

# Cours Création de Site Web

PAR Ir. KARIREKIMANA Dyna

# Matière: Programmation Web

- Objectifs du cours
- Présenter les systèmes d'information dans le contexte Internet. Le module initie à la programmation Web via les langages HTML et PHP.

# Programme du Cours

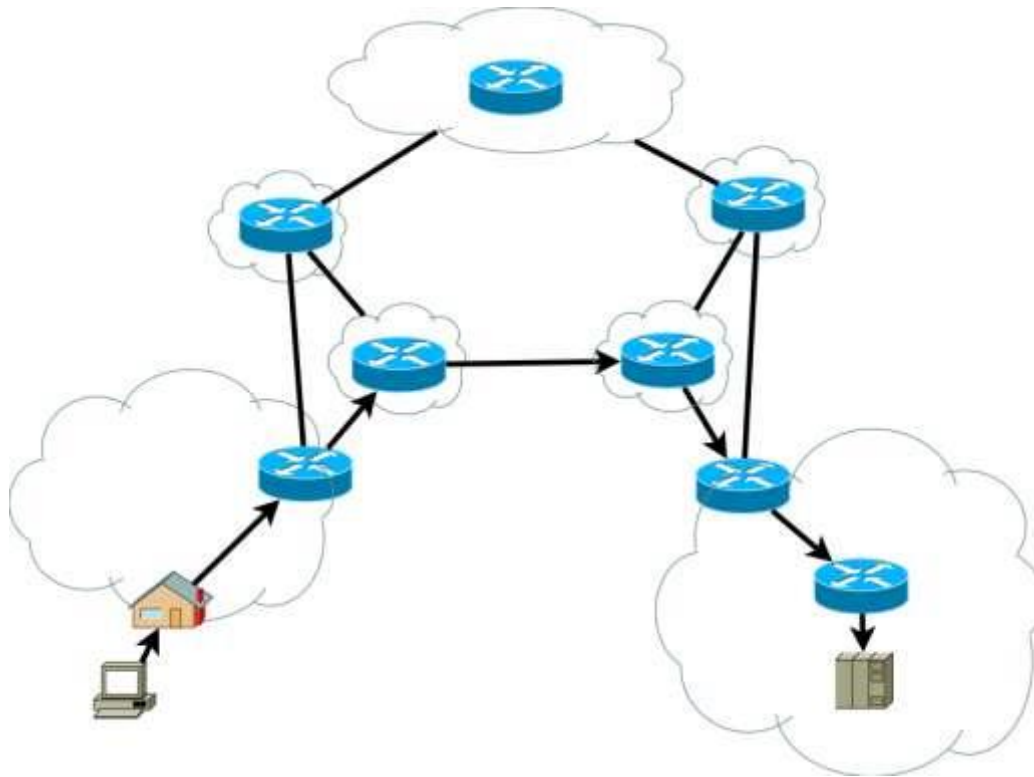
- Chap 1: Introduction au World Wide web
- Chap 2: Le langage HTML
- Chap 3: Le langage CSS
- Chap 4: Création d'un site web statique
- Chap 5: Les Langages PHP/My SQL et leur interaction
- Chap 6: Création d'un site web dynamique

# CHAP 1: INTRODUCTION AU WORLD WIDE WEB

1. Définition et Historique
2. Architecture Client/Serveur
3. Protocole http
4. Notions de base de Web 2.0

# Définition et Historique

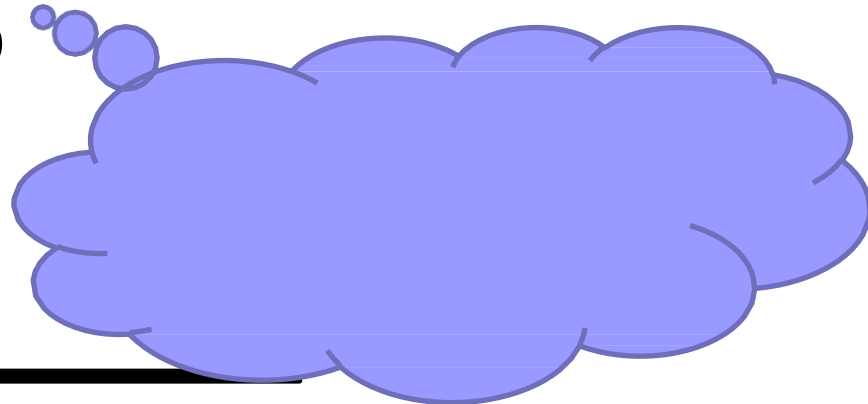
- Que signifie le terme internet?



**Le réseau Internet** est un réseau de machines (ordinateurs, serveurs, etc.) interconnectées et qui offre une variété de services

# Services Internet

- Courrier électronique
- Transfert de fichiers
- Messagerie instantané (skype, Whatsapp)
- Accès à distance (documents/machines)
- Radios, Télévision, multimédias
- World Wide Web



# C'est quoi le World Wide Web ?

- Le ***World Wide Web*** ( littéralement la « toile (d'araignée) mondiale », abrégé **WWW** ou le **Web**), la **toile mondiale** ou la **toile**,
- est un système HyperText public fonctionnant sur Internet.
- Le Web permet de consulter, avec un navigateur, des pages accessibles sur des sites.
- L'image de la toile d'araignée vient des hyperliens qui lient les pages web entre elles.

- communément appelé le Web, le web parfois la Toile ou le WWW, est un système hypertexte public fonctionnant sur Internet qui permet de consulter, avec un navigateur, des pages accessibles sur des sites.
- Le Web n'est qu'une des applications d'Internet. D'autres applications d'Internet sont le courrier électronique, la messagerie instantanée, ...etc.

Web  $\neq$  Internet

- Internet = support de communication
- Web = une partie des contenus circulant sur l'internet



# Internet Versus Web

- Le World Wide Web est une collection de documents.
- Ces documents sont formés avec ce qu'on appelle « HTML », qui signifie HyperText Markup Language.
- HTML est la base pour presque toutes les pages Web, même s'il peut y avoir des:
  - ☐ documents,
  - ☐ images,
  - ☐ vidéos,
  - ☐ etc.

# Historique du Web



## **Tim Berners-Lee**

- employé du CERN
- Inventeur du WWW (1989)

• Le concept du web était conçu initialement pour faciliter la communication au sein de l'entreprise CERN

# Bref historique d'Internet

- 1959-1968 : Programme ARPA
    - le ministère américain de la défense lance un réseau capable de supporter les conséquences d'un conflit nucléaire
  - 1969 : ARPANET, l'ancêtre d'Internet
    - les universités américaines s'équipent de gros ordinateurs et se connectent au réseau ARPANET
  - 1970-1982 : Ouverture sur le monde
    - premières connexions avec la Norvège et Londres
  - 1983 : Naissance d'Internet
    - protocole TCP/IP : tous les réseaux s'interconnectent
    - les militaires quittent le navire
  - 1986 : Les autoroutes de l'Information
    - la National Science Fondation déploie des super-ordinateurs pour augmenter le débit d'Internet
  - 1987-1992 : Les années d'expansion
    - les fournisseurs d'accès apparaissent
    - les entreprises privées se connectent au réseau
  - 1993-2003 : L'explosion d'Internet
    - ouverture au grand public
    - avènement du WEB et du courrier électronique
- } marché considérable

# Naissance du Web (1989-1991)

- ❑ Mars 89 : projet de création d'un hypertexte documentaire distribué sur le réseau du CERN
  - Origine : Tim Berners-Lee, puis Robert Cailliau (1990)
  - Choix des technologies TCP/IP et ouverture de la première connexion du CERN avec Internet
  - Mise au point des **3 technologies de base du Web : URL, HTML et HTTP**
- ❑ Septembre 90 : 1er site Web fonctionnel (mode texte)
  - 1<sup>er</sup> serveur Web : nxoc01.cern.ch
  - 1<sup>er</sup> navigateur Web : WorldWideWeb (rebaptisé plus tard Nexus), développé en Objective C
- ❑ Août 1991 : publication du projet WorldWideWeb dans un message sur UseNet
- ❑ Décembre 91 : démonstration publique à la conférence Hypertext'91 (San Antonio)



Le premier serveur Web,  
un NeXT Cube  
(source : Wikipédia)



# Historique du web

- ❑ 1993 : Mosaic : premier navigateur « grand public »
  - Marc Andreessen, NCSA : plateformes X, puis Mac et Windows
  - affichage d'images (GIF et XBM) dans les pages Web
  - prise en charge de formulaires interactifs
- ❑ 01/10/1994 : création du W3C
  - à l'initiative du CERN (Genève) et du MIT (Boston)
  - président : Tim Berners-Lee
  - but : standardisation et développement du Web
- ❑ 1994 : Apparition des navigateurs privés
  - M. Andreessen crée Netscape Communications Corp.
- ❑ 1995 : Microsoft lance la « guerre des navigateurs »
  - Apparition d'Internet Explorer pour Windows 95
- ❑ 1995 : Altavista : premier « gros » moteur de recherche
- ❑ 1996 : Navigateur Opera
- ❑ 1998 : Apparition de Google
- ❑ 2003 : Apple lance Safari
- ❑ 2004 : Première version de Mozilla Firefox
- ❑ 2004 : première conférence « Web 2.0 »
- ❑ 2008 : Google lance Google Chrome
- ❑ Actuellement : explosion du Web mobile
- ❑ En cours de développement : Web sémantique, de données, des objets...

## Nombre de sites référencés

1990 : 1 (CERN)

1991 : premier site hors d'Europe (SLAC, Stanford)

1992 : 26

Juin 1993 : 130

Juin 1994 : 2738

Juin 1995 : 23 500

Janvier 1996 : ~ 100 000

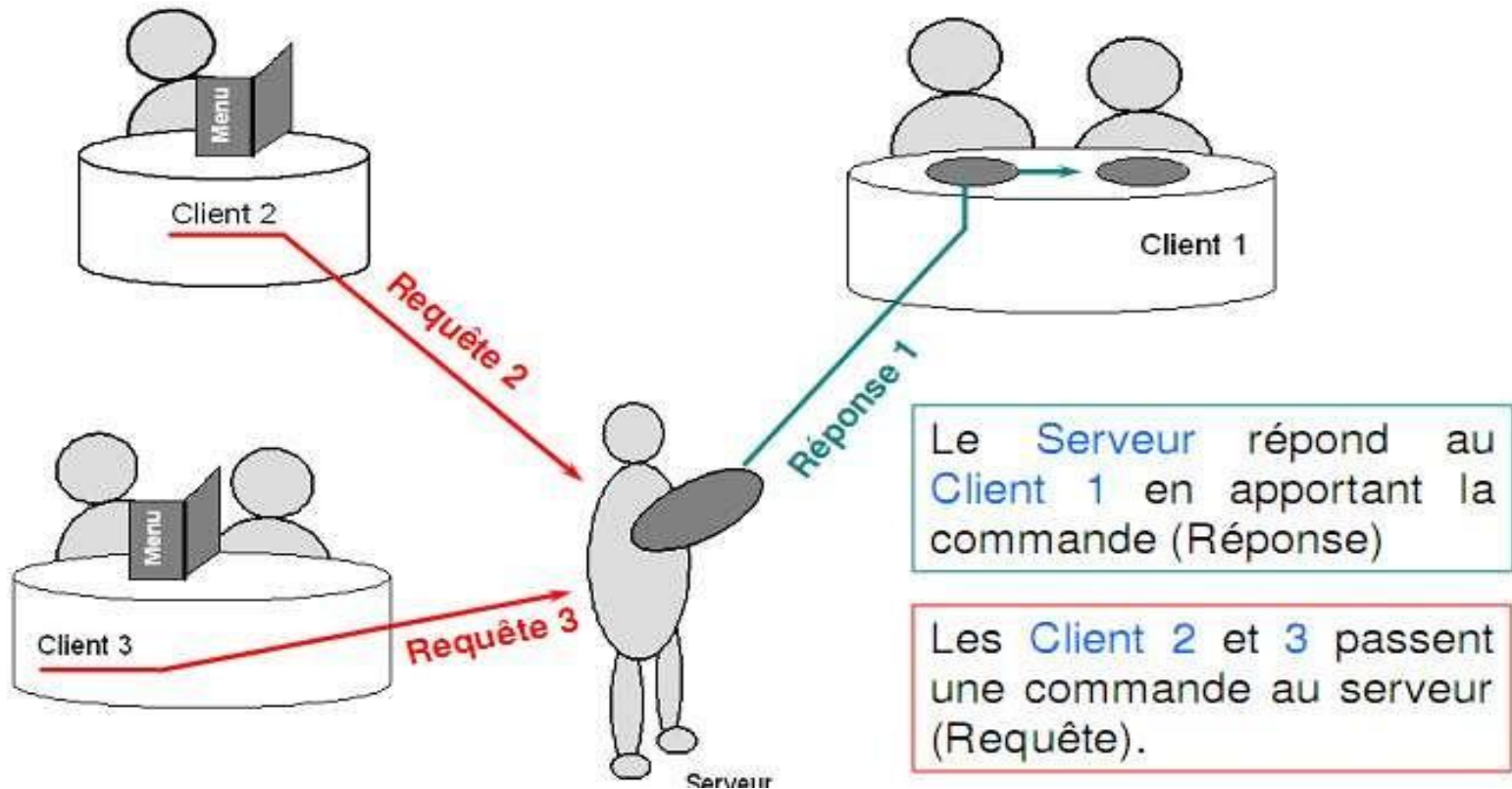
Avril 1997 : > 1 million

Février 2000 : > 11 millions

Février 2007 : > 100 millions

Février 2009 : ~ 216 millions

# Internet et le Modèle Client Serveur



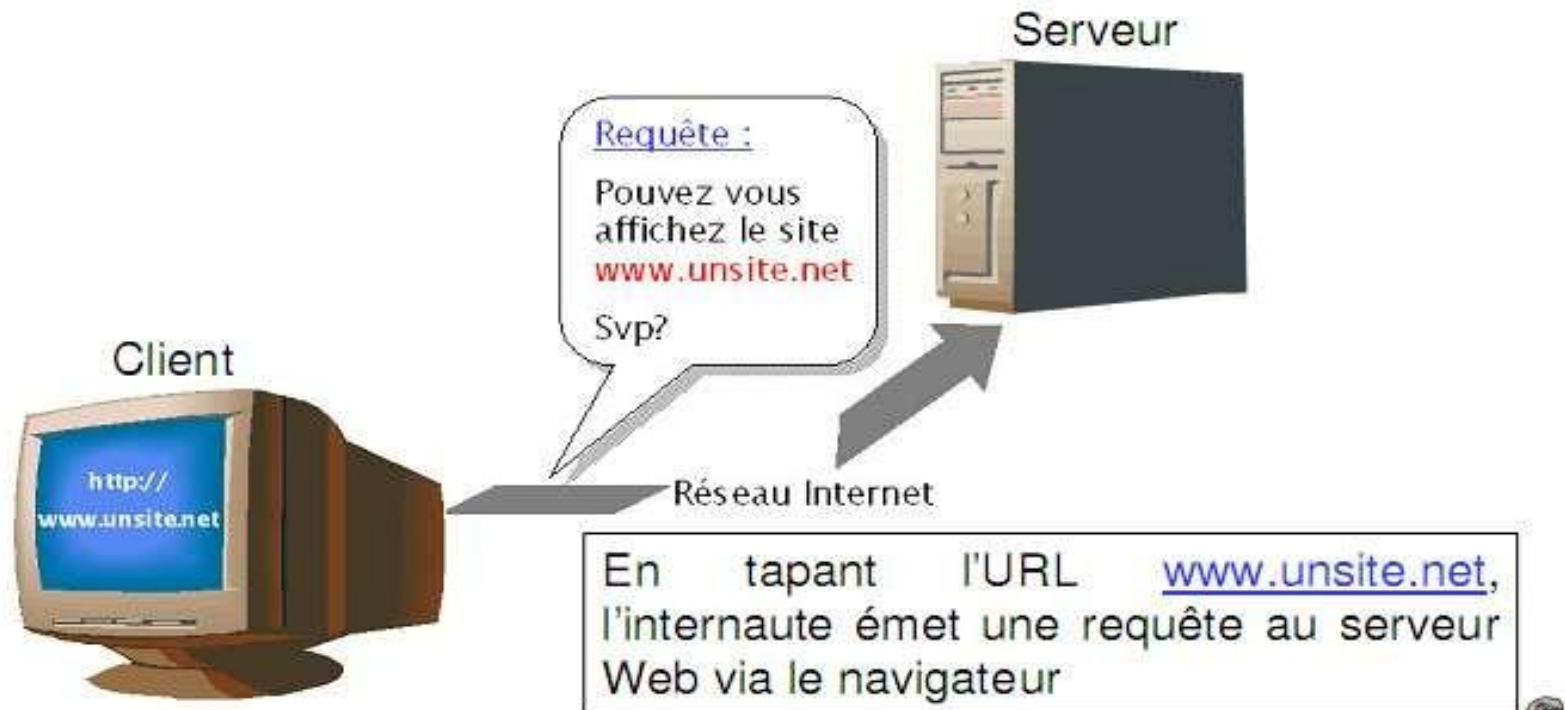
# Internet et le Modèle Client Serveur

L'accès à un site Internet nécessite les éléments suivants:

- Une application cliente( navigateur Web: Mozilla Firefox, Netscape Navigator, Internet Explorer) auquel nous fournissons l'adresse du site recherché du type `http://www.brb.bi`
- Une application serveur (serveur web: nestacape Entreprise server, ApacheHTTP server, IIS-internet Information server, Tomcat).
- Un protocole HTTP

# Internet et le Modèle Client Serveur

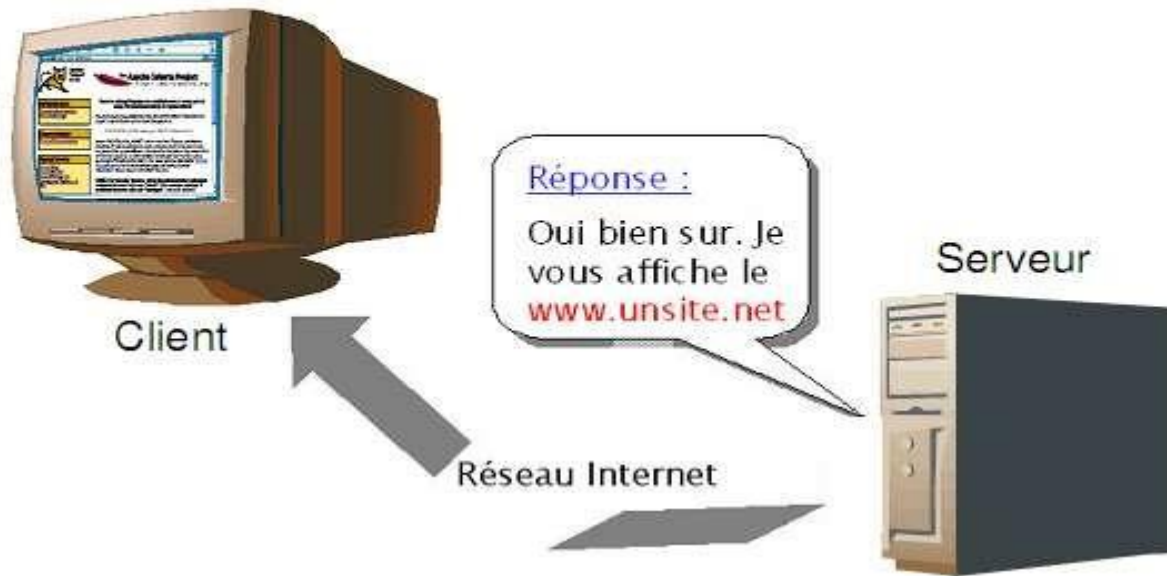
- L'application Cliente envoie une requête au serveur Web désigné par l'**URL** (**U**niform **R**esource **L**ocator) .





# Internet et le Modèle Client Serveur

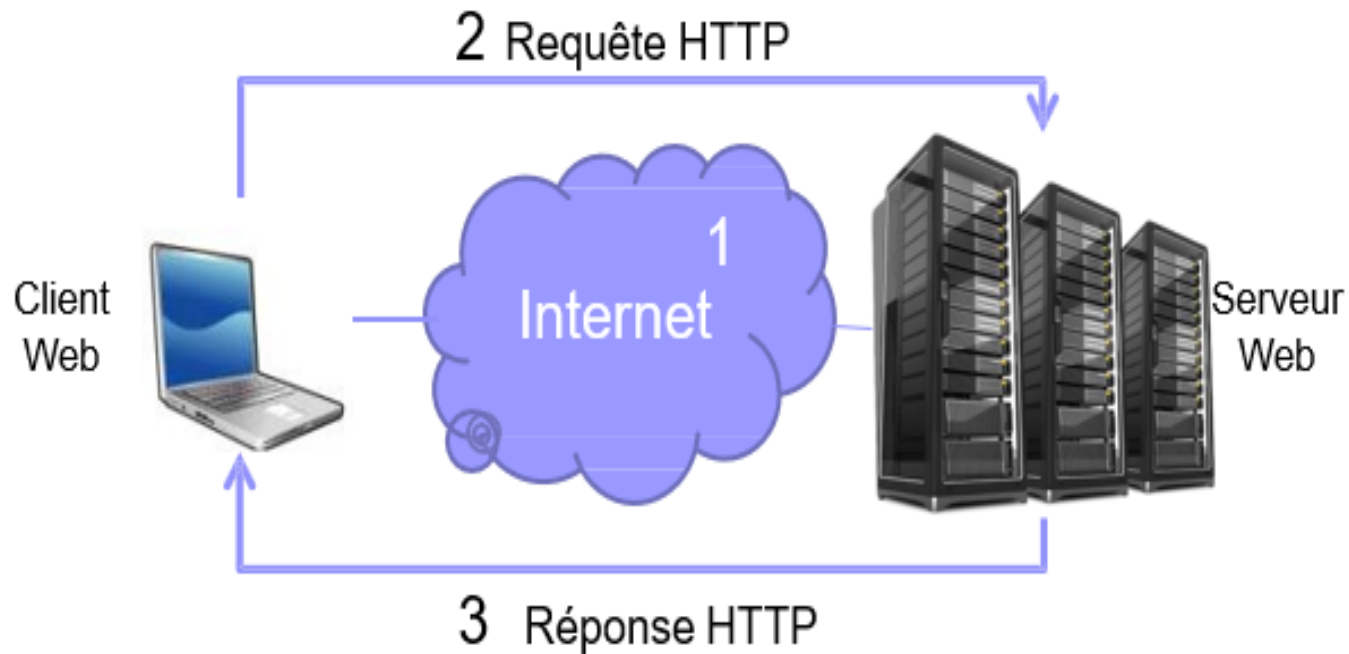
- Le serveur **Web** Contacté répond au client en affichant l'ensemble des informations stockées et organisées sur son disque dur à l'**URL** donnée.



# Internet et le Modèle Client Serveur

- Un client est une application qui se connecte à un autre ordinateur pour obtenir ou modifier des informations à l'aide de requêtes.
- Un serveur est une application située sur un ordinateur très puissant, capable de gérer un grand nombre de requêtes simultanément.
- Un serveur est toujours en attente de requête.

# Comment ca fonctionne?



# Client Web

- **Un Client Web** est l'outil permettant d'accéder aux documents disponibles sur le Web.
- Exemple de client Web: ***Les Navigateurs.***



Firefox  
(Mozilla)



Chrome  
(Google)



Opera  
(Opera)

# Client Web

- **Le fonctionnement d'un navigateur** est de la façon suivante :
  - Le navigateur désigne par une adresse **URL** (*Uniform Resource Locator*)
  - Le serveur répond aux demandes en envoyant les pages au navigateur Web.
  - Le navigateur interprète et affiche alors les pages à l'utilisateur.

# URL

- URL = **U**niform **R**esource **L**ocator
- URL est un **identifiant unique**
- Désigne une **page** ou une **ressource** sur un site web.
- Exemple : <http://www.exemple.com/test-dir/index.html>
- **http://** : Le protocole de communication
- [www.exemple.com](http://www.exemple.com) : Le nom du domaine (DNS, unique)
- **Test-dir** : Répertoire sur la machine
- **Index.html** : Nom du document à afficher sur le navigateur

# Serveur Web

- Un serveur web est un serveur informatique utilisé pour héberger des sites web sur internet.
- La fonction principale d'un serveur Web est de stocker et délivrer des pages Web qui sont généralement écrites en HTML.

# Protocole http

- Le protocole *HTTP* (*HyperText Transfer Protocol*) est le protocole le plus utilisé sur Internet depuis 1990.
- Le but du protocole HTTP est de permettre un transfert de fichiers (essentiellement au format HTML) localisés grâce à un URL entre un navigateur (le client web) et un serveur Web.



# LANGAGES DE PROGRAMMATION POUR LE WEB

La rédaction d'un cahier des charges

- **Première des questions:** un site, pour quoi faire ? Pour quels utilisateurs ?
  - Quel est l'objectif du concepteur du site ?
  - Pour satisfaire cet objectif, il faut satisfaire les besoins des utilisateurs: un utilisateur revient sur un site parce qu'il a trouvé un **contenu** satisfaisant (et non pour la beauté du site)
  - Si trop grande diversité des utilisateurs, aiguillage à l'entrée ?
  - Quel degré d'expertise des utilisateurs ?
    - Un site web doit être en principe conçu pour l'utilisateur le **moins** expert
  - Quel degré d'expertise du **futur** *webmaster* ?
- Principe clé: partir d'abord des besoins des utilisateurs (et pas de ses propres envies !)

- Définir l'**architecture** du site: un site doit être structuré en **rubriques**
  - Son architecture doit apparaître immédiatement au visiteur
- **Cerner les besoins**: éviter un site sur/sous dimensionné
  - Site sur-dimensionné: difficulté à retrouver l'information, site trop sophistiqué par rapport aux besoins des utilisateurs (*information overload*).
  - Site sous-dimensionné: difficulté parfois à faire évoluer le site par la suite

- Du contenu à l'image...
  - La « charte graphique » du site
  - Le choix d'une page modèle
    - Quelle structure ?
      - principe général: il faut que quel que soit la page visitée, l'utilisateur puisse 1) savoir où il est; 2) savoir où trouver une information si il n'est pas à la bonne page
      - Principe du « 3 clics maximum »

# Qu'est-ce qu'un Site Web ?

- Un site web, est un ensemble de pages web et de ressources reliées par des hyperliens, défini et accessible par une adresse web. Un site est développé à l'aide de langages de programmation web, puis hébergé sur un serveur web accessible via le réseau mondial Internet, un intranet local.

# Qu'est-ce qu'un Hyperlien ?

- Un hyperlien ou lien hypertexte, est une référence dans un système hypertexte permettant de passer automatiquement d'un document consulté à un autre document.

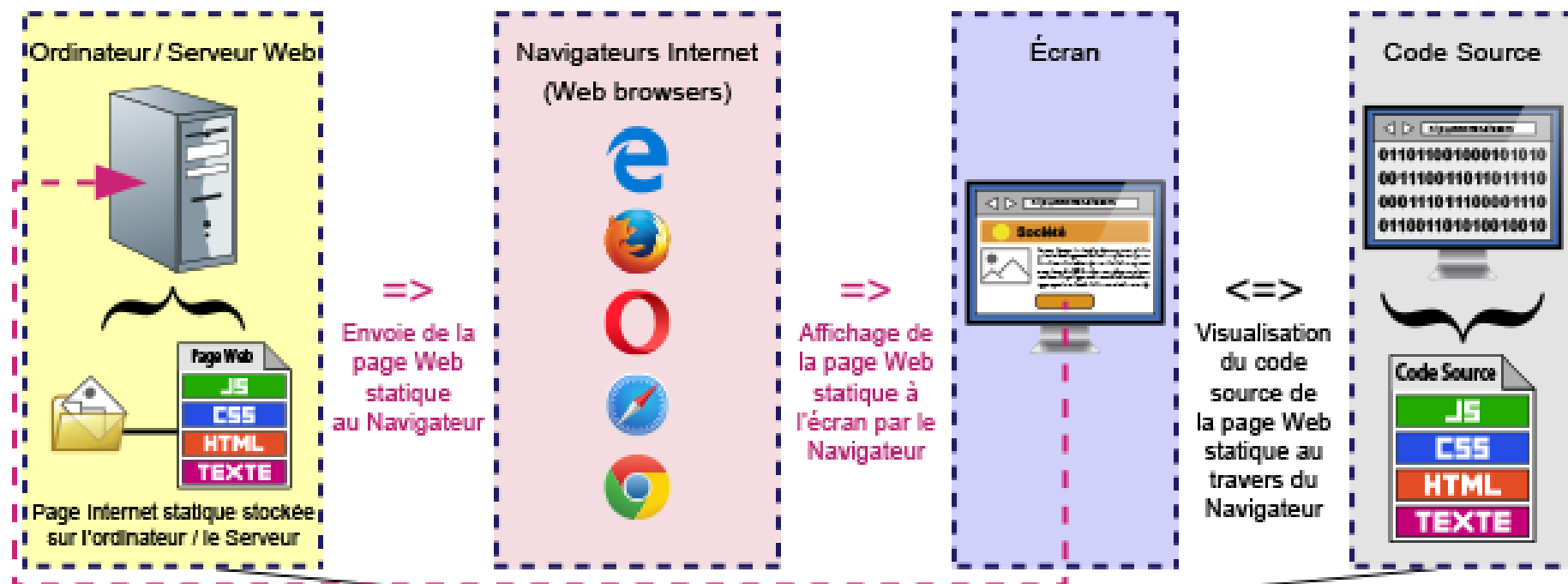
# Page web Statique

- Une page web statique est une page web dont le contenu ne varie pas en fonction des caractéristiques de la demande, c'est-à-dire qu'à un moment donné tous les internautes qui demandent la page reçoivent le même contenu.



# Site web Statique

## Circuit d'une page web statique



### APPEL DE LA PAGE

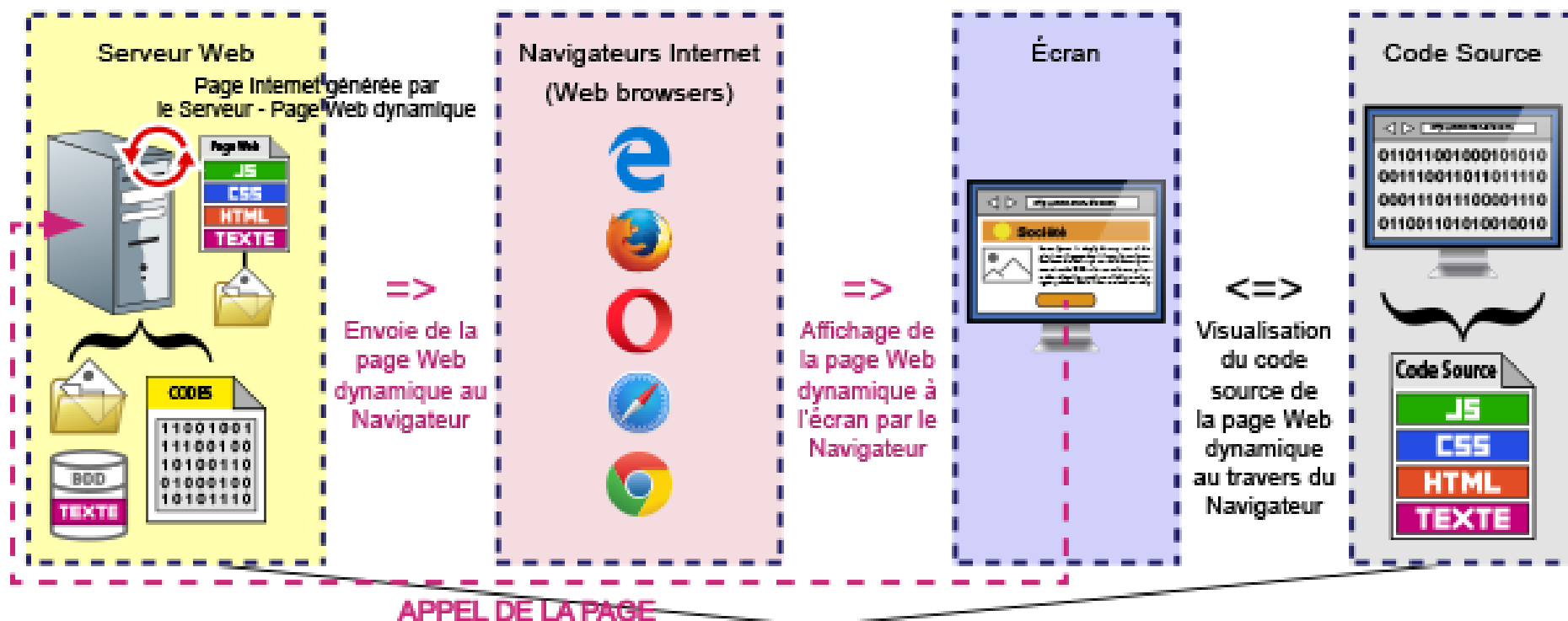
Code de la page Internet stockée sur le Serveur = Code de la page affichée à l'écran au travers du Navigateur  
( Page programmée = Page affichée )

# Page Web Dynamique

- Une page web dynamique est une page web générée à la demande, par opposition à une page web statique. Le contenu d'une page web dynamique peut donc varier en fonction d'informations (heure, nom de l'utilisateur, formulaire rempli par l'utilisateur, etc.) qui ne sont connues qu'au moment de sa consultation. À l'inverse, le contenu d'une page web statique est a priori identique à chaque consultation.

# Page Web Dynamique

## Circuit d'une page web dynamique

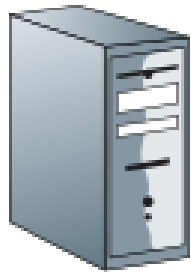


Code(s) servant à générer la page Internet sur le Serveur  $\neq$  Code de la page affichée à l'écran au travers du Navigateur  
( Ce qui est programmée  $\neq$  Page générée, Page générée = Page affichée )

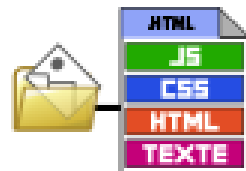
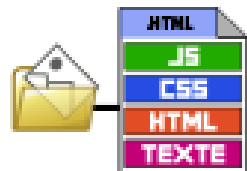
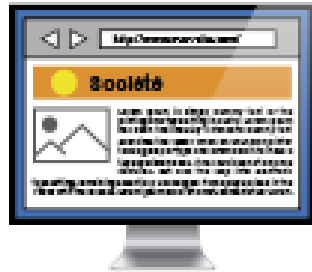
# Comparaison

## Page Statique / Page Dynamique

Ordinateur / Serveur  
Web



Client



Page Web stockée  
sur le Serveur

Page Web affichée

Page Web statique

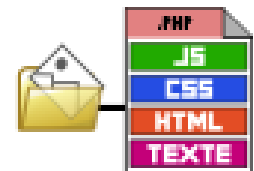
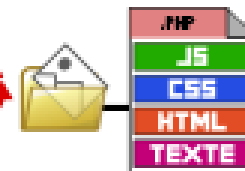
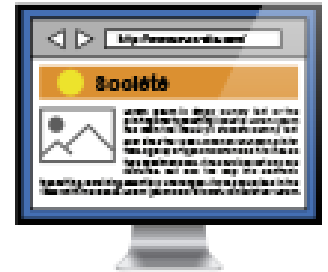
Serveur Web



Génération de la  
page Web  
dynamique lors  
de sa demande



Client



Page Web générée  
par le Serveur Web

Page Web affichée

Page Web dynamique

# LANGAGES DE PROGRAMMATION POUR LE WEB

## Chap 2: HTML

# Le langage HTML + CSS

- HTML = HyperText Markup Language 1991
- Fonction : mise en forme, structurer le contenu et lui donner du sens.
- Le langage HTML est un langage informatique qui permet de créer et structurer des pages Web.
- Par définition, toutes les pages Web sont faites de code HTML, le langage hypertexte.

# Fondations du HTML

## ELEMENTS, BALISES & ATTRIBUTS

- Éléments =
  - Définissent des objets dans notre page web
    - L'élément p définit un paragraphe,
    - Les éléments h1, h2, ... , h6 définissent des titres,
    - L'élément a définit un lien...
  - Généralement constitués d'une paire de balises :
    - Balise ouvrante : <p>
    - Balise fermante : </p>
    - Exception : balises orphelines comme <br/>

# Fondations du HTML

## ELEMENTS, BALISES ET ATTRIBUTS

- Attributs =
  - Propriétés utilisées pour donner des indications supplémentaires aux éléments.
  - Ex: Indiquer la cible d'un lien.





# Fondations du HTML

## STRUCTURE D'UNE PAGE EN HTML5

- Doctype : `<!DOCTYPE html>`
- Éléments :
  - html,
  - head,
  - title,
  - meta,
  - body.

# LES BALISES

Balise	Description
<code>&lt;html&gt;</code>	Balise principale
<code>&lt;head&gt;</code>	En-tête de la page
<code>&lt;body&gt;</code>	Corps de la page

## Balises d'en-tête

Ces balises sont toutes situées dans l'en-tête de la page web, c'est-à-dire entre `<head>` et `</head>` :

Balise	Description
<code>&lt;link /&gt;</code>	Liaison avec une feuille de style
<code>&lt;meta /&gt;</code>	Métadonnées de la page web (charset, mots-clés, etc.)
<code>&lt;script&gt;</code>	Code JavaScript
<code>&lt;style&gt;</code>	Code CSS
<code>&lt;title&gt;</code>	Titre de la page

## Balises de structuration du texte

Balise	Description
<code>&lt;abbr&gt;</code>	Abréviation
<code>&lt;blockquote&gt;</code>	Citation (longue)
<code>&lt;cite&gt;</code>	Citation du titre d'une œuvre ou d'un évènement
<code>&lt;q&gt;</code>	Citation (courte)
<code>&lt;sup&gt;</code>	Exposant
<code>&lt;sub&gt;</code>	Indice
<code>&lt;strong&gt;</code>	Mise en valeur forte
<code>&lt;em&gt;</code>	Mise en valeur normale
<code>&lt;mark&gt;</code>	Mise en valeur visuelle
<code>&lt;h1&gt;</code>	Titre de niveau 1
<code>&lt;h2&gt;</code>	Titre de niveau 2
<code>&lt;h3&gt;</code>	Titre de niveau 3
<code>&lt;h4&gt;</code>	Titre de niveau 4
<code>&lt;h5&gt;</code>	Titre de niveau 5
<code>&lt;h6&gt;</code>	Titre de niveau 6
<code>&lt;img /&gt;</code>	Image
<code>&lt;figure&gt;</code>	Figure (image, code, etc.)
<code>&lt;figcaption&gt;</code>	Description de la figure

<code>&lt;audio&gt;</code>	Son
<code>&lt;video&gt;</code>	Vidéo
<code>&lt;source&gt;</code>	Format source pour les balises <code>&lt;audio&gt;</code> et <code>&lt;video&gt;</code>
<code>&lt;a&gt;</code>	Lien hypertexte
<code>&lt;br /&gt;</code>	Retour à la ligne
<code>&lt;p&gt;</code>	Paragraphe
<code>&lt;hr /&gt;</code>	Ligne de séparation horizontale
<code>&lt;address&gt;</code>	Adresse de contact
<code>&lt;del&gt;</code>	Texte supprimé
<code>&lt;ins&gt;</code>	Texte inséré
<code>&lt;dfn&gt;</code>	Définition
<code>&lt;kbd&gt;</code>	Saisie clavier
<code>&lt;pre&gt;</code>	Affichage formaté (pour les codes sources)
<code>&lt;progress&gt;</code>	Barre de progression
<code>&lt;time&gt;</code>	Date ou heure

## Balises de listes

Cette section énumère toutes les balises HTML permettant de créer des listes (listes à puces, listes numérotées, listes de définitions...)

Balise	Description
<code>&lt;ul&gt;</code>	Liste à puces, non numérotée
<code>&lt;ol&gt;</code>	Liste numérotée
<code>&lt;li&gt;</code>	Élément de la liste à puces
<code>&lt;dl&gt;</code>	Liste de définitions
<code>&lt;dt&gt;</code>	Terme à définir
<code>&lt;dd&gt;</code>	Définition du terme

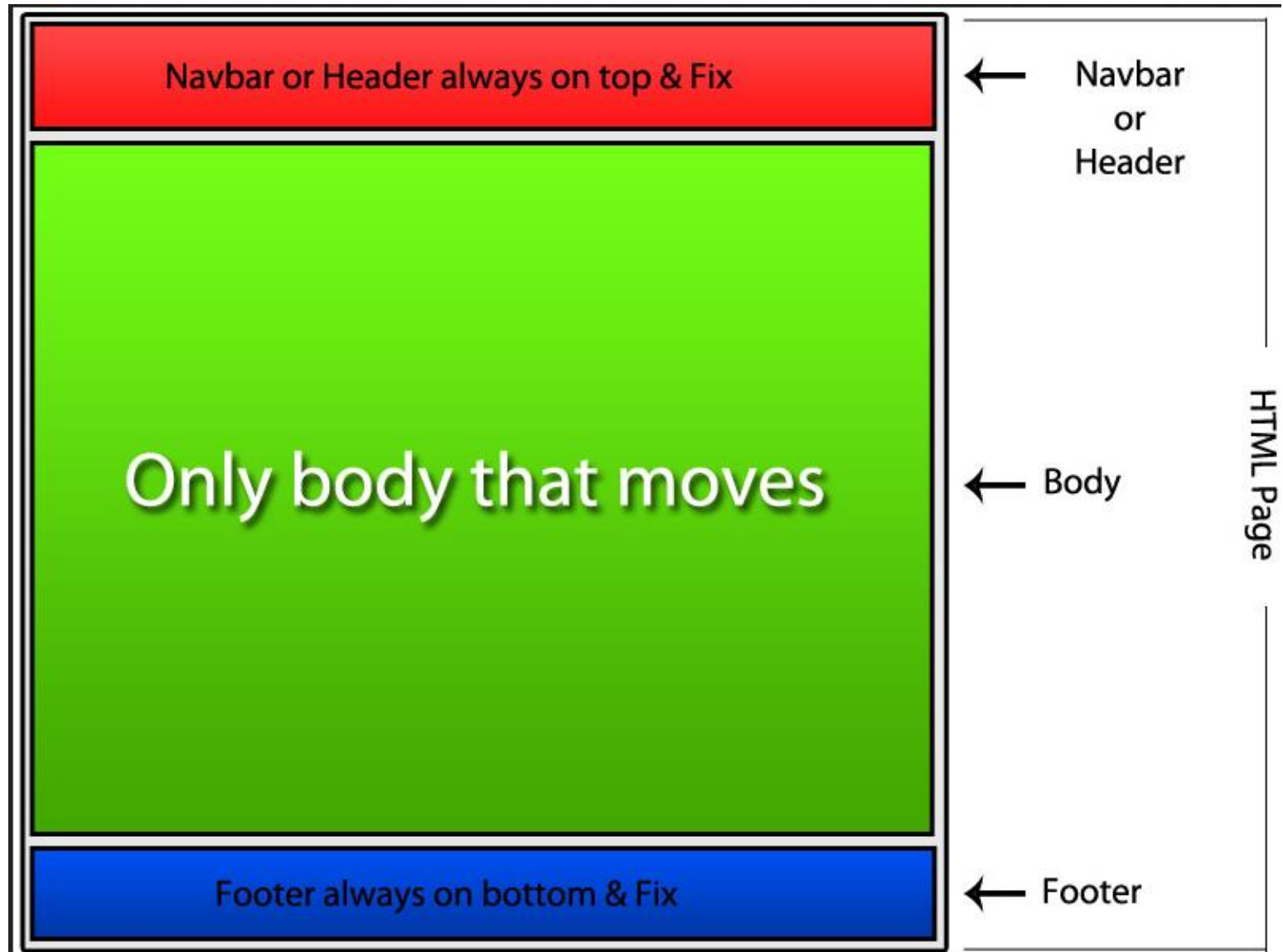
## Balises de tableau

Balise	Description
<code>&lt;table&gt;</code>	Tableau
<code>&lt;caption&gt;</code>	Titre du tableau
<code>&lt;tr&gt;</code>	Ligne de tableau
<code>&lt;th&gt;</code>	Cellule d'en-tête
<code>&lt;td&gt;</code>	Cellule
<code>&lt;thead&gt;</code>	Section de l'en-tête du tableau
<code>&lt;tbody&gt;</code>	Section du corps du tableau
<code>&lt;tfoot&gt;</code>	Section du pied du tableau

## Balises de formulaire

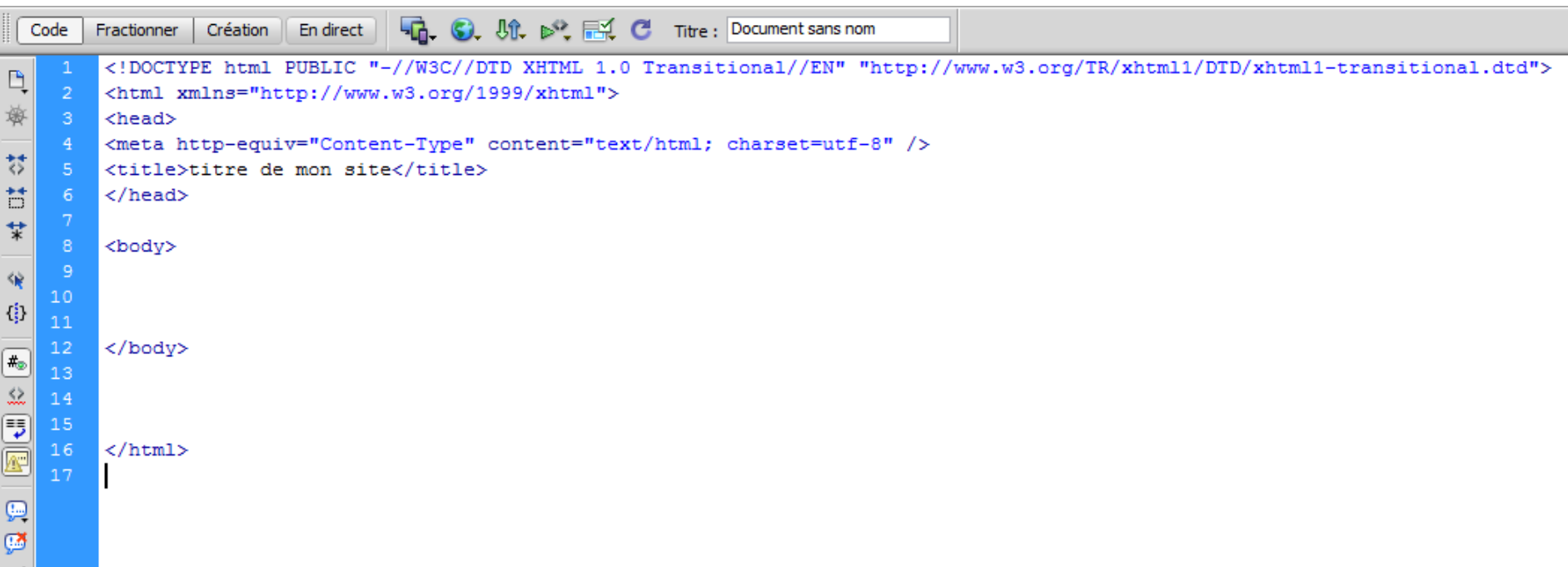
Balise	Description
<code>&lt;form&gt;</code>	Formulaire
<code>&lt;fieldset&gt;</code>	Groupe de champs
<code>&lt;legend&gt;</code>	Titre d'un groupe de champs
<code>&lt;label&gt;</code>	Libellé d'un champ
<code>&lt;input /&gt;</code>	Champ de formulaire (texte, mot de passe, case à cocher, bouton, etc.)
<code>&lt;textarea&gt;</code>	Zone de saisie multiligne
<code>&lt;select&gt;</code>	Liste déroulante
<code>&lt;option&gt;</code>	Élément d'une liste déroulante
<code>&lt;optgroup&gt;</code>	Groupe d'éléments d'une liste déroulante

# Structure d'une page html





# Structure d'un fichier html



The screenshot shows a web editor interface with a toolbar at the top and a code editor below. The toolbar includes buttons for 'Code', 'Fractionner', 'Création', and 'En direct', along with various icons for file operations and preview. The code editor displays the following HTML structure:

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
2 <html xmlns="http://www.w3.org/1999/xhtml">
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
5 <title>titre de mon site</title>
6 </head>
7
8 <body>
9
10
11
12 </body>
13
14
15
16 </html>
17 |
```

# Tp1 HTML

```
8 <body>
9 <h1> Mon premier titre de taille h1 </h1>
10
11 <h2 style="font-size:36px; color:#C00; text-align:center"> Mon premier titre de taille h2 </h2>
12 <h3> Mon premier titre de taille h3 </h3>
13 <h4> Mon premier titre de taille h4</h4>
14 <h5> Mon premier titre de taille h5</h5>
15 <h6> Mon premier titre de taille h6</h6>
16
17
18 <p> mon premier paragraphe </p>
19
20
21
22 <h1> les listes </h1>
23 <h4> liste non numerotée UL </h4>
24 <ul>
25 <li> j'apprends le html et le css </li>
26 <li> j'apprends le php </li>
27 <li> j'apprends le mysql via xampp</li>
28 </ul>
29 <h4> liste numerotée OL </h4>
30 <ol>
31 <li> j'apprends le html et le css </li>
32 <li> j'apprends le php </li>
33 <li> j'apprends le mysql via xampp</li>
34 </ol>
35 <h4> liste de definition DL</h4>
36 <dl>
37 <dt> HTML : </dt>
38 <dd> Le HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web. </dd>
39 </dl>
40 <h4> Tableau </h4>
41 <table border="solid 1px">
42 <tr>
43 <td> Nom </td>
44 <td> Prénom </td>
45 <td> Age </td>
46 </tr>
47 <tr>
48 <td> Lekhlifi </td>
49 <td> Ramzy </td>
50 <td> 29 </td>
51 </tr>
52 </tr>
53 </table>
54 </body>
55 </html>
```

# Mon premier titre de taille h1

## Mon premier titre de taille h2

### Mon premier titre de taille h3

#### Mon premier titre de taille h4

#### Mon premier titre de taille h5

##### Mon premier titre de taille h6

mon premier paragraphe

## les listes

#### liste non numerotée UL

- j'apprends le html et le css
- j'apprends le php
- j'apprends le mysql via xampp

#### liste numerotée OL

1. j'apprends le html et le css
2. j'apprends le php
3. j'apprends le mysql via xampp

#### liste de definition DL

#### HTML :

Le HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web.

#### Tableau

Nom	Prénom	Age
Lekhlifi	Ramzy	29

# Tp 2 page d'accueil site formation

Formation DevWeb @

ACCUEIL ÉTUDIANTS FORMATION CONNEXION



## HTML

Le HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web.



## CSS

Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML.



## MY SQL

MySQL est un système de gestion de bases de données relationnelles. Il est distribué sous une double licence GPL et propriétaire.



## PHP

PHP: Hypertext Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques.



- Dans ce tp nous allons créer la page d'accueil de notre site formation.
- Il sera composé d'un fichier html pour la structure du site relié à un fichier css pour le design
- Nous utiliserons aussi des images pour illustrer nos propos.

- Crée le fichier html page\_formation.html
- Crée le fichier css design\_site\_formation.css
- Relié le fichier css au html au niveau du head du fichier html via la balise link

**<link rel="stylesheet"  
href="design\_site\_formation.css"/>**

**Code du fichier page\_formation.html**

# Le Header

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<link rel="stylesheet" href="design_site_formation.css"/>
<title>Page de la formation</title>
</head>
<body>
<!-- header c'est l'entête-->
<header>
<div class="groupe">
<h1>Formation DevWeb <span class="orange">@</span></h1>
<nav>
<ul>
<li><a href="#">Accueil</a></li>
<li><a href="#">étudiants</a></li>
<li><a href="formation.html">formation</a></li>
<li><a href="#">connexion</a></li>
</ul>
</nav>
</div>
</header>
<!-- fin du header -->
```



# Image principale

```
<!-- image principale -->
<section id="image_principale">
<div class="groupe">
    <h2>Apprenez à créer votre <br /> <strong>site web</strong></h2>
</div>
</section>
<!-- fin image principale -->
```

# Partie liste des technologies (langages)

```
<!-- partie des langages de programmations -->
<section id="langages">
<div class="groupe">
<ul>
<li id="L1">
<h4>HTML</h4>
<p>Le HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web.</p>
</li>
<li id="L2">
<h4>CSS</h4>
<p>Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML.</p>
</li>
<li id="L3">
<h4>MySQL</h4>
<p>MySQL est un système de gestion de bases de données relationnelles. Il est distribué sous une double licence GPL et propriétaire.</p>
</li>
<li id="L4">
<h4>PHP</h4>
<p>PHP: Hypertext Preprocessor, plus connu sous son sigle PHP, est un langage de programmation libre, principalement utilisé pour produire des pages Web dynamiques.</p>
</li>
</ul>
</div>
</section>

<div class="clear"></div>

</div>
```

# Partie Objectif de la formation

```
<section id="objectif">

  <div class="invisible">
    <article style="background-image: url(image/circuit-page-web-statique.png)">
      <div class="transparent">
        <h4>site web statique</h4>
        <p>
          <small>
            Une page web statique est une page web dont le contenu ne varie pas en fonction des caractéristiques de la demande
          </small>
        </p>
        <a href="#" class="bouton1">Plus d'infos</a>
      </div>

    </article>

    <article style="background-image: url(image/circuit-page-web-dynamique.png)">
      <div class="transparent">
        <h4>site web Dynamique</h4>
        <p>
          <small>
            Une page web dynamique est une page web générée à la demande, par opposition à une page web statique.
          </small>
        </p>
        <a href="#" class="bouton1">Plus d'infos</a>
      </div>

    </article>
    <div class="clear"></div>
  </section>

<footer>

</footer>

</body>
</html>
```

**Code du fichier design\_site\_formation.css**

```
@charset "utf-8";
/* CSS Document */
/* annulé les marge par défaut*/

*
{
    margin: 0;
    padding: 0;
}

/* police generale */
body
{ font-family: Arial, Helvetica, sans-serif;
  font-size:15px;
}

.groupe
{
    width: 940px;
    margin: 0 auto;
    padding: 0 10px }
/* partie du style qui concerne le header-----*/
header
{ height:120px;
}

header h1
{ /*alignement*/
  float:left;
  margin-top:30px;
}
header nav
{ /*alignement*/
  float: right;
  margin-top:50px; }

h1
{ font-size: 45px;
  color:#444;
}

.orange
{
    color: #ff7a00;
}
```

```

/* enlevé les point noire de la liste */
ul
{ list-style: none;
  }
/* pour alligné les elements du nav*/
header nav ul li
{ display: inline-block;
  float: left;
  }
/* mettre les liens en maj*/
header nav ul li a
{ text-transform:uppercase;
  font-weight: bold;
  margin-right:20px;
}
/* enlevé le surlignement*/
a
{ text-decoration:none;
  color: #444;  }

/* partie du style image principale-----*/
#image_principale
{ height:580px;

background: url('image/mesa-trabajo-equipos-trabajo-tener-espacio-copia-idea-diseno-su-concepto-trabajo_35956-731.jpg') center no-repeat;
}
h2 {
  color:#444;
  font-size:55px;
}
#image_principale h2
{ /* enlevé le gras par default*/
  font-weight: normal;
  text-transform:uppercase;
  text-align:center;
  padding:350px 0 40px 0;
  color:#FFC;
}

/*----- partie langages-----*/

#langages ul
{
  margin: 80px 0;
}

```

```
#langages ul
{
    margin: 80px 0;
}
#langages ul li
{ /*pour aligner les li*/
    width: 225px;
    float:left;
    /* pour descendre le contenu*/
    padding-top: 100px;
    text-align:center;
    margin-right:10px;
}

h4
{ font-size:24px;
  color: #444;}

p{ line-height:20px;
   color:#777;
   font-weight: 400;}
#langages h4
{ text-transform:uppercase;
  margin-bottom:20px;
}

#L1
{
    background:url(image/html5.png) no-repeat top center;
}
#L2
{
    background: url(image/css-512.png) no-repeat top center;
}
#L3
{
    background: url(image/MySQL-512.png) no-repeat top center;
}
#L4
{
    background: url(image/Webysther_20160423_-_Elephpant.svg.png) no-repeat top center;
}
```

```
/*objectif*/

.clear
{
    clear: both;
}

.invisible
{
    width: 940px;
    margin: 0 auto;
}

#objectif
{
    background-color: #efefef;
    padding: 60px 0;
}

#objectif article
{
    width : 460px;
    height: 270px;
    float: left;
    border-radius: 10px;
}

#objectif article:first-child
{
    margin-right: 20px;
}

.transparent
{
    background: rgba(255,255,255,0.80);
    height: 230px;
    width:190px;
    padding: 20px;
    border-radius: 10px 0 0 10px;
    text-align: center;
}
```



```
article h4
{
    border-bottom: 2px solid orange;
    padding-bottom: 20px;
    text-transform: uppercase;
    margin-bottom: 20px;
    text-align: center;
}

#objectif p
{
    text-align: center;
    margin-bottom: 20px;
}

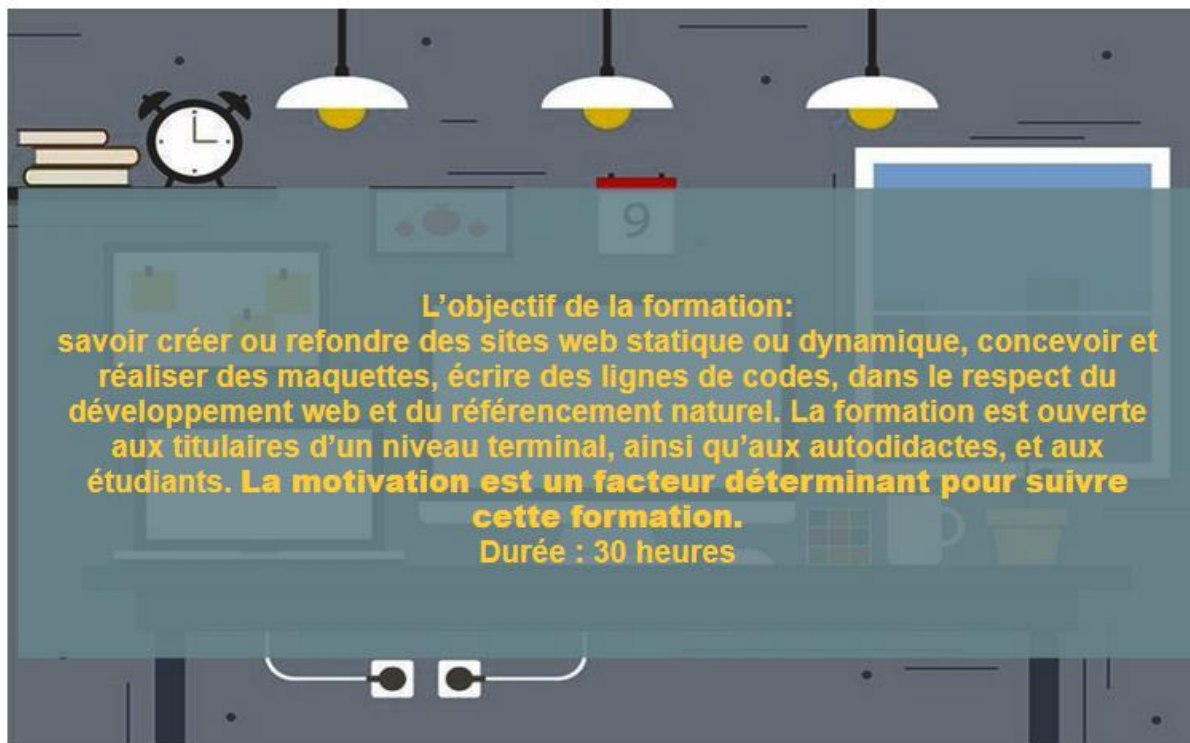
small
{
    color: black;
    font-size: 13px;
    font-style: italic strong;
}

.bouton1
{
    color: #fff;
    background-color: #ff7a00;
    padding: 6px 20px;
    border-radius: 3px;
}

.bouton1:hover
{
    background-color: cadetblue;
}
```

# Tp3 page formation

- Vous aller essayer de crée la page formation en ayant juste une maquette
- Tout ce que vous avez utiliser dans le tp2 vous sera utile



LANGAGES ----- HEURES	HTML/CSS	PHP	MYSQL	PROJET
3H	COURS THÉORIQUE	INTRODUCTION À L'ALGORITHMIE (VARIABLES, CONDITIONS, BOUCLES)	INTRODUCTION AU LANGAGE SQL (CREATION DE TABLE, REMPLISSAGE, MANIPULATION DES DONNÉES)	CRÉATION DE SITE E COMMERCE (GESTION DE STOCK ET GESTION DE VENTE)
3H	MANIPULATION DES LANGAGES (CRÉATION D'UNE PAGE STATIQUE)	INTRODUCTION À L'ALGORITHMIE (FONCTIONS, VARIABLES DYNAMIQUES)	MANIPULATION DES BASES DONNÉES AVEC PHP	CRÉATION DE SITE E COMMERCE (GESTION DE STOCK ET GESTION DE VENTE)
3H	MANIPULATION DES LANGAGES (REPRODUCTION D'UN VISUEL D'UNE PAGE)	MANIPULATION DU LANGAGE (CRÉATION DE PETITS PROGRAMMES)		

# Importance des formulaires

- Toute page HTML peut être enrichie de formulaires interactifs, qui invitent vos visiteurs à renseigner des informations : saisir du texte, sélectionner des options, valider avec un bouton.
- C'est là que langage HTML est limité car il faut ensuite pouvoir analyser les informations que le visiteur a saisies... et cela ne peut pas se faire en langage HTML. Comme nous allons le voir, le traitement des résultats doit s'effectuer dans un autre langage, par exemple le PHP.

# Champ de saisie avec label « Nom »

```
<form method="POST">  
  <label for="nom" >Nom :</label>  
  <input id="nom" type="text"  
    placeholder="Entrer votre nom"  
    required="required"/>  
</form>
```

# Champ de saisie email avec label « email »

```
<form method="POST">  
<label for="email">Email :</label>  
<input id="email" type="email" />  
</form>
```

# Champ de saisie mot de passe avec label « mot de passe »

```
<form method="POST">  
  <label for="pass">mot de passe</label>  
  <input id="pass" type="password"  
    placeholder="Entrer votre mot de passe"  
    required="required"/>  
</form>
```

# Checkbox

```
<form method="POST">
```

```
<input type="checkbox" value="html" /> html
```

```
<input type="checkbox" value="css" /> css
```

```
</form>
```



# Bonton radio

```
<input type="radio" value="h" name="genre" />  
      homme
```

```
<input type="radio" value="f" name="genre" />  
      femme
```

# TextArea avec label « commentaire »

```
<label for="commentaire">
    commentaire</label>
<textarea id="commentaire" cols="20"
    rows="20" placeholder="Entrer votre
    commentaire">
    </textarea>
```

# liste

```
<form method="POST">
```

```
<select>
```

```
<option value=« Buj">Bujumbura</option>
```

```
<option value=« Bub">Bubanza</option>
```

```
<option value=« Kay">Kayanza</option>
```

```
<option value=« Git">Gitega</option>
```

```
</select>
```

```
</form>
```

# Submit pour envoyer les informations

```
<input type="submit" value="ok" />
```

## Balises sectionnantes

Ces balises permettent de construire le squelette de notre site web.

Balise	Description
<code>&lt;header&gt;</code>	En-tête
<code>&lt;nav&gt;</code>	Liens principaux de navigation
<code>&lt;footer&gt;</code>	Pied de page
<code>&lt;section&gt;</code>	Section de page
<code>&lt;article&gt;</code>	Article (contenu autonome)
<code>&lt;aside&gt;</code>	Informations complémentaires

## Balises génériques

Les balises génériques sont des balises qui n'ont pas de sens sémantique.

En effet, toutes les autres balises HTML ont un *sens* : `<p>` signifie « Paragraphe », `<h2>` signifie « Sous-titre », etc.

Parfois, on a besoin d'utiliser des balises génériques (aussi appelées **balises universelles**) car aucune des autres balises ne convient. On utilise le plus souvent des balises génériques pour construire son design.

Il y a deux balises génériques : l'une est inline, l'autre est block.

Balise	Description
<code>&lt;span&gt;</code>	Balise générique de type inline
<code>&lt;div&gt;</code>	Balise générique de type block

# <div> </div>

- L'élément HTML <div> (qui signifie division du document) est un conteneur générique qui permet d'organiser le contenu sans représenter rien de particulier. Il peut être utilisé afin de grouper d'autres éléments pour leur appliquer un style (en utilisant les attributs class ou id)

# <section></section>

- L'élément HTML <section> représente une section générique d'un document, par exemple un groupe de contenu thématique. Une section commence généralement avec un titre.



# <span></span>

- L'élément HTML <span> est un conteneur générique en ligne (inline) pour les contenus phrasés. Il ne représente rien de particulier. Il peut être utilisé pour grouper des éléments afin de les mettre en forme (grâce aux attributs class ou id et aux règles CSS)

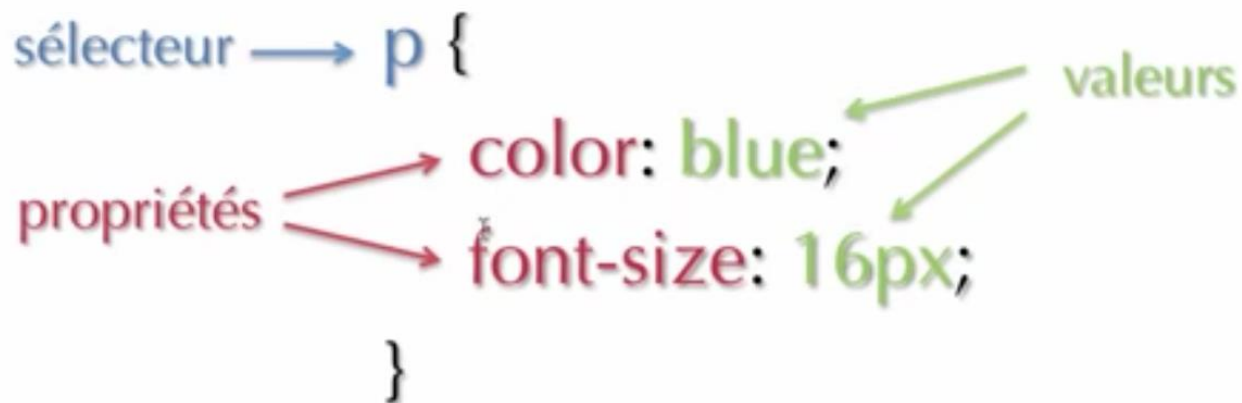
# Fondations du CSS

## **SELECTEURS, PROPRIETES, VALEURS**

- Un sélecteur détermine à quel élément un style doit être appliqué.
- Une propriété détermine le style qui va être appliqué à un élément.
- Une valeur détermine le comportement d'une propriété.

# SELECTEURS, PROPRIETES, VALEURS

- Exemple:



The diagram illustrates the components of a CSS rule. A blue arrow points from the label 'sélecteur' to the 'p' in 'p {'. Two red arrows point from the label 'propriétés' to 'color' and 'font-size'. Two green arrows point from the label 'valeurs' to 'blue' and '16px'.

```
sélecteur → p {  
propriétés → color: blue;  
              font-size: 16px;  
}
```

# OÙ ECRIRE LE CSS ?

- 3 possibilités :
  - Dans l'élément head du document HTML,
  - Dans la balise ouvrante d'un élément,
  - Dans un fichier CSS séparé.

Il est recommandé d'utiliser la dernière méthode.

- Pour lier un fichier HTML avec un fichier CSS, on écrit :  
`<link rel=« stylesheet » href=« fichiercss.css »/>`

# L'HERITAGE

- Les éléments en HTML « héritent » des styles de leurs parents. D'où le « cascading » de CSS.
- En cas de conflit, le style le plus proche de l'élément en question sera appliqué.

# MySQL

- MySQL est un système de gestion de bases de données relationnelles SQL open source développé et supporté par Oracle.
- Une base de données n'est qu'une collection structurée de données qui est organisée pour en faciliter l'utilisation et la récupération.

- Une information est stocké dans la base donnée avec un type qu'il faut spécifié.

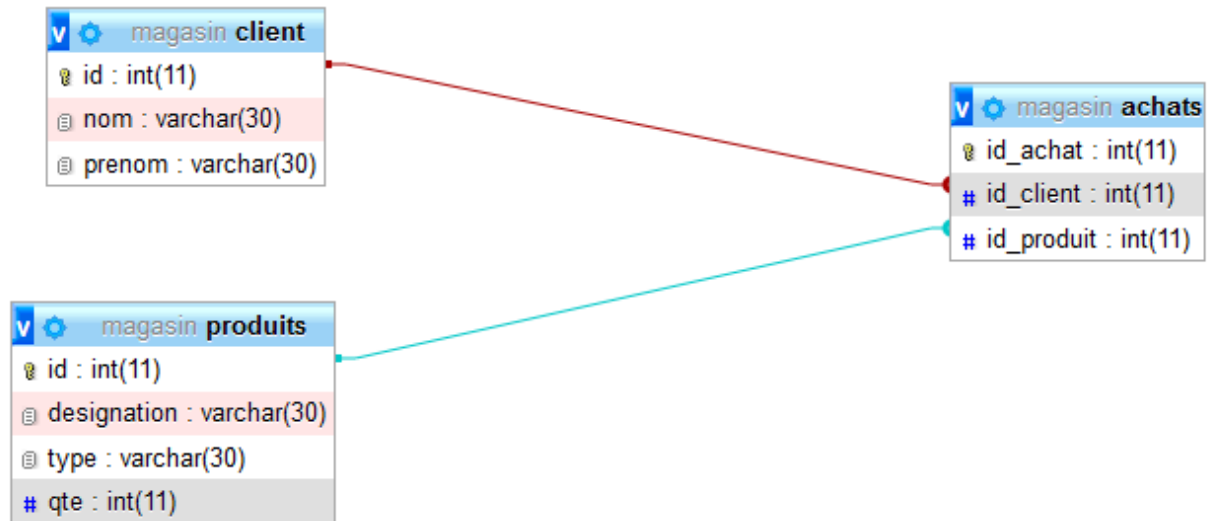
DATE TYPE	SPEC	DATA TYPE	SPEC
CHAR	String (0 - 255)	INT	Integer (-2147483648 to 214748-3647)
VARCHAR	String (0 - 255)	BIGINT	Integer (-9223372036854775808 to 9223372036854775807)
TINYTEXT	String (0 - 255)	FLOAT	Decimal (precise to 23 digits)
TEXT	String (0 - 65535)	DOUBLE	Decimal (24 to 53 digits)
BLOB	String (0 - 65535)	DECIMAL	"DOUBLE" stored as string
MEDIUMTEXT	String (0 - 16777215)	DATE	YYYY-MM-DD
MEDIUMBLOB	String (0 - 16777215)	DATETIME	YYYY-MM-DD HH:MM:SS
LONGTEXT	String (0 - 4294967295)	TIMESTAMP	YYYYMMDDHHMMSS
LOBLOB	String (0 - 4294967295)	TIME	HH:MM:SS
TINYINT	Integer (-128 to 127)	ENUM	One of preset options
SMALLINT	Integer (-32768 to 32767)	SET	Selection of preset options
MEDIUMINT	Integer (-8388608 to 8388607)	BOOLEAN	TINYINT(1)



- Dans une base de données relationnelle, une **clé primaire** est la donnée qui permet d'identifier de manière unique un enregistrement dans une table. Une **clé primaire** peut être composée d'une ou de plusieurs colonnes de la table.

- Une **clé étrangère**, dans une base de données relationnelle, est une contrainte qui garantit l'intégrité référentielle entre deux tables. Une **clé étrangère** identifie une colonne ou un ensemble de colonnes d'une table comme référençant une colonne ou un ensemble de colonnes d'une autre table (la table référencée).

# Exercice magasin



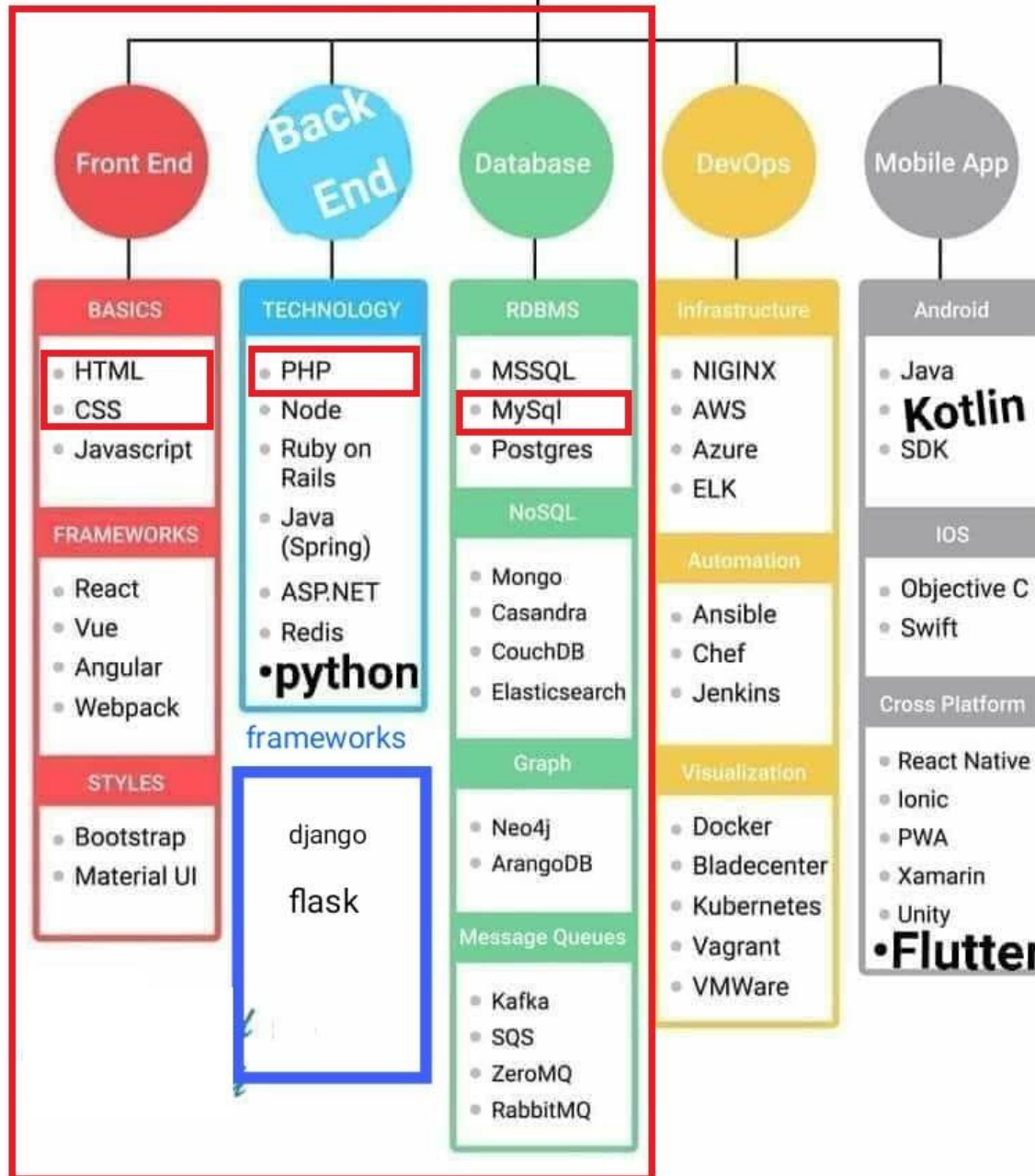
# Exercice sql Etudiant/avis



formation_web etudiants	
ID : int(11)	
NOM : varchar(50)	
PRENOM : varchar(50)	
age : int(11)	
niveau : varchar(30)	

formation_web avis	
num : int(11)	
avis : varchar(50)	
id_e : int(11)	

# FULL STACK DEVELOPER



# PHP

- PhP = PhP Hypertext Preprocessor
- Le PhP s'exécute côté serveur et permet de dynamiser un site.
- Site statique = doit être actualisé par quelqu'un pour changer
- Site dynamique = change tout seul en fonction du visiteur ou de n'importe quel autre paramètre (heure, saison, etc.)



# Relation Client/Serveur

- Client = Ordinateur qui demande une page web
- Serveur = Méga-ordinateur, qui ne s'arrête jamais et qui va envoyer la page web demandée au client
- Pour un site dynamique, le serveur génère potentiellement différemment une page donnée pour chaque visiteur qui la demande.



# Le PHP Génère du HTML/CSS

- Un client n'est pas capable de lire du PhP.
- Rôle principal et intérêt du PhP = générer et envoyer une page en HTML (ou en CSS) différente pour chaque visiteur.





- PhP n'est pas le seul langage permettant de créer des sites dynamiques, il possède de nombreux concurrents.
- Concurrents : Django, JEE, Ruby, etc.
- Pourquoi le PhP ?
  1. Simplicité de prise en main
  2. Grande communauté
  3. Fonctions prêtes à l'emploi



- Travailler en local = travailler « hors-ligne », sur sa machine. A opposer au travail sur serveur.
- Intérêt du travail en local ?
  1. gratuité
  2. possibilité de tester, de préparer des modifications
  3. Possibilité de travailler n'importe où, n'importe quand



- Pour travailler en local, obligation de « transformer » son ordinateur en serveur. En effet, les clients ne savent pas lire le PHP !
- Pour cela, il suffit d'installer...
  - WAMP
  - MAMP
  - XAMPP



- Cette fonction nous permet d'afficher à l'écran des chaînes de caractères, qui peuvent être définies directement par l'utilisateur comme lorsque l'on fait un **echo 'Bonjour ';** ou qui peuvent être des contenus de variables (lorsque l'on fait echo avec une variable).

```
<?php
/* fonction d'affichage */

echo "ceci est un affichage a l'aide de la fonction echo de php";
?>
```

# Les variables

A quoi servent les  
variables ?

- Dans un programme informatique, on va avoir en besoin de stocker provisoirement des valeurs. Il peut s'agir de données issues du disque dur ou fournies par l'utilisateur (frappées au clavier).
- Ces données peuvent être de plusieurs types : elles peuvent être des nombres, du texte, etc. Toujours est-il que dès que l'on a besoin de stocker une information au cours d'un programme, on utilise une variable.

```
8 <?php
9 /* les variable */
10 /*declaration des variables*/
11 $nom = "lekhlifi";
12 $prenom = "ramzy";
13 $age = 28;
14
15 /* affigage des information contenu dans les variable */
16
17 echo '<br> Bonjour ' . $nom . ' ' . $prenom . ' vous avez ' . $age . ' ans';
18
19 ?>
```



```
<?php

/* operations arithmetiques en utilisant les variables*/

$nombre1 = 20;

$nombre2 = 40;

/* l'addition du contenu des variable sera atocké dans la variable $addition */

$addition = $nombre1 + $nombre2;

/* affichage du resultat */

echo '<br> l\'addition de '.$nombre1.' et '.$nombre2.' est '.$addition;

/* autre affichage */

echo ' <br> '.$nombre1.' + '.$nombre2.' = '.$addition;

/* faire la même chose avec la division et la soustraction */

?>
```

- Faire la même chose avec la division, la soustraction et la multiplication.

# Les Conditions

Au quotidien, nous devons faire des choix. Et les actions réalisées dépendent de ces choix ou de ces conditions. Comme par exemple :

- s'il fait beau dimanche
  - alors j'irai à la plage,
  - sinon j'irai au cinéma.
- informatique, cela est fréquent.
- Le programme va prendre des directions différentes en fonction des conditions.

# L'instruction conditionnelle if

- La syntaxe de l'instruction if est décrite de la façon suivante:

```
if (expression)
    { instruction1;
      instruction2; }
```

# L'instruction conditionnelle else

```
if (expression)
    { instruction1; }
else
    { instruction2; }
```

# Les instructions conditionnelles imbriquées

- PHP permet d'écrire ce type de structure sous la forme :

```
if (expression1)
    { bloc1; }
else if (expression2)
    { bloc2; }
else if (expression3)
    { bloc3; }
else { bloc4;}
```

```
<?php
```

```
/* variable du temps */
```

```
$temps ="moche";
```

```
/*si il fait beau*/
```

```
if ($temps =="beau")
```

```
{
```

```
    echo "<br> je vais à la plage";
```

```
}
```

```
/* si il fait moche*/
```

```
else if ($temps =="moche")
```

```
{
```

```
    echo "<br> je vais au cinema";
```

```
}
```

```
?>
```

- Crée un variable **\$genre** qui prend soit **“homme”** ou **“femme”**
- **Si** c’est un homme afficher **“bonjour monsieur”**
- **Si** c’est un femme afficher **“bonjour madame”**



- Ajouter a l'exercice précédent la variable **\$heure**
- Si **\$heure > 17** et **\$genre == "homme"**
- Afficher **Bonsoir monsieur**
- Si **\$heure < 17** et **\$genre == "homme"**
- Afficher **Bonjour monsieur**
- Même chose si **\$genre=="femme"**

# L'instruction Switch

- `switch(variable){  
    case valeur1 : instr1; break;  
    case valeur2 : instr2; break;  
    case valeurN : instrN; break;  
    default : instr; break;  
}`

```
<?php
```

```
/* structure de controle switch */
```

```
$note = 10;
```

```
switch ($note)
```

```
{
```

```
    case 0 : {
```

```
        echo " <br> sans commentaire";
```

```
        break;
```

```
    }
```

```
    case 10 : {
```

```
        echo " <br> moyen";
```

```
        break;
```

```
    }
```

```
    case 15 : {
```

```
        echo " <br> très bien";
```

```
        break;
```

```
    }
```

```
    case 20 : {
```

```
        echo " <br> excellent ";
```

```
        break;
```

```
    }
```

```
}
```

```
?>
```

# Les Boucles

Le principe des **boucles**, ou calcul itératif, consiste à répéter les mêmes instructions un certain nombre de fois.

# La Boucle For

- La boucle for est une structure employée pour exécuter un bloc d'instructions un nombre de fois en principe connu à l'avance. Elle utilise la syntaxe suivante :

```
for ( initialisation; test; incrémentation)  
    { instructions; }
```

```
25  for($x=1;$x<=10;$x++)
26  {
27      echo 'ceci est le nombre :'.$x.'<br>';
28  }
29
```

```
30 for ($x=10; $x>=1; $x--)  
31 {  
32     echo 'ceci est le nombre :'.$x.'<br>';  
33 }  
34
```

# La Boucle While

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



```
10 <?php
11 $x =1;
12 while ($x<=10)
13 {
14
15     echo 'ceci est le nombre :'.$x.'<br>';
16     $x= $x+1;
17 }
18
19 ?>
20 |
```

# La Boucle do..While

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

```
1 <?php
2
3 do {
4     echo 'ceci est le nombre :'.$x.'<br>';
5     $x= $x+1;
6
7     }while ($x<=0)
8 ?>
9
```

- Crée une boucle qui calcule et affiche une des table de multiplication exemple :

$$5*1=1$$

$$5*2 =10$$

$$5*3 =15$$

$$5*4 =20$$

$$5*5 =25$$

etc...

# Variable dynamique avec des formulaires

- Pour donner la main a l'utilisateur nous utiliserons les formulaires pour lui proposer un champ de saisi une liste des bouton radio...etc...
- Les formulaires appartiennent au html mais le traitement de l'information choisie ou insérer se fera avec du PHP

# La fonction \$\_POST

Transmet les informations du formulaire de manière masquée

# La fonction isset

Détermine si une variable est définie et est différente de **NULL**.

```
<form method="POST">
<label for="nom">Nom</label>
<input type="text" name="nom" id="nom" required="required"/>
<input type="submit" name="valider" value="valider"/>

</form>
```

```
<?php
if(isset($_POST['valider'])) {
    if(isset($_POST['nom'])) {
        echo ' <br> Bonjour ' . $_POST['nom'];

    }

}

?>
```



```
<form method="POST">
<input type="radio" id="mr" name="choix" value="Monsieur" /> Monsieur
<input type="radio" id="m" name="choix" value="Madame" /> Madame
<input type="submit" name="ok" value="ok"/>
```

```
<?php
if(isset($_POST['ok'])) {
    if(isset($_POST['choix'])) {
        if(($_POST['choix']=="Monsieur")) {
            echo ' <br> Bonjour Monsieur';
        }
        else {
            echo ' <br> Bonjour Madame';
        }
    }
}

?>
```

- Exercice : l'utilisateur introduit un nombre entier et votre programme doit afficher la table de multiplication de ce dernier.

# Contrôle de la base de donnée avec PHP

- Pour manipuler la base de donnée a travers le PHP il faut déjà se connecté au serveur local puis à la base de donnéé
- A l'aide de la fonction **MySQLi** (abréviation pour MySQL Improved en anglais, c'est-à-dire MySQL Amélioré) est un pilote qui permet d'interfacer des programmes écrits dans le langage de programmation PHP avec les bases de données MySQL.

```
/* nom d'utilisateur et mdp de xampp */  
$u= "root";  
$p= "";  
$db='e_commerce';  
$db= new mysqli('localhost',$u,$p,$db) or die("unable to connect");
```

- Une fois la connexion établie nous allons exécuter une requête
- Pour le faire il faut affecté la requête a une variable en tant que chaine de caractère
- Puis l'exécuter en utilisant la fonction **mysql\_query** le résultat sera affecté a une variable \$resultat
- Pour afficher le résultat qui sera un ensemble de ligne nous utiliserons la fonction **mysqli\_fetch\_assoc** qui retourne un tableau associatif qui correspond à la ligne récupérée ou **null** s'il n'y a plus de ligne.

```
$sql="SELECT * FROM utilisateur";  
$resultat = mysqli_query($db,$sql) or die("bad query");  
  
while($row=mysqli_fetch_assoc($resultat))  
{  
    echo "{$row['id']}{$row['nom']}{$row['prenom']}{$row['email']}{$row['mdp']}<br>";  
}
```

- Pour savoir si le résultat de notre requête est vide ou non nous utiliserons la fonction `mysqli_num_rows` sur le résultat qui retourne le nombre de lignes dans un jeu de résultats