

HTML, CSS formulaire, frame PHP

P. Barger, J.-P. Boufflet

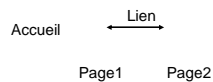
Contenu

- Hypertexte
- HTML origine
- Architecture sites web
- HTML
- CSS
- formulaire
- frame
- PHP

2

Hypertexte

- Mélange de corps du texte et de liens
- Séparation en unités (pages, chapitres, etc.)
- Liens entre ces unités
- Notion d'accueil : première unité



3

HTML HyperText Markup Language

- HyperText Markup Language (HTML)
 - Adapté à l'hypertexte
- Document HTML contient :
 - du texte
 - des **balises** (ou *tags*) : directives de mises en forme
 Exemple : `<i>Ce texte est en italique</i>`
- HTML : version allégée de SGML
- SGML (Standard Generalized Markup Language) créé en 1986
 - format de document des professionnels de l'édition et de la gestion documentaire
 - norme ISO 8879
 - langage très complet
 - suite bureautique SGML (éditeur + navigateur) très lourde à gérer

4

SGML

SGML (Standard Generalized Markup Language)

- "généralisé" signifie que ses balises décrivent le contenu d'un document et non sa présentation / définit une syntaxe sans lui associer de sémantique.
- "standard" pour décrire d'autres langages de balisage
- historique
 - années 60 : Charles Goldfarb mène pour IBM des recherches sur un système d'information intégré pour un cabinet d'avocats
 - crée une méthode "GML" permettant aux sous-systèmes d'information de formater, d'éditer et de partager des documents
 - 1986 : SGML adopté comme standard
- principes de fonctionnement
 - 1 document équivaut à 3 types d'information :
 - données (texte, graphique, son ...)
 - structure (sections, sous-sections ...)
 - format (apparence, mise en page)
 - SGML reconnaît données, structure et format comme trois types d'info. différents et dissociables
 - SGML préserve données et structure mais ne spécifie pas le format qui peut être défini par le destinataire du doc. lors de sa réception

5

SGML

- avantages
 - protection du capital informationnel par rapport aux risques d'utilisation restreinte à un type de configuration
 - ré-utilisabilité des données
 - interopérabilité des opérations (ne contraint pas l'architecture et les fonctions des applications qui peuvent donc être très diverses)
- conclusion
 - information plus accessible
 - réduction des coûts et erreurs relatives aux procédés de conversion des données
 - garantie que le document respecte scrupuleusement les spécifications de l'auteur
- balisage SGML en 3 étapes
 - on définit des classes de document (ex : classe livre)
 - pour chaque classe, on définit un modèle générique de structure (DTD)
 - on balise le document selon les règles définies dans sa DTD (Document Type Definition)

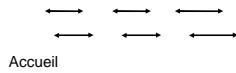
6

Architecture sites web

■ Lecture linéaire

□ Ex. : livre

- Page1
- Page2
- Page3
- ...



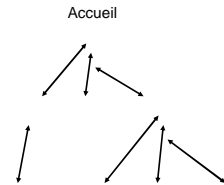
7

Architecture sites web

■ Lecture hiérarchisée

□ Ex. : guide

- Cours Tronc Commun
- Cours Branche Info
- Cours Branche Bio



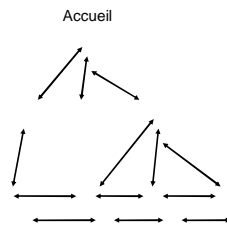
8

Architecture sites web

■ Combinaison linéaire & hiérarchisée

□ Ex. : livre de cours

- Chapitre1
- Chapitre2
- Chapitre3



9

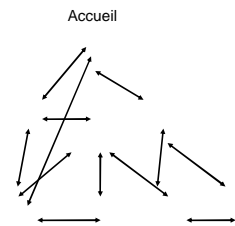
Architecture sites web

■ Toile d'araignée

□ Ex. : catalogue

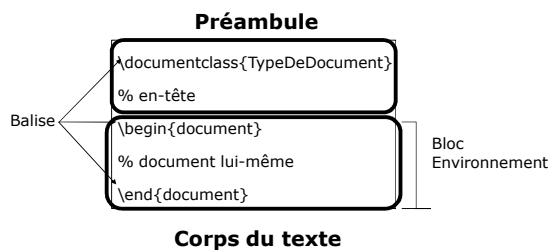
- Matériaux1
- Finition2
- Couleur3

□ Web



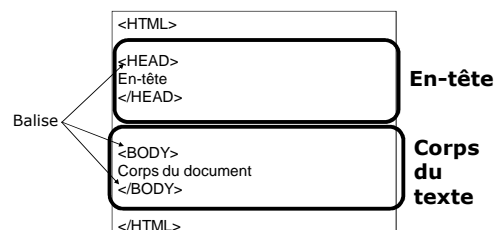
10

Document de base : LaTeX



11

Document de base : HTML



12

HTML Balisage

- 2 types de balises
 - autonome < ... > (ex. : <!DOCTYPE>)
 - délimitant une zone
 - balise de début de zone < ... > (ex. : <HTML>)
 - balise de fin de zone </ ... > (ex. : </HTML>)
- identifiants de balise non sensibles à la casse
 - <BODY> ≡ <body>
- certaines balises sont associées à un ou plusieurs attributs, chaque attribut à un identifiant et une valeur (entre "" et après =)
 -

13

HTML Balisage

- commentaires <!-- ceci est un commentaire -->
- encodage de caractère & ... ;
 - < > & donnent respectivement < > &
 - caractères accentués & **lettre accent** ;
 - ´ ` ô donnent respectivement é à ô

14

Entête HTML

- Informations sur le document
- <TITLE>un titre</TITLE>
 - **titre** du document / apparaît dans la barre supérieur du navigateur
- <META NAME="propriété" CONTENT="valeur">
 - **propriété** du document / non affichée / plusieurs propriétés

```
<HEAD> peuvent être déf.
<TITLE>Bienvenue sur la page web de Blanche
Neige</TITLE>
<META NAME="auteur" CONTENT="Le prince charmant">
<META NAME="contenu" CONTENT="Recettes de cuisine de 7
nains">
<META NAME="mis à jour" CONTENT="tous les 7 ans">
</HEAD>
```

15

Corps HTML

Balises principales

- <BODY attr1="val1" ... attrn="valn"> ... </BODY>
 - attributs possibles **valeurs par défaut** en cas d'absence
 - BGCOLOR : couleur de fond
 - TEXT : couleur du texte
 - BACKGROUND : URL de l'image de fond d'écran
 - LINK : couleur des liens non encore visités
 - VLINK : couleur des liens visités

16

Corps HTML

Balises principales

- <H1>Titre</H1>, <H2>, <H3>, <H4>, <H5>, <H6>
 - attribut possible ALIGN="left|center|right"
- <P> ... </P> paragraphe
-
 passage à la ligne
- <HR> trait horizontal
- ... gras
- <I> ... </I> italique

17

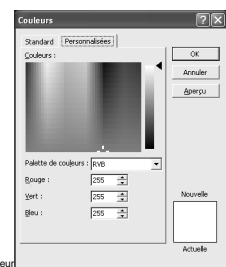
Couleurs

- 2 solutions
 - identificateurs prédéfinis :
 - green, yellow, purple, blue, red, ...
 - nombre de 6 chiffres hexadécimaux
 - ex. : #1A7FC0
 - codant les intensités de rouge/vert/bleu
 - intensité
- Equivalence entre les 2
 - blue ≡ #0000FF
 - purple ≡ #800080

La plupart des éditeurs HTML fournissent des outils de type "palette de couleur"

Exemple :

```
<center><h1><font color="#FF0000"><font size=+4>NF92</font></font></h1></center>
```



18

Liens Hypertexte

- Un lien permet de définir :
 - une région sensible au clic souris dans un document
 - l'endroit où l'on va se retrouver après ce clic.
- Trois pas :
 1.
 2. lien
 3.

Diagram illustrating a link between Page1 and Page2. A box labeled "lien" has an arrow pointing to "Page2". A warning icon with the text "Chemin d'accès" is also shown.

19

Liens Hypertexte

Chemin d'accès par défaut :
~nf92/public_html/index.html
Page d'accueil de votre compte

```
<html>
<head>
<title>Page de liens</title>
</head>
<body>
<a href="http://www4.utc.fr/~nf92/P10">
La page de NF92
</a>
</body>
</html>
```

20

Liens Hypertexte

- Portion de texte permettant d'atteindre un document désigné par une **URL**

texte du lien

- Lien absolu LIP6
- Lien relatif
 - sommaire
 - plan d'accès
 - accueil
- Lien exécutable (fichiers en téléchargement, déclenchement d'animations ou de programmes -plug-in)

21

Lien mailto

```
<html>
<head>
<title>Page de liens</title>
</head>
<body>
<a href="http://tice.utc.fr/moodle">
La page de NF92
</a><br>
<a href="mailto:bargerpa@hds.utc.fr">
Envoyer un mail au responsable de l'UV
</a>
</body>
</html>
```

22

URL

Uniform Ressource Locator

Diagram illustrating the components of the URL `http://tuxa.sme.utc/~nf92/P08/index.html`:

- Protocole: `http`
- Nom de l'hôte: `tuxa.sme.utc`
- Répertoires et nom de fichier: `/~nf92/P08/index.html`

23

URL

Éléments constitutifs (optionnels) d'une URL

- protocole:// protocoles utilisables : HTTP, FTP, MAILTO, FILE, NEWS
- utilisateur@ login pour les pages protégées
- serveur nom ou adresse IP (ex. : 163.136.27.6) du serveur
- :numéro numéro de port TCP du serveur (transmission control protocol)
- /répertoire chemin d'accès dans la hiérarchie de répertoires
- fichier nom du document à atteindre
- #signet signet dans le document à atteindre
- ?options information transmise au serveur (ex. : formulaire)

Exemple

- `http://bob@12.7.6.1:8080/pub/cv.html#diplomes`
- `ftp://anonymous@ftp.lip6.fr`
- `http://www.altavista.com/query.html?jussieu`

24

Listes

Listes numérotées (...) ou non (...)

 balise de début de liste
 Rouge balises pour un élément de la liste
 Vert balises pour un élément de la liste
 ...
 balise de fin de liste

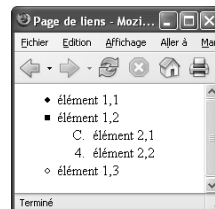
Attributs possibles

<UL TYPE="disc|circle|square"> : type de puce
 <OL TYPE="I|i|A|a"> : type de numérotation
 START="99"> : départ de la numérotation
 <LI TYPE="disc|circle|square"> : type de puce (liste UL)
 <LI TYPE="I|i|A|a"> : type de numérotation (liste OL)

25

Listes exemple

```
<UL>
<LI> élément 1,1 </LI>
<LI type="square"> élément 1,2 </LI>
<OL type="A" start="3">
<LI> élément 2,1 </LI>
<LI type="1"> élément 2,2 </LI>
</OL>
<LI type="circle"> élément 1,3 </LI>
</UL>
```



Images

Insertion d'une image (gif ou jpeg) dans un document

Attributs possibles

- HEIGHT="99" / WIDTH : hauteur/largeur en pixels de l'image
 - ALT=" ..." : légende associée à l'image
 - HSPACE="99" / VSPACE : espacement en pixels horiz./vert. autour de l'image
 - ALIGN="bottom|middle|top|left|right" : alignement de l'image / au texte

27

Image comme lien

```
<a href="ma_page.html">

</a>
```



Tableaux

Tableaux à 2 dimensions (<TABLE> ... </TABLE>)

<TABLE> balise de début de tableau
 <TR> balise de début de ligne (row)
 <TD> cellule </TD> balises pour une colonne (data)
 ...
 </TR> balise de fin de ligne
 ...
 </TABLE> balise de fin de tableau

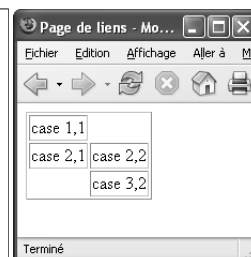
Attributs possibles pour <TABLE>

- BORDER="99" (0) : épaisseur en pixels des bordures du tableau
 - ALIGN="left|right" : alignement du tableau / au texte
 - WIDTH="99|99%" : largeur du tableau en pixels / en % largeur fenêtre
 - CELSPACING="99" (2) : espacement en pixels entre les cellules
 - CELLPADDING="99" (1) : marge intérieure en pixels des cellules

29

Tableaux exemple

```
<TABLE BORDER="1">
<TR> <TD> case 1,1 </TD> </TR>
<TR> <TD> case 2,1 </TD>
<TD> case 2,2 </TD> </TR>
<TR> <TD></TD>
<TD> case 3,2 </TD> </TR>
</TABLE>
```



30

Tableaux

Attributs possibles pour <TR>

- ALIGN="left|right|center|justify" : alignement horiz. du texte (≠ <TABLE>)
- VALIGN="middle|top|bottom" : alignement vertical du texte

Attributs possibles pour <TD>

- ALIGN/VALIGN : idem <TR>, supprime <TR ALIGN/VALIGN >
- WIDTH="99|99%" : largeur de la colonne en pixels | en % largeur tableau
- COLSPAN="99" : agrège plusieurs cellules horizontalement
- ROWSPAN="99" : agrège plusieurs cellules verticalement

31

Récapitulatif HTML à connaître

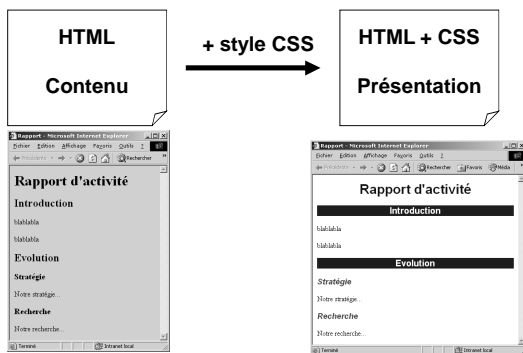
```
<html>...</html>
<head>...</head>
<body>...</body>
<title>...</title>
<h1>...</h1>
<h2>...</h2>
<p>...</p>
<ol>...</ol>
<ul>...</ul>
<i>
<!-- ... -->
<a href="...">...</a>

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

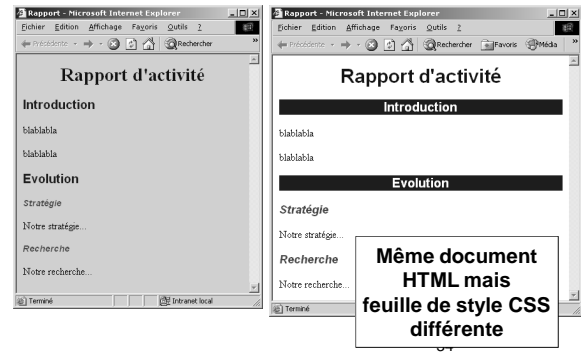
Pour visualiser le code d'une page :
"affichage" puis "Code source de la page"
ou encore <CTRL> U

32

CSS (Cascading Style Sheets)



CSS (Cascading Style Sheets)



CSS (Cascading Style Sheets)

Permettent de factoriser des éléments de présentation pour 1 ou plusieurs pages

Définition des styles dans un fichier ".css"

```
balise1 { prop1:val1; ... ; propn:valn; }
```

```
balise2 { prop1:val1; ... ; propn:valn; }
```

...

exemple : H1 { font-style: italic; color: blue; }

Liaison entre le fichier .html et le fichier .css par la balise <LINK>

35

CSS (Cascading Style Sheets)

ex : fichier f.css

```
H1 { font-style: italic; color: blue; }
```

```
<HTML>
<HEAD>
<LINK REL="stylesheet"
HREF="f.css" type="text/css">
</HEAD>
<BODY> <H1>Titre en ... </H1>
</BODY>
</HTML>
```



36

Classes CSS

Classes de style prédéfinies

A:link : style des **liens non encore visités** (par défaut bleu)
 A:active : style des **liens sélectionnés** (par défaut rouge)
 A:visited : style des **liens visités** (par défaut rouge)
 P:first-line : style de la **1ère ligne d'un paragraphe**
 P:first-letter : style de la **1ère lettre d'un paragraphe**

ex : A:link { color: green; }

37

CSS méthode à appliquer

Créer un fichier texte contenant les feuilles de style et dont l'extension est **.css**, par exemple **style.css** :

```
body {background-image: home.gif;}
LI {font: 13px Verdana;}
B {font: 14px Verdana; font-weight: bold;}
A { font:12px Verdana; font-weight: bold;
color=black; text-decoration: none; }
H1 {font: 16px Arial;font-weight: bold;color=black;}
H2 {font: 14px Arial;font weight:bold;color=black;}
```

CSS méthode à appliquer

Ensuite créer dans chaque page HTML qui utilise ces styles le raccourci vers cette page de définition de style :

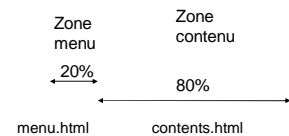
```
<HTML> <HEAD> <LINK rel="stylesheet" type="text/css" href="style.css"> </HEAD> ...
```

- La balise **<LINK>** avertit le navigateur qu'il doit chercher un document situé à l'extérieur de la page HTML.
- L'attribut **rel="stylesheet"** précise que le document en question est une feuille de style externe.
- L'attribut **type="text/css"** précise le type de feuille de style.
- L'attribut **href="URL"** donne l'URL de la feuille de style, c'est-à-dire son emplacement sur Internet.

Frame : multipages

Les framesets permettent de diviser une page en plusieurs parties, pour ensuite insérer dans chaque partie une page à part entière.

```
<FRAMESET COLS="20%,80%">
<FRAME SRC="menu.html" NAME="table">
<FRAME SRC="contents.html" NAME="contenu">
</FRAMESET>
```



40

Frame : multipages

■ frames (cadres)

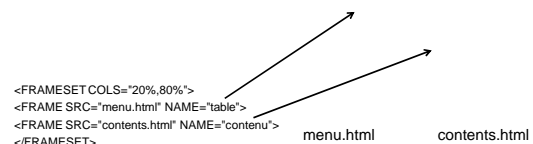
```
<FRAMESET COLS="20%,80%">
<FRAME SRC="menu.html" NAME="table">
<FRAME SRC="contents.html" NAME="contenu">
</FRAMESET>
```

NAME=nom du cadre
 définition du nom qui pourra être utilisé dans target
 SRC=url
 spécifie l'adresse du document qui sera contenu
 NORESIZE
 booléen interdisant l'utilisateur de modifier la taille si présent
 SRCOLLING=auto | yes | no
 scroll : si besoin ou systématique ou absent
 FRAMEBORDER=0 | 1
 1 : le cadre est séparé de ses voisins ; 0 : pas de séparation
 MARGINWIDTH=nb. pixels
 espace neutralisé à gauche et à droite du cadre
 MARGINHEIGHT=nb. pixels
 espace neutralisé en haut et en bas du cadre

41

Frame : multipages

Chaque frame a un nom par l'attribut « NAME », ce qui permet lorsque vous faites un lien d'afficher le résultat dans la frame ad hoc



Comportement : si la page "menu.html" contient des liens comme ` truc ` alors par défaut la page "ma_page.html" s'affichera dans la colonne de gauche de 20% en lieu et place de la page d'origine "menu.html". La page invoquée "hérîte" de l'endroit d'où elle est appelée (frame par défaut).

Frame : multipages

On résout ce problème en utilisant l'attribut "target" de l'élément
` truc `

de cette façon :

` truc `

Ce qui permet de diriger un document vers un cadre autre que celui qui contient le lien activé.

Formulaires

Permettent de récupérer de l'information saisie par l'utilisateur de la page

```
<FORM>
First name : <INPUT NAME="fname"><BR>
Last name : <INPUT NAME="lname"><BR>
<INPUT NAME="genre" TYPE="RADIO"
VALUE="Male">Male <BR>
<INPUT NAME="genre" TYPE="RADIO"
VALUE="Female">Female<BR>
<INPUT VALUE="Send" TYPE="SUBMIT">
<INPUT VALUE="Reset" TYPE="RESET"><BR>
</FORM>
```

44

Formulaires

- Texte
 - `<input type="text" name="firstname" />`
- Boutons Radio
 - `<input type="radio" name="sex" value="male" /> Male`
- Boutons à cocher
 - I have a bike:
`<input type="checkbox" name="vehicle" value="Bike" />`
- Bouton Soumettre
 - `<input type="submit" value="Submit" />`



Traitement de formulaires : sujet hors HTML, voir PHP, JavaScript, Java, CGI

45

Formulaires récapitulatif

Élément	Attributs	Description
<FORM>	METHOD, ACTION, ENCTYPE	spécifie l'adresse indiquée, spécifie le type de codage utilisé
<INPUT>	TYPE, NAME, VALUE	spécifie l'ACTION dans le balise
text		simple ligne de texte dont la longueur est donnée par l'attribut size
reset		efface le contenu du formulaire
radio		bouton radio
checkbox		case à cocher
NAME		Nom
SIZE		taille du texte
TEXTAREA	NAME, ROWS, COLS	Zone de texte
SELECT	NAME	Choix 1
OPTION	VALUE	Choix 2
MULTIPLE		Plusieurs choix possibles
SELECTED		option sélectionnée
VALUE		valeur fournie

46

Formulaires

■ Exemple

47

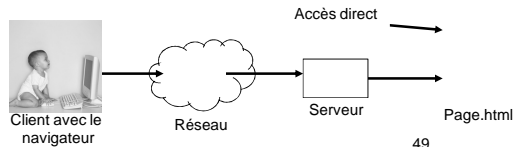
Formulaires

```
<FORM method="post" action="cgi-bin/script.pl">
Enregistrement d'un utilisateur
<TABLE BORDER="0">
<TR>
<TD>Nom</TD>
<TD><INPUT type="text" name="nom"></TD>
</TR>
<TR>
<TD>Prénom</TD>
<TD><INPUT type="text" name="prenom"></TD>
</TR>
<TR>
<TD>Sexe</TD>
<TD><INPUT type="radio" name="sex" value="M"> Homme :
<INPUT type="radio" name="sex" value="F"> Femme :
</TD>
</TR>
<TR>
<TD>Fonction</TD>
<TD><SELECT name="fonction">
<OPTION VALUE="enseignant">Enseignant</OPTION>
<OPTION VALUE="etudiant">Etudiant</OPTION>
<OPTION VALUE="ingenieur">Ingénieur</OPTION>
<OPTION VALUE="retraite">Retraité</OPTION>
<OPTION VALUE="autre">Autre</OPTION>
</TD>
</TR>
<TR>
<TD>Commentaires</TD>
<TD><TEXTAREA rows="3" name="commentaires">
Tapez ici vos commentaires</TEXTAREA>
</TD>
</TR>
<TR>
<TD colspan="2"><INPUT type="submit" value="Envoyer">
</TD>
</TR>
</TABLE>
</FORM>
```

48

Comment ça marche

- Accès aux pages Web via navigateurs
 - Navigateurs sont des logiciels qui interprètent le HTML, PHP, CSS, etc.
- 2 types d'accès
 - Accès direct au fichier (pour le visualiser, éditer)
 - Accès à travers un serveur web (typiquement Apache)
 - **Client** (vous) accède au **document HTML** sur le **serveur** grâce à un **protocole** (ex. http)



49

Client serveur

Mode de communication entre plusieurs ordinateurs d'un réseau qui distingue les clients du serveur. Un logiciel client envoie des requêtes au serveur. Il existe des serveurs d'applications, de fichiers, ou de messagerie électronique.

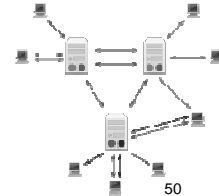
Caractéristiques du serveur :

- passif en attente d'une requête
- à l'écoute, prêt à répondre aux requêtes envoyées par les clients
- traite la requête d'un client dès qu'elle arrive et renvoie une réponse

Caractéristiques du client :

- actif
- envoie des requêtes
- attend et reçoit les réponses

ATTENTION clients et serveur utilisent le même protocole



50

Impact client/serveur comparatif succinct entre PHP et javascript deux langages « du web »

caractéristiques	PHP	javascript
exécution	serveur	client
interpréteur	sur serveur	dans navigateur
Manipulation fichiers	lecture/écriture texte et binaire sur serveur	incapable
Données issues formulaire (POST)	Récupération données formulaires	Accède aux champs du formulaire tant qu'il est dans la page affichée
Données passées par URL (GET)	Ok mais + facile	Ok
Manipulation de BDD	Oui	Non
Réagir aux événements clients	Non (car exécution sur serveur)	Oui

51

PHP pages web dynamiques

- **PHP: Hypertext Preprocessor**
- PHP langage de script sur le serveur, comme ASP (active server pages)
- PHP est exécuté sur le serveur, pas chez le client
- PHP accès bases de données (MySQL, Informix, Oracle, Sybase, Solid, PostgreSQL, Generic ODBC, etc.)
- PHP : open source
- PHP gratuit

52

PHP génère donc du HTML

fichier essai.php

```
<html>
<body>
<?php echo "Hello World"; ?>
</body>
</html>
```

génère

```
<html>
<body>
Hello World
</body>
</html>
```



Ce fichier s'appellera essai.php et pas essai.html car c'est un fichier exécutable sur le serveur



Il devra avoir des droits en exécution rwx (voir cours linux)

53

Liaison PHP HTML

- HTML peut envoyer des informations vers PHP


Fichier essai.html

```
<html>
<body>

<form action="welcome.php" method="post">
Name: <input type="text" name="fname" /> <br />
Age: <input type="text" name="age" /> <br />
<input type="submit" value="traiter" />
</form>

</body>
</html>
```

54



Liaison PHP HTML

Traitement de la page précédente :


Fichier welcome.php

```
<html>
<body>

Welcome <?php echo $_POST["fname"]; ?>!<br />
You are <?php echo $_POST["age"]; ?> years old.

</body>
</html>
```

55



Liaison PHP HTML

```
<html>
<body>
<?php
echo "Welcome <br />";
$variable1 = $_POST["fname"];
$variable2 = $_POST["age"];
if (empty($variable1) && empty($variable2) )
{
    print("<br/>les deux champs sont vides<br/>");
}
else
{
    $affiche = "Hello ".$variable1."<br/> You are ". $variable2." years old<br/>";
    print($affiche);
}
?>
</body>
</html>
```

Fichier welcome.php plus proche de la réalité

56