tr>
</table>
```



63

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Fusion de lignes

La fusion de lignes fonctionne exactement de la même manière que la fusion de colonnes. On utilise cette fois l'attribut rowspan (row signifiant « ligne » en anglais).

```
<table>
<caption>Jeux vidéos pour les filles et les garçons</caption>
<tr>
<th>Nom du jeu</th>
<td>Resident Evil</td>
<td>Les Sims</td>
<td>Super Mario Kart</td>
</tr>
<tr>
<th>Pour les filles ?</th>
<td>Trop terrifiant</td>
<td>Oui</td>
<td rowspan="2">Pour tout le monde !</td>
</tr>
<tr>
<th>Pour les garçons ?</th>
<td>Oui</td>
<td>Trop ringard</td>
</tr>
</table>
```



64



## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire

#### La balise de formulaire

Un formulaire se place à l'intérieur des balises **<form></form>**. Ces balises doivent être situées à l'extérieur des paragraphes, comme ceci :

```
<p>Paragraphe avant le formulaire.</p>
<form>
</form>
<p>Paragraphe après le formulaire.</p>
```

#### L'envoi des données

Une fois qu'un visiteur aura rempli le formulaire, il cliquera sur le bouton Envoyer. Mais où ces informations vont-elles être envoyées et par quel moyen ?

**method** : cet attribut permet d'indiquer la méthode d'envoi des données, voici les deux méthodes possibles : **Post & Get**

65

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire

- **action** : indique l'adresse de la page vers laquelle doit être redirigé le visiteur une fois qu'il a cliqué sur le bouton Envoyer. Cette page doit être une page de script et donc comporter par exemple l'extension .php .asp etc

```
<form method="post" action="traitement.php">
</form>
```

66

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire

#### Zone de texte monoligne

La création d'une zone de texte à une seule ligne requiert la balise `<input />` : c'est une balise auto-fermante

#### Attributs obligatoires

- **Type** : Une zone de texte monoligne se construit avec l'attribut `type` auquel il faut donner la valeur `text`.
- **name** est un attribut particulièrement important (et que vous réutiliserez pour tous les éléments du formulaire). Il permet de donner un nom à la zone de saisie, ce qui vous permettra plus tard de l'identifier.

```
<form method="post" action="traitement.php">
  <p><input type="text" name="pseudo" /></p>
</form>
```

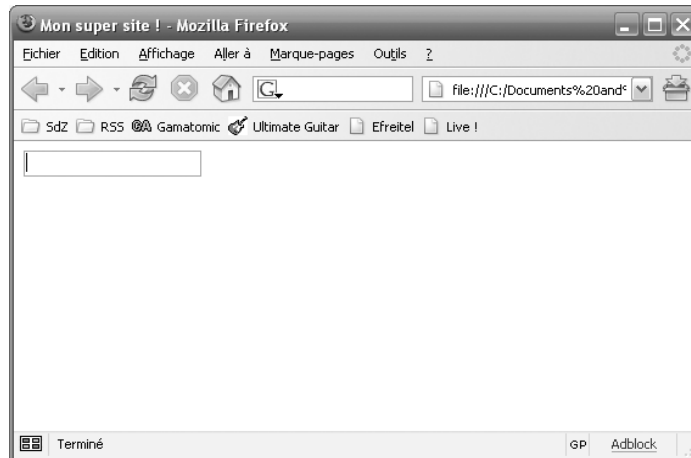
#### Astuce Masquer les caractères des mots de passe

Pour ce faire, il suffit simplement de donner la valeur `password` à l'attribut `type`, comme ceci : `<input type="password" />`

67

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire



68

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire

#### Attributs facultatifs

En plus des attributs indispensables que nous venons de voir, il y a quelques autres attributs qui vous intéresseront probablement. Voici les plus utiles d'entre eux :

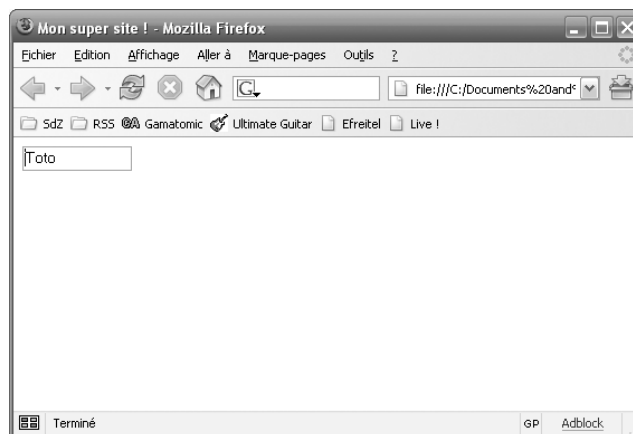
- **value** : indique la valeur par défaut du champ. La zone de texte sera préremplie avec la valeur de votre choix. Par exemple : `value="Votre pseudonyme"`.
- **size** : indique la largeur du champ. Par exemple : `size="12"`.
- **maxlength** : indique le nombre maximal de caractères

```
<form method="post" action="traitement.php">
  <p>
    <input type="text" name="pseudo" size="12" value="Toto" />
  </p>
</form>
```

69

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire



70

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire

#### Créer un libellé

Un libellé est un court texte à placer entre des balises `<label></label>`. Il permet d'indiquer au visiteur ce que doit contenir le champ

```
<form method="post" action="traitement.php">
  <p>
    <label for="pseudo">Entrez votre pseudonyme :</label>
    <input type="text" name="pseudo" id="pseudo" value="Toto" />
  </p>
</form>
```

l'attribut **for** du `<label>` doit être le même que l'id de l'`<input />`. L'ordinateur saura ainsi que le libellé correspond au champ de texte pseudo.

71

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire



72

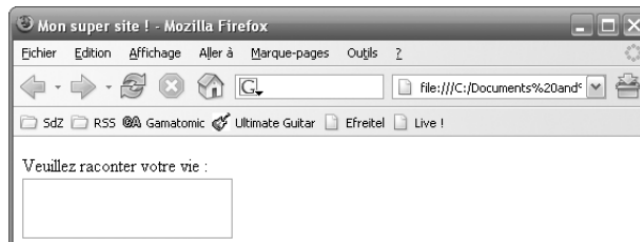
## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire

#### Zone de texte multiligne

La création d'une zone de texte multiligne requiert la balise `<textarea>`.

```
<form method="post" action="traitement.php">
  <p>
    <label for="votre_vie">Veuillez raconter votre vie :</label>
    <br />
    <textarea name="votre_vie" id="votre_vie"></textarea>
  </p>
</form>
```



73

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire

#### Zone de texte multiligne

on peut ajouter les attributs `rows` et `cols` qui indiquent respectivement le nombre de lignes et le nombre de colonnes dont le champ est constitué.

```
<form method="post" action="traitement.php">
  <p>
    <label for="votre_vie">Veuillez raconter votre vie :</label>
    <textarea name="votre_vie" rows="10" cols="45"></textarea>
  </p>
</form>
```



74

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire

#### Les options

Il existe trois types d'options :

- les cases à cocher
- les zones d'options
- les listes déroulantes

#### Les cases à cocher

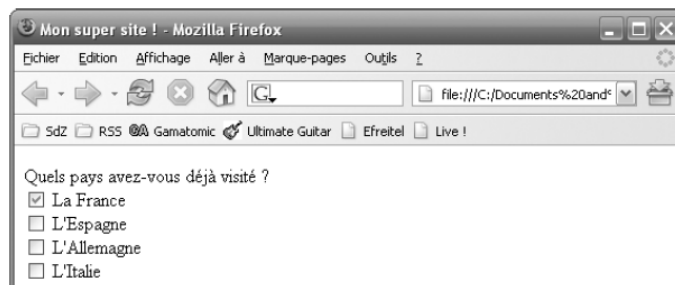
On utilise là encore la balise `<input />`. Mais nous devons changer son attribut `type`. Cette fois, on lui donnera la valeur `checkbox`.

```
<form method="post" action="traitement.php">
  <p>
    Quels pays avez-vous déjà visités ?<br />
    <input type="checkbox" name="france" id="france"
checked="checked" />
    <label for="france">La France</label><br />
    <input type="checkbox" name="espagne" id="espagne" />
    <label for="espagne">L'Espagne</label><br />
    <input type="checkbox" name="allemagne" id="allemagne" />
    <label for="allemagne">L'Allemagne</label><br />
    <input type="checkbox" name="italie" id="italie" />
    <label for="italie">L'Italie</label>
  </p>
</form>
```

75

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire



76

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire

#### Les zones d'options

Les zones d'options sont un peu plus délicates à gérer car elles fonctionnent par groupe. En effet, contrairement aux cases à cocher, on ne peut choisir qu'une seule valeur par groupe.

Pour créer un bouton d'option simple : `<input type="radio" />`

En effet, les éléments d'un même groupe doivent tous avoir le même nom : l'attribut **name** doit avoir une valeur identique.

#### Comment faire alors pour différencier les éléments ?

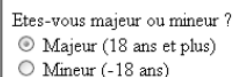
cela se fera grâce à l'attribut **value**.

77

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire

```
<form>
  <p>
    Etes-vous majeur ou mineur ?<br />
    <input type="radio" name="age" value="majeur" id="majeur" />
    <label for="majeur">Majeur (18 ans et plus)</label><br />
    <input type="radio" name="age" value="mineur" id="mineur" />
    <label for="mineur">Mineur (-18 ans)</label>
  </p>
</form>
```



En résumé, le fait de classer les boutons d'options par groupes permet d'éviter qu'ils interfèrent entre eux.

78

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire

#### Les listes déroulantes

Les listes déroulantes fonctionnent sur le même principe que les zones d'options : le visiteur doit faire un choix (et un seul) parmi une liste de possibilités.

Une liste déroulante se définit à l'intérieur d'une balise `<select></select>`.

À l'intérieur de cette balise, vous devez placer les différentes valeurs possibles. Pour ajouter une valeur possible, utilisez la balise `<option></option>`. Vous donnerez un nom à la valeur à l'aide de l'attribut `value`.

79

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire

```
<form method="post" action="traitement.php">
<h1>Agence de voyages Tourisk</h1>
  <p>
    <label for="destination">Où souhaitez-vous partir en vacances cet été ?</label>
    <br />
    <select name="destination" id="destination">
      <option value="papouasie">En Papouasie-Nouvelle-Guinée</option>
      <option value="fidji">Aux Îles Fidji</option>
      <option value="creuse">Dans la Creuse</option>
    </select>
  </p>
</form>
```

80



## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire



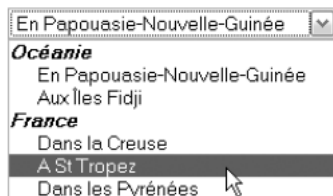
Les listes déroulantes sont utiles lorsqu'il y a beaucoup de valeurs possibles et que vous ne voulez pas prendre trop de place.

81

## HTML / XHTML / CSS

### Eléments HTML/XHTML : Formulaire

#### Une liste déroulante avec des groupes



Il est possible de regrouper ces options entre elles à l'aide de la balise `<optgroup>`/`</optgroup>`. L'attribut **label** permet de donner un nom au groupe d'options. Attention, cette fois, le nom du groupe sera visible par le visiteur

```
<select name="destination" id="destination">
  <optgroup label="Océanie">
    <option value="papouasie">En Papouasie-Nouvelle-Guinée</option>
    <option value="fidji">Aux îles Fidji</option>
  </optgroup>
  <optgroup label="France">
    <option value="creuse">Dans la Creuse</option>
    <option value="sttropez">A St Tropez</option>
    <option value="pyrenees">Dans les Pyrénées</option>
  </optgroup>
</select>
```

82

## HTML / XHTML / CSS

### CSS : Introduction

CSS (*Cascading Style Sheets* : feuilles de style en cascade) est un langage informatique qui sert à décrire la présentation des documents HTML et XML.

La première question que vous devez vous poser est :  
où doit-on placer le code CSS ?

Dans le fichier .html, entre des balises spéciales ou dans  
un autre fichier ?

### La réponse

vous avez le choix ! En effet, il y a 3 façons différentes d'insérer du code CSS. Certaines sont plus ou moins adaptées selon le cas. Nous allons étudier chaque méthode.

83

## HTML / XHTML / CSS

### CSS : Introduction

La technique la plus courante (et la plus conseillée !) consiste à créer un fichier spécial pour placer le code CSS.

On peut appliquer un style à toutes les balises, que ce soit des paragraphes <p>, des titres <h1>, des citations <q>, etc.

**Exemple** : Les accolades délimitent le style. Tout le code CSS qui se trouve à l'intérieur sera appliqué à toutes les balises.

```
*
{
  text-align: center;
}
```

84

## HTML / XHTML / CSS

### CSS : Introduction

```

<!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" xml:lang="fr" >
<head>
<title>Tests du langage CSS</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<link rel="stylesheet" media="screen" type="text/css" title="Mon style" href="style.css" />
</head>
<body>
<h1>Bienvenue sur mon site web !</h1>
<h2>Présentation</h2>
<p> Ceci est le site de Jean Suiffaire, un passionné de Volley Ball. </p>
<p>
    Revenez dans quelques jours, il est encore en construction ! Vous saurez bientôt tout sur
    les règles du Volley et pourrez suivre au jour le jour les résultats du championnat du monde !<br />
    En attendant, je fais quelques tests avec le langage
    CSS...
  </p> </body> </html>

```

85

## HTML / XHTML / CSS

### CSS : Introduction



86

## HTML / XHTML / CSS

### CSS : Introduction

**En résumé** : il faut séparer la propriété de sa valeur par le symbole deux points (:) et terminer par un point-virgule (;) après la valeur.

Schématiquement, notre code CSS devrait donc toujours ressembler à ceci :

```
*
{
  propriete: valeur;
  propriete: valeur;
  propriete: valeur;
}
```

87

## HTML / XHTML / CSS

### CSS : Appliquer un style à une balise

Si nous voulons centrer nos titres **<h1>** uniquement, il nous faudra donc écrire :

```
h1
{
  text-align: center;
}
```



88

## HTML / XHTML / CSS

### CSS : Appliquer un style à plusieurs balises

Il est aussi possible d'appliquer un même style à plusieurs balises à la fois. Pour faire cela, il suffit de séparer les noms des balises par des virgules.

Par exemple, si nous voulons centrer nos titres `<h1>` et `<h2>` :

```
h1, h2
{
    text-align: center;
}
```



89

## HTML / XHTML / CSS

### CSS : Appliquer un style à des balises imbriquées

### Qu'est-ce qu'une balise imbriquée ?

C'est une balise qui en contient une autre. Petit exemple :

```
<blockquote>
<p>
    Le voile du matin sur les monts se déploie.<br />
    Vois, un rayon naissant blanchit la vieille tour ;<br />
    Et déjà dans les cieux s'unit avec amour,<br />
    Ainsi que la gloire à la joie,<br />
    Le premier chant des bois aux premiers feux du jour.
</p>
</blockquote>
```

Vous pouvez créer une règle CSS qui ne centrera que les paragraphes situés à l'intérieur de balises `<blockquote>`.

Il suffit d'écrire les balises dans cet ordre précis en les séparant par un espace :

```
blockquote p
{
    text-align: center;
}
```

90

## HTML / XHTML / CSS

### CSS : Appliquer un style à des balises imbriquées

On peut même faire une triple imbrication si on veut :

```
blockquote p strong
{
  text-align: center;
}
```

Cette fois, cela signifie « Centrer tous les textes importants (strong) situés dans des paragraphes (p) eux-mêmes situés dans des citations longues (blockquote) ».

91

## HTML / XHTML / CSS

### CSS : Appliquer un style à certaines balise

id : un identifiant unique

**Comment faire pour changer l'apparence d'un seul paragraphe en particulier ?**

En fait, il s'agit d'identifier un paragraphe précis en CSS. Et pour cela, il faut donner un nom à la balise qui nous intéresse.

```
<p id="introduction">Ceci est le site de Jean Suiffaire, un
passionné de Volley Ball.</p>
```

Pour le code CSS :

```
#introduction
{
  text-align: center;
}
```

92

## HTML / XHTML / CSS

### CSS : Appliquer un style à certaines balise



93

## HTML / XHTML / CSS

### CSS : class (un identificateur réutilisable)

Le défaut de l'identificateur id, c'est qu'on ne peut l'utiliser qu'une seule fois. Il donne un nom unique.

Heureusement, il existe un autre attribut, lui aussi utilisable sur la plupart des balises. Il s'agit de class.

```
<h1 class="maclasse">Bienvenue sur mon site web !</h1>
<h2>Présentation</h2>
<p class="maclasse">Ceci est le site de Jean Suiffaire, un
passionné de Volley Ball.</p>
```

Le code CSS :

```
.maclasse
{
    text-align: center;
}
```

94

## HTML / XHTML / CSS

### CSS : class ( un identificateur réutilisable )



95

## HTML / XHTML / CSS

### CSS : Les commentaires

Il est possible d'insérer des commentaires en CSS.

```
/* Tous les paragraphes situés dans des citations longues
seront centrés */
blockquote p
{
    text-align: center;
}
```

96



## HTML / XHTML / CSS

### CSS : Mise en forme de texte

L'alignement : Aligner tout le texte

```
h1
{
  text-align: center;
}
```

→

- left : à gauche ;
- center : centré ;
- right : à droite ;
- justify : justifié.

```
p
{
  text-align: justify;
}
```

la valeur justify permet d'obliger le texte à prendre la même largeur sur chaque ligne. Cet alignement est très utilisé dans le monde de l'imprimé.

### Les alinéas

Par exemple, pour un alinéa de 30 pixels, vous devrez taper le code suivant :

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

Le Lorem Ipsum est simplement du faux texte employé dans la composition et la mise en page avant impression. Le Lorem Ipsum est le faux texte standard de l'imprimerie depuis les années 1500, quand un peintre anonyme assembla ensemble des morceaux de texte pour réaliser un livre spécimen de polices de texte.

97

## HTML / XHTML / CSS

### CSS : Mise en forme de texte

#### Les polices

Le langage CSS nous permet de changer la police du texte. Pour le moment, c'est une police par défaut qui est utilisée.

Arial  
**Arial Black**  
 Comic Sans MS  
 Courier New  
 Georgia  
**Impact**  
 Times New Roman  
 Trebuchet MS  
 Verdana

La propriété CSS précisant la police est **font-family**. Si vous voulez que vos paragraphes de texte s'affichent en **Verdana**, vous taperez le code suivant :

```
p
{
  font-family: Verdana;
  text-align: justify;
}
```

#### Proposer plusieurs polices

En règle générale, on n'indique jamais un seul nom de police. On en propose plusieurs, que l'on sépare par des virgules :

```
p
{
  font-family: Elephant, "Arial Black", Georgia, Verdana;
}
```

98

## HTML / XHTML / CSS

### CSS : Mise en forme de texte

#### La taille du texte

Par exemple, si vous souhaitez obtenir une taille de 18 pixels, vous écrirez :

```
p
{
    font-size: 18px;
}
```

- `xx-small` : minuscule ;
- `x-small` : très petit ;
- `small` : petit ;
- `medium` : moyen ;
- `large` : grand ;
- `x-large` : très grand ;
- `xx-large` : mmh... énorme ?

Texte en xx-small

Texte en x-small

Texte en small

Texte en medium

Texte en large

Texte en x-large

Texte en xx-large

99

## HTML / XHTML / CSS

### CSS : Mise en forme de texte

#### Gras, italique, souligné, etc.

Nous allons voir ici comment réaliser les effets suivants :

- mise en gras ;
- mise en italique ;
- mise en majuscules/minuscules ;
- soulignement, clignotement...

#### Mise en gras

Pour mettre en gras du texte grâce au CSS, on utilise la propriété **fontweight**. Celle-ci peut prendre 2 valeurs :

- **bold** : le texte sera écrit en gras.
- **normal** : le texte sera écrit normalement (valeur par défaut).

```
q, blockquote
{
    font-weight: bold;
}
```

100

## HTML / XHTML / CSS

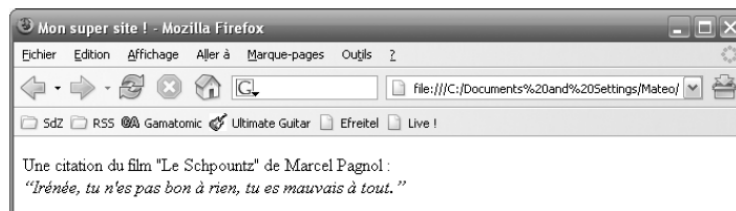
### CSS : Mise en forme de texte

#### Mise en italique

En CSS, l'italique est appliqué par le biais de la propriété `font-style`. Cette balise peut prendre 3 valeurs :

- `italic` : le texte sera mis en italique.
- `oblique` : le texte sera penché de manière oblique.
- `normal` : texte normal (valeur par défaut).

```
q, blockquote
{
  font-style: italic;
}
```



101

## HTML / XHTML / CSS

### CSS : Mise en forme de texte

#### Mise en majuscules/minuscules

La propriété `text-transform` propose plusieurs valeurs intéressantes :

- `uppercase` : le texte sera écrit en majuscules.
- `lowercase` : le texte sera écrit en minuscules.
- `capitalize` : la première lettre de chaque mot sera écrite en majuscule (usage anglo-saxon).
- `none` : le texte ne sera pas modifié (valeur par défaut).

```
q, blockquote
{
  text-transform: capitalize;
}
```



102

## HTML / XHTML / CSS

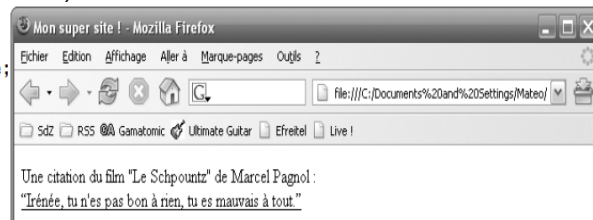
### CSS : Mise en forme de texte

#### Souligner, barrer, faire clignoter

La propriété **text-decoration** permet de décorer son texte de plusieurs façons différentes :

- **underline** : souligné ;
- **line-through** : barré ;
- **overline** : ligne au-dessus ;
- **blink** : clignotant. Attention, cette propriété ne fonctionne pas sur Internet Explorer ;
- **none** : normal (valeur par défaut).

```
q, blockquote
{
  text-decoration: underline;
}
```



103

## HTML / XHTML / CSS

### CSS : Couleur et fond

Le langage CSS vous offre un large choix de couleurs et il ne tient qu'à vous de les utiliser au mieux.

la propriété CSS qui permet de changer la couleur du texte est : **color**.

```
p
{
  color: votre_couleur;
}
```

#### Indiquer la couleur avec un nom

En effet, seules 16 couleurs possèdent un nom. Les noms des couleurs doivent être écrits en anglais.

104

## HTML / XHTML / CSS

### CSS : Couleur et fond

Couleur	Traduction	Couleur	Traduction
white	Blanc	yellow	Jaune
silver	Argent (gris léger)	olive	Olive (jaune foncé)
gray	Gris (gris foncé)	aqua	Bleu clair
black	Noir	blue	Bleu
red	Rouge	navy	Marine (bleu foncé)
maroon	Marron	fuchsia	Fuchsia (rose)
lime	Citron vert (vert clair)	purple	Violet
green	Vert	teal	Bleu-vert

Si nous désirons mettre les textes importants (**<strong>**) en rouge pour qu'on ne puisse pas les manquer, nous pouvons les écrire en rouge :

```
strong
{
  color: red;
}
```

105

## HTML / XHTML / CSS

### CSS : Mise en forme de texte

#### Indiquer la couleur en RGB

RGB est l'abréviation de « Red, Green, Blue », soit « Rouge, Vert, Bleu » en anglais. C'est une notation pour désigner les couleurs.

```
strong
{
  color: rgb(247, 128, 40);
}
```

Si on écrit :

```
rgb(255, 0, 0);
```

Cela représente une couleur rouge. En effet, il y a 255 (le maximum) de rouge, et 0 pour le vert et 0 pour le bleu.

106

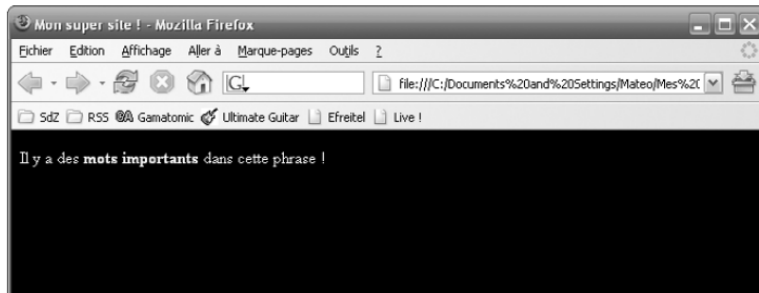
## HTML / XHTML / CSS

### CSS : Mise en forme de texte

#### La couleur de fond de la page

Pour appliquer une couleur de fond à la page, il suffit de modifier le style de la balise qui englobe tout le contenu de la page, c'est-à-dire la balise **<body>** :

```
body
{
    color: white;
    background-color: black;
}
```



107

## Plan du cours

- ◆ Introduction
- ◆ Serveur Web et protocole HTTP
- ◆ HTML / XHTML / CSS
- ◆ Javascript
- ◆ HTML 5

108

## JAVASCRIPT

### JavaScript : Introduction

#### Description :

- Créé en 1995 par Netscape et Sun Microsystems
- **But** : interactivité dans les pages HTML/XHTML, traitements simples sur le poste de travail de l'utilisateur
- **Moyen** : introduction de scripts dans les pages HTML/XHTML
- **Norme** : <http://www.ecma-international.org/publications/standards/Ecma-262.htm>

#### Programmation locale :

- **sans javascript** : programmation exécutée sur le serveur
- **avec javascript** : inclusion de programmes dans les pages HTML/XHTML afin de les exécuter sur le poste client

109

## JAVASCRIPT

### JavaScript : Introduction

- ◆ Insertion d'instructions de programmation directement dans le code des pages HTML/XHTML
- ◆ Exécution de code sur le poste client pour
  - améliorer l'interactivité (temps de réponse plus court)
  - améliorer les débits des réseaux (éviter les envois erronés, limiter les requêtes)
  - proposer des pages dynamiques (animation, personnalisation)
- ◆ Exemples :
  - test d'un formulaire avant envoi
  - animation type texte défilant
  - affichage dynamique
- ◆ Javascript ≠ Java !

110

# JAVASCRIPT

## JavaScript : Comparaison Javascript/Java

### Comparatif

#### Javascript

- interprété
- à base d'objets prédéfinis (sans héritage)
- code intégré (visible)
- typage faible
- inexistant hors Web
- débogage difficile

#### Java

- compilé
- orienté objets (classes, héritage)
- code dans applets (invisible)
- typage fort
- langage à part entière
- environnements de développement

NB : Communication possible entre Java et Javascript grâce au plug'in LiveConnect (Netscape) ou aux contrôles Active X (Microsoft)

111

# JAVASCRIPT

## JavaScript : Avantages et inconvénients

### ◆ Points forts :

- langage de programmation structurée
- enrichit le HTML/XHTML (intégré ) interprété par le client),
- Javascript partage les mêmes objets DOM (Document Object Model) que ceux de document HTML/XHTML -> manipulation dynamique possible
- gestionnaire d'événements

### ◆ Limitations :

- ce n'est pas un langage de programmation à part entière, indépendant
- c'est un langage de script, dépendant de HTML/XHTML
- ce n'est pas véritablement un langage orienté objet (ni héritage, ni polymorphisme)

Le Document Object Model (ou DOM) est une recommandation du W3C qui décrit une interface indépendante de tout langage de programmation et de toute plate-forme, permettant à des programmes informatiques et à des scripts d'accéder ou de mettre à jour le contenu, la structure ou le style de documents XML.

112



## JAVASCRIPT

### JavaScript : Domaines d'application

#### ◆ Javascript permet :

- de programmer des actions en fonction d'évènements utilisateurs (déplacements de souris, focus, etc.)
- d'accéder aux éléments de la page HTML/XHTML (traitement de formulaire, modification de la page)
- effectuer des calculs sans recours au serveur

#### ◆ Domaines d'application :

- petites applications simples (calculatrice, conversion, calendrier, etc.)
- aspects graphiques de l'interface (évènements, fenêtrage, etc.)
- tests de validité sur des interfaces de saisie

113

## JAVASCRIPT

### JavaScript : code Javascript

#### ◆ Utilisation de la balise `<script>...</script>` :

- déclaration de fonctions dans l'en-tête HTML/XHTML (entre `<head>` et `</head>`)
- appel d'une fonction ou exécution d'une commande Javascript dans `<body>...</body>`
- insertion d'un fichier externe (usuellement '.js')

#### ◆ Utilisation dans une URL, en précisant le protocole Ex : `<a href="javascript:instructionJavascript;">Texte</a>`

#### ◆ Utilisation des attributs de balise pour la gestion événementielle : `<balise onEvenement="instructionJavascript">...</balise>`

114

# JAVASCRIPT

## JavaScript : Inclusion dans le code

### Compatibilité avec la majorité des navigateurs

- En HTML4 :

```
<script type="text/javascript">
// <!--
Code Javascript
// -->
</script>
```

- En XHTML :

```
<script type="text/javascript">
// <![CDATA[
Code Javascript
// ]]>
</script>
```

- Code alternatif :

```
<noscript>
Message à afficher en cas d'absence de Javascript
</noscript>
```

115

# JAVASCRIPT

## JavaScript : Syntaxe générale

### Caractéristiques

- ◆ Description de la syntaxe
  - Variables faiblement typées
  - Opérateurs et instructions identiques au C/C++/Java
  - Des méthodes
    - globales (associées à tous les objets)
    - fonctions définies par l'utilisateur
  - Des "objets" (des prototypes)
    - prédéfinis (String, Date, Math, etc.)
    - liés à l'environnement
    - définis par l'utilisateur
  - Commentaires : // ou /\*...\*/
  - Séparateur d'instruction : `;`

116

## JAVASCRIPT

### JavaScript : Syntaxe générale

#### Opérateurs

- ◆ Opérateurs identiques à ceux du C/C++/Java
- ◆ opérateurs arithmétiques : + - \* / %
- ◆ in/décrémentation : var++ var-- ++var --var
- ◆ opérateurs logiques : && || !
- ◆ comparaisons : == === != !== <= < >= >
- ◆ concaténation de chaîne de caractères : +
- ◆ affectation : = += -= \*= ...

117

## JAVASCRIPT

### JavaScript : Syntaxe générale

#### Variables

- ◆ Utilisation de variables
  - Déclaration : var nom[=valeur];
    - déclaration optionnelle mais fortement conseillée
    - 'undefined' si aucune valeur à l'initialisation
    - aucun type !
  - Distinction de la localisation des variables (locale ou globale)
  - Sensible à la casse
  - Typage dynamique (à l'affectation) ) transtypage

#### Exemples

```
var philosophe      = "Diogene";
var anneeNaissance = -413;
var nonRien;       // vaut undefined
```

118

## JAVASCRIPT

### JavaScript : Syntaxe générale

#### Tests et boucles

##### Si-sinon-alors :

```
if(condition) {
  instructions
}
[else if {
  instructions
}]
[else {
  instructions
}]
```

##### Switch-case :

```
switch(variable) {
  case 'valeur1':
    Instructions
    break;
  ...
  default:
    Instructions
    break;
}
```

##### Boucles for :

```
for(i=0; i<N; i++) {
  Instructions
}

for(p in objet) {
  Instructions
}
```

##### Boucles while :

```
while(condition) {
  Instructions
}

do {
  Instructions
} while(condition);
```

119

## JAVASCRIPT

### JavaScript : Syntaxe générale

#### Fonctions et Procédures

##### Déclaration d'une fonction/procédure

```
function nom(arg1, ..., argN) {
  Instructions
  [return valeur;]
}
```

##### Remarques

- Arguments et valeur en retour non typés
- Nombre d'arguments non fixé par la déclaration
- Passage des paramètres par référence

120

## JAVASCRIPT

### JavaScript : Syntaxe générale

#### Les tableaux

- ◆ Déclaration :
  - `var nom = new Array([dimension]);`
  - `var nom = new Array(o1, ..., on);`
- ◆ Accession avec `[]` (`tableau[i]`)
  - les indices varient de 0 à N-1
  - les éléments peuvent être de type différent
  - la taille peut changer dynamiquement
  - les tableaux à plusieurs dimensions sont possibles
- ◆ Propriétés et méthodes : `length`, `reverse()`, `sort()`, `toString()`, `push(element)`, etc.

121

## JAVASCRIPT

### JavaScript : Syntaxe générale

#### Boîtes de dialogue et fenêtres

- ◆ Boîte de dialogue type "pop-up" : `window.alert("Message à afficher");`
- ◆ Boîte de saisie simple : `reponse = window.prompt("texte", "chaîne par défaut");`
- ◆ Ouverture d'une fenêtre fille : `fenetre = window.open("page", "titre");`

**Remarque : ces 3 fonctions sont des méthodes de la classe window.**

122

## JAVASCRIPT

### JavaScript : Programmation événementielle

#### Événements reconnus par Javascript

- ◆ onclick : un clic du bouton gauche de la souris sur une cible
- ◆ onmouseover : passage du pointeur de la souris sur une cible
- ◆ onblur : une perte de focus d'une cible
- ◆ onfocus : une activation d'une cible
- ◆ onselect : sélection d'une cible
- ◆ onchange : une modification du contenu d'une cible
- ◆ onsubmit : une soumission d'un formulaire
- ◆ onload : un chargement d'une page
- ◆ onunload : la fermeture d'une fenêtre ou le chargement d'une page
- ◆ .....

123

## JAVASCRIPT

### JavaScript : Programmation événementielle

#### Fonctionnement

- ◆ Le navigateur intercepte les événements (interruptions) et agit en conséquence
- ◆ Action → Événement → Capture → Action
- ◆ Actions associées aux cibles par les balises HTML : `<balise onEvenement="Action">`

#### Exemple

```
<a href="" onclick="alert('Merci!');">
```

124

## JAVASCRIPT

### JavaScript : Programmation événementielle

#### Événements liés à Window

- ◆ L'objet Window possède plusieurs méthodes spécifiques pour la gestion d'un compte à rebours :
  - **setTimeout**(instruction, temps) permet de spécifier un compteur de millisecondes associé à une instruction. Après l'intervalle de temps spécifié, une interruption est produite et l'instruction est évaluée.
  - **setInterval**(instruction, temps) permet de spécifier un compteur de millisecondes associé à une instruction. L'instruction est évaluée à intervalles réguliers.
  - **clearTimeout()** et **clearInterval()** annulent un compte à rebours.

#### Exemple

```
setTimeout("window.alert('Hello !');", 1000);
```

125

## JAVASCRIPT

### JavaScript : Programmation événementielle

#### Fonction wait

La fonction wait n'existe pas par défaut



Nécessité de la définir dans une fonction propre

#### Exemple

```
function wait(delay) {
  var date = new Date();
  var curDate = null;
  do {
    curDate = new Date();
  } while (curDate - date < delay);
}
```

126

## JAVASCRIPT

### JavaScript : Les objets

#### L'objet Date

##### ◆ Objet Date

- Pas de propriété
- Liste des méthodes :

<code>getDate ()</code>	<code>setDate ()</code>
<code>getDay ()</code>	
<code>getHours ()</code>	<code>setHours ()</code>
<code>getMinutes ()</code>	<code>setMinutes ()</code>
<code>getMonth ()</code>	<code>setMonth ()</code>
<code>getSeconds ()</code>	<code>setSeconds ()</code>
<code>getTime ()</code>	<code>setTime ()</code>
<code>getFullYear ()</code>	<code>setYear ()</code>
<code>getTimezoneOffset ()</code>	<code>...</code>

127

## JAVASCRIPT

### JavaScript : Les objets

#### L'objet String

- Lorsqu'on définit une constante ou une variable chaîne de caractères, Javascript crée d'une façon transparente une instance String
- 1 propriété : `length`
- Liste (non exhaustive) des méthodes :  
`bold ()` `italics ()` `fontcolor ()` `fontsize ()` `small ()` `big ()` `toUpperCase ()` `toLowerCase ()` `sub ()` `sup ()`  
`substring ()` `charAt ()` `indexOf ()` ...

#### Objet Math

- Propriétés : `Math.PI` et `Math.E`
- Méthodes :  
`atan ()` `acos ()` `asin ()` `tan ()` `cos ()` `sin ()` `abs ()` `exp ()`  
`max ()` `min ()` `pow ()` `round ()` `sqrt ()` `floor ()` `random ()` `log ()`

128



## Plan du cours

---

- ◆ Introduction
- ◆ Serveur Web et protocole HTTP
- ◆ HTML / XHTML / CSS
- ◆ Javascript
- ◆ HTML 5