

Systemes d'Information

Guillaume Rivière
g.riviere@estia.fr

30 mai 2007

support de cours réalisé par
Fabrice Depaulis

Mis à jour par
Guillaume Rivière

Introduction

✓ Continuité ...

- I1 – I2 : Mécanismes
- I3 : Mise en place, utilisation (prestataire), ...

✓ I2 : Echange de données

Comprendre les principes sur lesquels reposent les outils de gestions des systèmes d'information.

1. Développement d'outils ad hoc de recueils, traitements et restitutions d'information.
2. Intégration d'outils dans l'entreprise ou E.A.I, Entreprise Application Integration.

PHP : un outils simple et puissant

Pour générer dynamiquement du contenu Web



Forum



www.phpbb.fr

Gestion de BDD



www.phpmyadmin.net

Nombreux CMS



www.spip.net/fr

Organisation du module SI

- ✓ Les SI dans l'entreprise
- ✓ Les Architectures Client/Serveur supports aux SI

- ✓ L'intégration des technologies hétérogènes au coeur du S.I.

CM : $2 \times 2h$

TP : $3 \times 2h$

1. Gestion et traitement des infos

Développement d'outils de recueils, traitements et restitutions d'information.

TP : Mise en place d'un formulaire avec accès sécurisé.

2. Intégration d'outils dans l'entreprise

E.A.I, Entreprise Application Integration.

TP : Intégration du logiciel de CAO Pro-Engineer de chez PTC et du logiciel de calcul formel Mupad, logiciel libre du Mupad Group, (c) SciFace Software GmbH & Co. KG.

Plan du cours

✓ L'Internet et le WWW



✓ Notions de XHTML et HTML

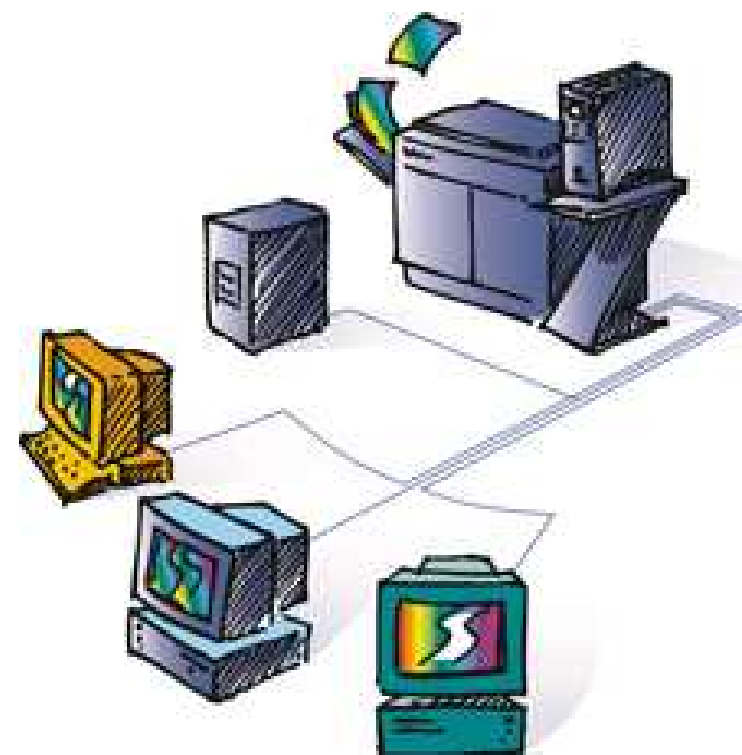


✓ Introduction au
langage php



L'Internet

Définition : un réseau est un ensemble de machines éventuellement hétérogènes reliées entre elles.



But

- Partager des ressources matérielles (imprimantes, disque dur, ...)
- Partager des ressources logicielles (programmes, données, ...)
- Echanger des informations

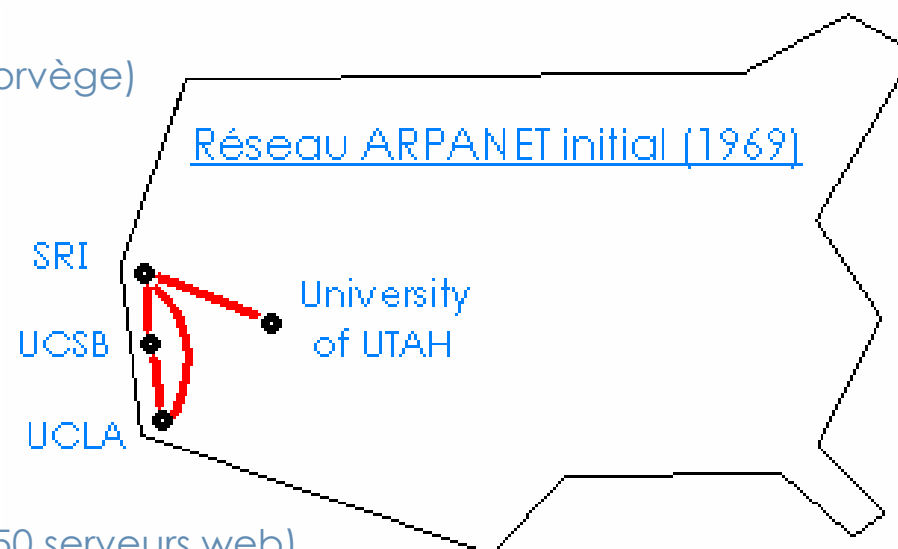
L'Internet

- ✓ InterconnectedNetwork : Réseau des réseaux
- ✓ Historique
 - 1957 : Spoutnik
 - 1969 : ARPANET (Advanced Research Projects Agency NETwork) pour la défense américaine
 - 1970 : Apparition du protocole TCP/IP (2 couches OSI)
 - 1980 : Connexion des réseaux US ARPANET : NSFNet (National Science Foundation Network).
 - 1990 : Connexion des différents réseaux mondiaux (Ca*Net, ...)

L'Internet en statistiques

Nombre de machines

1969 --	4
1971 --	23
1973 --	35 (UK, Norvège)
1976 --	111
1982 --	235
1983 --	562
1984 --	1 024
1985 --	1 961
1986 --	5 089
1987 --	28 000
1988 --	50 000
1989 --	100 000
1990 --	300 000
1992 --	1 000 000 (dont 50 serveurs web)
1994 --	4 000 000
1998 --	45 000 000



Nombre de sites web

1992 --	26
juin 1993 --	130
déc 1993 --	600
juin 1994 --	2 738
déc 1994 --	10 000
1995 --	19 000
1996 --	400 000
1997 --	1 400 000
1998 --	3 200 000
1999 --	4 370 000
2000 --	21 200 000
2001 --	32 400 000
2002 --	35 750 000
2003 --	42 300 000
2004 --	55 000 000
2005 --	70 000 000

1971 -- création de l'email (SNDMSG & READMAIL)

1973 -- email représente 75% du trafic ARPANET

1977 -- Internet devient réalité (D'autres réseaux sont connectés à ARPANET)

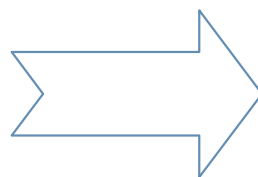
1991 -- arrivée du www (web, w3, toile)

1995 -- www devient le service le plus important de l'internet

Le World-Wide Web



TCP/IP
couche basse de
communication



1970s
Normalisation des
échanges (email, ftp, ...)

WWW : Service de l'Internet

- ✓ Conçu en 1989 au CERN par Tim Berners Lee
 - Projet interne au CERN
 - Besoin de partage d'informations de natures diverses (texte, données, images) au niveau mondial entre communautés de physiciens

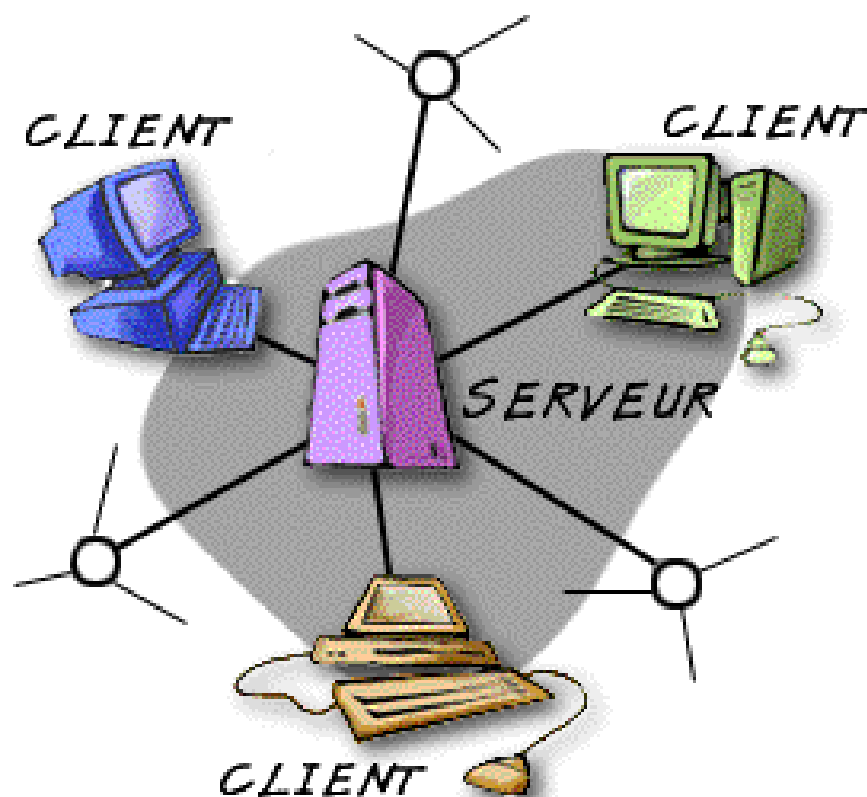
Ne pas confondre Internet et W3

Le World-Wide Web

Rappel : Modèle Client/Serveur

Serveur

Ordinateur connecté au réseau et fournissant au moins un **service**



Service (daemon)

Application tournant sur un **serveur** et capable d'accomplir une tâche en réponse à la requête d'un **client**, laquelle respecte un **protocole** donné

Apports du WWW

✓ HTML

- Langage de structuration de documents (« *Hyper-Text Markup Language* »)
- Permet de définir des liens entre documents

✓ HTTP

- Protocole de transfert de documents HTML (« *Hyper-Text Transfer Protocol* »)
- Associé à un service (HTTPd)
- Modèle **client-serveur**

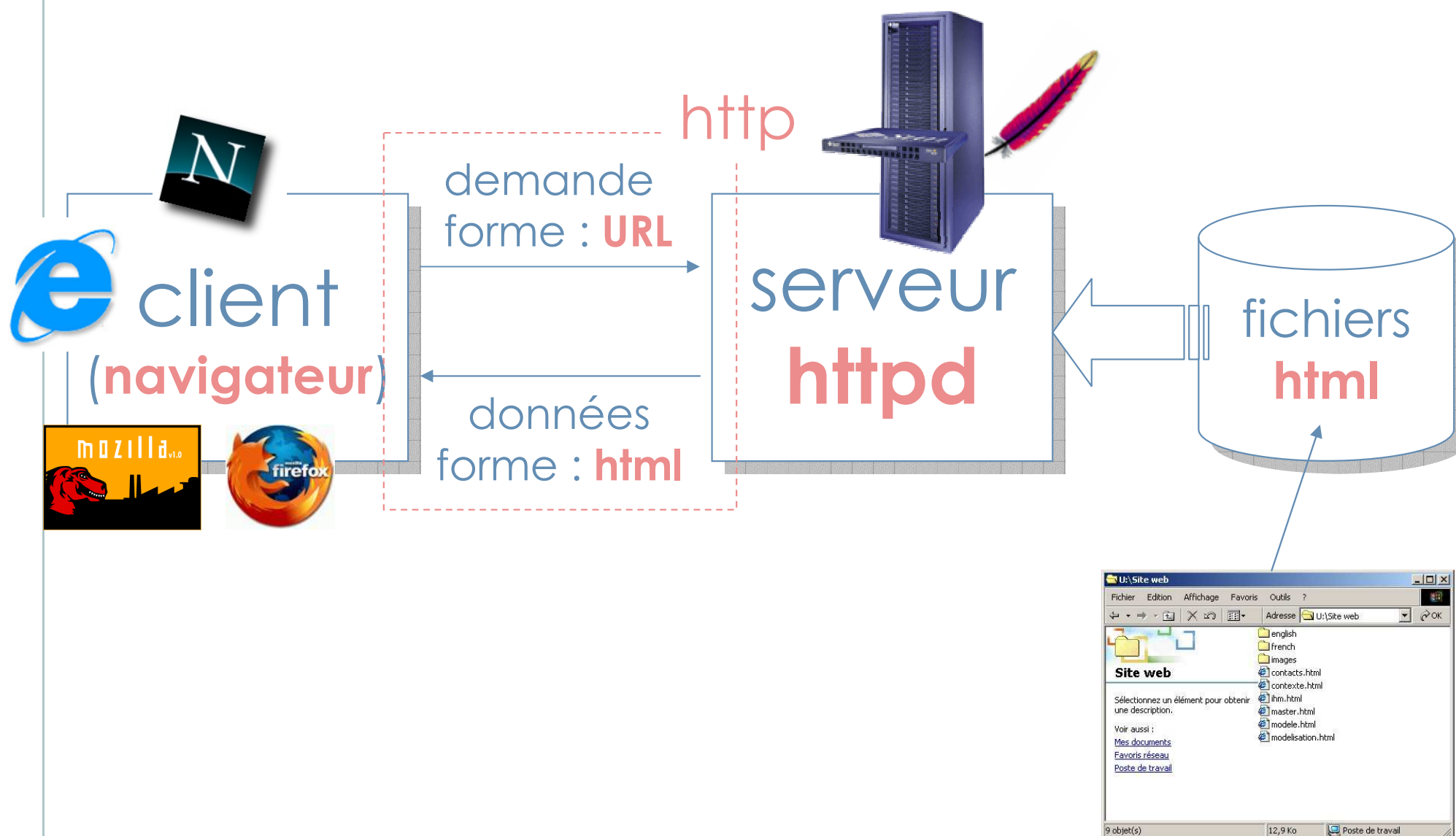
✓ URL

- Format d'adressage universel des ressources de l'Internet (« *Uniform Resource Locator* »)
 - Documents
 - Services existants : mail, news, telnet, ftp, ...

✓ Navigateur / Butineur / ...

- Logiciel graphique de lecture et de navigation au sein du WWW (**client**)
- Propose une interface pour chaque service pour lequel il existe un type d'URL

Apports du WWW



Le protocole HTTP

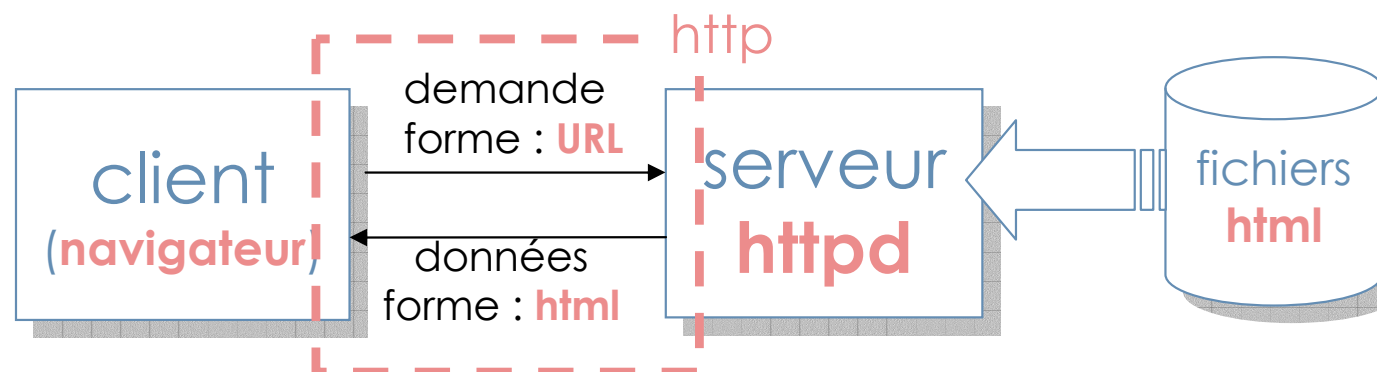
Http est un protocole de dialogue entre le navigateur et le serveur de documents HTML

But

Permettre un transfert de fichiers (HTML) localisés grâce à une chaîne de caractères appelée **URL** entre un navigateur (le client) et un serveur Web (http)

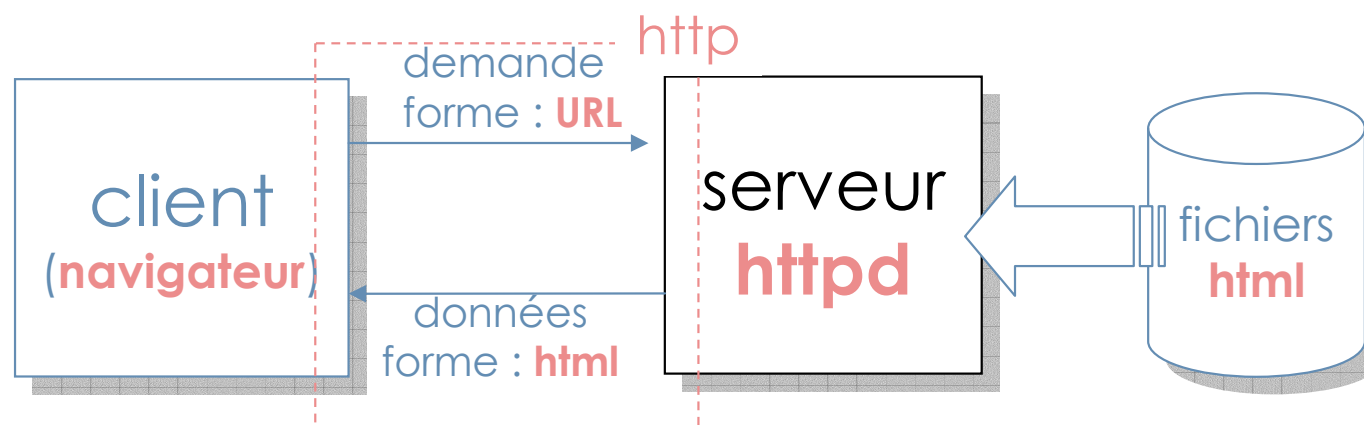
Caractéristiques

- ✓ Client-Serveur : seul le navigateur a l'initiative de la communication
- ✓ Non persistant : la connexion client/serveur ne dure que le temps de l'envoi du document demandé



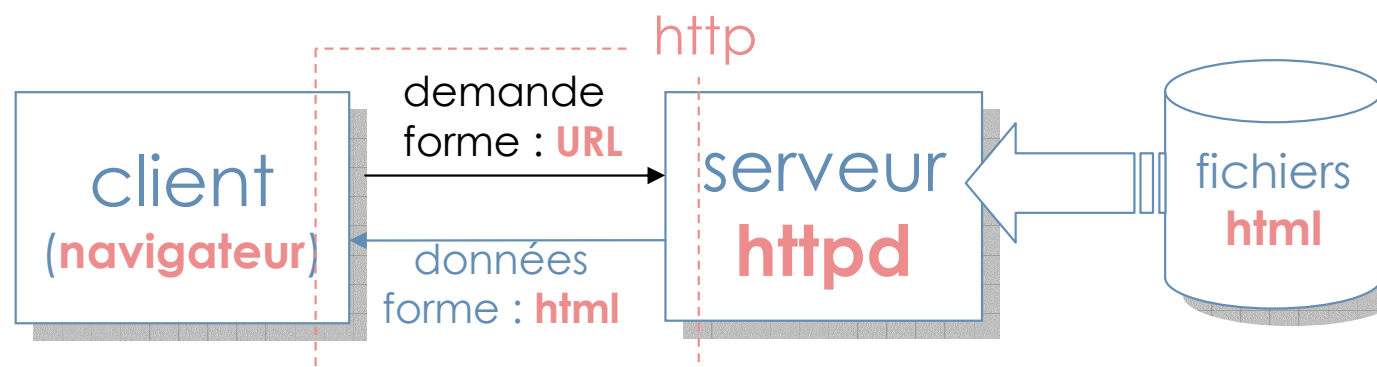
Le serveur HTTP

- ✓ En permanence à l'écoute des requêtes des clients (c'est un « démon » au sens Unix, d'où `httpd`)
- ✓ Renvoie un flot HTML correspondant au document demandé, qui peut être :
 - issu d'un fichier HTML stocké localement
 - généré par un programme spécifique (CGI)



URL

- ✓ Format d'adressage universel des ressources du web (« *Uniform Resource Locator* »)
- ✓ Permet au navigateur :
 - D'identifier le serveur et la localisation de données (locales ou distantes, document HTML ou autres)
 - D'accéder localement à certains services Internet : courrier, nouvelles, connexion à distance...
 - De demander au serveur l'exécution de programmes spécifiques (CGI, interprétation de script php)



URL

✓ Une URL est constituée :

- D'un nom de **protocole** (http par défaut)
- D'informations d'authentification (optionnelles)
- D'un nom ou d'une **adresse IP** de machine
- D'un numéro de **port** (80 par défaut)
- D'un **chemin local** à la machine
- Du **nom du document**
- D'**arguments** optionnels pour les programmes CGI

protocole://**username:password**@**machine**:**port**/**chemin**/**document**?**arguments**

Ex : `http://www.estia.fr/index.php?id=29`
`http://www.google.fr/search?q=ESTIA`
`http://www.estia.fr/uploads/media/general_estia.pdf`

`ftp://ftp.belnet.be/packages/damnsmaillinux/current/dsl-2.3.iso`

`mms://vipmms.canalplus.fr/canalplus/ba/brice-de-nice_060328_ba.wmv`

Plan du cours

✓ L'Internet et le WWW



✓ Notions de XHTML et HTML



✓ Introduction au
langage php



XHTML : le successeur de HTML

- ✓ XHTML = eXtensible HyperText Markup Language



<http://validator.w3.org/>

- ✓ XHTML n'est pas un langage sémantiquement figé mais au contraire un langage ouvert (*métalangage*)
- ✓ L'auteur d'un document XHTML peut créer ses **propres** balises.

XHTML : langage à balises

- ✓ Une balise est un mot clef, encadré par les caractères **<** et **>**

exemple :

la balise **<instrument>** peut être définie pour désigner un instrument de musique.

Cela s'écrirait de la façon suivante :

```
<instrument>
```

```
Guitare
```

```
</instrument>
```

XHTML : règles

- ✓ Dans un document XHTML, on ne doit pas tenir compte de la mise en forme.
- ✓ La mise en forme est réalisée par la feuille de style (*feuille CSS, cf 1ère année*)
- ✓ On prend en compte seulement du contenu de celui-ci que l'on peut, que l'on doit **structurer**.

XHTML : structuration

exemple

```
<vehicule>
  <typemine> ... </typemine>
  <chassis> ... </chassis>
  <moteur>
    <puissance> ... </puissance>
    <cylindree> ... </cylindree>
    <nbsoupapes> ... </nbsoupapes>
  </moteur>
  <couleur> ... </couleur>
  etc ...
</vehicule>
```

XHTML : du "HTML propre"

- ✓ XHTML 1.0 (Second Edition)
a reformulation of **HTML** 4 in XML 1.0



Premières différences de XHTML :

- mettre un **doctype** en haut du fichier
- fermer toutes les balises : ** ... **
- pour les balises simples, les fermer "de l'intérieur"
(comme **
** ou ****),
- toutes les balises en minuscules,
- les arguments entre guillemets,
- on définit ses propres balises

(!) Vision un peu réductrice

HTML

- ✓ Langage de structuration de documents (« *Hyper-Text Markup Language* »)
- ✓ HTML permet de
 - Publier des documents en lignes contenant du texte, des tableaux, des listes, ...
 - Lier des pages par des liens hypertextes
 - Concevoir des formulaires permettant d'effectuer des traitements d'informations
 - Insérer des documents dans d'autres formats : video, images, ...
- ✓ HTML définit le contenu et la structuration des informations au sein du document
- ✓ HTML ne définit pas l'apparence du document

HTML



Page Web

Ma première page.

Fichier html

interprété

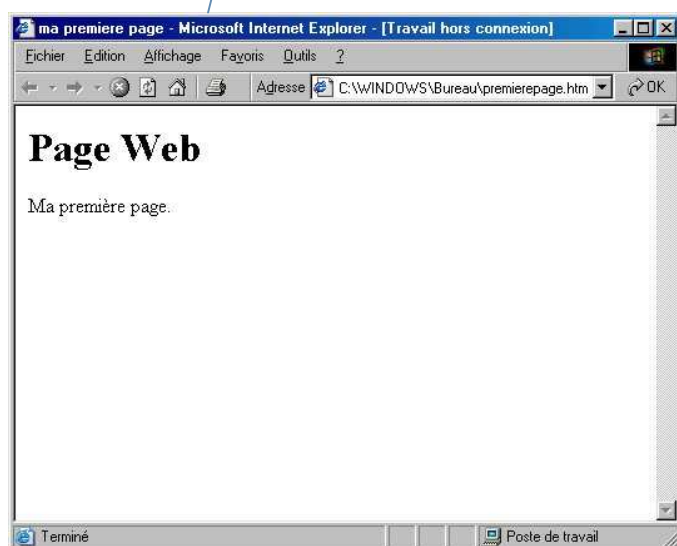
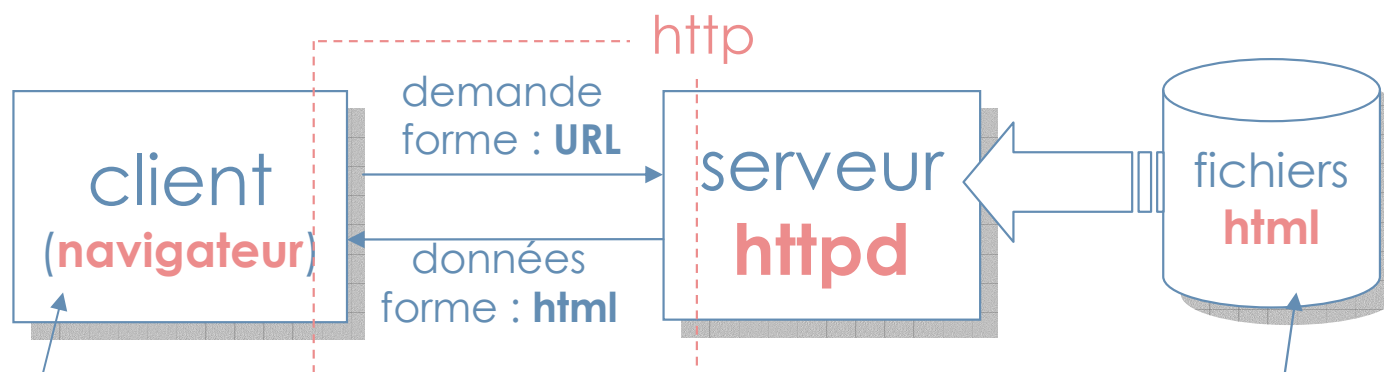
brut

```
<html>
<head>
<title>ma premiere page</title>
<meta http-equiv="Content-Type" content="text/html" />
</head>

<body>
<h1>Page Web</h1>
<p>Ma premi&egrave;re page.</p>
</body>
</html>
```

HTML

C'est le navigateur qui interprète le code HTML et lui donne son aspect



```
<html>
<head>
<title>ma premiere page</title>
<meta http-equiv="Content-Type" content="text/html" />
</head>

<body>
<h1>Page Web</h1>
<p>Ma premi&egrave;re page.</p>
</body>
</html>
```

Balises HTML

- ✓ HTML est un langage de balises
- ✓ Une balise est un mot clef, encadré par les caractères **<** et **>**
- ✓ L'effet d'une balise commence à **<nom_balise>** et se termine à **</nom_balise>**

`<i>texte en italique</i>`

texte en `gras`

`<i>texte en gras et italique</i>`

texte en italique

texte en gras et italique

texte en gras

Balises HTML

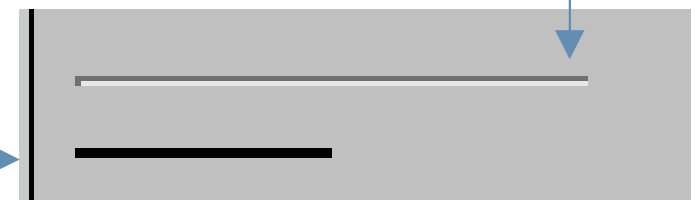
✓ Attributs d'une balise

- Situés entre le mot-clé de la balise et le « > »
- Constitués d'un identificateur et, si nécessaire, de la valeur de celui-ci, séparés par un « = »
- Les différents attributs d'une balise sont séparés par des espaces

Rmq : pas de `</hr>`

```
<hr style="text-align: left; width: 200px;" />  
<hr style="text-align: left;  
width: 100px" noshade />
```

(hr = Horizontal Rule)



Commentaires

Tout texte commençant par « <!-- » et se terminant par « --> » est considéré comme étant un commentaire

- Non interprété par le navigateur
- Non affiché

```
<html>
<head>
<title>ma premiere page</title>
</head>
<body>
<h1>Ma Page</h1>
<p>ma premi&egrave;re page web</p>
<!-- ne pas afficher cette partie -->
</body>
</html>
```



Format d'un document HTML

- ✓ Tout document HTML commence par la balise `<html>` et finit par la balise `</html>`
- ✓ Tout document HTML contient
 1. Un en-tête, délimité par les balises `<head>` et `</head>`
 2. Un corps, délimité par les balises `<body>` et `</body>`

```
<html>
<head>
<title>ma premiere page</title>
</head>
<body>
<h1>Ma Page</h1>
<p>ma premi&egrave;re page web</p>
</body>
</html>
```



1. En-tête `<head>`

- ✓ Délimité par les balises `<head>` et `</head>`
- ✓ Contient des informations générales sur le document, toujours chargées avant le corps
 - Titre du document `<title>`
 - Informations sur le contenu du document `<meta>`
 - Variables et fonctions des scripts JavaScript `<script>`
 - Les références aux feuilles de style `<link>`
 - ...
- ✓ Les balises utilisées sont spécifiques à l'entête
- ✓ Pas d'affichage dans le navigateur (en théorie)

Balise <title>

- ✓ Contenue dans l'en-tête du document
- ✓ Définit le titre du document, terminé par la balise </title>
- ✓ Le titre doit être court et explicite car il apparaît :
 - Comme titre de la fenêtre du navigateur
 - Dans la liste des signets (« bookmarks »)
 - Utilisé par les moteurs de recherche

```
<html>
<head>
<title>ma premiere page</title>
</head>
<body>
...
```



Balise **<meta>**

- ✓ Donne des informations précises sur le document
- ✓ Ces informations sont utilisées par les moteurs de recherche, par le navigateur pour authentifier le document ...

```
<html>
<head>
<title>ma premiere page</title>
<meta name="author" content="Fabrice" />
<meta name="description" lang="fr" content="il
s'agit d'une page de test" />
...
</head>
<body>
...
```

2. Corps `<body>`

- ✓ Délimité par les balises `<body>` et `</body>`
- ✓ Contient les informations affichables du document
 - Texte du document
 - Instructions

Titrage

Les balises h1, h2, ... h6 permettent de baliser un paragraphe comme étant un titre d'un certain niveau

```
<html>
<head>
<title>ma premiere page</title>
</head>
<body>
<h1>Ma Page</h1>
<h2>Titre1</h2>
<h3>Titre2</h3>
<p>Du texte normal</p>
</body>
</html>
```



Balise <p>

- ✓ Débute un paragraphe, terminé par </p>
- ✓ Un début de paragraphe provoque :
 - Un passage à la ligne
 - Un décalage d'une hauteur d'environ une ligne

Balise `
`

- ✓ Provoque un retour à la ligne dans la fenêtre d'affichage
- ✓ En l'absence des balises `
`, le texte n'est mis à la ligne que lorsque le bord de la fenêtre courante est atteint

```
Les sanglots longs<br/>  
Des violons<br/>  
De l'automne<br/>  
<br/>  
Blessent mon cœur...<br/>
```

```
Les sanglots longs  
Des violons  
De l'automne  
  
Blessent mon cœur...
```

```
Les sanglots longs Des violons De l'automne Blessent mon cœur...
```

Tableaux

- ✓ Balise `<table></table>`
- ✓ ~~Très utilisés pour la mise en page du texte et des images~~
- ✓ Les cellules des tableaux peuvent contenir toutes les balises courantes (et donc d'autres tableaux)
- ✓ Début du tableau : balise `<table>`
- ✓ Attributs : ~~align~~, `width`, ~~cols~~, `bgcolor`, `border`, `cellspacing`, `cellpadding`, ...

Tableaux

ligne

```
<body>
<table width="75%" border="1">
  <tr>
    <td>cel1</td>
    <td>cel2</td>
  </tr>
  <tr>
    <td>cel3</td>
    <td>cel4</td>
  </tr>
</table>
</body>
```



colonnes

Images

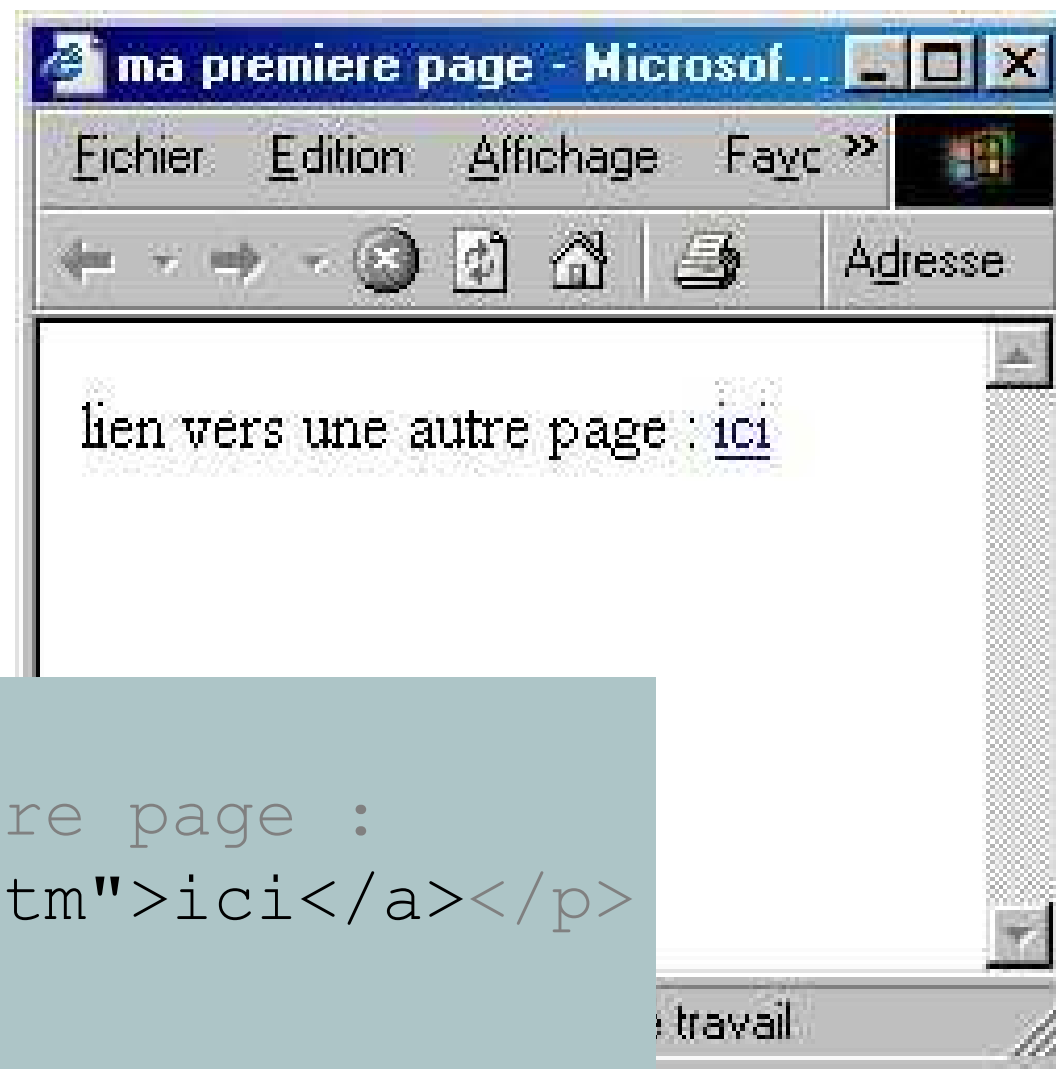
- ✓ La balise `` permet de placer une image sur le document
- ✓ Attributs :
 - `src` : url de l'image
 - `width` : largeur
 - `height` : hauteur

```
<body>  
<p>une image </p>  
<p>  
</p>  
</body>
```



Hyperliens

- ✓ Tout l'intérêt du HTML
- ✓ Balise `<a>`
- ✓ Attribut principal :
`href`



```
<body>
<p>lien vers une autre page :
<a href="autrepage.htm">ici</a></p>
</body>
</html>
```

Formulaires

- ✓ Les formulaires permettent à l'utilisateur d'être actif et de fournir de l'information au serveur, afin que celui-ci en tienne compte
- ✓ Le serveur traite les formulaires au moyen de «programmes» spécifiques

ma premiere page - Microsoft Internet Explorer...

Fichier Edition Affichage Favoris Outils ?

Adresse C:\

Calcullette F/E

Somme en francs:

Valeur en euros:

Calculer

Terminé Poste de travail

ma premiere page - Microsoft Internet Explorer...

Fichier Edition Affichage Favoris Outils ?

Adresse C:\

Mon formulaire

Email:

Abonnement à la newsletter ? ☒

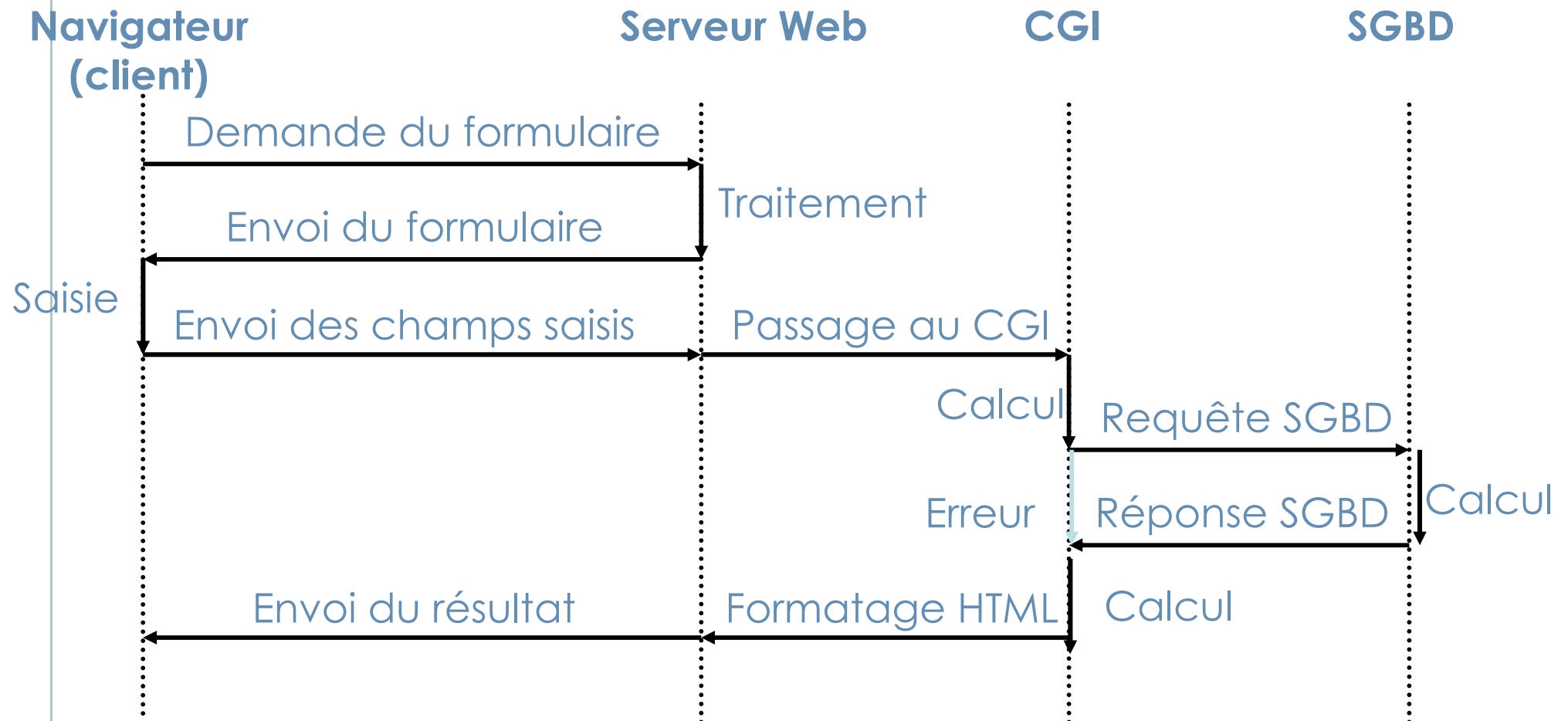
Envoyer Rétablir

Terminé Poste de travail

Formulaires

- ✓ Les formulaires
 - Définissent un ensemble de champs de saisie
 - Permettent de soumettre l'ensemble des valeurs des champs, en une fois, au serveur
- ✓ Le serveur
 - Traite ces informations au moyen d'un script php (par exemple)
 - Génère en retour un flot HTML qui peut encore être un formulaire (ré-essai ou procédure suivante)

Schéma d'une transaction type



Construction d'un formulaire

- ✓ Au moyen de balises HTML
- ✓ Définition des zones du formulaire
 - Nom
 - Type : texte, boutons, boutons radio, cases à cocher, ...
 - Valeurs par défaut

Balise `<form>`

Début un formulaire, terminé par `</form>`

Attributs :

`method=` : Méthode de transfert des valeurs des champs entre le navigateur et le serveur

- `get` : Les données sont accolées à l'URL du script CGI
- `post` : Les données sont envoyées à part (plus sûr)

`action=` : URL du script CGI chargé de traiter le formulaire

`id=` : Nom du formulaire

Quand on valide un formulaire : on appelle la page spécifiée par ACTION. Les paramètres sont les objets (zone de texte, bouton, ...) du formulaire, qui ont pour valeur les valeurs des objets du formulaire.

Exemple

ma premiere page - Microsoft Internet Explorer...

Fichier Edition Affichage Favoris Outils ?

Adresse C:\ OK

Calculatrice F/E

Somme en francs:

Valeur en euros:

Calculer

Terminé Poste de travail

```
<form action="monscript.php"
  method="post" id="calculatrice">
  <p> Somme en francs:
    <input name="francs" type="text" />
  </p>
  <p> Valeur en euros:
    <input name="euros" type="text" />
  </p>
  <p>
    <input type="submit" name="valider" value="Calculer" />
  </p>
</form>
```


Éléments de contrôle d'un FORM

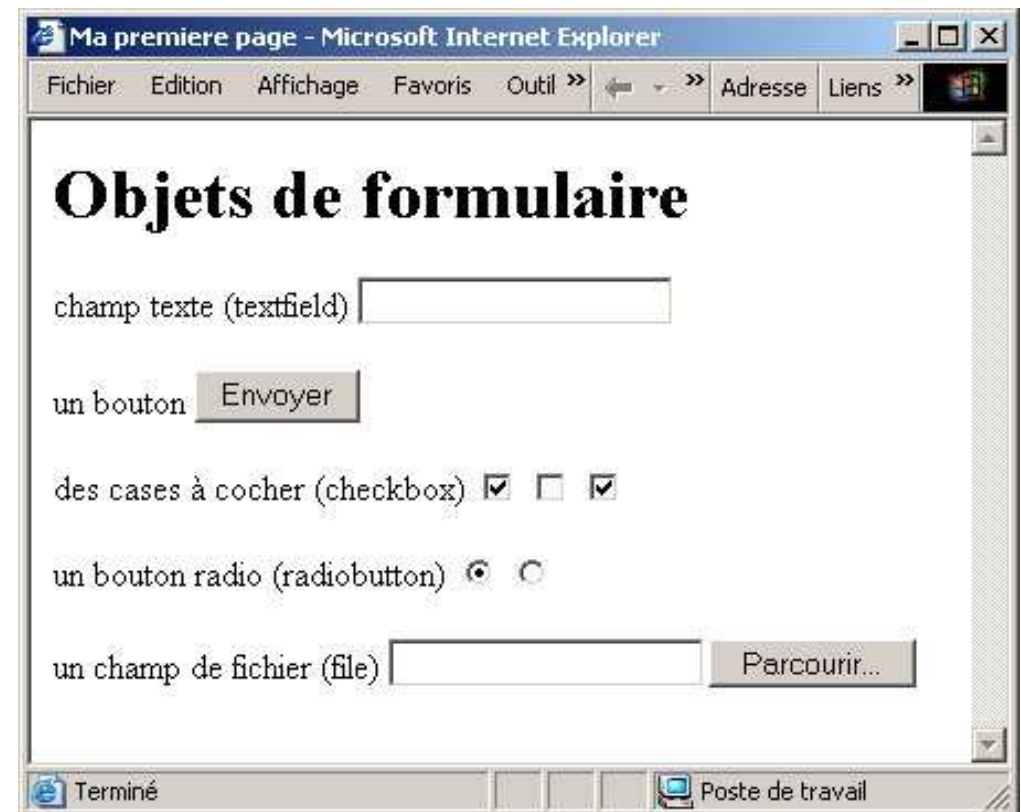
- ✓ Balise : `<input />`
- ✓ Aspect et comportement défini par l'intermédiaire de l'attribut « `type` »

✓ Autres attributs

name : nom du champ dans le formulaire

size : taille de la zone pour les saisies de texte

value : valeur par défaut



Balise `<input type="text" />`

- ✓ Définit une zone simple d'entrée de texte
- ✓ Le « `type="text"` » est optionnel
 - **name=** : Nom du champ dans le formulaire, permet au programme CGI d'identifier le champ
 - **size=** : Taille visible de la zone en caractères
 - **value=** : Valeur par défaut



```
<form action="monscript.php" method="post" id="form1">  
  <p>champ texte (textfield)  
    <input type="text" name="textfield" />  
  </p>  
</form>
```

Balise `<input type="password" />`

- ✓ Sert à la saisie de mots de passe
- ✓ Est identique à la zone `text`, à la différence que les caractères écrits s'affichent comme des étoiles « * »
 - **name=** : Nom du champ dans le formulaire
 - **size=** : Taille de la zone en caractères
 - **value=** : Valeur par défaut

Attention
ne pas utiliser
la méthode GET

```
<form action="monscript.php"
      method="post" id="form1">
  <p>champ texte (textfield)
    <input type="password"
          name="textfield" />
  </p>
</form>
```



Balise `<input type="submit" />`

Bouton de soumission du formulaire

✓ **name=** : Nom du bouton

- Identification du bouton (s'il y en a plusieurs par ex)
- Identification du formulaire auprès du script (le même script peut traiter plusieurs formulaires)

✓ **value=** : Texte affiché sur le bouton

```
<form action="monscript.php"
      method="post"
      id="calcullette">
  <p> Somme en francs:
  <input name="francs" type="text" /></p>
  <p> Valeur en euros:
  <input name="euros" type="text" /></p>
  <p>
    <input type="submit"
          name="valider"
          value="Calculer" />
  </p>
</form>
```



Validation d'un formulaire

Rappel

Quand on valide un formulaire : on appelle la page spécifiée par l'attribut `action` de la balise `<form>`. Les paramètres sont les objets (zone de texte, bouton, ...) du formulaire, qui ont pour valeur les valeurs des objets du formulaire.

Nom des paramètres = attribut `name` des objets

Valeur des paramètres = attribut `value` des objets

Valeurs possibles pour l'attribut `method` de la balise `<form>`

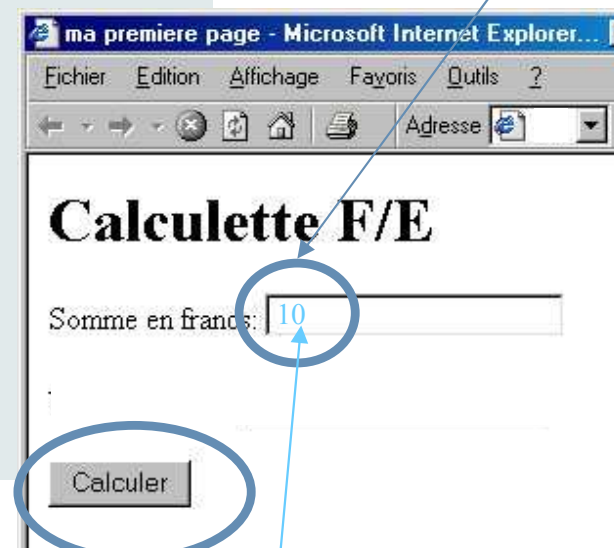
`get` : Les données sont accolées à l'URL du script CGI

`post` : Les données sont envoyées à part (plus sûr)

Exemple

```
<form action="http://www.estia.fr/monscript.php"
      method="get"
      id="calcullette">
  <p> Somme en francs:
  <input name="francs" type="text" /></p>
  <p>
  <input type="submit"
        name="valider"
        value="Calculer" /></p>
</form>
```

saisi par l'utilisateur



clic

<http://www.estia.fr/monscript.php?francs=10&valider=Calculer>



1.5244901723741037

envoi de la nouvelle page
(cheminement expliqué + loin)

Balise `<input type="reset" />`

Bouton de réinitialisation du formulaire

- ✓ `name=` : Nom du bouton
- ✓ `value=` : Texte affiché sur le bouton



état initial



saisie



clic sur Rétablir
(retour à l'état initial)

```
<input type="reset" name="Annuler" value="Rétablir" />
```


Bilan

- ✓ XHTML est le successeur de HTML
- ✓ C'est du "HTML propre" et plus encore
- ✓ XHTML et HTML permettent de créer des pages
- ✓ Pages interprétées par un navigateur
- ✓ Valider le code pour respecter les normes

<http://validator.w3.org/>

Bilan

- ✓ Langage de balise définissant des éléments de présentation des informations. 2 parties :
 - Entête : informations non visibles dans le navigateur
 - Corps : informations visibles directement dans le navigateur
- ✓ Permet de créer des formulaires pour rendre les pages interactives :
 - Comportement prédéfini des composants
 - Pas de traitement des données par HTML mais gère l'envoi

Suite du cours

Traiter ces données pour effectuer des traitements

Plan du cours

✓ L'Internet et le WWW



✓ Notions de XHTML et HTML



✓ Introduction au
langage php



php

Qu'est-ce que le php ?

- ✓ PHP : **P**ersonal **H**ome **P**ages / **P**eople **H**ate **P**erl / **H**ypertext **P**re**P**rocessor
- ✓ PHP est un langage de programmation interprété côté serveur
- ✓ Inventé par Rasmus LEDOORF (v1 : 1994, v2 : 1996, ... v5 : 2004)
- ✓ Il est dédié au Web : traitement des formulaires, communication avec des bases de données (souvent couplé à MySql)

Avantages

- Il est simple (pas typé, interprété, ...)
- Il est gratuit (?)
- Il est fait pour le Web
- Il est TRES répandu

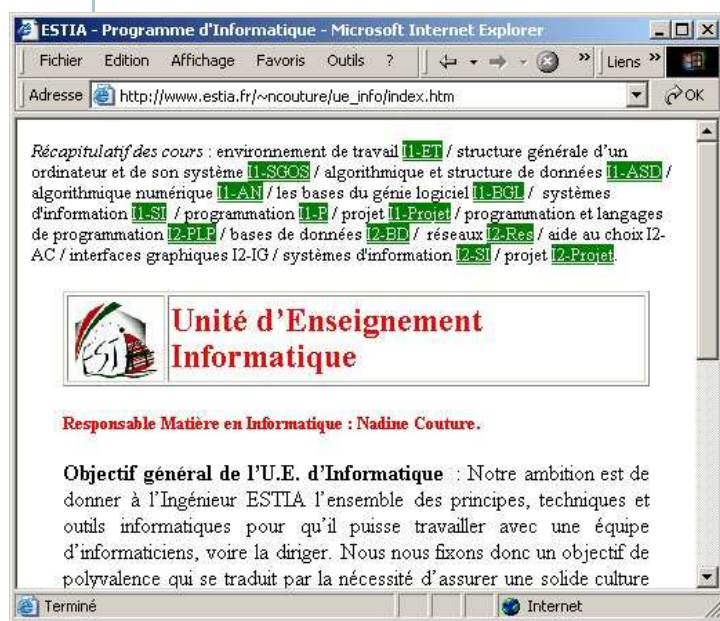


Inconvénient

- Code difficilement maintenable

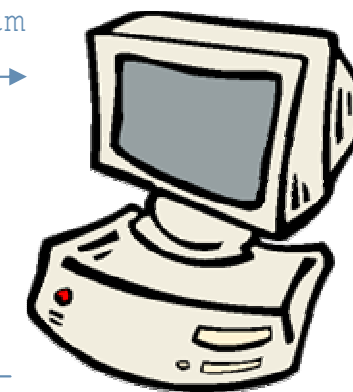
Comment ça marche ?

Rappel : le Web repose sur le modèle Client/Serveur



URL

`http://www.estia.fr/~ncouture/ue_info/index.htm`



Serveur http

Flot HTML correspondant à
"index.htm"

Client http

Comment ça marche ?

- ✓ Interprété côté serveur
- ✓ Code php dans la page HTML (balises spéciales)
- ✓ php produit du code HTML

Client http

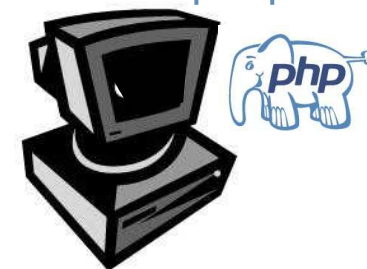


URL

`http://www.estia.fr/index.php`

Flot HTML correspondant à
"index.php" traité

Serveur php



index.php
traité

original
index.php



Serveur http

Remarques

- ✓ Extension du fichier doit être .php
- ✓ Le navigateur ne comprend que l'HTML
- ✓ Le code "php" inclus dans la page HTML est transformée par le serveur php
- ✓ Le "serveur" php et le "serveur" http peuvent être sur le même ordinateur
- ✓ Le code "php" est invisible dans la page HTML affichée

Un exemple pratique

Pour faire fonctionner un script php il faut :

- un serveur http
- un serveur php
- un client http (un navigateur)



Un exemple pratique : Hello World

```
<html>
  <head>
    <title>Ma premiere page</title>
  </head>
  <body>
    <h1>Hello World</h1>
  </body>
</html>
```

helloworld.htm



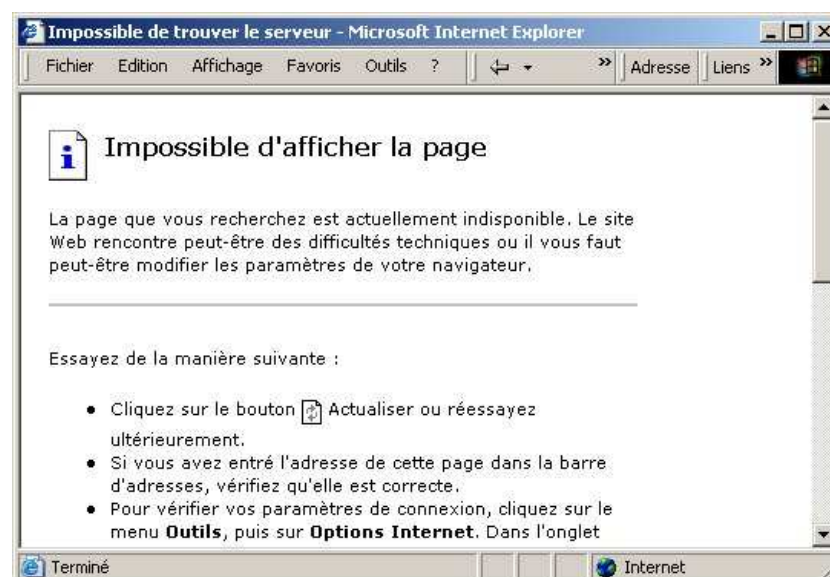
```
<html>
  <head>
    <title>Ma premiere page</title>
  </head>
  <body>
    <?
      echo "<h1>Hello World</h1>";
    ?>
  </body>
</html>
```

helloworld.php

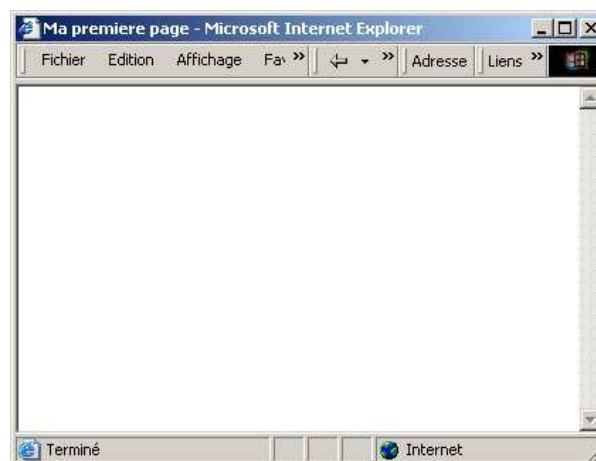


Un exemple pratique : remarques

- ✓ Si on "éteint" le serveur web



- ✓ Si on met une extension .html au fichier php



Éléments de syntaxe

La présentation des éléments de syntaxe qui suit est inspirée de celle du site :

www.developpez.com

Documentation complète en français :

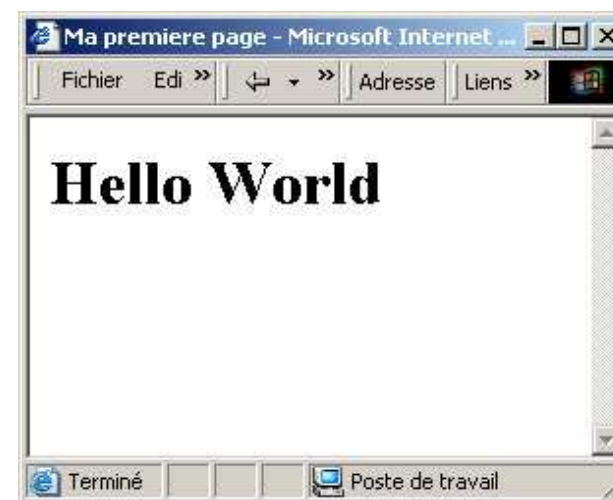
www.nexen.net

Balise

Les scripts php étant destinés à être utilisés à l'intérieur de pages HTML, il faut une balise spéciale permettant au "parser / interpréteur" php de savoir à quel endroit se trouve le code.

Un code php est toujours compris entre les balise "<?" d'une part et "?>" d'autre part.

```
[...]
<body>
    <?php
        echo "<h1>Hello World</h1>";
    ?>
</body>
[...]
```



Commentaires

✓ Comme en C ou en Java

```
<?php  
  
// commentaire de fin de ligne  
  
/* commentaire  
sur plusieurs  
lignes */  
  
?>
```

✓ Tout ce qui se trouve dans un commentaire est ignoré par le serveur php

Variables

- ✓ php est interprété et faiblement typé.
- ✓ Il n'est pas nécessaire de déclarer le type d'une variable.
- ✓ Les identificateurs de variable sont précédés du symbole « \$ » (dollar).

```
<body>
  <?php
    $toto = 5;
    echo $toto;
  ?>
</body>
```

- ✓ Types :
 - entier (**integer**)
 - réel (**double**)
 - tableau (**array**)
 - objet (**object**)
 - booléen (**boolean**)
 - chaîne de caractères (**string**)

Variables

✓ Portée

Limitée au bloc dans lequel elle a été créée.

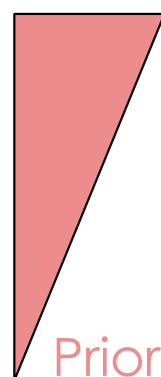
Une variable créée dans un bloc n'est pas connue dans les autres.

Les variables de fonctions ne sont connues que dans la fonction.

✓ Opérateurs

Par ordre de priorité

- ++, --, !
- +, -, *, /, %
- ==, <, >, <=, >=, !=
- &&, ||, and, or
- =



Priorité plus forte

Priorité plus faible

Constantes

- ✓ Les constantes ont une valeur fixe qui ne peut pas être modifiée.
- ✓ Les identificateurs de constante ne portent pas le symbole.

define("var",valeur)

*définit la constante **var** de valeur **valeur***

```
<body>
  <?php
    define("PI", 3.14);
    echo PI;
  ?>
</body>
```



Chaînes de caractères

Une chaîne de caractères est limitée par des cotes simples ' ou doubles ''.

✓ Les variables et les caractères spéciaux ne sont évalués que dans les doubles cotes.

```
<body>
  <?php
    $ma_chaine = "Yoko";
    $ta_chaine = "John";
    echo "la mienne : $ma_chaine <br/>";
    echo 'la tienne : $ta_chaine';
  ?>
</body>
```



✓ Caractères spéciaux : \n (nouvelle ligne), \t (tabulation horizontale), \\ (antislash), ...

Chaînes de caractères

Opérateur de concaténation de chaînes :
 . (point)

```
<body>
  <?php
    $ma_chaine = "Yoko";
    echo 'nos deux chaines :'.$ma_chaine."<br/>";
  ?>
</body>
```



Chaînes de caractères

Fonctions

- **echo(\$str)** : affiche la chaîne **\$str** dans le navigateur
- **strlen(\$str)** : retourne le nombre de caractères d'une chaîne
- **strtolower(\$str)** : retourne la chaîne **\$str** en minuscules
- **strtoupper(\$str)** : retourne la chaîne **\$str** en majuscules
- **trim(\$str)** : supprime les éventuels espaces en début et fin de **\$str**
- **substr(\$str,\$long,\$debut)** : retourne une sous chaîne de **\$str** de taille **\$long** et débutant à la position **\$debut**
- **strcmp(\$str1,\$str2)** : retourne 0 si **\$str1** et **\$str2** contiennent la même chaîne, -1 si **\$str1** est plus petit que **\$str2** et 1 sinon

Fonctions

La syntaxe de déclaration d'une fonction est de la forme :

```
<?php
    function nomFonction($param1, $param2, ..., $param3) {
        /* Bloc de traitement de la fonction */
        return valeur; // Cette instruction est optionnelle
    }
?>
```

Remarques

- On ne spécifie pas le types des paramètres ;
- Les paramètres sont séparés par des virgules ;
- S'il n'y a pas de paramètre, il faut tout de même placer les () ;

```
<?php
    // Déclaration
    function double($un_nombre) {
        $var = $un_nombre *2;
        return $var;
    }
?>
```

```
<?php
    // Appel
    echo double(500);
?>
```

Fonctions

- ✓ Les paramètres d'une fonction sont passés par copie.
- ✓ Il est possible de modifier ce comportement afin d'assurer une transmission par référence. Dans ce cas, les modifications apportées au paramètre dans la fonction affecteront la variable utilisée lors de l'appel. Pour ça on utilise le caractère &.

```
<?php
    function change(&$var) {
        $var += 100;    // incrémentatation de +100
    }
    ...
    $toto = 12;        // $toto vaut 12
    change($toto);
    echo $toto;        // $toto vaut 112
    ...
?>
```

Condition

Permet d'exécuter ou non un bloc d'instructions.

```
if (condition1) {
    /* bloc d'instructions lorsque
       condition1 est vraie */
} elseif (condition2) {
    /* autre bloc d'instructions, lorsque
       condition2 est vraie */
} ... {
} else {
    /* bloc d'instructions lorsque les
       conditions antécédentes sont fausses */
}
```

Rmq : la condition est une expression booléenne (l'évaluation de son résultat rend vrai ou faux)

L'opérateur switch

- ✓ Variante du **if ... else ...**
- ✓ Permet d'éviter une imbrication de **else if ...**
- ✓ Fonctionne sur les types « discrets » (en fait : byte, int, short, char, boolean)

```

switch (expression) {
    case valeur1 :
        /* bloc instructions */
        break;
    case valeur2 :
        /* bloc instructions */
        break;
    ...
    default :
        /* bloc instructions */
        break;
}

```

Boucle for

Permet d'exécuter plusieurs fois un bloc d'instructions donné (nombre d'itérations connu)

```
for (initialisation ; condition_pour_continuer ; modification) {  
    /* bloc d'instructions lorsque la  
       condition d'arrêt n'est pas vérifiée */  
}
```

```
/* typiquement : */  
for ($i=0; $i<10; $i++){  
    /* traitement */  
}
```

Boucle while

Permet d'exécuter plusieurs fois un bloc d'instructions donné (nombre d'itérations inconnu)

```
while (condition_pour_continuer) {  
    /* bloc d'instructions lorsque la condition  
       est vérifiée */  
}
```

Rmq : la condition est une expression booléenne (l'évaluation de son résultat rend vrai ou faux)

- si vrai : le bloc d'instructions est exécuté
- si faux : la boucle n'est plus itérée, et l'exécution continue à la suite la boucle

Inclusions

On peut inclure dans un script php le contenu d'un autre fichier.

✓ **require**

insert dans le code le contenu du fichier spécifié même si ce n'est pas du code php.

✓ **include**

évalue et insert à chaque appel (même dans une boucle) le contenu du fichier passé en argument.

```
<?php
    function change(&$var) {
        $var += 100;
    }
?>
```

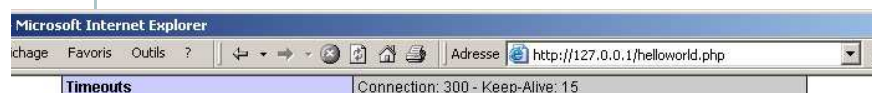
utile.php

```
<body>
<?php
    require ('utile.php');
    $var = 40;
    change($var);
    echo $var;
?>
<body>
```

mapage.php

Variables d'environnement

Pour connaître toutes les variables d'environnement :
`phpinfo()`



Directive	Local Value	Master Value
child_terminate	0	0
engine	1	1
last_modified	0	0
xbithack	0	0

Apache Environment

Variable	Value
COMSPEC	C:\WINNT\system32\cmd.exe
DOCUMENT_ROOT	p:/easyphp1-7/www
HTTP_ACCEPT	image/gif, image/x-bitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel, application/vnd.ms-powerpoint, application/msword, */*
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_ACCEPT_LANGUAGE	fr
HTTP_CONNECTION	Keep-Alive
HTTP_HOST	127.0.0.1
HTTP_USER_AGENT	Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
PATH	C:\WINNT\system32;C:\WINNT;C:\WINNT\System32\Wbem;C:\WINNT\system32\inls;C:\WINNT\system32\inls\ENGLISH;Y.;
REMOTE_ADDR	127.0.0.1
REMOTE_PORT	1859
SCRIPT_FILENAME	p:/easyphp1-7/www/helloworld.php
SERVER_ADDR	127.0.0.1
SERVER_ADMIN	admin@localhost
SERVER_NAME	127.0.0.1
SERVER_PORT	80
SERVER_SIGNATURE	<ADDRESS>Apache/1.3.27 Server at 127.0.0.1 Port 80</ADDRESS>
SERVER_SOFTWARE	Apache/1.3.27 (Win32) PHP/4.3.3

Quelques variables :

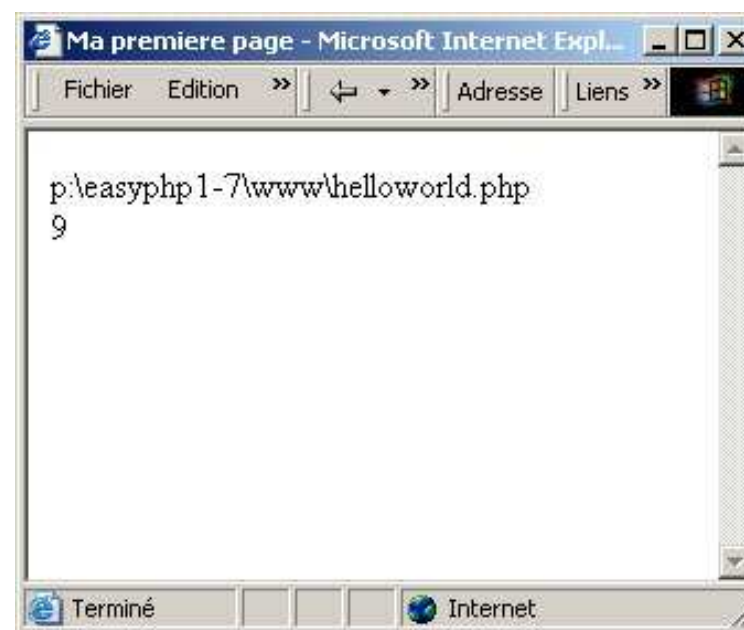
- **\$PHP_SELF** : nom du script en cours
- **\$HTTP_USER_AGENT** : signature du navigateur du client
- **\$REMOTE_ADDR** : adresse IP du client
- **\$QUERY_STRING** : chaîne au format URL contenant les paramètres passés à la page en cours
- **\$HTTP_REFERER** : URL de la source ayant renvoyée le client sur la page en cours (permet de connaître le moteur de recherche utilisé, les mots clés saisis, ...)

Constantes de PHP

Fournissent des informations sur le script en cours (par exemple) :

- **__FILE__** : nom du fichier en cours
- **__LINE__** : numéro de ligne en cours
- **PHP_VERSION** : version de PHP
- **PHP_OS** : nom du système d'exploitation qui est utilisé par la machine qui fait tourner le PHP
- **TRUE** : la valeur vraie booléenne
- **FALSE** : la valeur faux booléenne

```
<body>
  <?php
    echo __file__;
    echo "<br/>";
    echo __line__;
  ?>
</body>
```



Fichiers

Pour manipuler un fichier, on lui associe un identifiant dans le script php. Une fois un fichier ouvert, on peut effectuer des opérations qui déplacent un "curseur" imaginaire, représentant la position courante à l'intérieur du fichier.

Quelques fonctions:

- **fopen(\$file [, \$mode])** : ouverture du fichier identifié par son nom \$file et dans un mode \$mode particulier, retourne un identificateur \$fp de fichier ou FALSE si échec
- **fclose(\$fp)** : ferme le fichier identifié par le \$fp
- **fgets(\$fp, \$length)** : lit une ligne de \$length caractères au maximum
- **fputs(\$fp, \$str)** : écrit la chaîne \$str dans le fichier identifié par \$fp
- **fgetc(\$fp)** : lit un caractère
- **feof(\$fp)** : teste la fin du fichier
- **file_exists(\$file)** : indique si le fichier \$file existe
- **filesize(\$file)** : retourne la taille du fichier \$file
- **unlink(\$file)** : détruit le fichier \$file
- **copy(\$source, \$dest)** : copie le fichier \$source vers \$dest
- **rename(\$old, \$new)** : renomme le fichier \$old en \$new

Fichiers

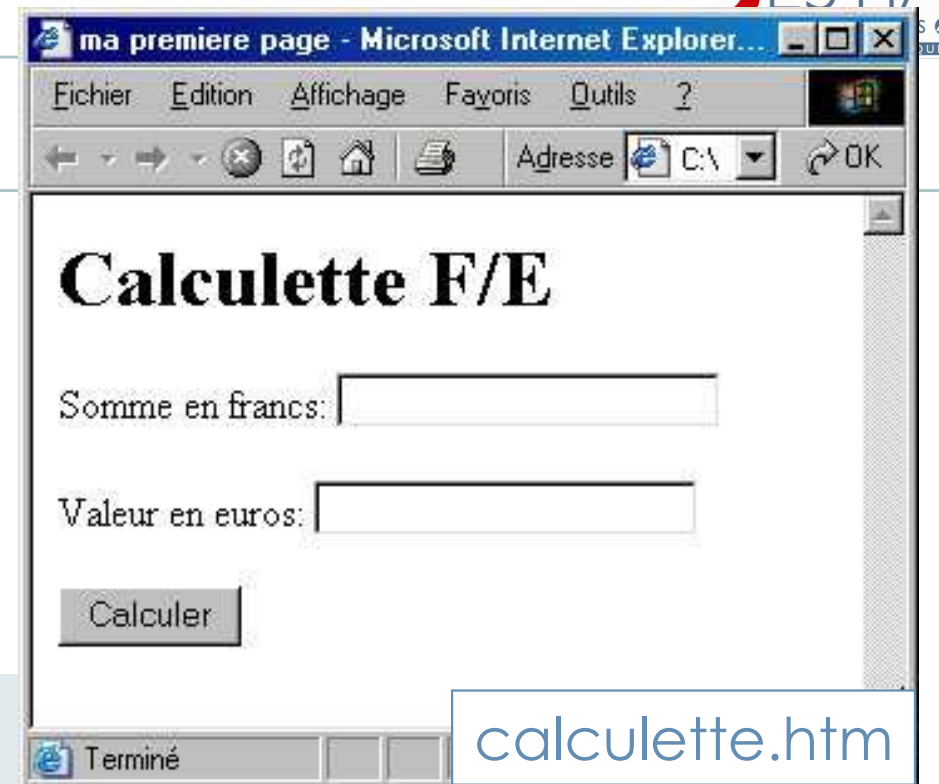
Exemple : afficher le contenu d'un fichier

```
<?php
$file = "fichier.txt" ;
if( $fd = fopen($file, "r")) { // ouverture du fichier
                                // en lecture
    while (!feof($fd)) { // teste la fin de fichier
        $str = $str . fgets($fd, 1024);
        /* lecture des (au plus) 1024 premiers
           caractères d'une ligne */
    }
    fclose ($fd);
    echo $str;
} else {
    die("Ouverture du fichier $file impossible.");
}
?>
```

PHP et formulaires

appel d'un script php

```
<form action="monscript.php"
        method="post" id="calcullette">
  <p> Somme en francs:
    <input name="francs" type="text" />
  </p>
  <p> Valeur en euros:
    <input name="euros" type="text" />
  <p>
    <input type="submit" name="valider" value="Calculer" />
  </p>
</form>
```

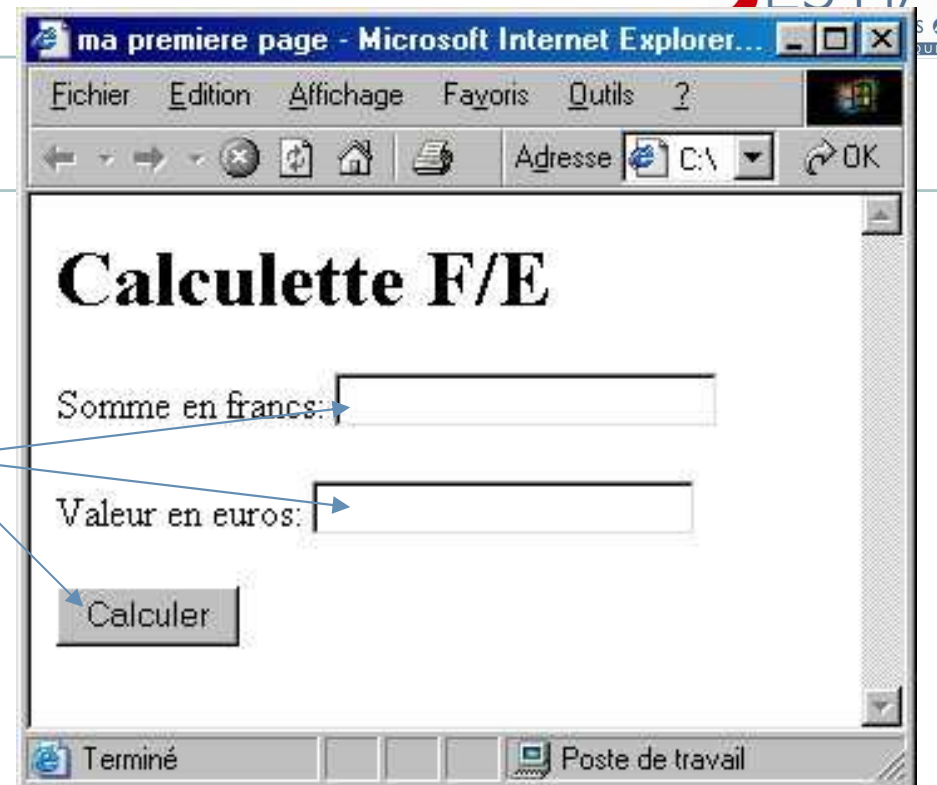


calcullette.htm

Php et formulaires

chaque élément du formulaire définit une variable pour le script appelé

```
<form action="monscript.php" ...
```



`$_POST['francs']`

`$_POST['euros']`

`$_POST['valider']`

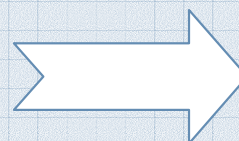
```
<input name="francs" type="text" />
```

```
<input name="euros" type="text" />
```

```
<input type="submit" name="valider" value="Calculer" />
```

Somme en francs: 10

calculatrice.htm



`$_POST['francs']=10`
monscript.php

Php et formulaires

Pour récupérer la valeur d'une entrée dans le formulaire à l'intérieur on utilise :

- `$_POST['Var']` : si le formulaire est en `post`
- `$_GET['Var']` : si le formulaire est en `get`

où **Var** représente la valeur de l'attribut "name" de la balise du formulaire

Un exemple complet

calculette.htm

```
<html>

<head>
  <title>Calculette FE</title>
</head>

<body>
  <h1>Calculette FE</h1>
  <form id="form1" method="post" action="convertir.php">
    <p>Somme en Francs :
      <input name="franc" type="text" id="franc" />
    </p>
    <p><input type="submit" name="convertirFenE" value="Convertir" /></p>
  </form>
</body>

</html>
```

Le formulaire d'entrée
calculette.htm



Un exemple complet

La page résultat, avec son script php

convertir.php

convertir.php

```
<html>

<head>
  <title>Convertir</title>
</head>

<body>
  <h1>R&eacute;sultat</h1>
  <p>
    <?php
      if (isset($_POST['franc'])) {
        echo "Valeur en euros = " . $_POST['franc']/6.55957;
      } else {
        echo "Erreur : la valeur en francs n'est pas définie";
      }
    ?>
  </p>
  <p><a href="calculatrice.htm">Convertir une nouvelle somme.</a></p>
</body>

</html>
```

Un exemple complet



Quelques références

PHP

- ✓ <http://www.php.net>
(Documentation > View Online > French)
- ✓ Programmation en PHP
Leon Atkinson, Editions Campus Press

XHTML

- ✓ <http://www.tuteurs.ens.fr/internet/web/html/>
- ✓ <http://validator.w3.org/>