

# Développement logiciel et Technologies Web

Ziguinchor Institut Polytechnique

Dut 2: ISD- RT

Année 2018 – 2019

Ismaïla Diallo

[cassoumaila@gmail.com](mailto:cassoumaila@gmail.com)

# Intégration dans la formation

- Objectif du module
  - Avoir les compétences nécessaires pour approfondir les principes d'une solution logicielle et les appliquer dans un environnement Web
- Compétences pédagogiques
  - Comprendre les bases du web (le vocabulaire, internet, web, architecture d'un site internet, ...)
  - Comprendre les bases du langage HTML
  - Comprendre les bases du langage PHP/MySQL.
  - Avoir les notions de base en programmation objet et utilisation de classe.
  - Procéder à l'hébergement et aux paramétrages.
- Prérequis:
  - Base de la programmation
  - Exploitation des données

# Comment ?

- Volume horaire : 45h
  - Cours Magistraux: 15h
  - Travaux Pratiques: 15h
  - Travaux Dirigés: 15h
- Évaluation
  - Contrôle sur table
  - Un projet à faire qui représentera votre note de TPE (travail personnel de l'étudiant) par binôme.

# Plan du cours

- Première partie
  - Les bases du Web
- Deuxième partie
  - Les bases du langage HTML
  - Les bases du Langage CSS
  - Les bases du langage PHP
- Troisième partie
  - Aller plus loin (Bootstrap)
  - Programmation orienté objet PHP

# Première Partie

- Les bases du Web
  - Internet
  - Web
  - Développement Web

# Que savez-vous d'internet?

# Que savez-vous du Web?

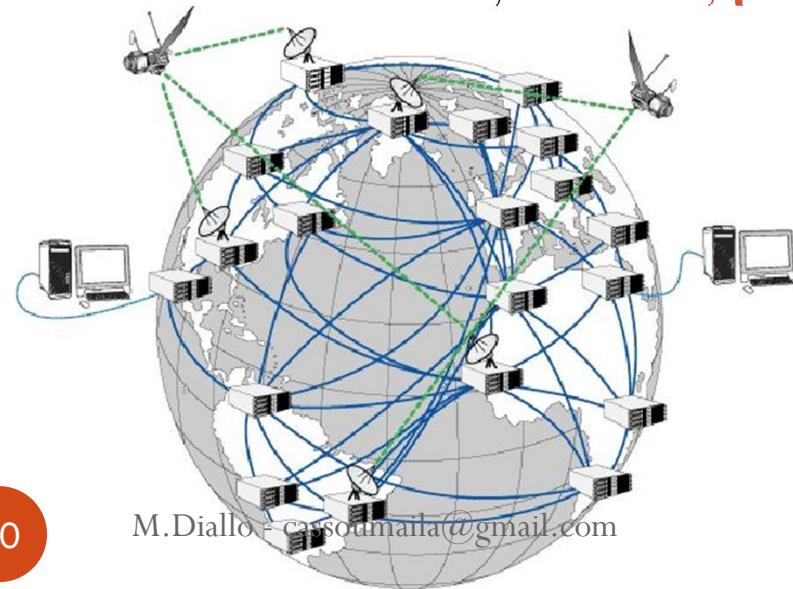
# L'internet et le Web

# L'internet

- Le terme internet est apparu en 1972 aux Etats Unis
- **L'internet est un réseau informatique mondial accessible au public.**
- Tous les réseaux (universités, réseaux privés etc.) sont connectés à internet
- Les ordinateurs, serveurs, imprimantes etc. d'internet ont des adresses uniques appelées **adresses IP** (192.162.1.1)

# LE WEB

- **World Wide Web (WWW) = web = toile** est une application d'internet
- D'autres applications sont l'email, la messagerie etc.
- Le web a été inventé par **Tim Berners-Lee en 1984** en Suisse (CERN)
- Le web est un système **hypertexte**



10

M.Diallo - casoumila@gmail.com



Sir Tim Berners-Lee, inventor of the worldwide web. Photograph: Sarah Lee for the Guardian

# Le Web

- *“I imagined the web as an open platform that would allow everyone, everywhere to share information, access opportunities, and collaborate across geographic and cultural boundaries.” Tim Berners-Lee*
- Les problèmes actuels:
  - Données privées
  - Fausses informations
  - Publicités

# Le Web

- Le Web est composé de pages Web sur des serveurs
- Chacune des pages est repérée par une adresse unique appelée URL (Uniform Ressource Locator) sa syntaxe est :
- **HTTP + :// + domaine + . + extension**
  - http (*HyperText Transfer Protocol*) est le protocole du web
  - https est la version sécurisée
- <http://zip-edu.sn>
- <http://facebook.com>
- <http://www.seneweb.com>
- <https://www.orange.sn>

# Exemple de page Web

The screenshot shows a web browser window with three tabs open:

- Télécom
- YouTube - mariah carey
- MySQL :: MySQL 5.0 Refer...

The main content area displays the YouTube search results for "mariah carey". The search bar contains "mariah carey". The results show approximately 111,000 videos, with the first few being official clips and others being fan-made content.

Key video entries include:

- Mariah Carey - Obsessed (clip officiel)**: Issu de l'album "Memoirs of an imperfect angel" (disponible). Duration 4:13, Views 702759 fois, Uploaded 3 months ago, UniversalMusicFrance.
- Mariah Carey on her new movie Precious'**: The diva talks about the latest film she is in with fellow singer Lenny Kravitz... Follow us on twitter at twitter.com. Duration 1:47, Views 93009 fois, Uploaded 6 months ago, En.
- Precious: Mariah Carey...**: Awards season is still a few months away, but one movie is already get... Duration 2:15, Views 2043 fois, tvguide.
- Mariah Carey Likes...**: www.mrpaparazzi.com Somewhere between landing at Heathrow airport and. Duration 2:56, Views 14198 fois, mrpaparazzi.
- Mariah Carey - I Want To Know What Love Is -**: Issu de l'album "Memoirs of an imperfect angel" (disponible). Duration 3:40, Views 101526 fois, MusicLionworld.
- Mariah Carey - Obsessed (clip officiel)**: Issu de l'album "Memoirs of an imperfect angel" (disponible). Duration 4:13, Views 702759 vues, Uploaded 3 mois, UniversalMusicFrance.
- Traduire Mariah Carey - Touch My Body**: Music video by Mariah Carey performing Touch My Body with Christopher Stewart [Producer], Terius Nash [Producer], Mariah Carey [Producer], Brett ... Duration 4:27, Views 33517760 vues, uploaded 1 an, universalmusicgroup.

The browser's toolbar includes "démarrer", "YouTube - mariah car...", "Mes Sites Web", and a search bar with the query "(02:06) [- 03:51]". The status bar at the bottom right shows "FR", "15/02/2019", and "15:13".

# Qu'est ce qu'une page Web?

- Une page Web est un document consultable (« texte » avec des images et des animations éventuelles)
- Une page Web peut être locale ou distante
  - Il est plus correct de parler de « Document HTML » que de « Page Web » mais l'usage fait parfois la loi...
- Une page Web est un fichier que le navigateur interprète en fonction de certains code ou « balises » (tags, marqueurs...)

# Où acquérir un nom de domaine ?

o <https://www.ovh.com/fr/>



The screenshot shows the OVH website homepage. At the top, there is a navigation bar with links for Cloud, Mobile Hosting, Web Hosting, Télécom, Programs, and À propos. Below the navigation bar, there is a banner featuring a man wearing glasses looking at a computer screen displaying a 3D model of a jet engine. The banner has a blue tint and contains the text "Serveurs dédiés GPU" and "Accélérez le traitement de vos calculs grâce au leader européen du cloud". It also includes a "Découvrir les offres GPU" button and a link to "Voir toute notre gamme de solutions cloud". In the bottom left corner, there is a yellow arrow pointing to a form field where you can enter a domain name. The form includes a text input field labeled "Saisir ici votre nom de domaine" and a blue "OK" button. Below the input field, there is a link "Renouveler votre nom de domaine". To the right of the form, there are two boxes: one for "Disques NVMe 450 Go Intel® P3520" with an image of a SSD, and another for "OVH Partners Tour Lyon". At the bottom right, there is a red footer bar with the OVH logo, a question mark icon, and the text "Besoin d'informations ?". The year "2019" is also visible in the bottom right corner.

# Raccourcir une adresse

- Certaines adresses web sont trop longues !
- On peut utiliser des raccourisseurs d'adresses
  - <http://bit.ly>
- Les adresses bit.ly sont utilisees sur les reseaux sociaux
- Exemple :
  - <http://bit.ly/jaimelire> est en fait :  
<http://www.mjangale.com/formationweb/ete2016/jaimelire/index.html>

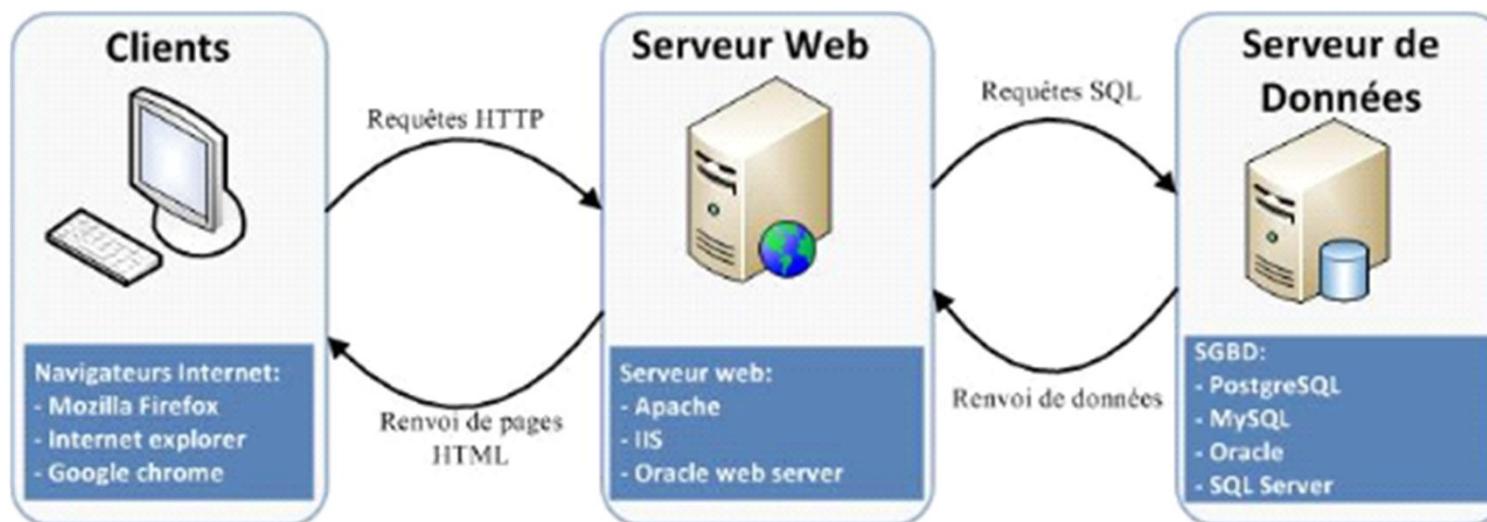
# navigateurs

# navigateurs

- Logiciel client capable d'interroger les serveurs ou les pages Web et d'exploiter leur résultats
- Permet de surfer sur le Net et d'afficher sur votre écran les pages qu'il a intercepté
  - Chrome (Google)
  - Firefox (Mozilla)
  - Internet Explorer (Microsoft)
  - Safari (Apple)



# Architecture du Web





# Boite à outils

# Boite à outils

- Editeurs de texte
  - Bloc note,
  - Notepad,
  - Notepad++,
  - SublimeText3...
- Navigateurs
  - Google Chrome,
  - Firefox...
- Un serveur Web
  - Wamp Server

# Deuxième Partie

---

- **Les bases du langage HTML**
- **Les bases du langage CSS**
- **Les bases du langage PHP/MySQL**

# Introduction

- Le langage HTML est un langage informatique qui permet de créer une page web. Il est l'abréviation de **HyperText Markup Language**
- HTML est un langage de description qui va nous permettre de décrire l'aspect d'un document web, d'y inclure des informations variées (textes, images, vidéo, sons, animation...) et d'établir des relations cohérentes grâce aux liens **Hypertexte**
- Les liens hypertextes permettent la navigation entre les pages Web

# Donc HTML est constitué de quoi?

- Le langage **HTML** n'est pas un langage de programmation proprement dit! Comme le langage C, pascal...
- Il est simplement composé de « **balises** » (ou « **tags** » en anglais) pour mettre en forme du texte et des images

# Structure d'une page HTML

```
<!DOCTYPE html>
<html lang='fr'>
  <head>
    <title> page simple HTML5 </title>
    <meta charset=<< utf-8 >> />
  </head>

  <body>
    Bonjour et bienvenu sur ma page Web
  </body>
</html>
```

# Allure d'un fichier HTML

```
<!-- commentaire -->
<!DOCTYPE>
<HTML>
  <HEAD>
    
  </HEAD>
  <BODY>
    
  <BODY>
</HTML>
```



# Qu'est ce qu'une balise

- Une balise est le moyen de communiquer avec le navigateur
- C'est une instruction comprises entre crochets (chevron) <  
> qui possède un nom et parfois des attributs
- La plupart doivent être fermée
  - Exemple <b> votre texte </b>
- Peuvent posséder des attributs
  - <balise attribut1="val\_num" attribut2="val\_alpha\_num"  
...>...</balise>

# Les balises « bloc »

- Servent à distinguer les parties entières de texte, comme des titres, des paragraphes, des listes, des divisions (**h1** à **h6**, **p**, **ul**, **div**)
  - À l'affichage, les blocs se placent toujours par défaut l'un en dessous de l'autre
- Les balises **bloc** peuvent servir de conteneur à du texte, des balises en ligne ou (exception faite des balise de **h1** à **h6**) à d'autres balises bloc

# Les balises « en ligne »

- Les balises « **en ligne** » sont insérées dans le texte pour l'enrichir comme un lien **hypertexte, italique, gras, image** (**a, em, strong, img**)
  - Elles interviennent dans le déroulement naturel du texte sans provoquer de saut de ligne
- Des **balises** « en ligne » ne peuvent contenir que des **balises** « en ligne »
- Exemples:
  - **<b> <div> blabla </div></b>** - **incorrect**
  - **<div> <b> blabla</b> </div>** - **correct**

# Attribut d'une balise

- Les attributs permettent de modifier le comportement par défaut d'une balise ou bien de spécifier des informations indispensables (par exemple la source d'une image)
- Certains attributs sont **obligatoires**, d'autres **optionnels**
- L'ordre dans lequel les attributs sont écrits n'est pas important et le séparateur est un espace
  - **<balise attribut1="val1" attribut2="val2">blabla</balise>**

# Règles de bases pour l'écriture en langage HTML

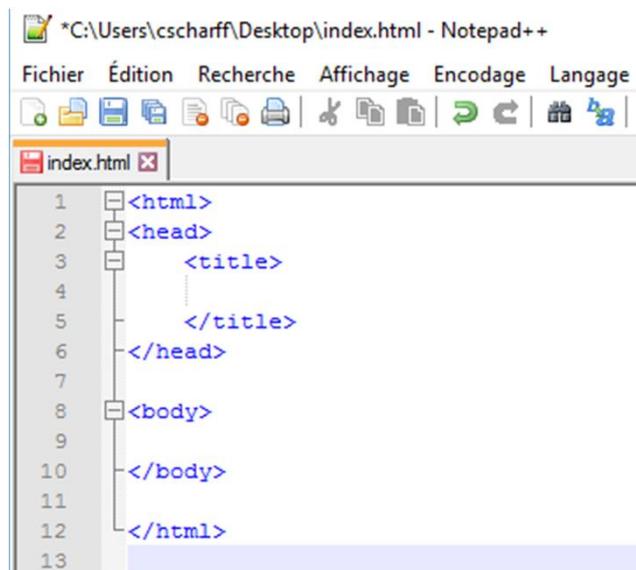
- Un document HTML standard doit impérativement commencer par **<HTML>** et finir par **</HTML>**
- Les commandes HTML ont une marque de début et une marque de fin. Certaines marques de fin sont facultatives
- Les commandes HTML utilisent les chevrons **<** et **>** comme délimiteurs
- Les balises peuvent être écrites en minuscule ou en majuscule

# Structure d'un document HTML

- <HTML> début d'un document en HTML
- </HTML> fin d'un document en HTML
- <HEAD> début de la zone d'en-tête
- </HEAD> Fin de la zone d'en-tête
- <TITLE> Début du titre de la page
- </TITLE> Fin du titre de la page
- <BODY> Début du document proprement dit
- </BODY> Fin du document proprement dit

# Exercice

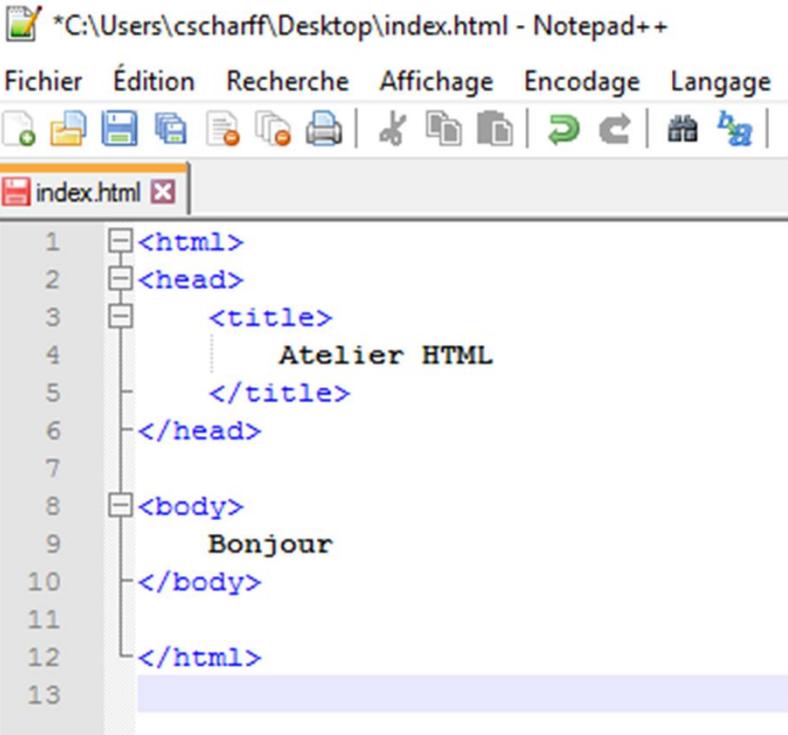
- Créer votre fichier HTML (index.html) et le squelette d'une page web
- Ouvrir le fichier dans le navigateur Chrome



```
*C:\Users\cscharff\Desktop\index.html - Notepad++
Fichier Édition Recherche Affichage Encodage Langage
index.html x
1 <html>
2 <head>
3   <title>
4   ...
5   </title>
6 </head>
7
8 <body>
9
10 </body>
11
12 </html>
13
```

# Première page web

- Ajouter un titre dans le header
- Ajouter du texte dans <body> ... </body>
- Ouvrir le fichier dans le navigateur chrome



```
*C:\Users\cscharff\Desktop\index.html - Notepad++
Fichier Édition Recherche Affichage Encodage Langage
index.html x
1 <html>
2 <head>
3   <title>
4     Atelier HTML
5   </title>
6 </head>
7
8 <body>
9   Bonjour
10 </body>
11
12 </html>
13
```

# Règles Générales

- Imbrication: le chevauchement des balises n'est pas permis
- ~~<h1><b> blabla </h1> </b>~~

# L'en-tête

- TITLE indique le nom du document HTML, qui sera afficher dans la barre de titre
- Exemple:
- <head> <title> Guide HTML </title></head>

# L'en-tête

- La balise **META** permet de donner des « méta-information » sur le document
- Exemple :
  - `<meta name="author" content="G. Chagon" />`
  - `<meta name="keywords" content="HTML, initiation" />`
  - `<meta name="description" content="cette page fournie une introduction au langage html" />`
  - `<meta name="date" content="2018-01-06" />`

# Le corps du document (body)

- Il contient tout ce qui apparaîtra dans la fenêtre du navigateur
- C'est ici où se trouvent les données à afficher (titre, images, vidéo...)
- **BACKGROUND**: permet de mettre une image dans le fond de la page
  - Exemple: <body background="mon\_image.png" >
- **BGCOLOR**: permet de changer la couleur de l'arrière plan
- **TEXT**: permet de changer la couleur du texte
- **LINK**: permet de changer la couleur des liens
- **ALINK**: change la couleur des liens sélectionnés
- **VLINK**: change la couleur des liens déjà visités
- Exemple    <body    **bgcolor**=“couleur”    **text**=“couleur”  
                    **alink**=“couleur” **vlink**=“couleur” >

# Groupement d'éléments

- Il est parfois nécessaire de grouper plusieurs éléments soit pour organiser le contenu, soit pour mettre en page à l'aide d'une feuille de style
  - Tel que par exemple des titres, paragraphe partageant un point commun
- Les éléments **div** et **span** sont là pour ça
  - **Div** est un élément de type bloc et peut contenir d'autres éléments de type bloc
  - **Span** est un élément en ligne

# Formatage du texte: les titres

- H1 à H6 définit 6 niveau pour les titres
- Chaque niveau correspond à son propre style plus le chiffre augmente plus la taille du texte est petite
- Paramètres: l'attribut **ALIGN** spécifie l'aligne horizontal d'un titre
- Align = left/right/center
  - Exemples : **<h1 align="center">mon titre</h1>**

# Formatage du texte: les paragraphes

- **P** construit un nouveau paragraphe = un double retour à la ligne
- Paramètres: l'attribut align spécifie l'alignement horizontal d'un paragraphe
  - **<p align="left">mon paragraphe </p>**
- Pour aligner du texte on peut utiliser l'attribut align avec la balise **<p>** lui affectant la valeur center, ou le tag **<center>**. Il existe une balise permettant d'aligner différents éléments: c'est le tag **<div>**
  - **<div align="left">...</div>**
  - **<div align="center">...</div>**
  - **<div align="right">...</div>**
- Le tag **blockquote** permet de décaler le paragraphe à droite

# Les caractères spéciaux

- Pour afficher les symboles <, >, &, “ sans qu'ils ne soient interprétés comme des délimiteurs il faut utiliser les codes:
  - &lt;      <
  - &gt;      >
  - &amp;    &
  - &quote;   “

# Les séparateurs

- **<br />** saut de ligne qui est une balise orpheline
- **<hr />** ligne horizontale
  - Paramètres:
    - Size: spécifie l'épaisseur de la barre
    - Width: spécifie la longueur de la barre
    - Align: pour l'alignement de la ligne
  - **<hr size=épaisseur width=largeur align=alignement color=couleur />**

# Le texte dans les pages web

```
index.html x
1 <!DOCTYPE html>
2 <html>
3     <head>
4         <title>
5             Ma premiere page Web
6         </title>
7     </head>
8     <!-- corps du document -->
9     <body>
10        text simple <br />
11        <b>texte en gras</b><br />
12        <strong>texte en gras bis</strong><br />
13        <i>text en italique</i> <br />
14        <em>texte en italique bis</em><br />
15        <b> <i>texte en gras et en italique</i></b> <br />
16        <font size="5">texte</font> <br />
17        <font color="#0000ff">texte en bleu</font> <br />
18        <sup>text mis en exposant</sup> <br />
19        <sub>text mis en indice</sub> <br />
20        <s>le texte sera baré</s>
21        <!-- cest fini -->
22    </body>
23 </html>
```

# Les polices

- Avec la balise **<font>** on peut jouer sur la taille (`size='10'`), la couleur (`color='green'`) et la police des caractères (`face='Arial'`)
- `<font size='5'>texte</font>`
- `<font color='blue' face='Arial'>coucou</font>`

# Quelques couleurs basiques

- Bleu #0000ff
- Blanc #ffffff
- Rouge #ff0000
- Gris clair #c0c0c0
- Vert #00ff00
- Violet #8000ff
- Jaune #ffff00
- Noir #000000

# Tableau de couleurs en hexadecimal

RGB Hex Triplet Color Chart															
E-mail-ware... What a concept!															
EEEEEE	FFFF66	FFCC66	FF9966	FF6666	FF3366	FF0066	FF00FF	FF00CC	FF0099	FF0000	FF0000	FF0000	FF0000	FF0000	FF0000
DDDDDD	FFFF33	FFCC33	FF9933	FF6633	FF3333	FF0033	FF0000								
CCCCCC	FFFF00	FFCC00	FF9900	FF6600	FF3300	FF0000									
BBBBBB	CCFFFF	CCCCFF	CC99FF	CC66FF	CC33FF	CC00FF	CC0000								
AAAAAA	CCFFCC	CCCCCC	CC99CC	CC66CC	CC33CC	CC00CC	CC0000								
999999	CCFF99	CCCC99	CC9999	CC6699	CC3399	CC0099	CC0000								
888888	CCFF66	CCCC66	CC9966	CC6666	CC3366	CC0066	CC0000								
777777	CCFF33	CCCC33	CC9933	CC6633	CC3333	CC0033	CC0000								
666666	CCFF00	CCCC00	CC9900	CC6600	CC3300	CC0000									
555555	99FFFF	99CCFF	9999FF	9966FF	9933FF	9900FF	990000	990000	990000	990000	990000	990000	990000	990000	990000
444444	99FFCC	99CCCC	9999CC	9966CC	9933CC	9900CC	990000	990000	990000	990000	990000	990000	990000	990000	990000
333333	99FF99	99CC99	999999	996699	993399	990099	990000	990000	990000	990000	990000	990000	990000	990000	990000
222222	99FF66	99CC66	999966	996666	993366	990066	990000	990000	990000	990000	990000	990000	990000	990000	990000
111111	99FF33	99CC33	999933	996633	993333	990033	990000	990000	990000	990000	990000	990000	990000	990000	990000
000000	99FF00	99CC00	999900	996600	993300	990000	990000	990000	990000	990000	990000	990000	990000	990000	990000
FF0000	66FFFF	66CCFF	6699FF	6666FF	6633FF	6600FF	660000	660000	660000	660000	660000	660000	660000	660000	660000
EE0000	66FFCC	66CCCC	6699CC	6666CC	6633CC	6600CC	660000	660000	660000	660000	660000	660000	660000	660000	660000
DD0000	66FF99	66CC99	669999	666699	663399	660099	660000	660000	660000	660000	660000	660000	660000	660000	660000
CC0000	66FF66	66CC66	669966	666666	663366	660066	660000	660000	660000	660000	660000	660000	660000	660000	660000
BB0000	66FF33	66CC33	669933	666633	663333	660033	660000	660000	660000	660000	660000	660000	660000	660000	660000
AA0000	66FF00	66CC00	669900	666600	663300	660000	660000	660000	660000	660000	660000	660000	660000	660000	660000
990000	33FFFF	33CCFF	3399FF	3366FF	3333FF	3300FF	330000	330000	330000	330000	330000	330000	330000	330000	330000
880000	33FFCC	33CCCC	3399CC	3366CC	3333CC	3300CC	330000	330000	330000	330000	330000	330000	330000	330000	330000
770000	33FF99	33CC99	339999	336699	333399	330099	330000	330000	330000	330000	330000	330000	330000	330000	330000
660000	33FF66	33CC66	339966	336666	333366	330066	330000	330000	330000	330000	330000	330000	330000	330000	330000
550000	33FF33	33CC33	339933	336633	333333	330033	330000	330000	330000	330000	330000	330000	330000	330000	330000
440000	33FF00	33CC00	339900	336600	333300	330000	330000	330000	330000	330000	330000	330000	330000	330000	330000
330000	00FFFFFF	00CCCCFF	009999FF	006666FF	003333FF	0000FF	000000	000000	000000	000000	000000	000000	000000	000000	000000
220000	00FFCC	00CCCC	0099CC	0066CC	0033CC	0000CC	000000	000000	000000	000000	000000	000000	000000	000000	000000
110000	00FF99	00CC99	009999	006699	003399	000099	000000	000000	000000	000000	000000	000000	000000	000000	000000
	00FF66	00CC66	009966	006666	003366	000066	000000	000000	000000	000000	000000	000000	000000	000000	000000
	00FF33	00CC33	009933	006633	003333	000033	000000	000000	000000	000000	000000	000000	000000	000000	000000
	00FF00	00CC00	009900	006600	003300	000000	000000	000000	000000	000000	000000	000000	000000	000000	000000

Copyright © 1995 Douglas R. Jacobson  
All Rights Reserved

M. Ndiaye

# Les listes non organisées

```
<ul>
    <lh>Entête en option</lh>
    <li>introduction</li>
    <li>Développement</li>
    <li>Conclusion</li>
</ul>
```

# Les listes organisées

```
<ol>
    <lh>Entête en option</lh>
    <li>introduction</li>
    <li>Développement</li>
    <li>Conclusion</li>
</ol>
```

# Les types de listes

<UL type="disc ">	Puces rondes pleine
<UL type="circle">	Puces rondes creuses
<UL type="square">	Puces carrées
<OL type=1>	(1,2,3)
<OL type=a>	(a,b,c)
<OL type=A>	(A,B,C)
<OL type=i>	(i,ii,iii)
<OL type=I>	(I,II,III)

# Exemple

```
index.html
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <title>
5              Nos Filières
6          </title>
7      </head>
8      <!-- corps du document -->
9      <body>
10         <h1>Informatique et télécom</h1>
11         <ol>
12             <li>Télécom et réseau</li>
13             <li>Développement logiciel</li>
14             <li>Maintenance</li>
15         </ol>
16     </body>
17 </html>
```

# L'hypertexte et les liens: internet

- Lien Externe:
  - <a href="URL cible">...</a>
    - <a href="http://zip-edu.sn" > Ziguinchor Institut Polytechnique </a>
- Lien Local
  - <a href="chemin/index.html">...</a>
  - Il existe deux types de chemins
    - Relatif: montre la structure des dossiers
      - dossier/page.html
      - ../dossier/page.html../
    - Absolue: se rapporte toujours au dossier parent du serveur (racine):  
C:///User/Pc/Mes Documentet/maPage.html

# Les ancrés ou signets

- Des liens peuvent aussi pointer vers un endroit précis du même document ou d'un autre fichier,
- C'est ce que l'on appelle les ancrés, ancrages ou pointeurs [Anchor]
- On veut se positionner dans exemple en haut de la page (le titre)
  - Identifier la cible avec
    - `<a name="titre" >Texte ou image </a>`
  - Utiliser le marqueur
    - `<a href="#titre" > pour y aller </a>`

# La structure du contenu

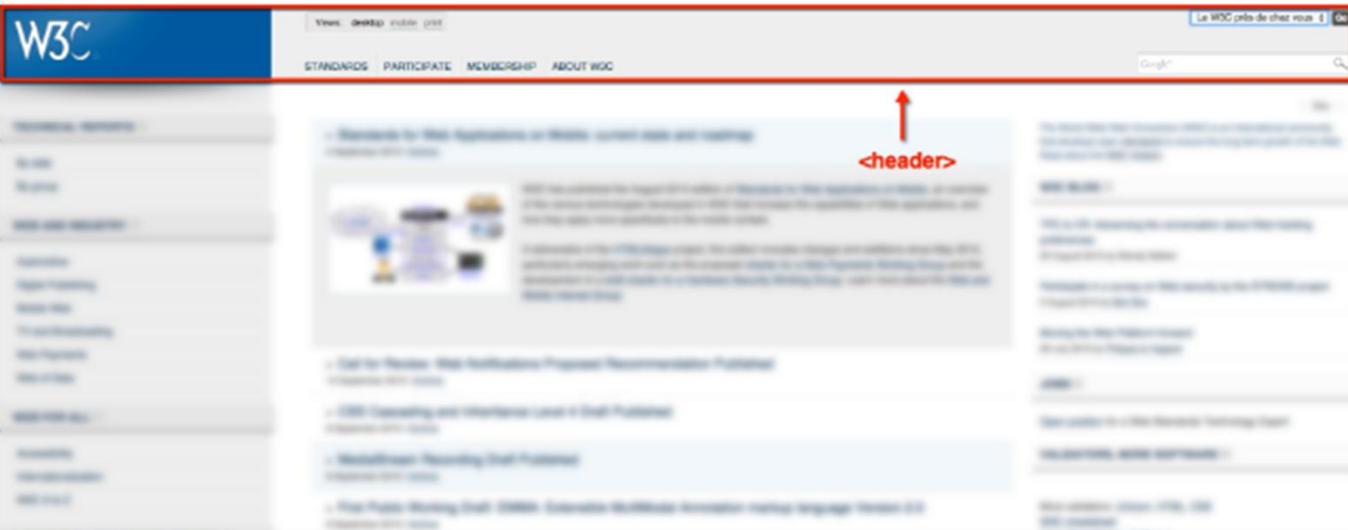
- En HTML 4 il n'y a pas de balises de structuration spécialisées, le contenu est structuré avec des **<div>**, **<span>** et autres conteneurs.
- HTML 5 introduit plusieurs balises pour aider à représenter la structure usuelle des documents.

# La structure du contenu

- <header>

```
1 <header>
2   <!-- Placez ici le contenu de l'en-tête de votre page -->
3 </header>
```

La figure suivante, par exemple, représente le site du W3C (qui se charge des nouvelles versions de HTML et CSS notamment). La partie encadrée en rouge correspondrait à l'en-tête :

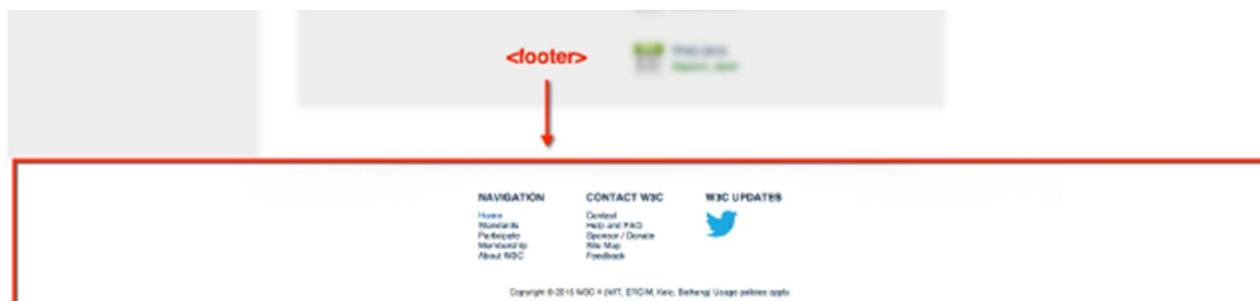


# La structure du contenu

- **<footer>** À l'inverse de l'en-tête, le pied de page se trouve en général tout en bas du document. On y trouve des informations comme des liens de contact, le nom de l'auteur, les mentions légales, etc.

```
1 <footer>
2   <!-- Placez ici le contenu du pied de page -->
3 </footer>
```

La figure suivante vous montre à quoi ressemble le pied de page du W3C.



# La structure du contenu

- **<section>** sert à regrouper des contenus en fonction de leur thématique. Elle englobe généralement une portion du contenu au centre de la page.

```
1 <section>
2   <h1>Ma section de page</h1>
3   <p>Bla bla bla bla</p>
4 </section>
```

Sur la page d'accueil du portail Free.fr, on trouve plusieurs blocs qui pourraient être considérés comme des sections de page (figure suivante).

## Actualités

A la une

Sport | Économie | People | Insolite | Rugby 2011 | Science | Musique |



- = Rugby: les All Blacks euphoriques, un point c'est tout
- = L'UE peine à définir sa réponse à la crise
- = Ski: Ligety triomphé à Sölden, juste devant Pinturault
- = Trois humanitaires européens enlevés en Algérie
- = Jean d'O, marque déposée !
- = La France ne renonce pas sur le FESF
- = La canonisation de trois religieux perturbée par un incident
- = Rugby: les All Blacks champions du monde pour un point
- = La police démantèle un gang de braqueurs "hors norme"
- = Séisme de 6,6 dans l'est de la Turquie

<section>

France | Monde | Insolite | Toute l'actu people du jour

» Toute l'actualité

## FHV



La vidéo à la demande en illimité  
à partir de **7,99€/mois**



Nouveautés de la semaine

1. Beatdown [\*\*<section>\*\*](#)
2. De l'amour à la folie
3. Patch Adams
4. Les allumés du golf
5. Un Amour de Coccinelle
6. La Belle et la Bête
7. Kiss Kiss, Bang Bang
8. Bobby Jones : naissance d'une

## Assistance



Installer votre Freebox Révolution

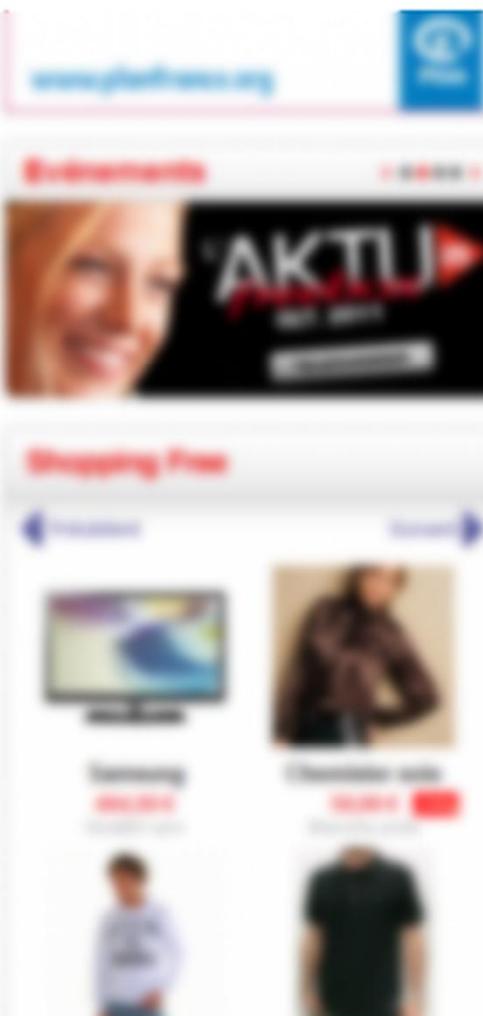
Pour une installation optimum,  
suiquez notre tutoriel

[\*\*<section>\*\*](#)

- Résoudre : Freebox bloque à étape X
- Internet : Configurer le WiFi
- Télévision : Prise en main du Freebox Player
- Téléphone : Brancher son téléphone Freebox

[\*\*► Visitez le site de l'assistance\*\*](#)

Rechercher sur le site l'assistance de Free :



Des sections de page sur le portail de Free

# La structure du contenu

- **<nav>** doit regrouper tous les principaux liens de navigation du site. Vous y placerez par exemple le menu principal de votre site

Généralement, le menu est réalisé sous forme de liste à puces à l'intérieur de la balise **<nav>** :

html

```
1 <nav>
2   <ul>
3     <li><a href="index.html">Accueil</a></li>
4     <li><a href="forum.html">Forum</a></li>
5     <li><a href="contact.html">Contact</a></li>
6   </ul>
7 </nav>
```

The screenshot shows the left sidebar of the W3C website, which is highlighted with a red border. A red arrow points from the text '<nav>' to the sidebar area. The sidebar contains several sections:

- TECHNICAL REPORTS**
  - By date
  - By group
- WEB AND INDUSTRY**
  - Automotive
  - Digital Publishing
  - Mobile Web
  - TV and broadcasting
  - Web Payments
  - Web of Data
- WEB FOR ALL**
  - Accessibility
  - Internationalization
  - W3C A to Z
- COMMUNITY AND BUSINESS GROUPS**
  - Current Groups
  - Proposed Groups
- WORKING GROUPS**
  - Specifications by group
  - Participant guidebook

The main content area displays a list of documents under the heading "Standards for Web Applications on Mobile, content reuse and re-use". The first item in the list is "HTML5 Application Cache (HTML5 Application Cache Specification)".

Le menu de navigation du W3C

# La structure du contenu

- <**article**> sert à englober une portion généralement autonome de la page.
  - C'est une partie de la page qui pourrait ainsi être reprise sur un autre site. C'est le cas par exemple des actualités (articles de journaux ou de blogs).

```
1 <article>
2   <h1>Mon article</h1>
3   <p>Bla bla bla bla</p>
4 </article>
```

Par exemple, voici un article sur le Monde :

The image shows a news article from Le Monde. The article is titled "Trente ans après, « Super Mario Bros. » casse toujours des briques". It discusses the game's enduring popularity and iconic nature. A red arrow points to the right side of the article frame, with the text "<article>" written next to it.

Trente ans après, « Super Mario Bros. » casse toujours des briques

Le jeu vidéo le plus emblématique des années 1980 continue d'alimenter activement la culture populaire, grâce à son univers iconique et son inépuisable richesse.

Ajouter à ma liste de lecture Partager Réagir Classer Partager Recommander Partager 27 personnes recommandent ça. Riez le premier parmi vos amis.

Le Monde.fr | 12.09.2015 à 20h35 Mis à jour le 13.09.2015 à 10h19 | Par William Audureau

Au diable les lois de la physique et de la biologie. Pour quiconque a déjà tenu une manette en main, la vérité ne fait aucun doute : les tortues les plus agressives sont rouges et ailées ; manger des fleurs permet de jeter des boules de feu ; des haricots magiques et des sous se cachent dans les briques ; et, par définition, les princesses sont toujours dans un autre château.

Ce dimanche 13 septembre, Super Mario Bros. et son univers losque filent leurs trente ans, et rares sont les jeux vidéo à pouvoir se vanter d'avoir traversé les générations tout en restant aussi iconiques, aussi modernes, aussi proches.

Un vivier à hommages

EN CONTINU 08.34 La canonisation du pétin polonais Jerry Popieluszko en route pour Rome

# La structure du contenu

- <**aside**> est conçue pour contenir des informations complémentaires au document que l'on visualise.
- Ces informations sont généralement placées sur le côté (bien que ce ne soit pas une obligation).

```
1 <aside>
2   <!-- Placez ici des informations complémentaires --&gt;
3 &lt;/aside&gt;</pre>
```

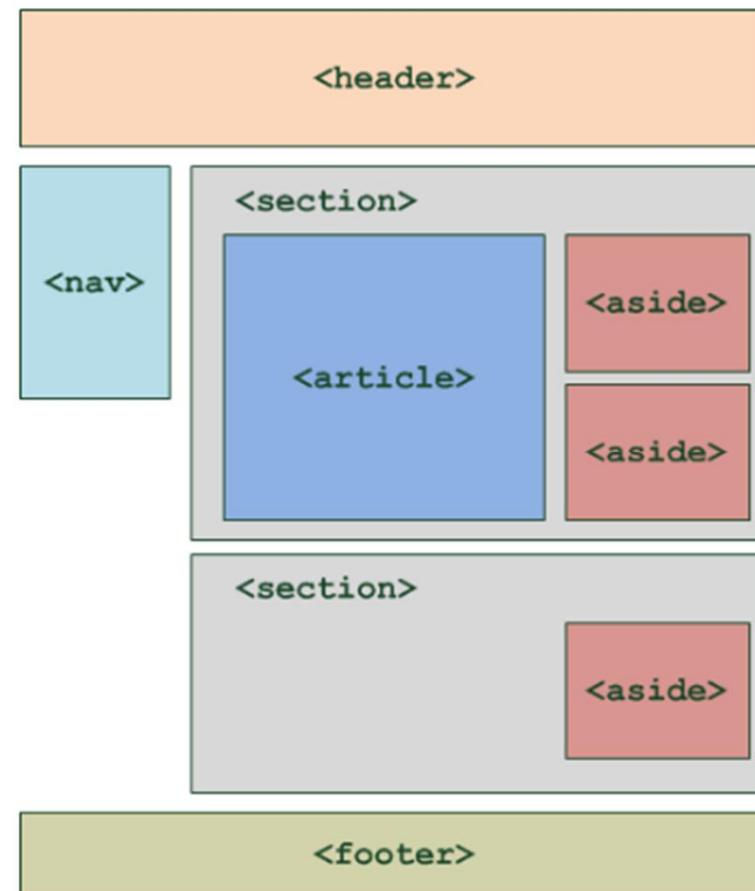
**<aside>**

The screenshot shows the French Wikipedia article for "Saturne (planète)". The page includes a sidebar with navigation links and a main content area with text about the planet. A red arrow points from the text "<aside>" to a callout box on the right. This callout box contains a title "Saturne ⓘ", a small image of the planet with its rings, and the caption "Saturne vue par la sonde Cassini en 2008". Below this is a section titled "Caractéristiques orbitales" with the following data:

Demi-grand axe	1 421 179 772 km (9,53707032 UA)
Aphélie	1 503 983 449 km (10,05350640 UA)
Périhélie	1 349 467 375 km (9,02093224 UA)
Circonférence orbitale	5 684 719 086 km (38 UA)
Excentricité	0,05415050
Période de révolution	10 757,7365 d (29 a 165 d 11,68 h)

Bloc d'informations complémentaires sur Wikipédia

# En résumé



# Exemple

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8" />
5          <title>Zozor - Le Site Web</title>
6      </head>
7
8      <body>
9          <header>
10             <h1>Zozor</h1>
11             <h2>Carnets de voyage</h2>
12         </header>
13
14         <nav>
15             <ul>
16                 <li><a href="#">Accueil</a></li>
17                 <li><a href="#">Blog</a></li>
18                 <li><a href="#">CV</a></li>
19             </ul>
20         </nav>
21
22         <section>
23             <aside>
24                 <h1>À propos de l'auteur</h1>
25                 <p>C'est moi, Zozor ! Je suis né un 23 novembre 2005.</p>
26             </aside>
27             <article>
28                 <h1>Je suis un grand voyageur</h1>
29                 <p>Bla bla bla (texte de l'article)</p>
30             </article>
31         </section>
32
33         <footer>
34             <p>Copyright Zozor - Tous droits réservés<br />
35             <a href="#">Me contacter !</a></p>
36         </footer>
37
38     </body>
39 </html>
```

# Les Images

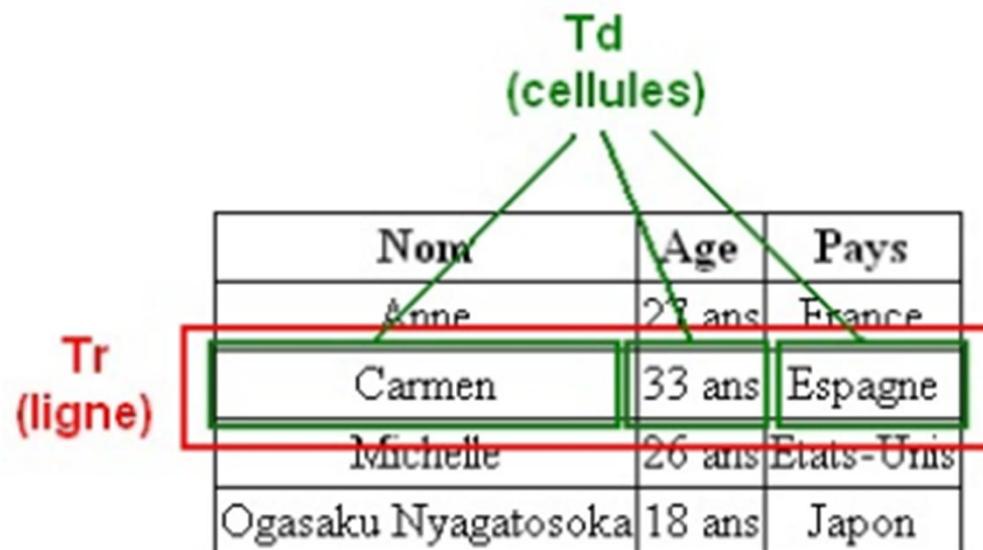
- ``
- ``
- ``
- `<a href="#" > </a>`

# Les tableaux

- Nous allons étudier les étapes à suivre pour créer un tableau avec le langage **HTML**
- La première balise à connaître est **<table> ...</table>** . C'est cette balise qui permet d'indiquer le début et la fin d'un tableau. Cette balise est de type bloc
  - **<tr> </tr>** indique le début et la fin d'une ligne du tableau ;
  - **<td>...</td>**: indique le début et la fin du contenu d'une cellule
- En HTML, un tableau se construit ligne par ligne. Dans chaque ligne (**<tr>**), on indique le contenu des différentes cellules (**<td>**)

# Les tableaux

- Schématiquement, un tableau se construit comme à la figure suivante



The diagram illustrates the structure of an HTML table. It shows a table with four rows and three columns. The columns are labeled "Nom", "Age", and "Pays". The rows are labeled "Anne", "Carmen", "Michelle", and "Ogasaku Nyagatosoka". A red box highlights the second row, labeled "Tr (ligne)". A green box highlights the third column, labeled "Td (cellules)".

Nom	Age	Pays
Anne	27 ans	France
Carmen	33 ans	Espagne
Michelle	26 ans	Etats-Unis
Ogasaku Nyagatosoka	18 ans	Japon

# Les tableaux

- Exemple

Code : HTML

```
<table>
  <tr>
    <td>Carmen</td>
    <td>33 ans</td>
    <td>Espagne</td>
  </tr>
  <tr>
    <td>Michelle</td>
    <td>26 ans</td>
    <td>États-Unis</td>
  </tr>
</table>
```

# Les tableaux

- La ligne d'entête
  - La ligne d'en-tête se crée avec un comme on l'a fait jusqu'ici, mais les cellules qu'elle contient sont, cette fois, encadrées par des balises **<th>** et non pas **<td>**

```
<table>
  <tr>
    <th>Nom</th>
    <th>Âge</th>
    <th>Pays</th>
  </tr>

  <tr>
    <td>Carmen</td>
    <td>33 ans</td>
    <td>Espagne</td>
  </tr>
  <tr>
    <td>Michelle</td>
    <td>26 ans</td>
    <td>États-Unis</td>
  </tr>
</table>
```

# Les tableaux

- Le titre du tableau
  - La balise est **<caption>...</caption>** elle se place juste avant l'entête. C'est elle qui porte le titre du tableau.

```
<table>
  <caption>Passagers du vol 377</caption>

  <tr>
    <th>Nom</th>
    <th>Âge</th>
    <th>Pays</th>
  </tr>
  <tr>
    <td>Carmen</td>
    <td>33 ans</td>
    <td>Espagne</td>
  </tr>
  <tr>
    <td>Michelle</td>
    <td>26 ans</td>
    <td>ÂEtats-Unis</td>
  </tr>
</table>
```

# Les tableaux : Diviser un gros tableau

- Si votre tableau est assez gros, vous aurez tout intérêt à le découper en plusieurs parties.
- Pour cela, il existe des balises HTML qui permettent de définir les trois « zones » du tableau :
  - l'en-tête (en haut) : il se définit avec les balises **<thead></thead>** ;
  - le corps (au centre) : il se définit avec les balises **<tbody></tbody>** ;
  - le pied du tableau (en bas) : il se définit avec les balises **<tfoot></tfoot>** .
- Que mettre dans le pied de tableau ?
  - Généralement, si c'est un long tableau, vous y recopiez les cellules d'en-tête. Cela permet de voir, même en bas du tableau, à quoi se rapporte chacune des colonnes

# Les tableaux : Diviser un gros tableau

- Schématiquement, un tableau en trois parties se découpe donc comme illustré à la figure suivante.

Passagers du vol 377			
En-tête du tableau	Nom	Age	Pays
	Carmen	33 ans	Espagne
	Michelle	26 ans	Etats-Unis
Corps du tableau	François	43 ans	France
	Martine	34 ans	France
	Jonathan	13 ans	Australie
	Xu	19 ans	Chine
Pied du tableau	Nom	Age	Pays

`<thead>`

`<tbody>` Un tableau découpé en plusieurs parties

`<tfoot>`

# Les formulaires (<form>)

- Pour insérer un formulaire il faut placer les marqueur **<form>...</form>** avec les attributs qui sont:
  - **Action:** il faut indiquer une URL indiquant l'adresse du script chargé de traiter le formulaire
  - **Method:** indiquer la méthode de transfert des données (**POST** ou **GET**)

# Les formulaires (<form>)

- Les éléments d'un formulaire:
  - **TEXTAREA**: pour insérer une zone de texte
  - **SELECT**: pour insérer une boite de sélection
  - **OPTION**: pour insérer une option de boite sélection
  - **<INPUT TYPE =“RADIO” />**: pour insérer un bouton radio
  - **<INPUT TYPE =‘CHECKBOX’ />**: pour insérer une case à cocher
  - **<INPUT TYPE =‘TEXT’ />**: pour insérer une zone d'édition de texte
  - **<INPUT TYPE =‘PASSWORD’ />**: pour insérer une zone d'édition de mot de passe

# Les formulaires (<form>)

- Les éléments d'un formulaire:
  - <INPUT TYPE =‘HIDDEN’ />: pour insérer un champ caché de données
  - <INPUT TYPE =‘FILE’ />: pour insérer un champ de fichier
  - <INPUT TYPE =‘IMAGE’ />: pour insérer un champ d'image
  - <INPUT TYPE =‘RESET ’ />: pour insérer un bouton de réinitialisation des données
  - <INPUT TYPE =‘SUBMIT’ />: pour insérer un bouton de transmission des données

# Nouveau type pour INPUT

Type	Type
<u>datetime</u>	<u>number</u>
<u>datetime-local</u>	<u>range</u>
<u>date</u>	<u>email</u>
<u>month</u>	<u>url</u>
<u>week</u>	<u>search</u>
<u>time</u>	<u>color</u>

# Deuxième Partie

---

- Les bases du langage HTML
- Les bases du langage CSS
- Les bases du langage PHP/MySQL

# Qu'est ce qu'une feuille de style

- Une feuille de style (Cascading Style Sheet CSS) est document numérique (d'extension **.css** ) qui vas pouvoir spécifier tous les caractéristiques de la mise en forme du document lié à la balise à laquelle elle s'applique

# Définition

- La syntaxe d'un style est toujours la même, elle précise la balise (sélecteur) à laquelle le style va s'appliquer, et les différents attributs d'un style
- Les attributs sont enfermés entre deux accolades (ouvrante et fermante) et chaque attribut est séparé par un point virgule (;
- Un sélecteur et ses propriétés constituent une règle de **CSS**
  - **Sélecteur{ propriétés de style: valeur;...}**

# Exemple

- H3 { font-family: Arial; font-style: italic }
- Donc ici la balise **h3** sera en **Arial** et en **Italic**
- Et dans votre document tous les balises **h3** auront comme style **Arial et Italic**
- Pour la lisibilité, vous pouvez toujours écrire vos style sur plusieurs lignes
  - H3{
    - **Font-family: Arial;**
    - **Font-style: italic**
  - }
- On peut attribuer un même style à plusieurs balises séparées par des virgules
  - **H3, h2{font-family:Arial; font-style: italic}**

# Déclaration interne

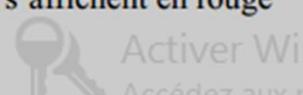
```
<html>  
  <head>  
    <style>  
      <!-- la feuille de style -->  
    </style>  
  </head>  
  <body>  
  </body>  
</html>
```

# Déclaration dans un document séparé

```
<html>  
  <head>  
    <link rel="stylesheet" href="style.css"  
          media="screen" />  
  </head>  
  <body>  
  </body>  
</html>
```

# Règles de CSS

Regles	Eléments HTML concernés par la règle	Explications
<code>p{color:gray;}</code>	<code>&lt;p&gt;ceci est un paragraphe&lt;/p&gt;</code>	Mise en forme de tous les éléments p
<code>H1, h2{padding-top:1em;}</code> <code>/ *Groupes*/</code>	<code>&lt;h1&gt;Grand titre&lt;/h1&gt;</code> <code>&lt;p&gt;Paragraphe introductif&lt;/p&gt;</code> <code>&lt;h2&gt;Première partie&lt;/h2&gt;</code>	Pour appliquer la même mise en forme à plusieurs types d'élément, on sépare les sélecteurs par une virgule
<code>/ *selecteur de classs*/</code> <code>.important {font-weight:bold;}</code>	<code>&lt;p&gt;Un paragraphe est important, sauf</code> <code>&lt;span class='important'&gt;ce mot&lt;/span&gt;</code> <code>&lt;/p&gt;</code> <code>&lt;p class='important'&gt;Et aussi un</code> <code>important du début à la fin&lt;/p&gt;</code>	Si les éléments d'information proposés par les éléments HTML ne sont pas précis, il faut créer une classe de style. Coté CSS une classe commence par un point (.), coté HTML, c'est un attribut.
<code>/ *selecteur d'identifiant*/</code> <code>#menu{</code> <code>position:fixed;</code> <code>Top:0px;}</code>	<code>&lt;p&gt;Contenu du document HTML&lt;/p&gt;</code> <code>&lt;p&gt;</code> <code>&lt;a href='index.html' id='menu'&gt;home&lt;/a&gt;</code> <code>&lt;a href='ex.html'&gt;exos&lt;/a&gt;</code> <code>&lt;/p&gt;</code>	Si l'information que l'on veut ajouter ne concerne qu'un seul élément dans tout le fichier HTML, on utilisera son identifiant unique (id attribut en html, identifiant précédé de # en CSS)
<code>/ *selecteur de pseudo-classs*/</code> <code>a:hover{color:red}</code>	<code>&lt;p&gt; Un paragraphe avec &lt;a href='lien.html'&gt;un</code> <code>lien &lt;/a&gt; dedans. Et puis aussi &lt;a</code> <code>href='lik.html'&gt;un autre liens&lt;/a&gt;&lt;/p&gt;</code> 	La sémantique des pseudo-classes est prédéfinie. Coté CSS, elle s'appliquent à un sélecteur en utilisant (:). Ici les liens survolés par la souris s'affichent en rouge



# Les styles de polices

- **Font-family**: définit un nom de police précise (Arial, Times, Helvetica, monospace...)
  - H2 {font-family: Arial;}
- **Font-style**: définit le style de l'écriture normal ou italique ou oblique
  - P {font-style: italic;}
- **Font-weight**: définit l'épaisseur de la police (normal, bold, bolder, lighter ou des valeurs numériques 100 | 200 ...)
  - P {font-weight: bold;}
- **Font-size**: définit la taille de la police (xx-small, x-small, small, medium, large, x-large, xx-large, ou en taille précise en points (pt), centimètre (cm) pixels (px) ou pourcentage (%)

# Les styles de Texte

- **Text-align**: définit l'alignement du texte (left, center, right)
  - H2 {text-align: left;}
- **Text-indent**: définit un retrait dans la première ligne d'un bloc souvent utilisé avec <p> spécifier en centimètre (cm) ou en pixels
  - P {text-indent: 1cm;}
- **color**: définit la couleur du texte par exemple en hexadécimal
  - P {text-color: #00FF00;}
- **Text-transform**: définit la casse du texte majuscule ou minuscule (uppercase, lowercase) ou capitalize qui met la première lettre en majuscule

# Les styles de Texte

- **word-spacing**: définit l'espace entre les mots exprimé en point (pt) centimètre (cm), pixel (px) ou en %
  - H2 {text-spacing: 1ptt;}
- **letter-spacing**: définit l'espace entre les lettres exprimé en point (pt) centimètre (cm), pixel (px) ou en %
  - P {letter-spacing: 1cm;}
- **Line-height**: définit l'interligne soit l'espace entre les lignes en cm, pt ou px
  - P {line-height: 5px;}
- **width**: définit la largeur d'un élément de texte ou d'une image (pt, cm, px, %)
- **Height**: définit la hauteur d'un élément de texte ou d'une image (pt, cm, px, %)

# Les arrières plans

- **Background-color**: définit une couleur de fond
  - H2 {background-color: #FF0000;}
- **Background-image**: définit une image d'arrière plan
  - body {background-image: image.png;}

# Les marges

- **Margin-top**: définit la valeur de la marge supérieur en unité de longueur ou auto
  - H2 {margin-top: 5px;}
- **Margin-left**: définit la valeur de la marge gauche en unité de longueur ou auto
  - H2 {margin-left: 5px;}
- **Margin-right**: définit la valeur de la marge droite en unité de longueur ou auto
  - H2 {margin-right: 5px;}
- **Margin-bottom**: définit la valeur de la marge inférieur en unité de longueur ou auto
  - H2 {margin-bottom: 5px;}

# Les bordures

- **Border-width**: définit l'épaisseur de la bordure (thin, medium, thick ou en unité de mesure)
  - H2 {border-width: thin;}
- **Border-color**: définit la couleur de la bordure
  - H2 {border-color: 5px;}
- **Border-style**: définit le style de la bordure (solid, dotted, dashed, double, groove, ridge, inset, ou outset)
  - H2 {border-style: solid;}

# Les tables

- **Border**: Pour spécifier le bordures de tableau en CSS
  - Table, tr, th {border: 1px solid black;}
  - Notez que la table dans l'exemple ci-dessus a des doubles frontières. En effet, la table et les éléments <th> et <td> ont des bordures distinctes.
- **Border-collapse**: définit si les bordures de la table doivent être réduite en une seule bordure
  - Table {border-collapse: collapse;}
  - Table, th, td, tr {border: 1px solid black;}
- **Tr:hover**: définir une règle au survole d'une ligne du tableau
  - Tr:hover {background-color: #f5f5f5;}
- **Tr:nth-child(even)**: pour manipuler les lignes (paires ou impaire du tableau)
  - Tr:nth-child(even) {background-color: #f5f5f5;}

# Les transformations

- Les transformations CSS vont nous permettre de faire tourner, de changer l'échelle ou encore de torde un élément de notre page
- De même il est possible de définir une perspective afin de simuler une position ou une animation dans l'espace 3D

# Transformations 2D

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>Transformation</title>
5      <link rel="stylesheet" type="text/css" href="style.css">
6  </head>
7  <body>
8  <div id="wrap">
9      <div class="transformation">
10         
11     </div>
12 </div>
13 </body>
14 </html>
```

```
1  #wrap{
2      width: 80px;
3      margin-right: 10px;
4  }
5  .transformation{
6      float: left;
7      margin: 10px;
8      border: 5px solid #fff;
9      border-radius: 10px;
10     transform:rotate(5deg);
11 }
```

# Transformation 2D

- Si on voulait modifier les proportions du div on aurait appliqué:
  - `Transform:scale(1.5);`
  - Représente la valeur 150% à la taille normale

# Transformation 3D

## • Les rotations

- `Rotate3d(x,y,z,angle)`: spécifie la rotation 3D autour de l'axe défini par les vecteur x,y et z
- `RotateX(angle)`: spécifie la rotation 3D autour de l'axe X
- `RotateY(angle)`: spécifie la rotation 3D autour de l'axe Y
- `RotateZ(angle)`: spécifie la rotation 3D autour de l'axe Z

# Transformation 3D

- Les translations
  - `translate3d(x,y,z,angle)`: spécifie une translation en 3D
  - `translateZ(z)`: spécifie une translation sur l'axe des z
- Les changements d'état
  - `scaleZ(x,y,z)`: spécifie un changement d'échelle en 3D
  - `scaleZ(x)`: spécifie un changement d'échelle sur l'axe Z

# Les transitions

Les transformations que nous venons de réaliser sont opérées de manière immédiate dans les navigateurs . Les nouveautés CSS3 permettent d'effectuer une transition dans les transformations, ce qui donne un effet visuel intéressant.

- Elle se définit de la manière suivante:
  - ***transition-property: prop1, prop2 ... propN***
- Il est possible de définir toutes ces informations sur une même ligne définie de la manière suivante:
  - ***transition : prop || durée || fonction || délai***

# Les transitions

- ❖ ***transition-duration*** : *Ns* : durée de la transition en seconde.
- ❖ ***transition-timing-function*** : *value* : type de fonction de transition. Les valeurs peuvent être:
  - *linear* : vitesse constante du début à la fin
  - *ease-in* : la vitesse de transition augmente
  - *ease-out* : la visesse de transition diminue
  - *ease-in-out* : la vitesse de transition est lente au début et à la fin
- ❖ ***transition-delay*** : *Ns* : temps d'attente avant déclenchement de la transition

# Les animations

Pour réaliser une animation nous aurons besoin de placer dans les propriétés de l'objet animé:

- ❑ ***animation-name*** : nom de l'animation
- ❑ ***animation-duration*** : durée de l'animation
- ❑ ***animation-iteration-count*** : nombre de fois où l'animation est réalisée

# Les animations

## ❑ ***animation-timing-function*** : type de fonction d'animation

- *linear* : vitesse constante du début à la fin
- *ease-in* : la vitesse de transition augmente
- *ease-out* : la visesse de transition diminue
- *ease-in-out* : la vitesse de transition est lente au début et à la fin

## ❑ ***animation-delay*** : délai d'attente avant exécution de l'animation.

# Les animations

Une fois ces paramètres réglés nous devrons définir l'animation en elle-même

***@-nom\_animation{***

***from {***

définition de la position de départ

***}***

***X% {*** définition de la position à X% de l'animation. La position X% n'est pas obligatoire mais est très pratique lorsque l'on veut faire une animation en boucle ou bien la rotation d'un objet pour lui donner un sens.

***}***

***to {***

définition de la position d'arrivée

***}***

***}***

# Propriétés de mise en forme

Propriété	Valeurs (exemples)	Description
<code>font-family</code>	<code>police1, police2, police3, serif, sans-serif, monospace</code>	Nom de police
<code>@font-face</code>	<i>Nom et source de la police</i>	Police personnalisée
<code>font-size</code>	1.3em, 16px, 120%...	Taille du texte
<code>font-weight</code>	bold, normal	Gras
<code>font-style</code>	italic, oblique, normal	Italique
<code>text-decoration</code>	underline, overline, line-through, blink, none	Soulignement, ligne au-dessus, barré ou clignotant
<code>font-variant</code>	small-caps, normal	Petites capitales
<code>text-transform</code>	capitalize, lowercase, uppercase	Capitales
<code>font</code>	-	Super propriété de police. Combine : <code>font-weight, font-style, font-size, font-variant, font-family</code> .

# Propriétés de mise en forme

<b>font</b>	-	Super propriété de police. Comme : <b>font-weight, font-style, font-size, font-variant, font-family</b> .
<b>text-align</b>	left, center, right, justify	Alignement horizontal
<b>vertical-align</b>	baseline, middle, sub, super, top, bottom	Alignement vertical (cellules de tableau ou éléments inline-block uniquement)
<b>line-height</b>	18px, 120%, normal...	Hauteur de ligne
<b>text-indent</b>	25px	Alinéa
<b>white-space</b>	pre, nowrap, normal	Césure
<b>word-wrap</b>	break-word, normal	Césure forcée
<b>text-shadow</b>	5px 5px 2px blue <i>(horizontale, verticale, fondu, couleur)</i>	Ombre de texte

# Propriétés de couleurs et de fond

Propriété	Valeurs (exemples)	Description
<b>color</b>	<i>nom, rgb(rouge,vert,bleu), rgba(rouge,vert,bleu,transparence), #CF1A20...</i>	Couleur du texte
<b>background-color</b>	<i>Identique à color</i>	Couleur de fond
<b>background-image</b>	<i>url('image.png')</i>	Image de fond
<b>background-attachment</b>	<i>fixed, scroll</i>	Fond fixe
<b>background-repeat</b>	<i>repeat-x, repeat-y, no-repeat, repeat</i>	Répétition du fond
<b>background-position</b>	<i>(x y), top, center, bottom, left, right</i>	Position du fond
<b>background</b>	<i>-</i>	Super propriété du fond. Combine : <b>background-image</b> , <b>background-repeat</b> , <b>background-attachment</b> , <b>background-position</b>
<b>opacity</b>	<i>0.5</i>	Transparence

# Propriétés des boîtes

Propriété	Valeurs (exemples)	Description
<b>width</b>	150px, 80%...	Largeur
<b>height</b>	150px, 80%...	Hauteur
<b>min-width</b>	150px, 80%...	Largeur minimale
<b>max-width</b>	150px, 80%...	Largeur maximale
<b>min-height</b>	150px, 80%...	Hauteur minimale
<b>max-height</b>	150px, 80%...	Hauteur maximale
<b>margin-top</b>	23px	Marge en haut
<b>margin-left</b>	23px	Marge à gauche
<b>margin-right</b>	23px	Marge à droite
<b>margin-bottom</b>	23px	Marge en bas
<b>margin</b>	23px 5px 23px 5px (haut, droite, bas, gauche)	Super-propriété de marge. Combine : <b>margin-top</b> , <b>margin-right</b> , <b>margin-bottom</b> , <b>margin-left</b> .

# Propriétés des boîtes

<code>padding-left</code>	23px	Marge intérieure à gauche
<code>padding-right</code>	23px	Marge intérieure à droite
<code>padding-bottom</code>	23px	Marge intérieure en bas
<code>padding-top</code>	23px	Marge intérieure en haut
<code>padding</code>	23px 5px 23px 5px <i>(haut, droite, bas, gauche)</i>	Super-propriété de marge intérieure. Combine : <code>padding-top, padding-right, padding-bottom, padding-left</code> .
<code>border-width</code>	3px	Épaisseur de bordure
<code>border-color</code>	<i>nom, rgb(rouge,vert,bleu), rgba(rouge,vert,bleu,transparence), #CF1A20...</i>	Couleur de bordure
<code>border-style</code>	<i>solid, dotted, dashed, double, groove, ridge, inset, outset</i>	Type de bordure
<code>border</code>	3px solid black	Super-propriété de bordure. Combine <code>border-width, border-color, border-style</code> . Existe aussi en version <code>border-top, border-right, border-bottom, border-left</code> .
<code>border-radius</code>	5px	Bordure arrondie
<code>box-shadow</code>	<i>6px 6px 0px black (horizontale, verticale, fondu, couleur)</i>	Ombre de boîte



# Propriétés de positionnement et d'affichage

Propriété	Valeurs (exemples)	Description
<b>display</b>	block, inline, inline-block, table, table-cell, none...	Type d'élément (block, inline, inline-block, none...)
<b>visibility</b>	visible, hidden	Visibilité
<b>clip</b>	rect (0px, 60px, 30px, 0px) rect ( <i>haut, droite, bas, gauche</i> )	Affichage d'une partie de l'élément
<b>overflow</b>	auto, scroll, visible, hidden	Comportement en cas de dépassement
<b>float</b>	left, right, none	Flottant
<b>clear</b>	left, right, both, none	Arrêt d'un flottant
<b>position</b>	relative, absolute, static	Positionnement
<b>top</b>	20px	Position par rapport au haut
<b>bottom</b>	20px	Position par rapport au bas
<b>left</b>	20px	Position par rapport à la gauche
<b>right</b>	20px	Position par rapport à la droite
<b>z-index</b>	10	Ordre d'affichage en cas de superposition. La plus grande valeur est affichée par-dessus les autres.

# Propriétés des listes

Propriété	Valeurs (exemples)	Description
<code>list-style-type</code>	disc, circle, square, decimal, lower-roman, upper-roman, lower-alpha, upper-alpha, none	Type de liste
<code>list-style-position</code>	inside, outside	Position en retrait
<code>list-style-image</code>	<code>url('puce.png')</code>	Puce personnalisée
<code>list-style</code>	-	Super-propriété de liste. Combine <code>list-style-type</code> , <code>list-style-position</code> , <code>list-style-image</code> .

# Propriétés des tableaux

Propriété	Valeurs (exemples)	Description
<b>border-collapse</b>	collapse, separate	Fusion des bordures
<b>empty-cells</b>	hide, show	Affichage des cellules vides
<b>caption-side</b>	bottom, top	Position du titre du tableau

# Deuxième Partie

---

- Les bases du langage HTML
- Les bases du langage CSS
- Les bases du langage PHP/MySQL

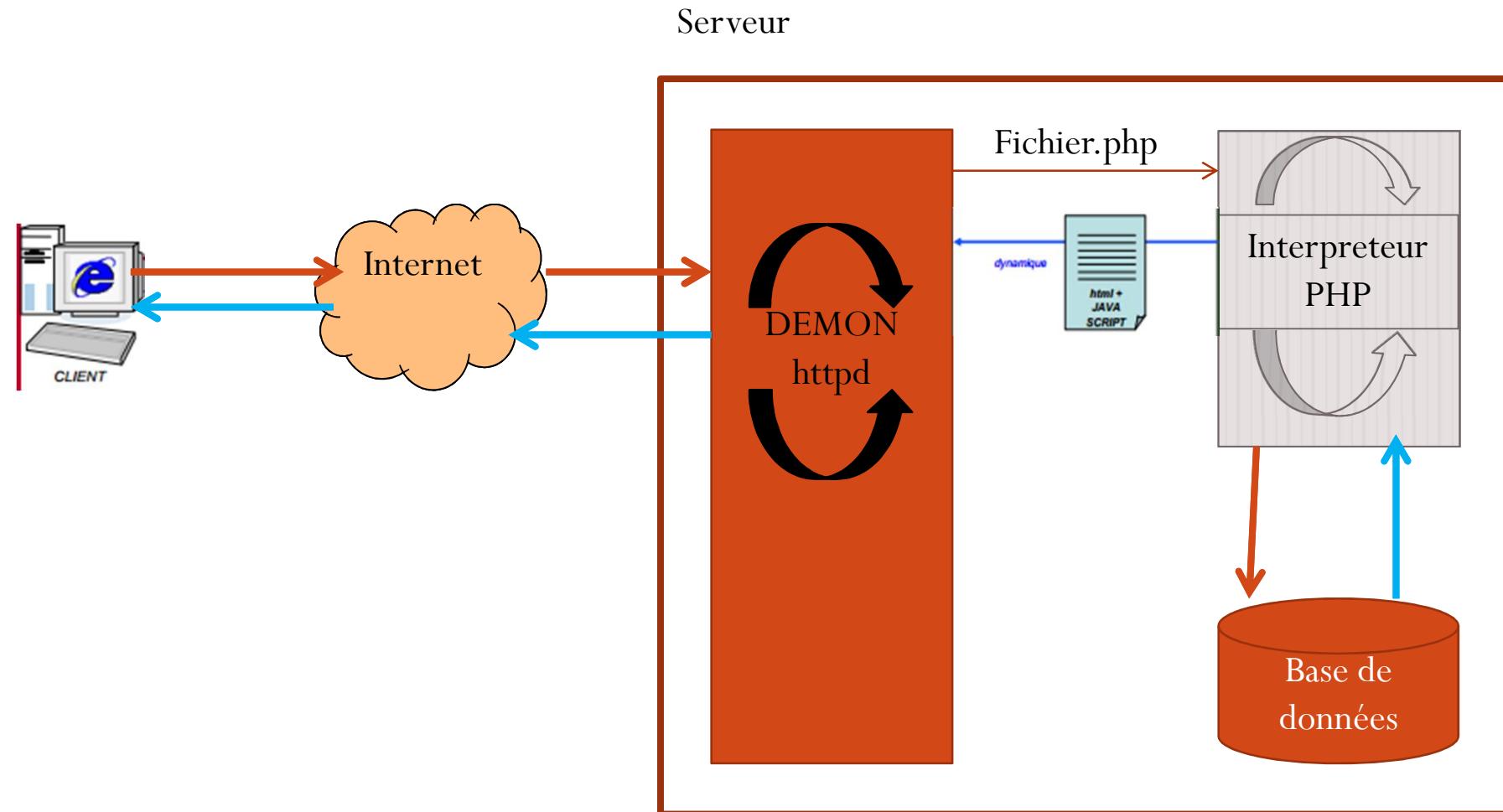
# Plan – Notion de base en PHP

- Introduction
- Les variables et Constantes
- Les chaines en PHP
- Les typages en PHP
- Les tableaux en PHP
- Les opérateurs
- Les Structures de contrôles
- Les fonctions
- Les Inclusions de fichiers externe
- Les formulaires
- Accès aux base de données
- Les Sessions
- Programmation orientée objet en PHP

# Introduction

- PHP (*Hypertext PreProcessor*) est un langage interprété orienté Web.
- Syntaxiquement, c'est un mélange de C et de Perl.
- Les scripts PHP sont lus et interprétés par le moteur PHP.
- PHP comporte plus de 500 fonctions.
- Il est fourni avec des bibliothèques offrant des fonctionnalités diverses
  - Accès aux bases de données
  - Fonctions d'image
  - Socket ...
- Les grands CMS Content Management Systems ou Systèmes de Gestion de Contenus utilisent PHP, comme :
  - Wordpress
  - Joomla
  - Drupal

# Principe de fonctionnement



# SYNOPSIS

- Un script PHP peut comprendre à la fois du code PHP et du code HTML, non interprété.
- On doit donc encadrer les parties comportant le code PHP entre 2 balises `<?PHP` et `?>`. Le reste de la page n'est pas interprété

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>PHP</title>
5  </head>
6  <body>
7  |   <?php echo "Bonjour le monde"; ?>
8  </body>
9  </html>
```

# Commentaires

- Pour commenter le code, on utilise :
  - Commentaire sur une ligne: // ou #
  - Commentaire sur plusieurs lignes: /\* ... \*/

```
1  <!DOCTYPE html>
2  <html>
3  <head>
4      <title>PHP</title>
5  </head>
6  <body>
7      <?php
8          //commentaire sur une ligne
9          #autre commentaire sur une ligne
10         echo "Bonjour le monde";
11         /* Commentaire sur
12             plusieurs lignes
13         */
14     ?>
15  </body>
16  </html>
```

# Les Variables

- Les variables sont précédées du signe \$, quelque soit leur type.

```
<?php  
    $variable = "Une variable en PHP";  
    //autres variable  
    $x = 1000;  
?>
```

- Existence de variables, la fonction **isset()**:
  - La fonction **isset()** permet de tester si une variable existe
  - La fonction **unset()** permet de supprimer une variable et de désallouer la mémoire utilisée

```
<?php  
    $a = 5;  
    if(isset($a)) echo "la variable a existe et a= $a";  
    unset($a);  
    echo "La variable a été supprimée";  
?>
```

# Les Variables

- La fonction **empty()** teste si la variable est vide ou pas.
- La fonction **gettype()** permet de connaître le type de la variable. Elle renvoie une chaîne : "**string**" ou "**integer**" ou "**double**" ou "**array**" ou "**object**".
- On peut également tester un type particulier a l'aide des fonctions **is\_array**, **is\_string**, **is\_int**, **is\_float**, **is\_object** .

```
<?php
    $a = 5;
    if(!empty($a)) echo "la variable a existe et a= $a
        et elle est non vide";
    echo gettype($a); //affiche "integer"
    echo is_int($a); // vrai
?>
```

# Les Constantes

- PHP permet de définir des constantes a l'aide de la fonction **define**.
  - **define("CONSTANTE", "rouge" );**
- Deux constantes sont prédéfinies par PHP :
  - **\_\_FILE\_\_** contient le nom du fichier
  - **\_\_LINE\_\_** le numéro de la ligne courante.

```
<?php
    define("PAGESUIVANTE", "script.php");
    echo "page courante ".__FILE__." page suivante ".
        PAGESUIVANTE;
?>
```

# Les chaines en PHP

- **Guillemets ou Cotes :**

```
<?php  
$var="Hello PHP";  
$machaine="le contenu de \$var est $var<br>";  
echo $machaine;  
//ou avec des ' ':  
$mystring='le contenu de $var est '.\$var;  
echo $mystring;  
?>
```

- **Dont le résultat sera toujours le même**
  - *le contenu de \$var est Hello PHP*
- La concaténation se fait à l'aide du **point (.)**
- la fonction **strlen()**: renvoi la longueur d'une chaîne
  - `<?php int lg=strlen($chaine);?>`
- **strtoupper()** pour obtenir des majuscules
- **strtolower()** pour mettre en minuscules
- **ucfirst()** pour mettre en majuscule la première lettre d'une chaîne
- **ucwords()** pour mettre en majuscule la première lettre de chaque mot dans une chaîne

# Les chaines en PHP

- **Guillemets ou Cotes :**

```
<?php
$var="Hello PHP";
$machaine="le contenu de \$var est $var<br>";
echo $machaine;
//ou avec des ' ':
$mystring='le contenu de $var est '.$var;
echo $mystring;
?>
```

- **Dont le résultat sera toujours le même**
  - *le contenu de \$var est Hello PHP*
  - La concaténation se fait à l'aide du **point (.)**

# Les typages en PHP

- *gettype()* renvoie l'un des résultats suivants
  - Intger, double, string, array, object, class
- *settype()* change le type d'un élément :
  - `<?php $a = 3,14; settype($a, 'integer');?>`
- Fonction de test:
  - *is\_int()*
  - *is\_long()*
  - *is\_double()*
  - *is\_array()*
  - *is\_object()*
  - *is\_string()*

# Les tableaux en PHP

- Tableaux associatifs - parcours avec boucle foreach :

```
<?php
    $jours=array("Lu"=>"Lundi", "Ma"=>"Mardi",
    "Me"=>"Mercredi", "Je"=>"Jeudi", "Ve"=>"Vendredi",
    "Sa"=>"Samedi", "Di"=>"Dimanche" );
    foreach($jours as $key=>$val) echo $key. " ". $val.
        "<br>\n";
?

```

Ce qui donne :

```
Lu Lundi
Ma Mardi
Me Mercredi
Je Jeudi
Ve Vendredi
Sa Samedi
Di Dimanche
```

# Les tableaux en PHP

- Affichage avec `print_r()` :

```
<?php  
    print_r($jours);  
?>
```

- On obtient un résultat brut qui affiche la structure du tableau

```
Array  
(  
    [Lu] => Lundi  
    [Ma] => Mardi  
    [Me] => Mercredi  
    [Je] => Jeudi  
    [Ve] => Vendredi  
  
    [Sa] => Samedi  
    [Di] => Dimanche  
)
```

# Les opérateurs

**PHP dispose des opérateurs classiques inspirés des langages C et Perl.**

## Comparaison

<code>==</code>	égalité
<code>&gt;</code>	inférieur strict
<code>&lt;</code>	supérieur strict
<code>&lt;=</code>	inférieur ou égal
<code>&gt;=</code>	supérieur ou égal
<code>!=</code>	négation

## Logiques

Les opérateurs logiques sont utilisés dans les tests, par exemple dans un « `if ( condition )` »

<code>&amp;&amp;</code>	et
<code>  </code>	ou
<code>xor</code>	ou exclusif
<code>!</code>	négation

**Remarque :** les opérateurs `and`, `or` , `not` sont également disponibles et font la même chose.

# Les opérateurs

## Arithmétiques

- + addition
- soustraction
- / division
- \* multiplication
- % modulo
- ++ incrément
- décrément

Remarque : l'opérateur / renvoie un entier si les 2 opérandes sont des entiers, sinon il renvoie un flottant.

## Affectation

- = affectation
- += addition puis affectation
- = soustraction puis affectation
- \*= multiplication puis affectation
- /= division puis affectation
- %= modulo puis affectation

```
$n = 0;  
$n += 2;      // $n vaut 2  
$n *= 6;      // $n vaut 12  
$r= $n % 5;   // 12 modulo 5 => $r = 2  
if( ++$n == 13 ) echo " pas de chance ";  
// pré-incrément le test renvoie vrai
```

# Les opérateurs

## Divers

- L'opérateur de concaténation est utilisable sur les chaînes scalaires.

```
$chaîne = "Votre nom est" ;  
$nom = "Toto";  
echo $chaîne . " " . $nom; // affiche "Votre nom est Toto"
```

- L'opérateur ? : ou *opérateur de test trinaire*. Sa syntaxe est [test logique] ? [expression si vrai] : [expression si faux]

```
$a= $b =1;  
( $a == $b ) ? $c= 10 : $c = 20; // effectue $c = 10;
```

- On peut également l'utiliser pour compacter les séquence de test / affectations

```
$réponse = ( $a == $b ) ? "a égal b" : "a différent de b" ;  
echo $réponse; // affiche "a égal b" car le test ( $a == $b ) renvoie vrai
```

# Les Structures de contrôles

## Les tests IF

Syntaxes :

Test if " basique " :

```
if( [condition] ) {  
    ...  
}
```

Test if-else :

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

Test if-elseif :

```
if( [condition] ) {  
    ...  
} elseif( [condition] ) {  
    ...  
}
```

Dans le cas de plusieurs tests successifs portant sur une Même variable, on utilisera plutôt **le test switch**.

**Remarque :** Si le corps du test ne comporte qu'une *instruction*, les accolades {} sont optionnels, ( contrairement au Perl).

# Les Structures de contrôles

## Le test SWITCH

Le **switch** n'a pas d'équivalent en Perl. Il est l'équivalent du *SELECT CASE* en Basic.

Il permet de confronter une variable à plusieurs valeurs prédéfinies.

Il permet un code plus compact et lisible qu'un test :

*if-elseif-elseif...*

### Syntaxe :

```
switch( [variable] ) {  
    case [valeur1] :  
        [bloc d'instructions]  
        break;  
    case [valeur2] :  
        [bloc d'instructions]  
        break;  
    ...  
    default:  
        [bloc d'instructions]  
}
```

# Les Fonctions

## Déclaration :

La syntaxe de déclaration s'appuie sur le mot clé **function**. Ce mot clé est immédiatement suivi du nom de la fonction par lequel on va l'appeler depuis n'importe quel endroit du code PHP, puis des parenthèses destinées à accueillir les éventuels paramètres.

```
function bonjour() {  
    echo " Bonjour ";  
}  
.....  
bonjour(); // Affiche " Bonjour " à l'écran.
```

# Les Fonctions

Les fonctions peuvent ou non renvoyer un résultat. on utilise l'instruction return. La variable renournée peut être de type quelconque. Elle est transmise par copie..

```
function bonjour2() {  
    return " Bonjour ";  
}  
.....  
echo bonjour2(); // Affiche " Bonjour " à l'écran.
```

Le mode de fonctionnement est sensiblement différent, la fonction *bonjour* affiche directement le mot " Bonjour " à l'écran, alors que s'affiche le résultat de *bonjour2*.

# Les inclusions de fichiers

- ***include*** :
  - Semblable aux *include* du C/C++
  - Réalise une inclusion physique du fichier demandé
- ***include\_once*** :
  - identique au include
  - protège contre d'éventuelles inclusions multiples
  - qui pourraient mener à des erreurs (redéclarations, etc.)

```
<?php include_once ("connect.php"); ?>
```

# Les inclusions de fichiers

- ***require* et *require\_once* :**
  - fonctionnent comme le include et le include\_once respectivement
  - mais le programme s'arrête si le fichier inclus n'existe pas
- ***dirname()***
  - Pour savoir dans quel répertoire on se trouve on peut utiliser la fonction PHP *dirname()*

```
<?php  
require("malib.php");  
require_once("connect.php");  
?>
```

# Formulaires

- Les superglobales **PHP \$ \_GET et \$ \_POST** sont utilisés pour collecter des données de formulaire.
- **GET** et **POST** créent tous deux un tableau
- Ce tableau contient des paires clé / valeur, où clés sont les noms des contrôles de formulaire (l'attribut **name**) et les valeurs, les données d'entrée de l'utilisateur.
- Les informations envoyées à partir d'un formulaire avec la méthode **GET** sont **visibles par tous** dans l'**URL**.
- **GET** a également des limites sur la quantité d'informations à envoyer. La limitation est d'environ 2000 caractères.
- **GET** peut être utilisé pour envoyer des données non sensibles.
- **Remarque:** **GET** ne doit JAMAIS être utilisé pour envoyer des mots de passe ou d'autres informations sensibles!

# Formulaires

- Les informations envoyées à partir d'un formulaire à l'aide de la méthode POST sont **invisibles pour** tous (tous les noms / valeurs sont incorporés dans le corps de la requête HTTP) et **ne limitent pas** le nombre d'informations à envoyer.
- De plus, POST prend en charge des fonctionnalités avancées telles que la prise en charge d'une entrée binaire en plusieurs parties lors du téléchargement de fichiers sur le serveur.

# Formulaires

- L'exemple ci-dessous affiche un simple formulaire HTML avec deux champs de saisie et un bouton d'envoi:

```
<form action="welcome.php" method="post">
    Name: <input type="text" name="name"><br>
    E-mail: <input type="text" name="email"><br>
    <input type="submit">
</form>
```

- Lorsque l'utilisateur remplit le formulaire ci-dessus et clique sur le bouton Envoyer, les données du formulaire sont envoyées pour traitement dans un fichier PHP nommé "**welcome.php**".
- Les données de formulaire sont envoyées avec la méthode HTTP POST.
- Pour afficher les données soumises, vous pouvez simplement faire écho à toutes les variables.

# Formulaires

- Le "welcome.php" ressemble à ceci:

```
Welcome <?php echo $_POST["name"]; ?><br>
Your email address is: <?php echo $_POST["email"]; ?>
```

- La sortie pourrait être quelque chose comme ceci:

Welcome John

Your email address is john.doe@example.com

# Programmation Orienté Objet en PHP

- Les objets existent en **PHP** à partir de la version 4
- Changements importants en **PHP 5** : Convergence vers le modèle objet de Java
  - Introduction comme en Java **d'interfaces** et de **classes abstraites**
  - emploi des modificateurs *private* et *public* comme en **java**
  - On retrouve aussi **`__toString()`**, **`__clone()`** et un mécanisme de traitement des exceptions semblable à celui de Java.
  - Les constructeurs s'appellent désormais : **`__construct()`** et les destructeurs **`__destruct()`**
  - les méthodes portent le mot clef ***function*** mais ne signalent pas leur type de retour
  - les commentaires de documentation se font à la manière de **Java**
- Les objets s'instancient avec le mot clef **`new`** comme en **Java** ou **C++** mais on utilise **`->`** pour signifier l'appel d'une **méthode**.
- Le **`.`** est déjà pris pour la concaténation des chaînes

# Programmation Orienté Objet en PHP

- Fabriquons maintenant un objet simple en PHP. Ecrivons un objet représentant un étudiant avec ses données :
  - Identifiant
  - Nom
  - date de naissance
- et des méthodes pour opérer sur ces données :
  - Constructeur
  - getters et setters
  - equals()
  - toString() pour affichage

# Programmation Orienté Objet en PHP

```
1 <?php
2     /**
3      *
4      */
5     class Etudiant
6     {
7         /** Identification unique d'un etudiant */
8         protected $etudiant_id;
9         /** Nom de l'etudiant */
10        protected $nom;
11        /** Date de naissance de l'etudiant */
12        protected $naissance;
13        function __construct($id,$nom,$naissance)
14        {
15            $this->etudiant_id = (int) $id; // cast vers integer
16            $this->nom = (string) $nom; // cast vers string
17            $this->naissance= (int) $naissance; // cast vers date(timestamp)
18        }
19        /**
20         * Fonction de comparaison simplifiee entre etudiants
21         * == comparera id, nom et naissance
22         */
```

# Programmation Orienté Objet en PHP

```
23     public function equals(etudiant $etudiant){
24         return ($this->getId() == $etudiant->getId());
25     }
26     public function getId(){
27         return $this->etudiant_id;
28     }
29     public function getNom(){
30         return $this->nom;
31     }
32     public function getNaissance(){
33         return $this->naissance;
34     }
35     public function __toString(){
36         setlocale(LC_TIME, "fr_FR");
37         $ne=strftime('%A %d %B %Y' , $this->naissance);
38         return 'etudiant: id=' . $this->getId() . ', nom=' .
39                $this->getNom() . " $ne";
40     }
41 }
42 /* Test : */
43 date_default_timezone_set('Europe/Paris' );
44 $etu=new Etudiant(234, "Talon", time());
45 var_dump($etu);
46 echo "<br/>";
47 echo $etu;
48 ?>
```

# Programmation Orienté Objet en PHP

- **Indication :**

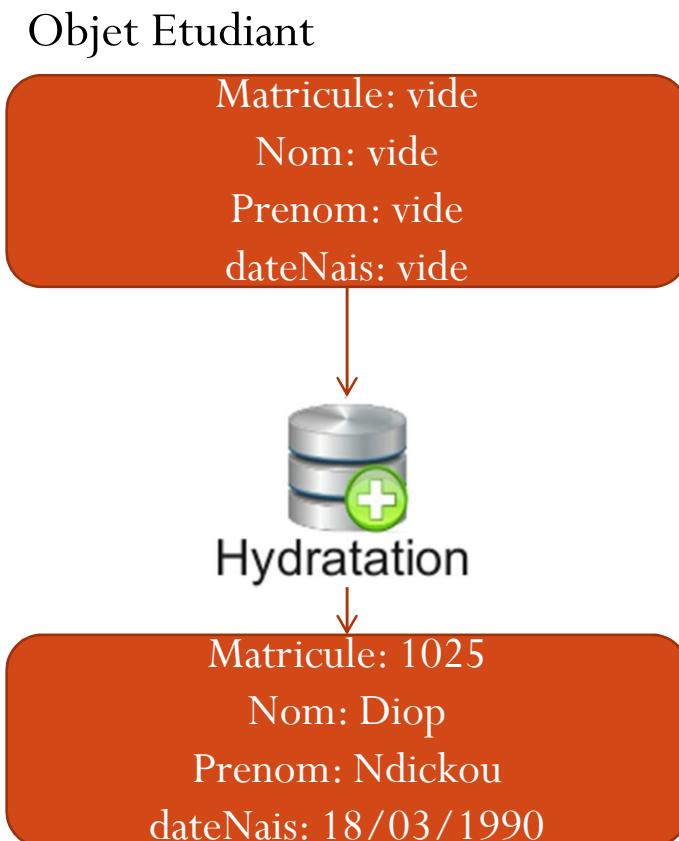
- Lorsqu'on utilise l'opérateur de comparaison `==` sur des objets, ceux-ci sont comparés en utilisant les règles suivantes :
  - deux objets sont égaux s'ils sont **instances de la même classe et ont les mêmes attributs et valeurs**, les valeurs étant comparées avec l'opérateur `==`.
- Lors de l'utilisation de l'opérateur d'identité `== ==`
- les objets sont identiques **uniquement s'ils font référence à la même instance de la même classe.**

# L'hydratation

- L'hydratation est un point essentiel dans le domaine de la **POO**, notamment lorsqu'on utilise des objets représentant des données stockées.
- Quand on vous parle d'hydratation, c'est qu'on parle d'« objet à hydrater ».
- Hydrater un objet, c'est tout simplement lui apporter ce dont il a besoin pour fonctionner.
- En d'autres termes plus précis, hydrater un objet revient à lui fournir des données correspondant à ses **attributs** pour qu'il assigne les valeurs souhaitées à ces derniers.
- L'objet aura ainsi des attributs valides et sera en lui-même valide.
- On dit que **l'objet** a ainsi été **hydraté**

# L'hydratation

- Schématique on a:



# L'hydratation

```
16     function __construct(array $donnee)
17     {
18         $this->hydrate($donnee);
19     }
20     /**
21      * la fonction hydrate|
22      */
23     public function hydrate(array $donnees){
24         foreach ($donnees as $key => $value)
25         {
26             // On récupère le nom du setter correspondant à l'attribut.
27             $method = 'set' . ucfirst($key);
28
29             // Si le setter correspondant existe.
30             if (method_exists($this, $method))
31             {
32                 // On appelle le setter.
33                 $this->$method($value);
34             }
35         }
36     }
```

# L'accès aux données MySQL

- Avec PHP, vous pouvez vous connecter et manipuler des bases de données.
- MySQL est le système de base de données le plus populaire utilisé avec PHP
- PHP 5 et versions ultérieures peuvent fonctionner avec une base de données MySQL en utilisant:
  - **Extension MySQLi** (le "i" signifie amélioré)
  - **PDO (PHP Data Objects)**
- Les versions précédentes de PHP utilisaient l'extension MySQL. Cependant, cette extension a été obsolète en 2012.

# L'accès aux données MySQL

- Avec PHP, vous pouvez vous connecter et manipuler des bases de données.
- MySQL est le système de base de données le plus populaire utilisé avec PHP
- PHP 5 et versions ultérieures peuvent fonctionner avec une base de données MySQL en utilisant:
  - **Extension MySQLi** (le "i" signifie amélioré)
  - **PDO (PHP Data Objects)**
- Les versions précédentes de PHP utilisaient l'extension MySQL. Cependant, cette extension a été obsolète en 2012.

# Création de base de donnée avec MySQLi et PDO

- Créer une base de données MySQL en utilisant MySQLi et PDO
- L'instruction CREATE DATABASE est utilisée pour créer une base de données dans MySQL.
- Les exemples suivants créent une base de données nommée "myDB":

# Création de base de donnée MySQLi

```
1 <?php
2     $servername = "localhost";
3     $username = "root";
4     $password = "";
5
6     // Create connection
7     $conn = new mysqli($servername, $username, $password);
8     // Check connection
9     if ($conn->connect_error) {
10         die("Connection failed: " . $conn->connect_error);
11     }
12
13     // Create database
14     $sql = "CREATE DATABASE myDB";
15     if ($conn->query($sql) === TRUE) {
16         echo "Database created successfully";
17     } else {
18         echo "Error creating database: " . $conn->error;
19     }
20
21     $conn->close();
22 ?>
```

# Création de base de donnée MySQLi

```
1 <?php
2     $servername = "localhost";
3     $username = "root";
4     $password = "";
5
6     // Create connection
7     $conn = new mysqli($servername, $username, $password);
8     // Check connection
9     if ($conn->connect_error) {
10         die("Connection failed: " . $conn->connect_error);
11     }
12
13     // Create database
14     $sql = "CREATE DATABASE myDB";
15     if ($conn->query($sql) === TRUE) {
16         echo "Database created successfully";
17     } else {
18         echo "Error creating database: " . $conn->error;
19     }
20
21     $conn->close();
22 ?>
```

# Créer une table MySQL avec MySQLi et PDO

- L'instruction CREATE TABLE permet de créer une table dans MySQL.
- Nous allons créer une table nommée « Clients », avec cinq colonnes: "id", "prénom", "nom", "email" et "reg\_date":

```
1 CREATE TABLE clients(
2     id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
3     firstname VARCHAR(30) NOT NULL,
4     lastname VARCHAR(30) NOT NULL,
5     email VARCHAR(50),
6     reg_date TIMESTAMP
7 )
```

# Créer une table MySQL avec MySQLi

```
2      $servername = "localhost";
3      $username = "root";
4      $password = "";
5      $dbname = "myDB";
6
7      // Create connection
8      $conn = new mysqli($servername, $username, $password, $dbname);
9      // Check connection
10     if ($conn->connect_error) {
11         die("Connection failed: " . $conn->connect_error);
12     }
13     // sql to create table
14     $sql = "CREATE TABLE clients (
15         id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
16         firstname VARCHAR(30) NOT NULL,
17         lastname VARCHAR(30) NOT NULL,
18         email VARCHAR(50),
19         reg_date TIMESTAMP
20     )";
21
22     if ($conn->query($sql) === TRUE) {
23         echo "Table clients created successfully";
24     } else {
25         echo "Error creating table: " . $conn->error;
26     }
```

# Créer une table MySQL avec PDO

```
1 <?php
2     $servername = "localhost";
3     $username = "root";
4     $password = "";
5     $dbname = "myDBPDO";
6     try {
7         $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
8         // set the PDO error mode to exception
9         $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
10        // sql to create table
11        $sql = "CREATE TABLE clients (
12            id INT(6) UNSIGNED AUTO_INCREMENT PRIMARY KEY,
13            firstname VARCHAR(30) NOT NULL,
14            lastname VARCHAR(30) NOT NULL,
15            email VARCHAR(50),
16            reg_date TIMESTAMP
17        )";
18        // use exec() because no results are returned
19        $conn->exec($sql);
20        echo "Table clients created successfully";
21    }
22    catch(PDOException $e){
23        echo $sql . "<br>" . $e->getMessage();
24    }
25    $conn = null;
26 ?>
```

Activer Windows  
Accédez aux paramètres po

# Insérer des données dans MySQL avec MySQLi et PDO

- Une fois la base de données et la table créées, nous pouvons commencer à y ajouter des données.
- Voici quelques règles de syntaxe à suivre:
  - La requête **SQL** doit être citée en **PHP**
  - Les valeurs de chaîne dans la requête **SQL** doivent être citées
  - Les valeurs numériques ne doivent pas être citées
  - Le mot **NULL** ne doit pas être cité
- L'instruction **INSERT INTO** est utilisée pour ajouter de nouveaux enregistrements à une table **MySQL**

# Insérer des données MySQLi

```
1 <?php
2     $servername = "localhost";
3     $username = "root";
4     $password = "";
5     $dbname = "myDB";
6
7     // Create connection
8     $conn = new mysqli($servername, $username, $password, $dbname);
9     // Check connection
10    if ($conn->connect_error) {
11        die("Connection failed: " . $conn->connect_error);
12    }
13
14    $sql = "INSERT INTO clients (firstname, lastname, email)
15          VALUES ('John', 'Doe', 'john@example.com')";
16
17    if ($conn->query($sql) === TRUE) {
18        echo "New record created successfully";
19    } else {
20        echo "Error: " . $sql . "<br>" . $conn->error;
21    }
22
23    $conn->close();
24 ?>
```

# Insérer des données PDO

```
1 <?php
2     $servername = "localhost";
3     $username = "root";
4     $password = "";
5     $dbname = "myDBPDO";
6
7     try {
8         $conn = new PDO("mysql:host=$servername;dbname=$dbname", $username, $password);
9         // set the PDO error mode to exception
10        $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
11        $sql = "INSERT INTO clients (firstname, lastname, email)
12            VALUES ('John', 'Doe', 'john@example.com')";
13        // use exec() because no results are returned
14        $conn->exec($sql);
15        echo "New record created successfully";
16    }
17    catch(PDOException $e)
18    {
19        echo $sql . "<br>" . $e->getMessage();
20    }
21
22    $conn = null;
23 ?>
```

# Obtenir le dernier Id insérer

- MySQLi

```
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com');

if ($conn->query($sql) === TRUE) {
    $last_id = $conn->insert_id;
    echo "New record created successfully. Last inserted ID is: " . $last_id;
} else {
    echo "Error: " . $sql . "<br>" . $conn->error;
}
```

- PDO

```
$sql = "INSERT INTO MyGuests (firstname, lastname, email)
VALUES ('John', 'Doe', 'john@example.com')";
// use exec() because no results are returned
$conn->exec($sql);
$last_id = $conn->lastInsertId();
echo "New record created successfully. Last inserted ID is: " . $last_id;
`
```

# Sélectionner des données d'une base de données MySQL

- L'instruction **SELECT** permet de sélectionner des données dans une ou plusieurs tables:
  - **SELECT column\_name(s) FROM table\_name**
- Ou nous pouvons utiliser le caractère **\*** pour sélectionner TOUTES les colonnes d'un tableau:
  - **SELECT \* FROM table\_name**

# Sélectionner des données d'une base de données MySQL

- L'instruction **SELECT** permet de sélectionner des données dans une ou plusieurs tables:
  - **SELECT column\_name(s) FROM table\_name**
- Ou nous pouvons utiliser le caractère **\*** pour sélectionner TOUTES les colonnes d'un tableau:
  - **SELECT \* FROM table\_name**

# Sélectionner des données d'une base de données MySQL

- MySQLi

```
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "
<br>";
    }
} else {
    echo "0 results";
}
```

# Sélectionner des données d'une base de données MySQL

- MySQLi

```
$sql = "SELECT id, firstname, lastname FROM MyGuests";
$result = $conn->query($sql);

if ($result->num_rows > 0) {
    // output data of each row
    while($row = $result->fetch_assoc()) {
        echo "id: " . $row["id"]. " - Name: " . $row["firstname"]. " " . $row["lastname"]. "
<br>";
    }
} else {
    echo "0 results";
}
```

# Http et la conservation d'informations sur les clients

- Problème
  - Le protocole **HTTP** est un protocole *sans mémoire*. Une requête d'un client ou les informations qui y sont rattachées ne sont pas mémorisées par défaut.
  - Plusieurs techniques ont été développées pour remédier à ce manquement :
    - Envoyer de l'information sur **l'URL**
    - Utiliser un champ caché **HTML**
    - Utiliser des **Cookies**
    - Utiliser des **Sessions**

# Http et la conservation d'informations sur les clients

- Envoi d'information par **URL**
  - Les variables ne transitent pas toujours via un formulaire mais bien souvent par l'**URL** via la méthode **GET**.
  - Dans ce cas, les variables et les valeurs qu'elles prennent sont déclarées directement dans l'**URL** c'est à dire via la balise de lien HTML **<a href="URL">**.
  - Les variables sont ensuite exploitable sur la page cible en **PHP**.
  - Exemple:
  - **<a href="page.php?nom=dia&prenom=toto">page</a>**
  - **Header('location:page.php?id=5')**

# Http et la conservation d'informations sur les clients

- Utiliser les **COOKIES**
  - L'information est stockée dans un petit fichier texte sur le navigateur du client.
  - On peut par exemple s'en servir pour stocker un identifiant de **session**, un **login**, un **compteur** de visites ou encore mesurer un **temps** de connexion.
  - La syntaxe est: **setCookie("variable","valeur");**
  - On utilise la variable superglobals pour récupérer le cookie:  
**\$\_COOKIE['variable']**

# Http et la conservation d'informations sur les clients

- Utiliser les **COOKIES**

```
<?php
    if (isset($_COOKIE[ 'compteur' ])){
        $message = "Vous etes deja venu ". $_COOKIE[ 'compteur' ] . " fois
        ·-><br/>\n";
        $valeur = $_COOKIE[ 'compteur' ] + 1;
    }
    else{
        $message = "Je vous met un petit cookie<br/>\n";
        $valeur = 1;
    }
    setCookie("compteur", $valeur);
    echo $message;
?>
```

# Http et la conservation d'informations sur les clients

- Utiliser les **SESSIONS**
  - Mais si on a des informations en plus grand nombre à stocker ou qui revêtent un caractère plus sensible, on préférera les stocker essentiellement côté serveur et utiliser le mécanisme plus complet des sessions.
  - Celui-ci consiste à utiliser le tableau associatif `_SESSION[]` qui permet de stocker toute sorte d'informations associées à ce client (données de type nombres ou chaînes, tableaux ou objets PHP)
  - **session\_start()**: Cette fonction devra être placée en tête de page (car envoyé dans le header) sur chaque page qui doit utiliser les sessions. Attention ! Aucun caractère en devra être affiché avant cette fonction.
  - **session\_destroy()**: Cette fonction permet de tuer toutes les variables de session créées.

# Http et la conservation d'informations sur les clients

- Utiliser les SESSIONS

```
<?php
// session1.php
session_start();
if (!isset($_SESSION['cpt']))
    $_SESSION['cpt']=0;
else
    $_SESSION['cpt]++;
echo "Vous avez vu cette page ".$_SESSION['cpt']."' fois <br/>\n";
echo "Le SID courant est " . session_id();
echo "<br/> <a href=\"session2.php\">Aller à la page suivante,
    ↵session2.php</a>";
?>
```

# Troisième Partie

---

Aller plus loin avec bootstrap

# introduction

- **À l'origine, projet interne lancé au sein de l'entreprise Twitter**
- **Août 2011 Bootstrap V1, Janvier 2012 Bootstrap V2, Août 2013 Bootstrap V3**
- **Bootstrap est un frameworks HTML/CSS**
- **Bibliothèque complète de fonctionnalités prêtes à l'emploi**
- **Concevoir un site web rapidement**
- **Sites web adaptatifs (Responsive Web Design)**

# plan

- **Mise en place**
- **Créer une première page**
- **Fonctionnalités**
- **Sites web adaptatifs (Responsive Web Design)**

# Mise en place

- Téléchargement sur <http://getbootstrap.com/>
- Ou Bootstrap CDN

```
<! -- Latest compiled and minified CSS -->
<link rel="stylesheet"
      href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/css/bootstrap.min.css">
<! -- Latest compiled and minified JavaScript -->
<script
      src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.5/js/bootstrap.min.js">
</script>
```

# Créer première page web avec bootstrap

```
1 <!DOCTYPE html>
2 <html lang="fr">
3     <head>
4         <meta charset="utf-8">
5         <meta name="viewport" content="width=device-width, initial-scale=1">
6         <title>Premiere page avec Bootstrap</title>
7         <link href="css/bootstrap.min.css" rel="stylesheet">
8         <script src="js/bootstrap.min.js"></script>
9     </head>
10    <body>
11        <div class="container">
12            <p>Premiere page avec Bootstrap</p>
13        </div>
14    </body>
15 </html>
```

# Typographie

- Bootstrap propose une large bibliothèque d'éléments standard, par exemple les titres, les paragraphes, les messages d'informations, les badges et les labels

```
<h1>h1. Bootstrap heading</h1>
<h2>h2. Bootstrap heading</h2>
<h3>h3. Bootstrap heading</h3>
<h4>h4. Bootstrap heading</h4>
<h5>h5. Bootstrap heading</h5>
<h6>h6. Bootstrap heading</h6>
```

h1. Bootstrap heading

Semibold 36px

h2. Bootstrap heading

Semibold 30px

h3. Bootstrap heading

Semibold 24px

h4. Bootstrap heading

Semibold 18px

h5. Bootstrap heading

Semibold 14px

h6. Bootstrap heading

Semibold 12px

# Les listes

- **Bootstrap** offre différentes possibilités de liste, par exemple les listes basiques, les listes en ligne et les listes de définition

```
<ul class="list-inline">
  <li>...</li>
  <li>...</li>
</ul>

  <dl>
    <dt>...</dt>
    <dd>...</dd>
  </dl>

  <dl class="dl-horizontal">
    <dt>...</dt>
    <dd>...</dd>
  </dl>
```



# Les Boutons

- Bootstrap propose une bibliothèque complète d'options afin de mettre en place des boutons élégants

```
<!-- Standard button -->
<button type="button" class="btn btn-default">Default</button>

<button type="button" class="btn btn-primary">Primary</button>
<button type="button" class="btn btn-success">Success</button>
<button type="button" class="btn btn-info">Info</button>
<button type="button" class="btn btn-warning">Warning</button>
<button type="button" class="btn btn-danger">Danger</button>
<button type="button" class="btn btn-link">Link</button>
```



# Les Formulaires

- Bootstrap permet de créer facilement des formulaires élégants

EXAMPLE

**Email address**

**Password**

**File input**

Aucun fichier choisi  
Example block-level help text here.

Check me out

# Les Formulaires

```
<form>
  <div class="form-group">
    <label for="exampleInputEmail1">Email address</label>
    <input type="email" class="form-control" id="exampleInputEmail1"
      placeholder="Email">
  </div>
  <div class="form-group">
    <label for="exampleInputPassword1">Password</label>
    <input type="password" class="form-control" id="exampleInputPassword1"
      placeholder="Password">
  </div>
  <div class="form-group">
    <label for="exampleInputFile">File input</label>
    <input type="file" id="exampleInputFile">
    <p class="help-block">Example block-level help text here.</p>
  </div>
  <div class="checkbox">
    <label>
      <input type="checkbox"> Check me out
    </label>
  </div>
  <button type="submit" class="btn btn-default">Submit</button>
</form>
```

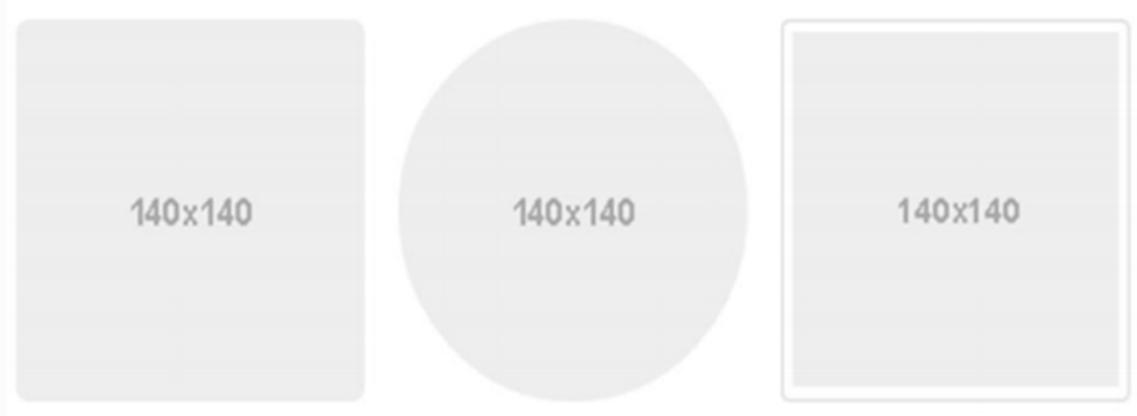
# Les images

- Bootstrap aide à mettre en forme les images sans avoir de connaissances approfondies en CSS

```
  
  

```

## EXAMPLE



# Les tableaux

- Bootstrap propose différentes possibilités de tableaux : **tableaux basiques, condensés et zébrés**

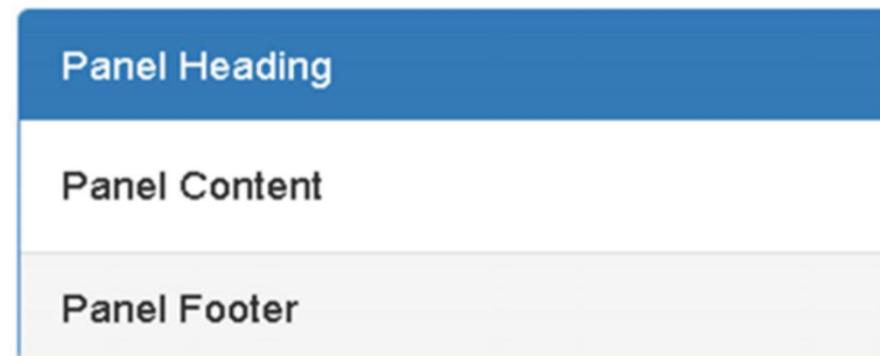
```
<table class="table">  
  ...  
</table>  
  
<table class="table table-condensed">  
  ...  
</table>  
  
<table class="table table-striped">  
  ...  
</table>
```

#	First Name	Last Name	Username
1	Mark	Otto	@mdo
2	Jacob	Thornton	@fat
3	Larry	the Bird	@twitter

# Les Panneaux

- Les panneaux permettent de représenter l'information à l'intérieur d'une boîte avec un entête et un pied

```
<div class="panel panel-primary">  
  
  <div class="panel-heading">  
    <p>Panel Heading</p>  
  </div>  
  <div class="panel-body">  
    <p>Panel Content</p>  
  </div>  
  <div class="panel-footer">  
    <p>Panel Footer</p>  
  </div>  
  
</div>
```



# Les Systèmes de grilles

- **Bootstrap** découpe le corps de la page en une grille composée de lignes de 12 colonnes de largeur égale



# Les Systèmes de grilles

```
<div class="row">
    <div class="col-md-8">.col-md-8</div>
    <div class="col-md-4">.col-md-4</div>
</div>

<div class="row">
    <div class="col-md-4">.col-md-4</div>
    <div class="col-md-4">.col-md-4</div>
    <div class="col-md-4">.col-md-4</div>
</div>

<div class="row">
    <div class="col-md-6">.col-md-6</div>
    <div class="col-md-6">.col-md-6</div>
</div>
```

# Responsive web design

- la balise `<meta>` permet à Bootstrap d'analyser la largeur de l'écran

`<meta name="viewport" content="width=device-width, initial-scale=1">`

Grille	Classes	Taille d'écran	Type d'écran
xs : extra small devices	.col-xs-*	< 768 pixels	Smartphone
sm : small devices	.col-sm-*	≥ 768 pixels	Tablette
md : medium devices	.col-md-*	≥ 992 pixels	Écran de bureau
lg : large devices	.col-lg-*	≥ 1200 pixels	Grand écran de bureau

# •Travaux Pratiques!!!

# Troisième Partie

---

Les CMS (WordPress et prestashop)