UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

Développement Web – PHP
Cours 1

Fabrice BOISSIER
Fabrice.Boissier@univ-paris1.fr

Ali JAFFAL
Ali.Jaffal@univ-paris1.fr

Bureau C.14.05
2018-2019

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

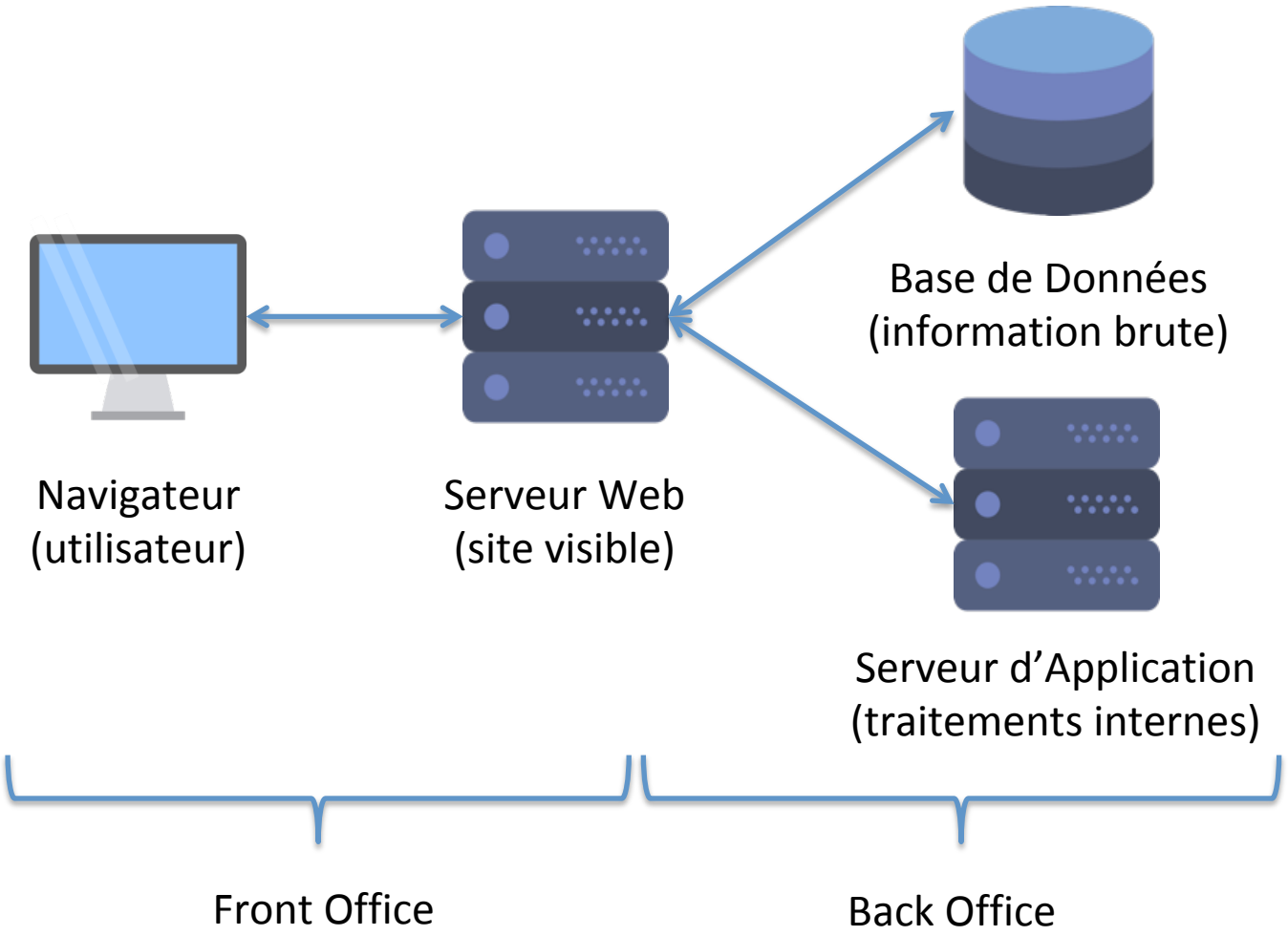
Organisation

- **12 séances**
- **Une séance = 1h30 cours + 1h30 TD, Projet**
- **Projet en binôme ou en solo (selon les groupes du premier semestre)**
- **1 Devoir Sur Table**
- **Interros Surprises 10 minutes**
- **Participation, travail continu notés sous forme de points bonus**
- **TD de la semaine, divers documents et informations diverses sur l'EPI**
- **Fin cours avril**
- **Evaluation**
  - Contrôle continu (50 %)
    - Devoir Maison
    - Projet
    - 1 Devoirs Sur Table
    - Interrogations surprises
    - Bonus Participation
  - Examen partiel (50 %)

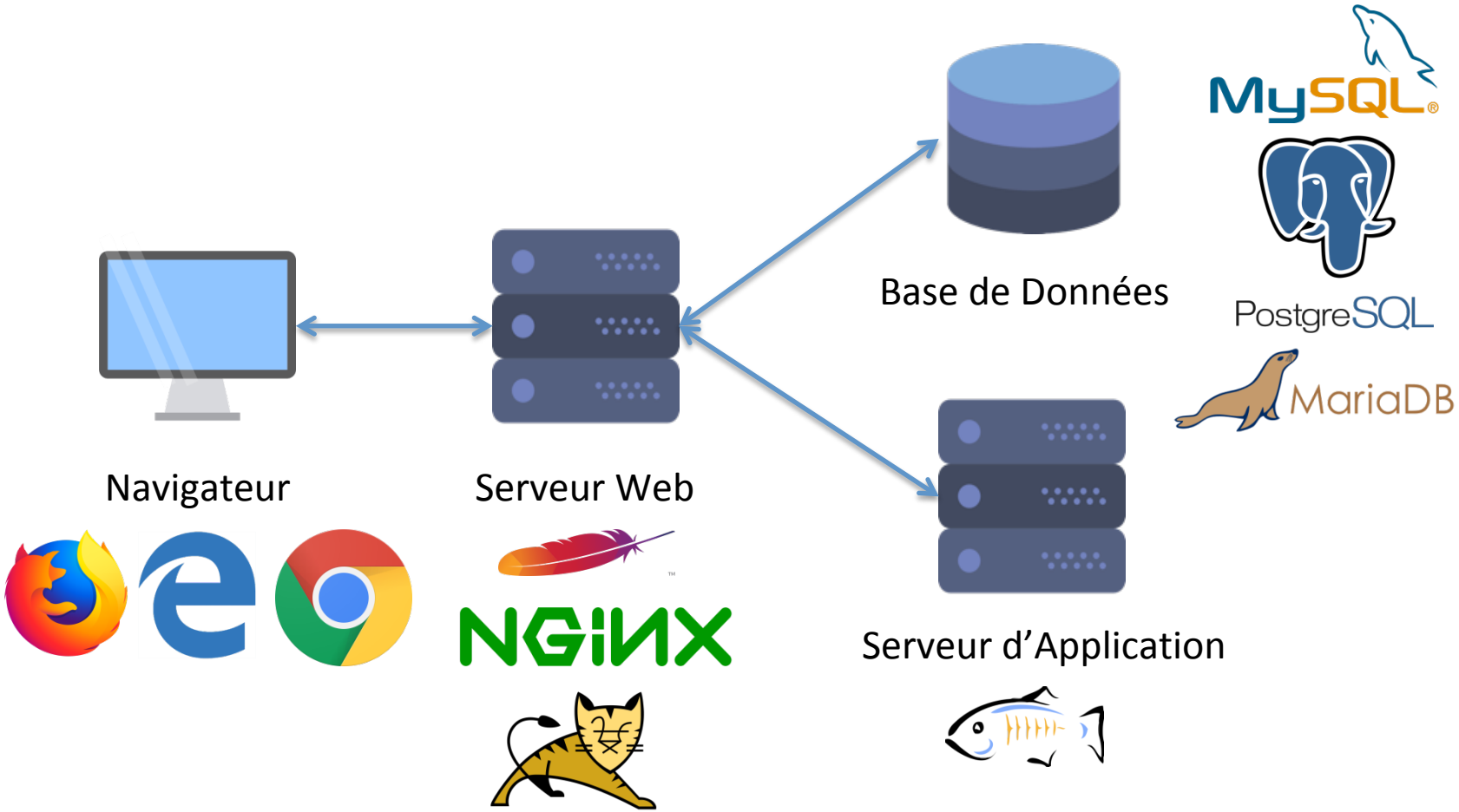
Programme du cours

- Introduction générale au web
  - Architectures techniques
  - Rappel sur HTML et la différence Statique/Dynamique
  - Explications sur l'environnement web
  - Premiers pas en PHP
  - Installation WAMP/MAMP/XAMP
- PHP
- PHP et MySQL

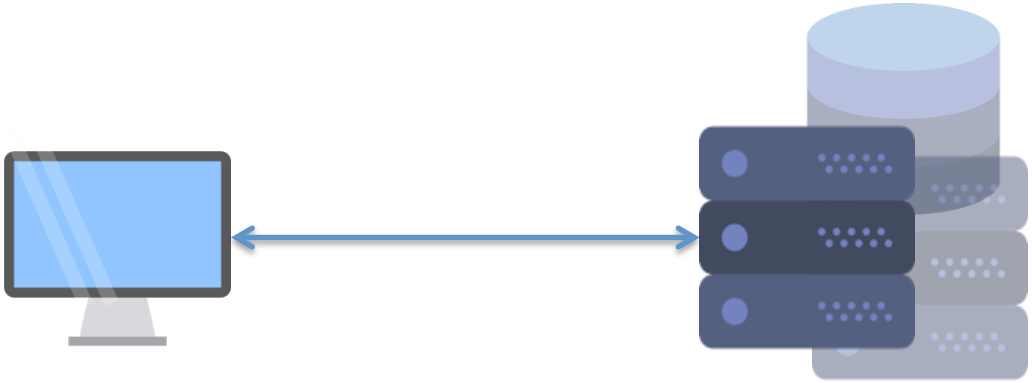
Architecture générale d'un site web



Architecture générale d'un site web



Architecture générale d'un site web



Navigateur

Serveur Web &
Base de Données

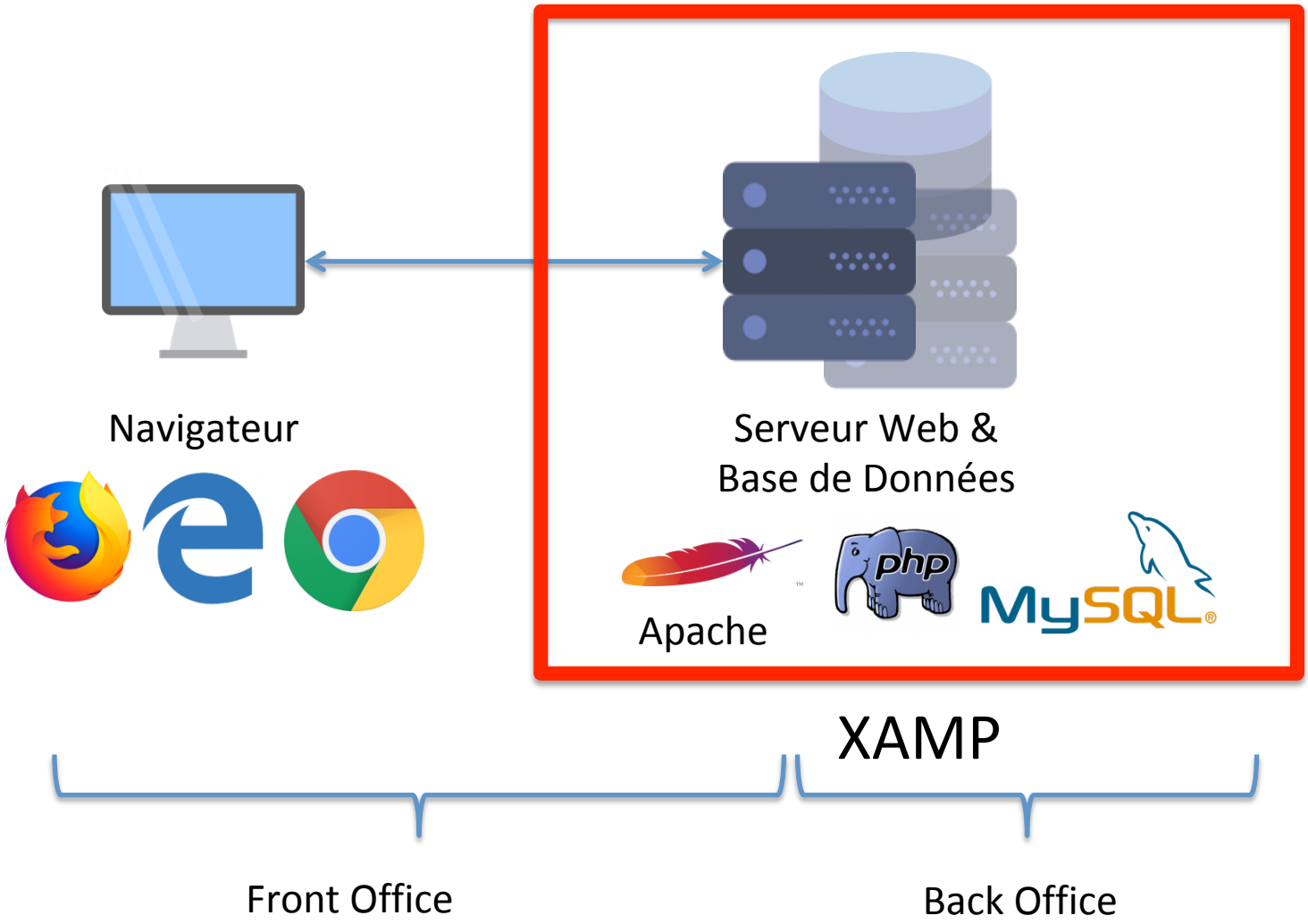
Apache

MySQL

Front Office

Back Office

Architecture générale d'un site web



Rappel des versions des technologies

Vérifiez bien les versions de tutoriaux que vous trouverez sur internet !

- HTML — *pas vraiment de version minimale hormis 1.1… les navigateurs interprètent ce qu'ils peuvent*
- HTML 5 — version actuelle *(4 passe aussi)*

- ~~PHP 4~~ — *obsolète et non-supporté (+ failles de sécurité)*
- PHP 5.6 — *supporté*
- PHP 7.3 — version actuelle

- MySQL 8 — version actuelle *(mais peu de soucis avec le langage, car standard SQL, excepté si tutorial <= MySQL 3.0)*

Rappel des versions des technologies

Vérifiez bien les versions de tutoriaux que vous trouverez sur internet !

- Documentation PHP : http://php.net/manual/fr/
- Exemple concret avec les fonctions de chaînes : http://php.net/manual/fr/ref.strings.php

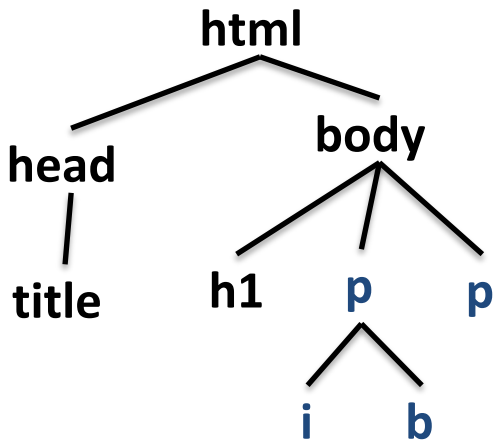
Rappel HTML

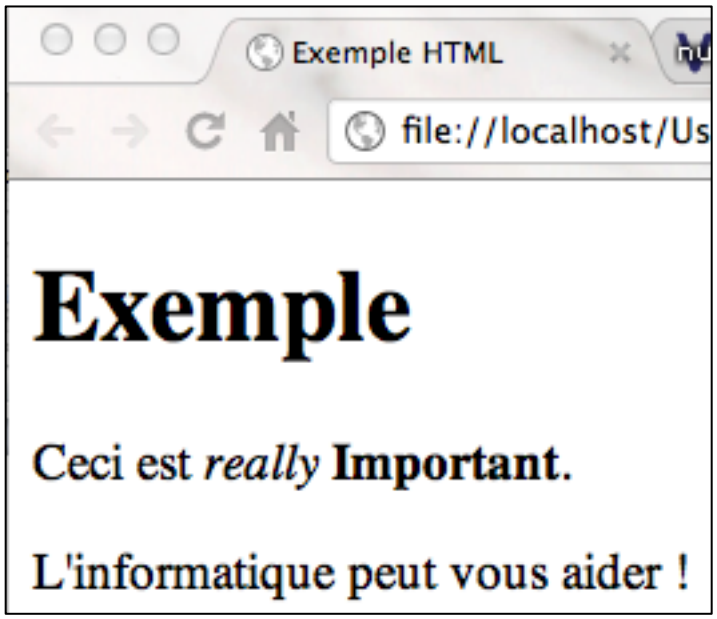
- Header / Body
- Listes
- Tableaux
- Images et Liens

HTML

Chaque balise ouverte doit être fermée
<balise> … </balise>

```
<html>
 <head>
  <title> Exemple HTML </title>
 </head>
 <body>
  <h1>Exemple</h1>
  <p>Ceci est  <i>really</i>
  <b>Important</b>.  </p>
   <p> L'informatique peut vous
  aider ! </p>
 </body>
 </html>
```





HTML

- Structure d'un document HTML

```
<!DOCTYPE html>
```

→ Indication « idiome » HTML

```
<html>
  <head>
    <meta name="author" content=
                "Manuele Kirsch Pinheiro" />
    <title> Exemple HTML </title>
  </head>
  <body>
    <h1>Exemple</h1>
    <p>Ceci est <i>really</i>
        <b>Important</b>.  </p>
    <p> L'informatique peut vous aider ! </p>
  </body>
</html>
```

Entête (head)
Informations générales sur le document

Corps (body)
Contenu du document

HTML

- Elément **DOCTYPE**
  - Indique au navigateur quelle version de HTML a été utilisée
  - **HTML 4.01**
    - Couramment compris par tous les navigateurs

<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" http://www.w3.org/TR/html4/loose.dtd>

  - **HTML 5**

<!DOCTYPE html>

    - En cours de définition
    - Reconnu uniquement par les navigateurs les plus récents (Google Chrome 16.0, Firefox 9.0, Internet Explorer 9…)

HTML

- **Eléments de l'entête (head)**
  - Informations **complémentaires** sur le document

```
<head>
   <meta name="author"
         content= "Manuele" />
   <title> Exemple HTML </title>
</head>
```

  - **Ce n'est pas le contenu du document**, donc ces informations ne sont **pas affichées** dans la page
  - Typiquement, informations pour les moteurs de recherche
- **Balises**
  - **<titre> … </titre>** : titre du document
  - **<meta … />** : métadonnées (descriptions) sur le document
  - *<link … />, <style> … </style>* : styles

HTML

- **Eléments de l'entête (head)**

Ouverture et **fermeture** de la balise

<meta *name="author"*
***content= "auteur"* />**

Attributs associés à la balise

Précisions sur une balise

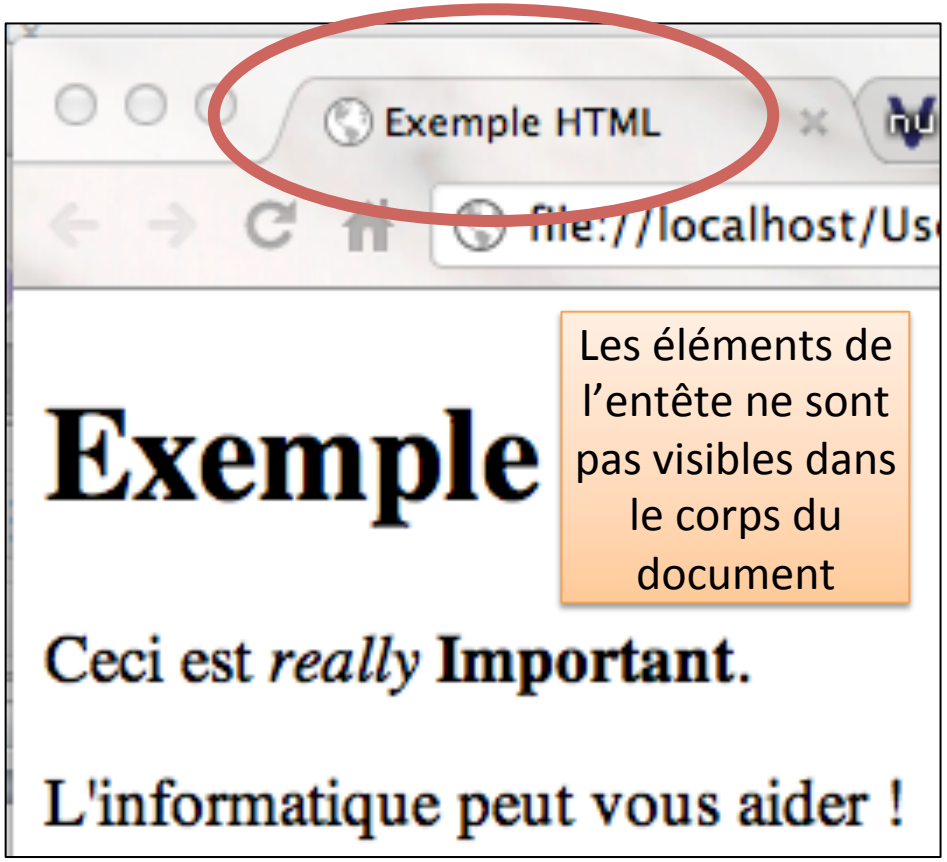
Chaque balise possède son ensemble d'attributs

<balise attribut = "valeur" … >

<meta name="description" value="…" />

<meta charset="ISO-8859-1">

< title > Exemple HTML **</ title >**



HTML

- **Eléments du corps (body)**
  - Contenu du document
  - Partie rendue visible par les navigateurs
- **Balises** : il y en a plein…
  - Titres : **<h1>, <h2> … <h6>**
  - Paragraphe et saut de ligne : **<p>** et **
**
  - Citations et mises en valeur **: <b>, <i>, <blockquote>**…
  - Images et liens **: <img>, <a …>** …
  - Listes : **<ol>, <ul>, **
  - Tableaux : **<table>, <tr>, <td>**…
  - Organisation du document : ** , <section>**…

```
<body>
  <h1>Exemple</h1>
  <p>Ceci est <i>really</i>
      <b>Important</b>.  </p>
  <p> L'informatique peut vous
aider ! </p>
</body>
```

HTML

- **Eléments du corps (body)**
- **Les titres : h1, h2, h3, h4, h5, h6**
  - Les éléments **h*x*** permettent de définir des **titres** de **différents niveaux**
    - **h1 correspond au titre principal**
  - Ils doivent apparaître dans l'ordre (**h1 avant h2**) avec **un seul titre principal** (h1)

```
<body>
  <h1>Exemple</h1>
  …
</body>
```

HTML

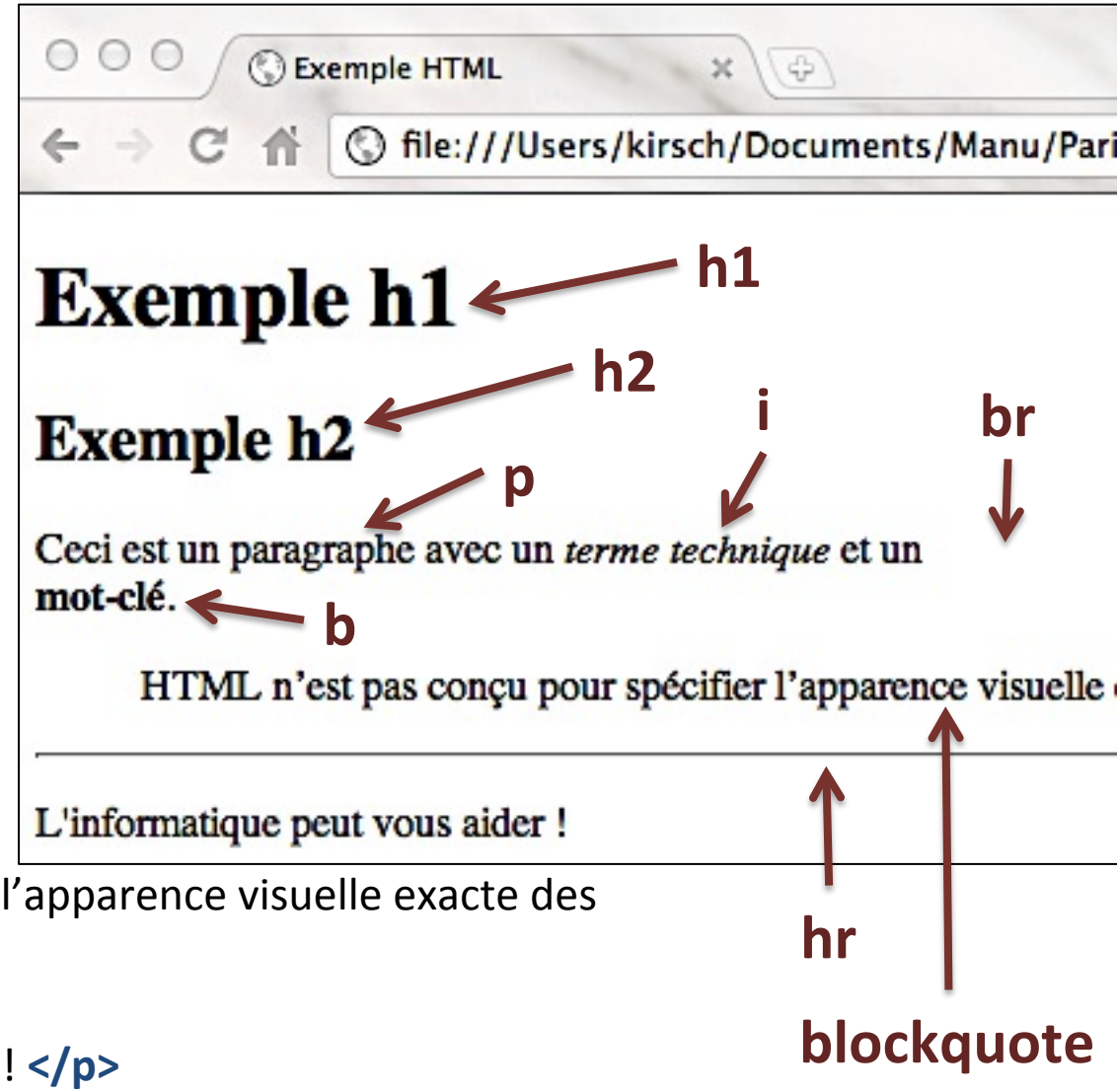
- **Eléments du corps (body)**
- **Paragraphe, saut de ligne et citation…**
  - La balise ** … ** indique un paragraphe
  - La balise **
** fait un simple saut de ligne
  - Les balises **…** et **…** mettent un texte en relief (en gras ou en italique)
  - La balise **<blockquote>…</blockquote>** permet de citer une autre page Web

    <blockquote *cite="http://source/"*> citation </blockquote>
  - La balise **<hr />** permet d'établir une séparation (ligne horizontal) dans le document

HTML

- **Eléments *body***

```
<html>
 <head> … </head>
 <body>
  <h1>Exemple h1</h1>
  <h2>Exemple h2</h2>
   <p>Ceci est un paragraphe avec un <i>terme technique</i> et un <br/> <b>mot-clé</b>. </p>
<blockquote cite="http://fr.wikipedia.org/wiki/Hypertext_markup_language">
HTML n'est pas conçu pour spécifier l'apparence visuelle exacte des documents. </blockquote>
 <hr/>
  <p> L'informatique peut vous aider ! </p>
</body>
</html>
```



HTML

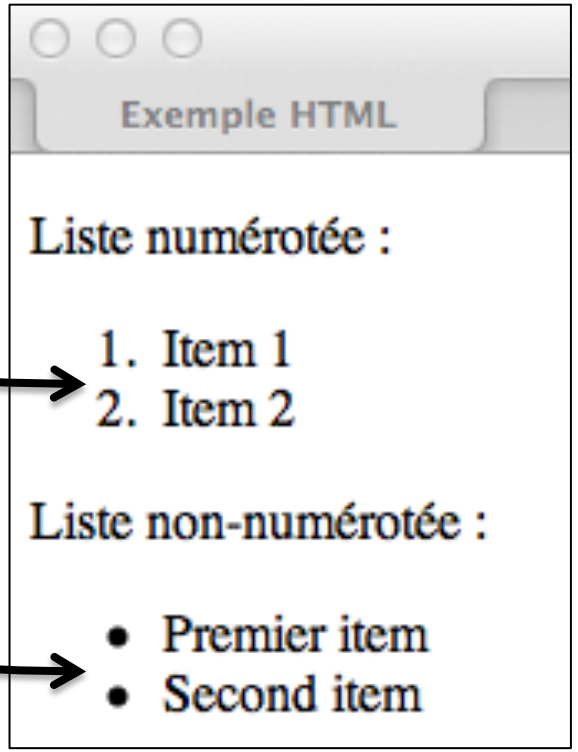
- **HTML**
  - Langage de balises, permettant la structuration des pages Web
  - Organisation en balises

    <balise attr="valeur"> … </balise>
  - Organisation du document
    - Entête : **head**
    - Corps du document : **body**
  - Différents types de balises possibles
    - Listes, tableaux, images, liens…

HTML : listes

- Plusieurs types de listes sont possibles
  - Listes numérotés : ** … **
  - Listes non-numérotés : ** … **
  - Peu importe la liste, un seul moyen d'indiquer les éléments : ** … **

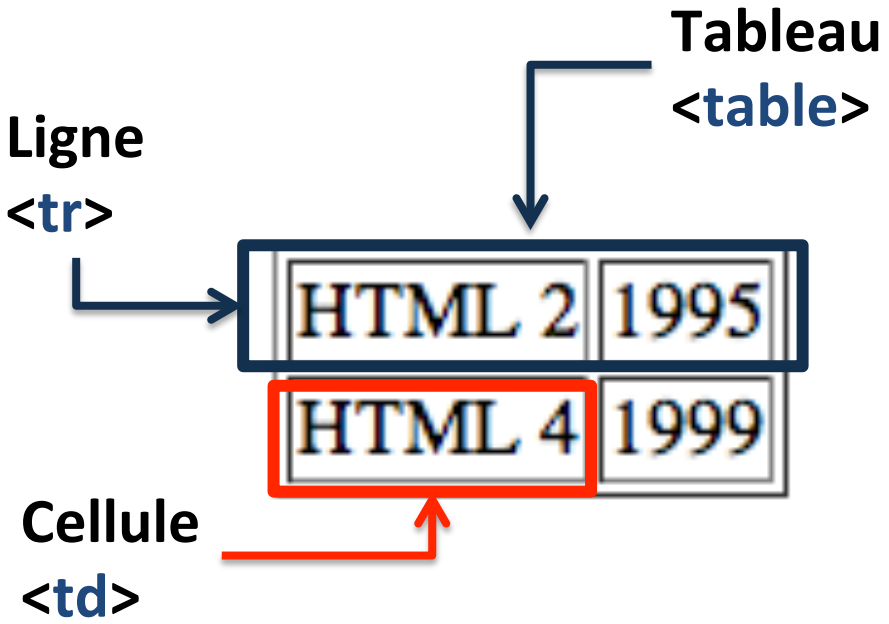
```
<ol>
  <li> Item 1 </li>
  <li> Item 2 </li>
</ol>

<ul>
   <li> Premier item </li>
   <li> Second item </li>
</ul>
```



HTML : tableaux

- Pour créer un tableau en HTML, on va combiner plusieurs balises :
  - **table, tr, td**, caption, th, thead, tbody

```
<table border="1">
 <tr>
  <td>HTML 2</td>
  <td>1995</td>
 </tr>
 <tr>
  <td>HTML 4</td>
  <td>1999</td>
 </tr>
</table>
```



HTML : tableaux

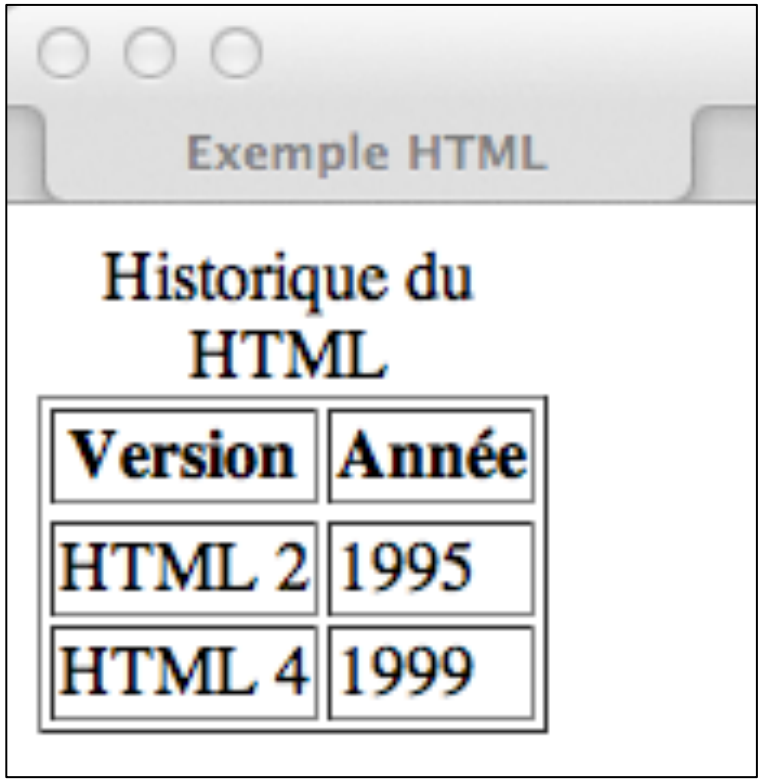
```
<table border="1">
<caption>Historique du HTML </caption>
 <thead>
 <tr>
   <th> Version</th>
   <th>Année </th>
 </tr>
 </thead>
 <tbody>
 <tr>
  <td>HTML 2</td> <td>1995</td>
 </tr>
 <tr>
  <td>HTML 4</td> <td>1999</td>
 </tr>
</tbody>
</table>
```

caption : légende

thead : Entête du tableau

th : Cellule de l'entête

tbody : corps du tableau



HTML : images

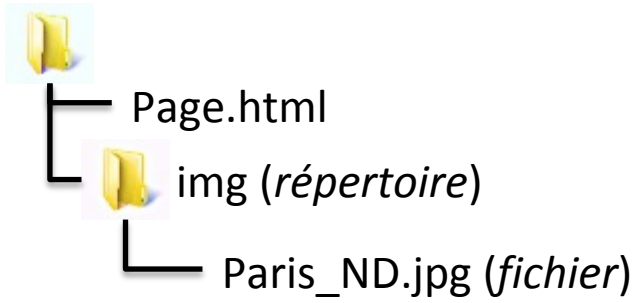
- Insertion d'images dans le texte : balise **img**

****

Texte alternatif
infobulle

Taille optionnelle
width= "80"
height="70"

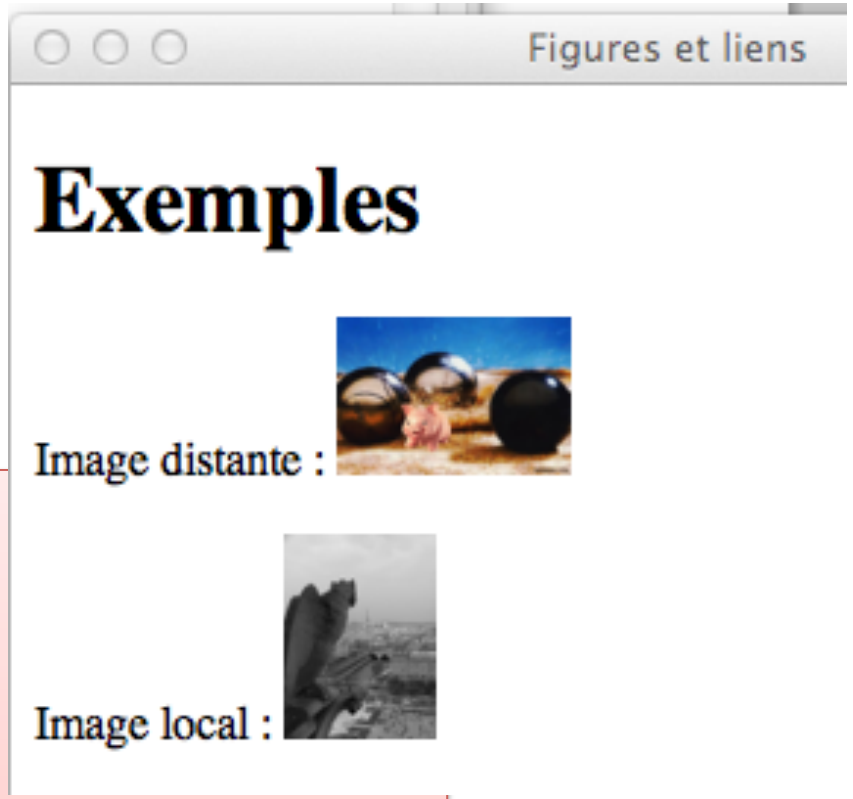
src : où se trouve l'image
Soit elle est avec la page Web
src=".\img\Paris_ND.jpg"



Soit elle est sur un serveur :
src="http://lsteffenel.fr/images/petanque-cochonnet.jpg"

HTML : images

- Balises HTML : Images




```
<html>
 <head> … </head>
<body>
  <h1>Exemples </h1>
  <p>Image distante : <img
src="http://lsteffenel.fr/images/petanque-cochonnet.jpg"
width="80" alt="Cochonnet et petanques" /> </p>
  <p>Image local : <img src=".\img\Paris_ND.jpg"
height="70"  alt="Vue de Paris" /> </p>
</body>
</html>
```

HTML : liens

- L'usage des **liens** permet de relier une page Web à d'**autres pages**, voire à d'**autres points** dans la page

** lien visible**

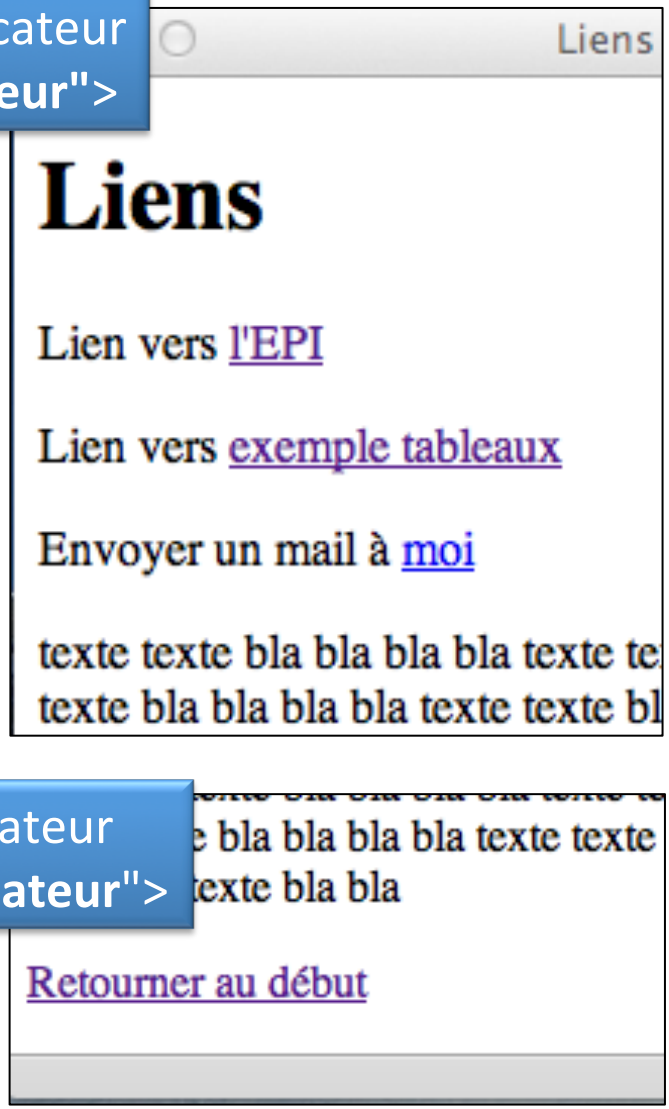
- L'attribut ***href*** indique vers où aller lorsqu'on clique sur le lien
  - Lien local :
    **** vers autre page ****
  - Lien distant:
     ailleurs ****
  - Envoyer un mail :
    ****envoyer mail****

HTML : liens

On attribue un identificateur
<balise **id="identificateur"**>

```
<h1 id="debut">Liens </h1>

<p>Lien vers <a href="http://epi.univ-paris1.fr">
 l'EPI </a></p>

<p>Lien vers <a href="coursHTML-5.html">
exemple tableaux </a></p>

<p>Envoyer un mail à
<a href="mailto:moi@mail.com"> moi </a></p>

<p> .... </p>

<p> <a href="coursHTML-7.html#debut"> Retourner
au début </a> </p>
```

Lien vers l'identificateur
<a href="**#identificateur**">



Statique vs Dynamique

- Statique :
  - HTML
  - CSS
  - *JavaScript (JS)*

Partie « fixe » des pages renvoyées au client : les parties qui ne changent pas quoi qu'il arrive

- Dynamique :
  - CGI
  - PHP
  - Java
  - JavaScript (JS)
  - …

Partie « variable » des pages renvoyées au client : les parties qui changent selon les requêtes et les informations disponibles

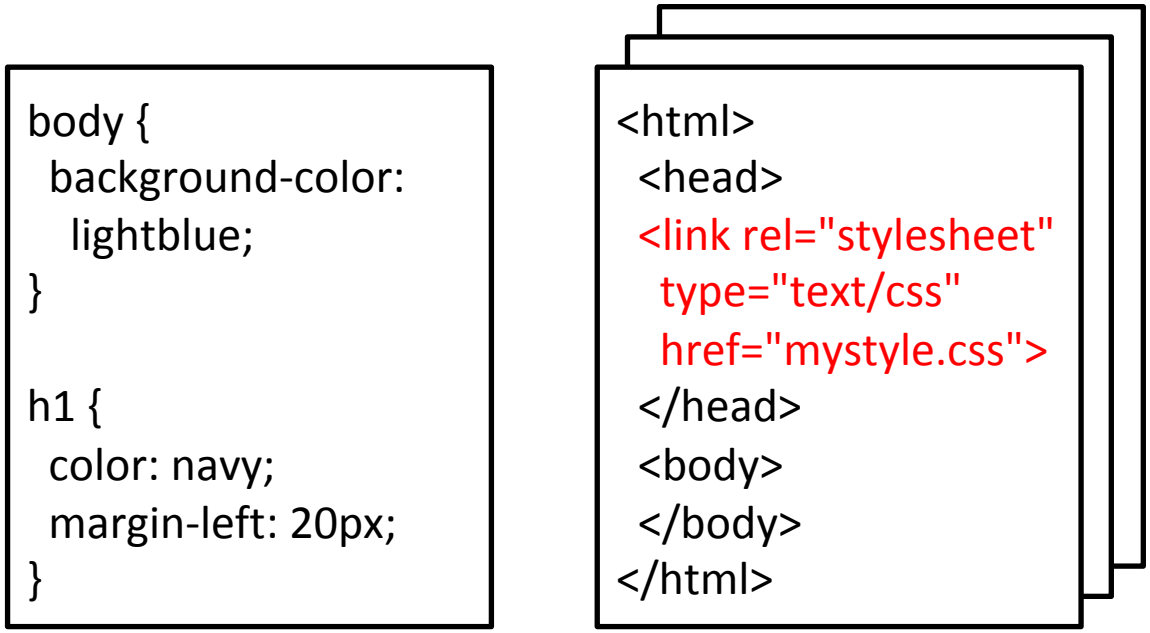
Statique vs Dynamique

- Site web statique :
  Aucun changement dans les pages lorsque l'on « rafraichit »/refait la même requête

- Dynamique :
  Les pages évoluent selon les informations externes (à chaque requête, en général)

Statique vs Dynamique

- Site web statique



Statique vs Dynamique

- Site web dynamique

```
body {
  background-color:
    lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```

CSS

```
<html>
 <head>
 <link rel="stylesheet"
   type="text/css"
   href="mystyle.css">
 </head>
```

HTML

```
<?php
include("head.html");
echo "<body>";
Var = 3 + 5;
echo "Valeur : $Var";

echo "</body>";
include("foot.html");
?>
```

PHP

Apache, URI/URL, DNS, PHP

- Serveur Web : Apache
(traite les connexions et transmet les requêtes)

- S'appuie sur HTTP, URI/URL, DNS
(pour communiquer ainsi que comprendre quelle ressource et quel site sont visés)

- Extension au serveur web : PHP
(lit les requêtes qu'Apache lui transmet, et fait les traitements demandés)

Apache & PHP

- Apache va traiter les connexions et requêtes
- PHP va construire la réponse en exécutant la logique métier (le code)

Navigateur

Serveur Web

Apache

Apache

- Apache :
  Serveur web libre et gratuit
  Dispose d'extensions pour se lier à PHP ou à d'autres outils pour traiter les requêtes

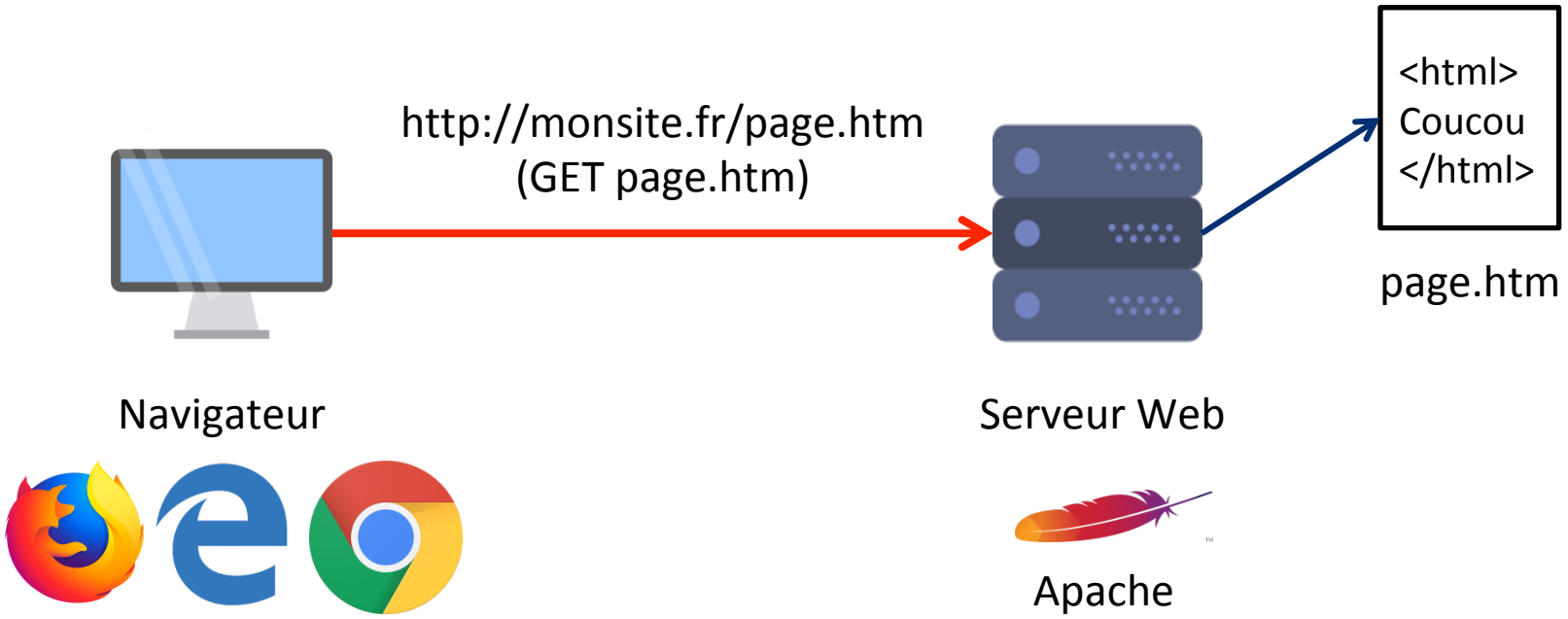
Navigateur

Serveur Web

Apache

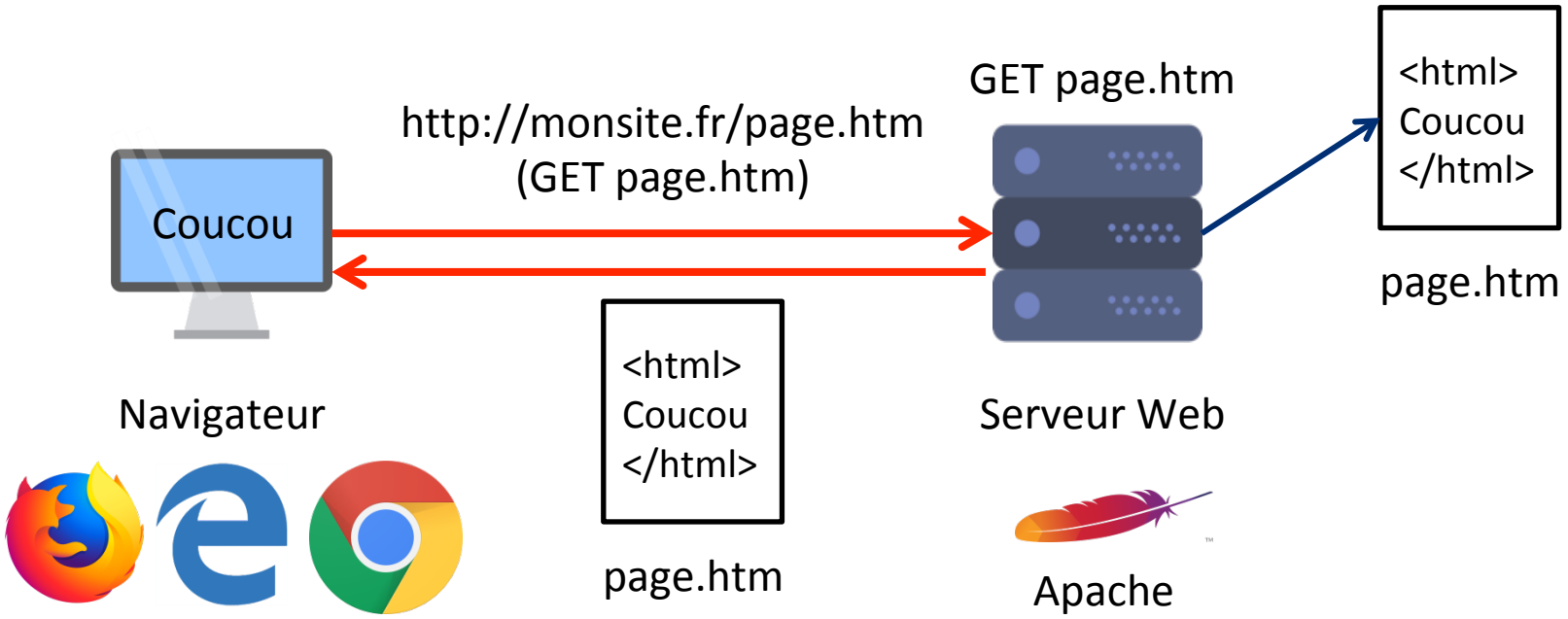
Apache

- Apache :
  Usage classique associe des fichiers à des requêtes



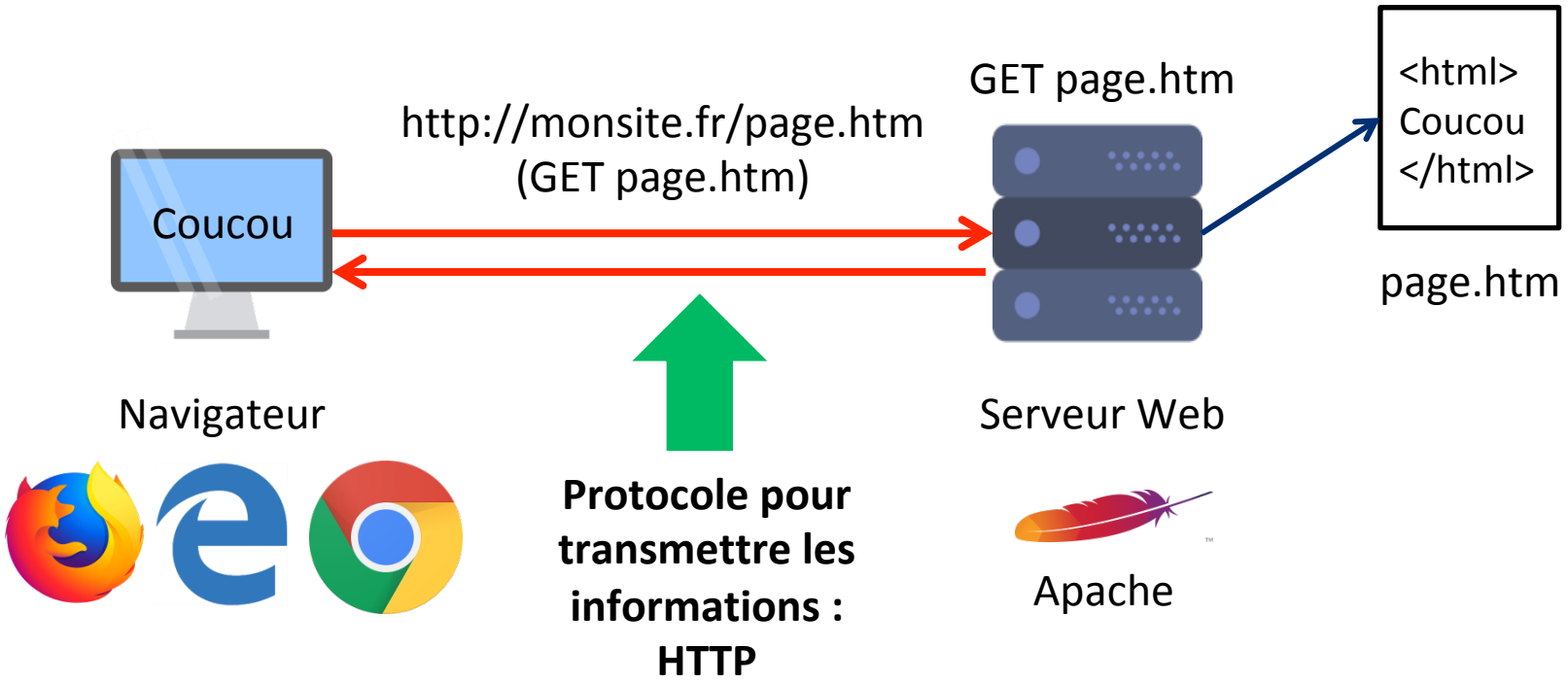
Apache

1. Client envoie une requête avec la ressource visée
2. Apache lit la requête, et cherche le fichier
3. Apache répond à la requête en envoyant le fichier



Apache

- Requête et réponse transmises avec :
  - HTTP 1.0 ou HTTP 1.1 ou HTTP/2
  - (HyperText Transfer Protocol)



Apache

```
$ telnet www.perdu.com 80
Trying 208.97.177.124...
Connected to www.perdu.com.
Escape character is '^]'.
```

Connexion au serveur par telnet

```
GET / http/1.1
Host: www.perdu.com
```

Requête HTTP

```
HTTP/1.1 200 OK
Date: Sat, 17 Aug 2013 11:59:04 GMT
Server: Apache
Accept-Ranges: bytes
X-Mod-Pagespeed: 1.1.23.1-2169
Vary: Accept-Encoding
Cache-Control: max-age=0, no-cache
Content-Length: 204
Content-Type: text/html
```

Réponse du serveur : headers

```
<html><head><title>Vous Etes Perdu ?</title></head><body><h1>Perdu sur l'Interne
t ?</h1><h2>Pas de panique, on va vous aider</h2><strong><pre>    * <----- vous
&ecirc;tes ici</pre></strong></body></html>
```

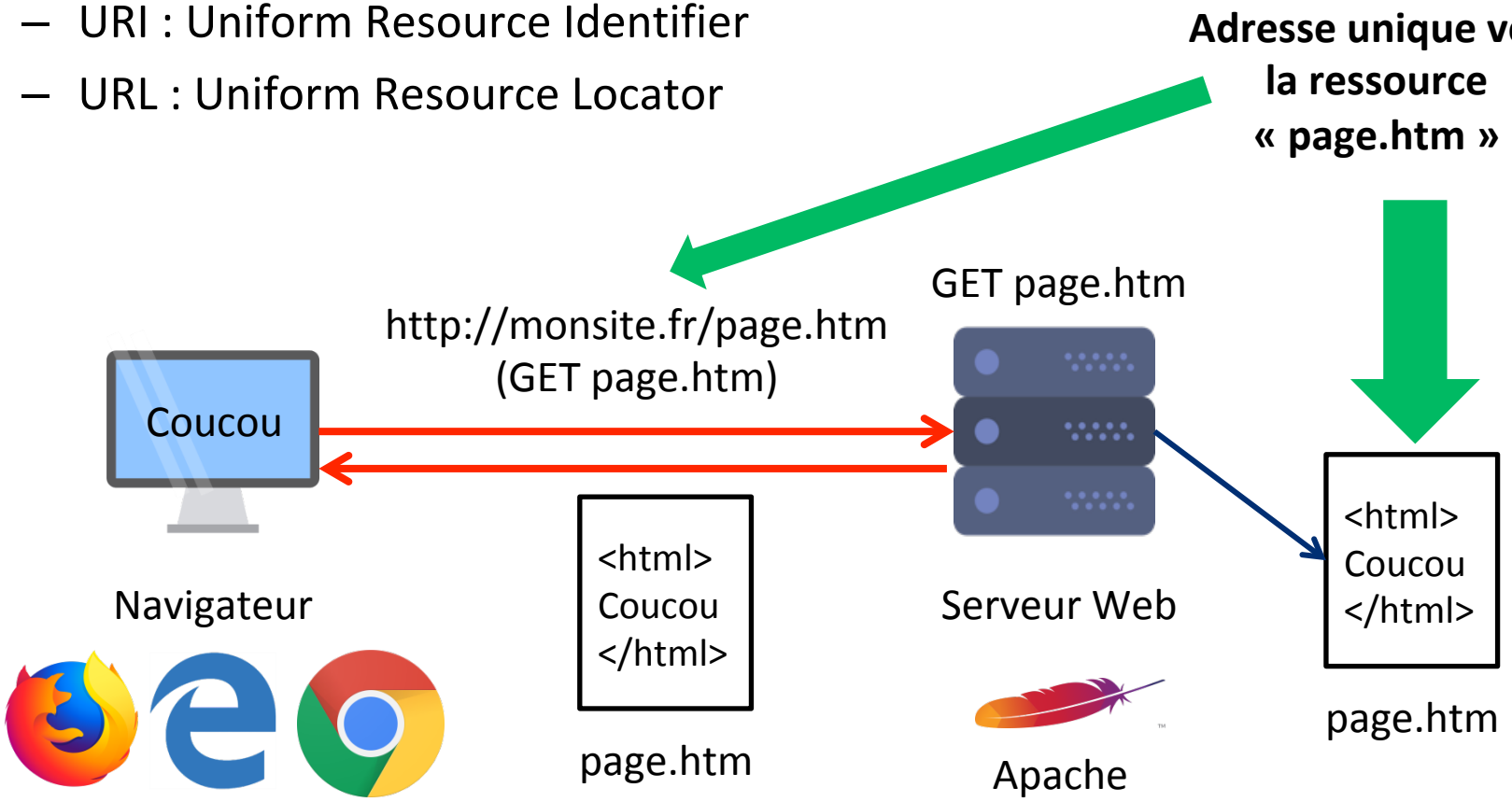
Réponse du serveur : body

Apache

- Serveur web s'appuie donc, sur :
  - Protocole HTTP (utilise généralement le port 80)
  - Protocole HTTPS (utilise généralement le port 443)
    [S pour « secure », en utilisant des certificats pour chiffrer la connexion ET pour s'assurer que l'hôte/host est bien celui que l'on cherche]
- Pour pouvoir communiquer avec les navigateurs

Apache & URI/URL

- Identification des ressources avec URI et URL
  - URI : Uniform Resource Identifier
  - URL : Uniform Resource Locator



Apache & URI/URL

- Identification des ressources avec URI et URL

- http://monsite.fr/page.htm
  - Ressource : « page.htm »
  - Dans le dossier : « / » (la racine)
  - Sur le site web : « http://monsite.fr »

- ftp://machine1.autresite.com/images/oiseau.jpg
  - Ressource : « oiseau.jpg »
  - Dans le dossier : « /images/ »
  - Sur la machine « machine1.autresite.com » accessible en « ftp:// »

Apache & URI/URL

```
$ telnet www.perdu.com 80                                  Connexion au serveur par telnet
Trying 208.97.177.124...
Connected to www.perdu.com.
Escape character is '^]'.

GET / http/1.1                                             Requête HTTP
Host: www.perdu.com

HTTP/1.1 200 OK                                            Réponse du serveur : headers
Date: Sat, 17 Aug 2013 11:59:04 GMT
Server: Apache
Accept-Ranges: bytes
X-Mod-Pagespeed: 1.1.23.1-2169
Vary: Accept-Encoding
Cache-Control: max-age=0, no-cache
Content-Length: 204
Content-Type: text/html

<html><head><title>Vous Etes Perdu ?</title></head><body><h1>Perdu sur l'Interne
t ?</h1><h2>Pas de panique, on va vous aider</h2><strong><pre>    * <----- vous
&ecirc;tes ici</pre></strong></body></html>
                                                           Réponse du serveur : body
```

Apache & DNS

- Chaque machine sur internet est identifiée par son IP

- Exemple IPV4 : 8.8.8.8 ou 127.0.0.1
- Exemple IPV6 : 2001:0db8:85a3:0000:0000:8a2e:0370:7334

- Pas très pratique pour visiter des sites web…
  …mais parfait pour rendre unique une machine

Apache & DNS

- DNS : Domain Name System
- Gère les « noms de domaine » (exemple : *univ-paris1.fr)*

- Permet de lier un nom de domaine à une ou des IP
- Permet de gérer des « sous-domaines »

- Exemple : univ-paris1.fr
  www.univ-paris1.fr ent.univ-paris1.fr

- « www » et « ent » sont des sous-domaines

Apache & DNS

```
$ telnet www.perdu.com 80
Trying 208.97.177.124...
Connected to www.perdu.com.
Escape character is '^]'.
```

Connexion au serveur par telnet

```
GET / HTTP/1.1
Host: www.perdu.com
```

Requête HTTP

```
HTTP/1.1 200 OK
Date: Sat, 17 Aug 2013 11:59:04 GMT
Server: Apache
Accept-Ranges: bytes
X-Mod-Pagespeed: 1.1.23.1-2169
Vary: Accept-Encoding
Cache-Control: max-age=0, no-cache
Content-Length: 204
Content-Type: text/html
```

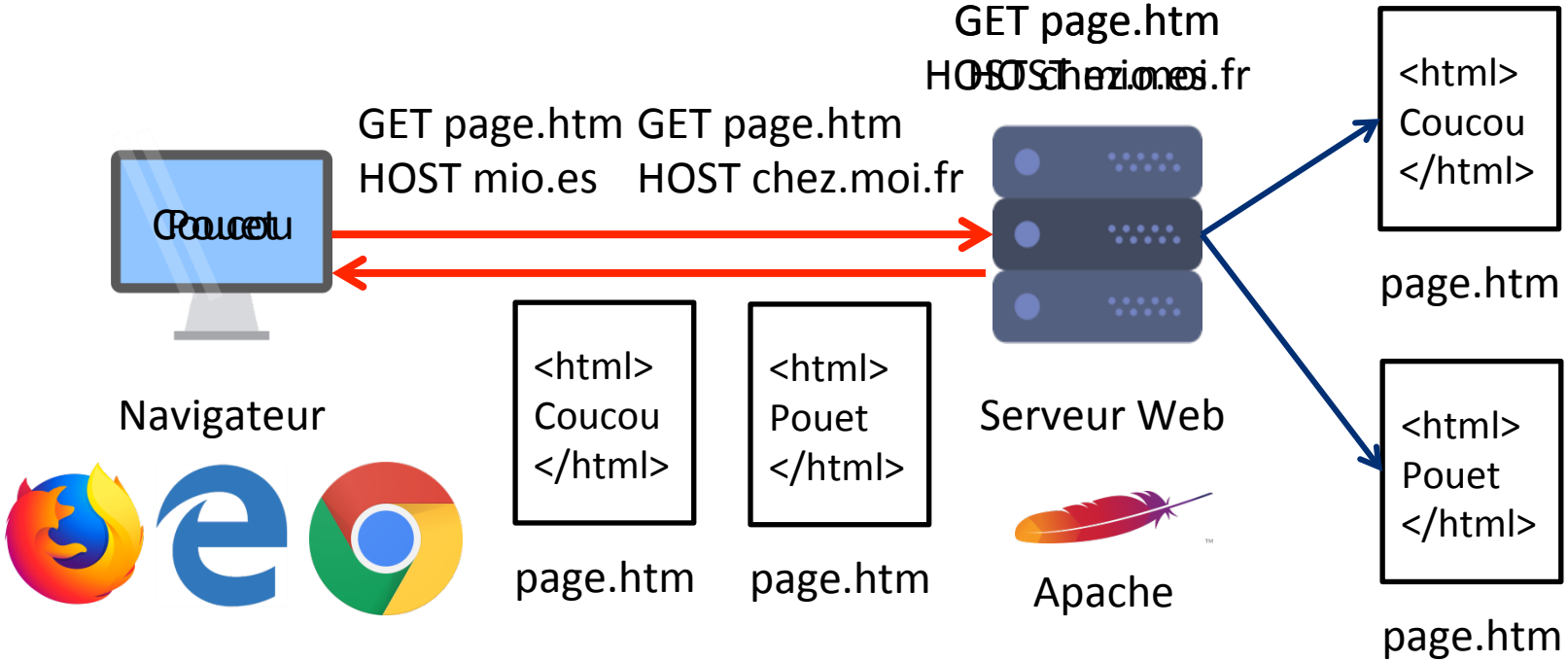
Réponse du serveur : headers

```
<html><head><title>Vous Etes Perdu ?</title></head><body><h1>Perdu sur l'Interne
t ?</h1><h2>Pas de panique, on va vous aider</h2><strong><pre>    * <----- vous
&ecirc;tes ici</pre></strong></body></html>
```

Réponse du serveur : body

Apache & DNS

1. Client envoie une requête avec la ressource visée
2. Apache lit la requête, et cherche le fichier
3. Apache répond à la requête en envoyant le fichier

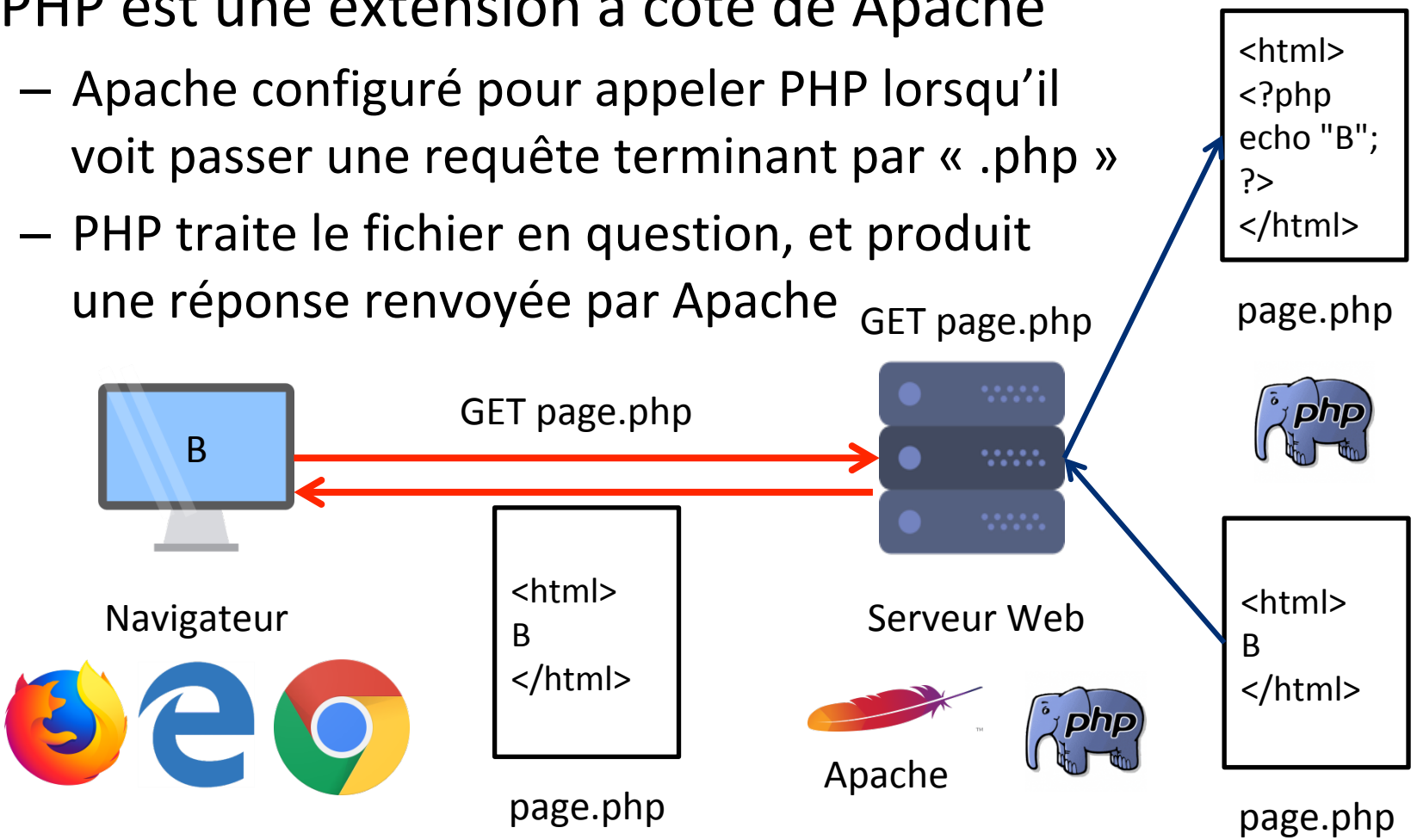


Apache & DNS

- Plusieurs noms de domaine peuvent renvoyer vers la même IP
- Le serveur web peut donc afficher des sites différents selon l'hôte demandé dans la requête
  - Chez Apache, on appelle cela des « virtual hosts » (vhosts)
  - Pour gérer cela, vous devez avoir accès à la configuration du serveur web. Ce qui n'est pas toujours le cas lorsque vous souscrivez à un service « d'hébergement web ».
- Pour avoir un « bon » site, il est utile de lui choisir un nom de domaine pertinent
  - Les certificats pour mettre du HTTPS se basant sur les noms de domaine, et pas sur les IP

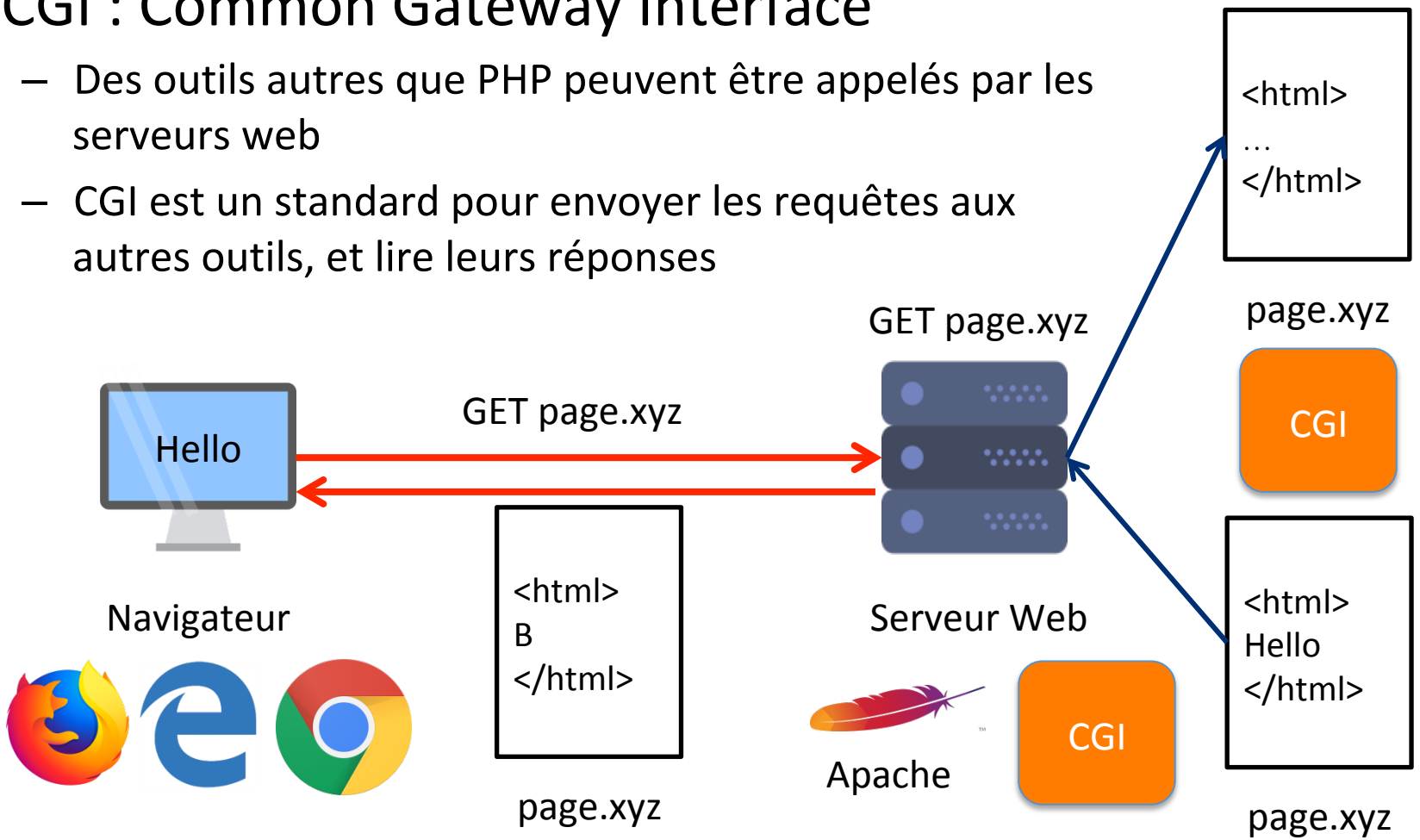
Apache & PHP

- PHP est une extension à côté de Apache
  - Apache configuré pour appeler PHP lorsqu'il voit passer une requête terminant par « .php »
  - PHP traite le fichier en question, et produit une réponse renvoyée par Apache



Apache & CGI

- CGI : Common Gateway Interface
  - Des outils autres que PHP peuvent être appelés par les serveurs web
  - CGI est un standard pour envoyer les requêtes aux autres outils, et lire leurs réponses



Objectifs du cours :

Création d'un site Web dynamique

PHP

PHP

- **PHP** est un **langage de programmation** utilisé pour la construction de **sites Web dynamiques**
  - **Pages PHP** : pages Web qui contiennent de PHP
    - On va **mélanger le PHP** au code **HTML / CSS**
    - Le code **PHP** va être **analysé par le serveur**
    - Le **résultat** va être une **nouvelle page Web** mise à jour automatiquement par le code PHP

Le code PHP est à l'intérieur de la balise **<?php … ?>** ou entouré par la balise **<script language="php"> … </script>**

coursPHP-1.php

```
<html> …
 <?php
     date_default_timezone_set("Europe/Paris");
     echo "<p style='font-style: italic;'> Paris, le "
          .date('d / m / Y')."</p>" ;
 ?>
… </html>
```

PHP

PHP : Php Hypertext Preprocessor.

- Langage interprété pour créer des sites dynamiques
- Langage de script côté serveur
- Langage faiblement typé
- Langage « Embedded HTML »
- Open source : PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook, Wikipédia, etc.

Commandes PHP de base

- Variables
- Types
- Opérateurs
- Fonctions
- echo, gettype, unset

Installation et configuration de PHP

Il suffit de télécharger la suite logicielle :

- WAMP : http://www.wampserver.com/
- MAMP : http://www.mamp.info/
- XAMPP : https://www.apachefriends.org/fr/

Et un éditeur de texte :

- Sublim Text (macOs): http://www.sublimetext.com/2
- Notepad++ (Windows) : https://notepad-plus-plus.org/fr/
- Autres (Linux/BSD/UNIX) : emacs, vim, nano, gedit, …

Introduction au PHP

Syntaxe de base :

```
<!doctype html>
<html>
    <head>
<title>Titre</title>
    </head>
<body>
    <?php echo "Hello World !"; ?>
</body>
</html>
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//
DTD HTML 4.01//EN" "http://
www.w3.org/TR/html4/strict.dtd">
```

Commentaires PHP

Commentaires :

Commentaires hérités du langage C et Perl

// Ceci est un commentaire sur une seule ligne

***/* Ceci est un commentaire sur plusieurs lignes**
***/**

Commentaire style shell

 # Ceci est un commentaire sur une seule ligne

Variables PHP

- **La notion de variable**
  - Une variable est **un conteneur de valeur**
  - On peut lui affecter une valeur, qu'on va utiliser plus tard

```
$variable = "PHP5" ;
```

Le « **$** » indique une variable

Le **nom de variable** commence toujours par une **lettre** ou un « _ », **sans espace**

Le « **=** » est une **affectation** On attribut une valeur à la variable

```
echo "... $variable ..." ;
```

On récupère la valeur gardée dans la variable par son nom

Variables PHP

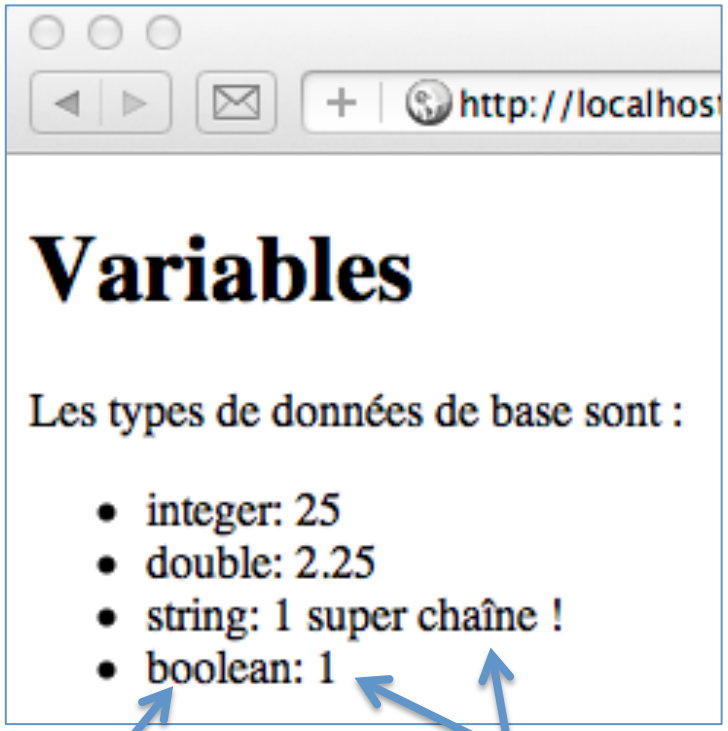
- **La notion de variable : les types des données**
  - Les variables peuvent garder de valeurs de différents types
    - **Nombres entiers** (**integer**) : 25
    - **Nombres décimaux** (**double** ou **float**) : 2.25
    - **Chaînes de caractères** (**string**) : « 1 super chaîne ! »
    - **Logique** (**boolean**) : « **true** » (1) ou « **false** »
  - La fonction **gettype($variable)** permet de savoir quelle type de valeur contient la variable
    - $entier = 25; gettype($entier) ➔ integer
    - $decimal = 2.25; gettype($decimal) ➔ double
    - $chaine = "1 super chaîne !"; gettype($chaine) ➔ string
    - $bool = true; gettype($bool) ➔ boolean

Variables PHP

- Exemple :

```
<?php
    $entier = 25;
    $decimal = 2.25;
    $chaine = "1 super chaîne !";
    $boolean = true;

    echo "<li>" . gettype($entier) . ": $entier </li>";
    echo "<li>" . gettype($decimal) . ": $decimal </li>";
    echo "<li>" . gettype($chaine) . ": $chaine </li>";
    echo "<li>" . gettype($boolean) . ": $boolean </li>";
?>
```

Définition d'une **variable**



On **récupère** la valeur de la variable **$boolean**

gettype() informe le type de la variable

Valeur de chaque variable

Types de données PHP

Opérateur sur les chaînes de caractères :

– concaténation : chaine1 . Chaine2

Opérateurs logiques :

- AND ou && (vrai si $a et $b vrais)
- OR ou || (vrai si $a ou $b sont vrais)

Opérateurs arithmétiques :

- addition : $a + $b,
- soustraction : $a - $b,
- multiplication : $a * $b,
- division : $a / $b,
- modulo (reste de la division entière) : $a % $b.

Types de données PHP

Opérateurs arithmétiques :

- Attention : lorsqu'une chaine de caractère est évaluée comme une valeur numérique, les règles suivantes s'appliquent :

- **$toto = 1 + "4.5" ; # $toto vaut 5.5**

- **$toto = 1 + "titi + 149" ; # $toto vaut 1 car la chaine vaut 0 si c'est du texte ou,**

- **$toto = 1 + "149 + titi" ; # $toto vaut 150 car la chaine vaut 149 (commence par une valeur numérique).**

Types de données PHP

Opérateurs de comparaison :

- égal à : **$a == $b**
- différent de : **$a != $b**
- supérieur à : **$a > $b**
- inférieur à : **$a < $b**
- supérieur ou égal à : **$a >= $b**
- inférieur ou égal à : **$a <= $b**

Exemple : **echo $toto == 0 ? "Vrai" : "Faux" ;**

Opérateurs PHP

- **Opérateurs**
  - Différents opérateurs permettent de manipuler des valeurs, qu'ils soient dans les variables ou pas

Opérateurs mathématiques	Opérateurs String	Opérateurs de comparaison	Opérateurs logiques
+ - * / %	. *(concaténation)*	== != <= < >= >	|| (OR) && (AND) ! (*not*)

```
<?php
    $a = 2 + 3 ;
    $b = 4 - $a ;
    $nom = "Toto";
    echo "Salut " . $nom;
    echo "<p> 4 - $a vaut $b  </p>";
 ?>
```

Exemple avec les opérateurs :

Salut Toto

4 - 5 vaut -1

Fonctions PHP

Date

Déclaration avec :

- DATETIME

```
$date = new DateTime('2000-01-05');
```

- DATE_CREATE : un Alias de DateTime::__construct()

```
$date2 = date_create('2000-01-01');
```

Extraction du Mois, année , …… :

Avec style procédural en utilisant DATE_FORMAT

```
echo date_format($date, "m");
```

Avec style orienté objet (OO) :

```
echo $date->format('Y');
```

Fonctions PHP

Date

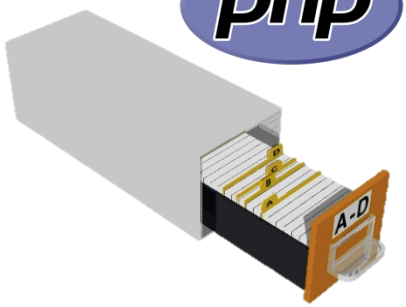
Exemples :

- **$date=date("d-m-y ");**
  echo " ceci est la date du jour " .$date ;

- **$heure = date("h:i:s");**

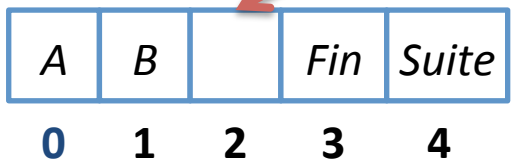
echo "c'est l'heure du jour " .$heure ;

Tableaux PHP

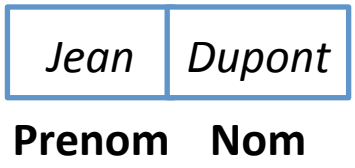


- **Tableaux**
  - Variables capables d'enregistrer plusieurs valeurs d'un type
- **Tableaux à indice :**
  - Chaque position est identifiée par un numéro (commençant par **0**)
    - **$tableau** [**0**] = "A";
    - **$tableau** [**1**] = "B";
    - $tableau [**3**] = "Fin";
    - **$tableau [ ]** = "Suite";

Attention a définir toutes les positions avant de les utiliser ou il y aura une **message d'erreur**.



- **Tableaux associatifs :**
  - Chaque position reçoit un identifiant (un label)
    - **$tableauAssoc** ["**Prenom**"] = "Jean";
    - **$tableauAssoc** ["**Nom**"] = "Dupont" ;



Tableaux PHP

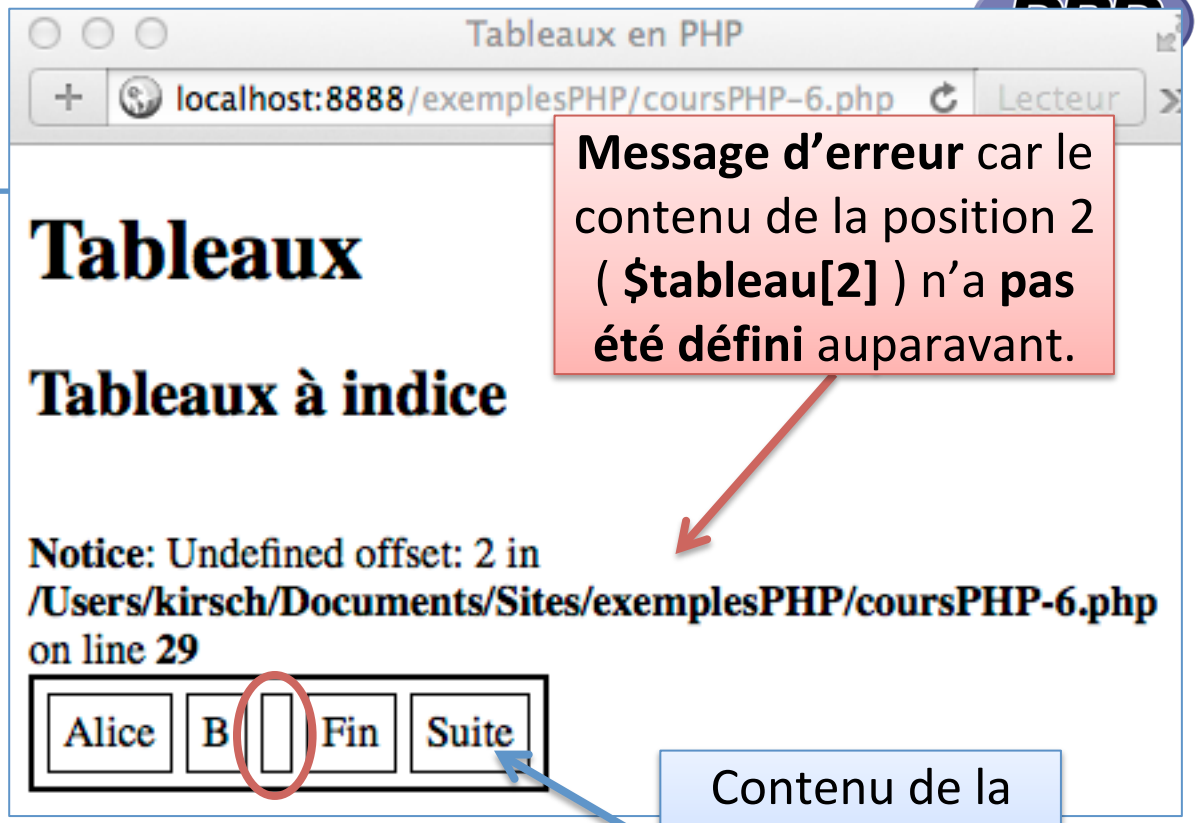
Les tableaux de PHP ressemblent aux tableaux associatifs (*hash-tables*).

- L'index est appelé ***clé***
- La valeur associée à la clé est appelée ***valeur***

On déclare un tableau de deux façons :

- Utiliser la fonction **array()** pour créer un tableau
- Ou affecter directement les valeurs au tableau

```
<head>  …
      <style>… </style>
</head>
<body>  …
 <h2>Tableaux à indice </h2>
 <table>
 <?php
     $tableau [0] = "A";
     $tableau [1] = "B";
     $tableau [3] = "Fin";
     $tableau [] = "Suite";

     echo "<tr> <td>". $tableau[0] . "</td> <td>".  $tableau[1]
         . "</td> <td>" . $tableau[2] . "</td><td>" . $tableau[3]
         . " </td><td>" . $tableau[4]  . "</td></tr>" ;
 ?>
 </table>
 …
```



Message d'erreur car le contenu de la position 2 (**$tableau[2]**) n'a **pas été défini** auparavant.

Contenu de la position 4 (**$tableau[4]**)

Tableaux PHP

```
...
<h2>Tableau associatif </h2>
 <table>
   <tr> <th> Nom </th> <th>Prénom </th> </tr>

  <?php
    $tableauAssoc ["Prenom"] = "Jean";
    $tableauAssoc ["Nom"] = "Dupont" ;

    echo "<tr> <td>" .  $tableauAssoc ["Nom"] . "</td>" ;
    echo "<td>" . $tableauAssoc ["Prenom"] . " </td></tr>" ;
  ?>

 </table>
</body>
```

Tableau associatif

Nom	Prénom
Dupont	Jean

Tableaux PHP

Fonctions sur les tableaux :

- **sizeof()** : retourne le nombre d'éléments d'un tableau, ou

- **count()** : retourne le nombre d'éléments d'un tableau s'il existe, 1 si la variable n'est pas un tableau et 0 si la variable n'existe pas.

Tableaux PHP

Exemple:

- $suite = array(1, 2, 3, 4) ;
- $tab[0] = 1 ;
- $tab[1] = "toto" ; # on peut mélanger les contenus
- $tab["chaine"] = " valeur" ; # on peut mélanger les clés.
- $personne = array("type" => "M.", "nom" => "Smith") ;

Tableaux PHP

Parcourir un tableau :

```
<?php
// On crée notre array $prenoms
$prenoms = array ('François', 'Michel', 'Nicole', 'Véronique',
'Benoît');
// Puis on fait une boucle pour tout afficher :
for ($numero = 0; $numero < 5; $numero++)
{
        echo $prenoms[$numero] . '<br />';
}
?>
```

Tableaux PHP

Parcourir un tableau :

```
<?php

$prenoms = array ('François', 'Michel', 'Nicole', 'Véronique',
'Benoît');

foreach($prenoms as $id => $valeur)

{

        echo "Case ($id) = $valeur";

}
?>
```

Tableaux PHP

Parcourir un tableau associatif :

```
<?php
$personne = array("type" => "M.", "nom" => "Smith") ;
foreach($personne as $cle => $valeur)
{
        echo "cle=" .$cle .“ valeur= " . $valeur ;
}
?>
```