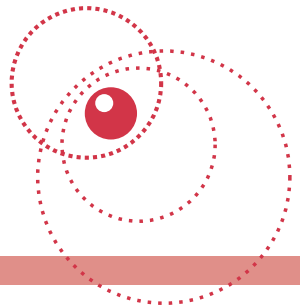

Ingénierie du Web (INF347)

Introduction aux technologies du Web, côté serveur

15 mai 2012

Licence de droits d'usage

Pierre Senellart

TELECOM ParisTech

Plan

Créer et héberger un site Web
- Planification
- Choisir une solution d'hébergement
- Administration
- Développement du site

HTTP

Formulaires HTML

Langages côté serveur

Un exemple : PHP

Licence de droits d'usage

Plan

Créer et héberger un site Web
- Planification
- Choisir une solution d'hébergement
- Administration
- Développement du site

HTTP

Formulaires HTML

Langages côté serveur

Un exemple : PHP

Licence de droits d'usage

Cahier des charges

- Objectifs du site
- Fonctionnalités :
  - Zones de contenu (nombre de pages, etc.)
  - Accès à des données d'une base existante
  - Fonctionnalités de recherche, de mise à jour, etc.
  - Interactions utilisateur : authentification, commentaires, profils, vente, etc.
  - Contenu multimédia
  - Besoins spéciaux d'interface (jeux, cartographie, etc.)
- Définir l'arborescence du site
- Calendrier de lancement
- Public visé : tous clients Web, navigateurs graphiques récents uniquement, nécessité d'un site spécial ou d'une application pour smartphone, etc.

Licence de droits d'usage

Maquette fonctionnelle

- Se baser sur le cahier des charges pour faire une maquette fonctionnelle (mock-up) de ce à quoi ressemblera le site
- Sans programmer :
  - Structure principale de chaque type de page
  - Contenu des formulaires
  - Informations présentées à chaque niveau
- Peut-être fait sur papier, avec des mock-ups HTML rapide, avec un logiciel de dessin vectoriel
- Inutile de se préoccuper des aspects graphiques à ce niveau, mais réflexion sur l'ergonomie

Licence de droits d'usage

Choix des technologies : en fonction de quoi ?

En se basant sur :

- le cahier des charges
- la maquette
- les compétences à disposition
- les contraintes d'hébergement
- les contraintes externes (p. ex., intégration avec un logiciel tiers)
- le public visé

Licence de droits d'usage

Choix des technologies

Choisir :

- Une plate-forme (Apache / IIS, Linux / Windows)
- Un format de stockage des données (fichiers plats, XML, base Oracle, base MySQL, etc.)
- Une ou plusieurs technologies côté serveur (CMS, framework d'applications, langage)
- Les technologies côté client (version d'HTML visée, framework JavaScript, technologies pour applications multimédia riches)
- HTTPS ?

Licence de droits d'usage

Dimensionnement

Calcul approximatif de la charge du serveur :

- Nombre de requêtes par seconde (en moyenne, en pleine charge)
- Volume de données transmis par seconde (en moyenne, en pleine charge)
- Calculs coûteux côté serveurs

Licence de droits d'usage

Exemple de dimensionnement

- Paramètres :
  - 1 000 visiteurs attendus par jour, avec des pointes à 10 000
  - Chaque visiteur visite une dizaine de pages en moyenne, et accède à une vidéo en moyenne dans 10% des cas
  - Chaque page fait appel (images, scripts, etc.) à une dizaine de fichiers et pèse de l'ordre de 500 kilo-octets
  - Une vidéo pèse de l'ordre de 200 méga-octets
  - Pas de traitements coûteux côté serveur
- Estimation :
  - $\frac{1\,000 \times 10 \times 10}{3\,600 \times 24} = 1{,}15$ accès par seconde en moyenne
  - $\frac{1\,000 \times (10 \times 500 + 0{,}1 \times 200\,000)}{3\,600 \times 24} = 289$ kilo-octets par seconde en moyenne (soit 2 mégabits/seconde)
  - 100 fois plus en charge (10 fois plus de visiteurs certains jours, non répartis uniformément sur la journée)

Licence de droits d'usage

Aspects légaux

- Dépôt de marque ?
- Constitution d'une société ?
- D'où vient le contenu du site ? Propriété intellectuelle
- Réfléchir aux conditions d'utilisation, de vente, de conservation des informations, etc.
- Déclaration à la CNIL en cas de base d'utilisateurs avec informations personnelles

Licence de droits d'usage

Plan

Créer et héberger un site Web
- Planification
- Choisir une solution d'hébergement
- Administration
- Développement du site

HTTP

Formulaires HTML

Langages côté serveur

Un exemple : PHP

Licence de droits d'usage

Auto-hébergement

Un serveur Web peut être hébergé directement sur un ordinateur de l'entreprise ou d'un particulier connecté à Internet, mais attention aux aspects suivants :

- Débit suffisant (en particulier en upload) ?
- Garanties de pouvoir garder le serveur Web allumé 24h/24 ? (refroidissement, interventions en cas de panne, etc.)
- Visibilité de l'ordinateur sur Internet : IP publique ? NAT ?

Licence de droits d'usage

Recours à un hébergeur

Les hébergeurs (p. ex., OVH, Gandi, Amen, Amazon, Orange, Free) proposent en général trois types d'offre :

- Hébergement Web mutualisé : le site est un hôte virtuel hébergé sur la même instance d'Apache ou IIS que d'autres sites. Restrictions d'espace disque, de débit, et de logiciels potentiellement utilisables. Convient pour des sites avec un petit nombre d'accès (moins d'un accès par seconde) et sans contrainte particulière.
- Serveur virtualisé : accès complet à une machine virtuelle physiquement située sur la même machine que d'autres serveurs virtuels. Apparaît comme une machine indépendante, mais typiquement moins puissante qu'un vrai serveur dédié. Convient à la plupart des sites.
- Serveur dédié. Pour les sites avec gros trafic ou les serveurs avec autre chose que du Web.

Licence de droits d'usage

Enregistrement du nom de domaine

- Le nom de domaine désiré doit être enregistré (acheté) auprès d'un *registrar* (*Go Daddy*, *Network Solutions*, etc.)
- La plupart des hébergeurs proposent également ce service
- Attention à la durée d'engagement
- Attention à ne pas manquer les dates de renouvellement
- Réfléchir au contenu de l'entrée Whois. Certains registrars permettent de la masquer en payant un supplément.
- Éventuellement une bonne idée d'enregistrer des noms de domaines proches

Licence de droits d'usage

Gestion des noms de domaine

- Besoin de (au moins) deux serveurs de noms qui répondront aux requêtes pour le nom de domaine enregistré
- Plusieurs possibilités :
  - le serveur Web lui-même, plus un autre serveur de l'entreprise
  - hébergeurs DNS dédiés, gratuits ou bon marché (FreeDNS, CloudDNS, ZoneEdit, etc.)
  - l'hébergeur Web fournit souvent gratuitement ou bon marché l'hébergement DNS
- Les serveurs de nom de domaine sont ensuite déclarés auprès du registrar

Licence de droits d'usage

Certificats SSL

En cas de site HTTPS, besoin d'un certificat SSL :

- Peut être autosigné, mais provoque des alertes du navigateur à la visite
- Peut être acheté à une autorité de certification (VeriSign, GeoTrust, DigiCert, etc.)
- Les hébergeurs Web fournissent souvent la possibilité de s'en occuper
- Valide seulement pour un nom de domaine donné !

Licence de droits d'usage

Plan

Créer et héberger un site Web

- Planification
- Choisir une solution d'hébergement
- Administration
- Développement du site

HTTP

Formulaires HTML

Langages côté serveur

Un exemple : PHP

Licence de droits d'usage

Installation du serveur

- Pas dans le cas d'un hébergement mutualisé
- Choix (et éventuellement achat de la license) de la version du système d'exploitation
- Installation (et éventuellement achat) de tous les logiciels nécessaires :
  - Serveur Web (Apache, IIS)
  - Accès distant (SSH, Remote Desktop, VNC, FTP)
  - SGBD (MySQL, Oracle, PostgreSQL, etc.)
  - Ce qui est requis par le langage côté serveur (module PHP, serveur d'applications Java, etc.)
  - CMS, frameworks côté serveur, etc.
  - Services non Web hébergés par le serveur (p. ex., email)

Licence de droits d'usage

Configuration d'Apache : Comment ?

Apache peut être configuré :

- par ses fichiers de configurations globaux (sous Linux, dans `/etc/apache2`) : uniquement pour les administrateurs du serveur. Seul endroit où de nouveaux hôtes virtuels peuvent être crées, ou où la configuration SSL peut être effectuée.
- par des fichiers `.htaccess` dans chacun des répertoires du site Web : doit être activé dans la configuration du serveur (`AllowOverride All`) – parfois impossible chez certains hébergeurs mutualisés.

Licence de droits d'usage

Configuration d'Apache : Quoi ?

- Jeu de caractère du site : `AddDefaultCharset utf-8`
- Redirections HTTP : `RedirectMatch permanent`
- Fichier d'index d'un répertoire : `DirectoryIndex toto.php`
- Authentification HTTP : `AuthUserFile`
- Négociation de contenu
- Types MIME
- cf. `http://httpd.apache.org/docs/`

Licence de droits d'usage

Serveur d'applications Java

Pour JSP, servlets Java, nécessité d'installer un serveur d'applications Java :

- Tomcat, JBoss, Geronimo (surcouche de Tomcat) : libres
- WebLogic Server, WebSphere : commerciaux
- Peuvent aussi servir de serveurs Web, ou connectés avec Apache : Tomcat se connecte avec Apache via mod_jk

Licence de droits d'usage

Optimisations

- Optimiser la base de données (index)
- Optimiser le code côté serveur
- Réduire le volume des pages téléchargées
- Ajouter un cache des pages Web les plus demandées :
  - Au niveau du serveur Web (mod_cache d'Apache)
  - Sur un serveur intermédiaire entre Internet et le serveur Web, servant de proxy
  - En utilisant des sociétés tiers pour héberger certains contenu (p. ex., vidéo) ou pour servir de proxy pour le site tout entier : Akamai, Google Web Accelerator
- Répartir la charge entre plusieurs serveurs Web :
  - Plusieurs adresses IP pour le même nom de domaine
  - Une adresse IP pointant vers un load balancer qui se contente de rediriger la requête vers de multiples serveurs
  - Possibilité d'optimisations géographiques

Licence de droits d'usage

Maintenance régulière

- Mises à jour de sécurité :
  - du système d'exploitation
  - du serveur Web
  - des modules du serveur Web et logiciels tiers
  - des CMS et frameworks d'application Web
  - du SGBD
- Surveillance régulière de la charge et de l'activité du serveur Web
- Surveillance de la structure de liens du site (pas de liens cassé, internes ou externes)
- Migrations ponctuelles : nouveau serveur, nouvelles versions de logiciels, etc.

Licence de droits d'usage

Plan

Créer et héberger un site Web
- Planification
- Choisir une solution d'hébergement
- Administration
- Développement du site

HTTP

Formulaires HTML

Langages côté serveur

Un exemple : PHP

Licence de droits d'usage

Développement par couches

- Successivement (entrelacement possible, mais chaque étape correspond à une progression logique dans le développement) :
  - Conception de la base de données
  - Développement de la logique de l'application côté serveur (modèle)
  - Développement des vues et du contrôleur côté serveur ; possible annotation sémantique dans les vues
  - Conception graphique, CSS
  - Interactions JavaScript
  - Contenu multimédia riche
  - Méta-informations : `robots.txt`, sitemap
- À chaque étape :
  - Démarche qualité : validation W3C, accessibilité, etc.
  - Tests utilisateurs, retours, pour infléchir le développement au besoin

Licence de droits d'usage

Lancement

- S’assurer de résister à la charge
- Basculement en production
- Marketing
- Surveiller l’activité très attentivement les premiers jours pour être près à corriger : performance, bugs, ergonomie, etc.

Licence de droits d’usage

Référencement

- S’assurer que le site est accessible aux robots avant le lancement
- Respecter les bonnes pratiques (cf. cours moteur de recherches)
- Marketing pour créer des liens vers le site
- Publicité ciblée : dans les moteurs de recherche, sur les sites tiers

Licence de droits d’usage

Plan

Créer et héberger un site Web

HTTP

Formulaires HTML

Langages côté serveur

Un exemple : PHP

Document License

HTTP (HyperText Transfer Protocol)

- Application protocol at the basis of the World Wide Web
- Latest and most widely used version: HTTP/1.1

- Client request:

```
GET /MarkUp/ HTTP/1.1
Host: www.w3.org
```

- Server response:

```
HTTP/1.1 200 OK
...
Content-Type: text/html; charset=utf-8

<!DOCTYPE html ...> ...
```

- Two main HTTP methods: GET and POST (HEAD is also used in place of GET, to retrieve meta-information only).
- Additional headers, in the request and the response
- Possible to send parameters in the request (key/value pairs).

Document License

GET

- Simplest type of request.
- Possible parameter are sent at the end of a URL, after a ‘?’
- Not applicable when there are too many parameters, or when their values are too long.
- Method used when a URL is directly accessed in a browser, when a link is followed, and for some forms.

Example (Google query)

URL: `http://www.google.com/search?q=hello`

Corresponding HTTP GET request:

```
GET /search?q=hello HTTP/1.1
Host: www.google.com
```

POST

- Method only used for submitting forms.

Example

```
POST /php/test.php HTTP/1.1
Host: www.w3.org
Content-Type: application/x-www-form-urlencoded
Content-Length: 100

type=search&title=The+Dictator&format=long&country=US
```

Parameter encoding

- By default, parameters are sent (with GET or POST) in the form: `name1=value1&name2=value2`, and special characters (accented characters, spaces...) are replaced by codes such as +, `%20`. This way of sending parameters is called `application/x-www-form-urlencoded`.
- For the POST method, another heavier encoding can be used (several lines per parameter), similar to the way emails are built: mostly useful for sending large quantity of information. Encoding named `multipart/form-data`.

Document License

Status codes

- The HTTP response always starts with a status code with three digits, followed by a human-readable message (e.g., 200 OK).
- The first digit indicates the class of the response:
  1. Information
  2. Success
  3. Redirection
  4. Client-side error
  5. Server-side error

Most common status codes

200	OK
301	Permanent redirection
302	Temporary redirection
304	No modification
400	Invalid request
401	Unauthorized
403	Forbidden
404	Not found
500	Server error

Document License

Virtual hosts

- Different domain names can refer to the same IP address, i.e., the same physical machine (e.g., `www.google.fr` and `www.google.com`)
- When a machine is contacted by TCP/IP, it is through its IP address
- No *a priori* way to know which precise domain name to contact
- In order to serve different content according to the domain name (virtual host): header `Host:` in the request (only header really required)

Example

```
GET /search?hl=fr&q=hello HTTP/1.1
Host: www.google.fr
```

Content type

- The browser behaves differently depending on the content type returned: display a Web page with the layout engine, display an image, load an external application, etc.
- MIME classification of content types (e.g., `image/jpeg`, `text/plain`, `text/html`, `application/xhtml+xml`, `application/pdf` etc.)
- For a HTML page, or for text, the browser must also know what character set is used (this has precedence over the information contained in the document itself)
- Also returned: the content length (can be used to display a progress bar)

Example

```
HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Content-Length: 3046
```


Document License

Client and server identification

- Web clients and servers can identify themselves with a character string
- Useful to serve different content to different browsers, detect robots...
- ... but any client can say it's any other client!
- Historical confusion on naming: all common browsers identify themselves as Mozilla!

Example

```
User-Agent: Mozilla/5.0 (X11; U; Linux x86_64; fr;
rv:1.9.0.3) Gecko/2008092510 Ubuntu/8.04 (hardy)
Firefox/3.0.3

Server: Apache/2.0.59 (Unix) mod_ssl/2.0.59 OpenSSL/0.9.8e
PHP/5.2.3
```

Authentication

- HTTP allows for protecting access to a Web site by an identifier and a password
- Attention: (most of the time) the password goes through the network uncrypted (but for instance, just encoded in Base64, revertible encoding)
- HTTPS (variant of HTTP that includes encryption, cryptographic authentication, session tracking, etc.) can be used instead to transmit sensitive data

Example

```
GET ... HTTP/1.1
Authorization: Basic dG90bzp0aXRp
```

Content negotiation

- A Web client can specify to the Web server:
  - the content type it can process (text, images, multimedia content), with preferrence indicators
  - the languages preferred by the user
- The Web server can thus propose different file formats, in different languages.
- In practice, content negociation on the language works, and is used, but content negociation on file types does not work because of bad default configuration of some browsers.

Example

```
Accept: text/html,application/xhtml+xml,application/xml;
q=0.9,*/*;q=0.8
Accept-Language: fr,fr-fr;q=0.8,en-us;q=0.5,en;q=0.3
```


Document License

Cookies

- Information, as key/value pairs, that a Web server asks a Web client to keep and retransmit with each HTTP request (for a given domain name).
- Can be used to keep information on a user as she is visiting a Web site, between visits, etc.: electronic cart, identifier, and so on.
- Practically speaking, most often only stores a session identifier, connected, on the server side, to all session information (connected or not, user name, data...)
- Simulates the notion of session, absent from HTTP itself

Example

```
Set-Cookie: session-token=RJYBsG//azkfZrRazQ3SPQhlo1FpkQka2;
path=/; domain=.amazon.de;
expires=Fri Oct 17 09:35:04 2008 GMT

Cookie: session-token=RJYBsG//azkfZrRazQ3SPQhlo1FpkQka2
```


Document License

Conditional downloading

- A client can ask for downloading a page only if it has been modified since some given date.
- Most often not applicable, the server giving rarely a reliable last modification date (difficult to obtain for dynamically generated content!).

Example

```
If-Modified-Since: Wed, 15 Oct 2008 19:40:06 GMT

304 Not Modified
Last-Modified: Wed, 15 Oct 2008 19:20:00 GMT
```

Originating URL

- When a Web browser follows a link or submits a form, it transmits the originating URL to the destination Web server.
- Even if it is not on the same server!

Example

```
Referer: http://www.google.fr/
```


Document License

Plan

Créer et héberger un site Web

HTTP

Formulaires HTML
- Généralités
- Les champs de saisie

Langages côté serveur

Un exemple : PHP

Document License

Plan

Créer et héberger un site Web

HTTP

Formulaires HTML
- Généralités
- Les champs de saisie

Langages côté serveur

Un exemple : PHP

Document License

Formulaires

- Permettent d'interagir avec l'utilisateur en lui proposant d'entrer des informations
- En HTML : uniquement l'interface de formulaire
- L'essentiel du travail sera fait par le script qui traitera la soumission du formulaire

Document License

Balise <form>

- Un formulaire HTML est placé à l'intérieur d'une balise `<form>`.
- Celle-ci prend les attributs suivants:

action	URL du script auquel sera soumis le formulaire.
method	Méthode HTTP, valant soit `"get"` soit `"post"`.
enctype	Encodage HTTP. Peut valoir `"application/x-www-form-urlencoded"` (valeur par défaut) ou `"multipart/form-data"`.

Example (Formulaire élémentaire)

```
<form action="action.php" method="get">
  <div><input type="submit"></div>
</form>
```

Ensembles de champ

En HTML, il est interdit de mettre des champs de formulaire directement à l'intérieur d'un `<form>`. Il faut d'abord les regrouper:

- Dans des paragraphes `<p>` si les champs de formulaires sont à l'intérieur de paragraphes de textes (rare).
- Dans des ensembles de champ `<fieldset>` pour regrouper des champs de formulaire de sémantique proche. On pourra alors donner une légende à l'ensemble de champs avec la balise `<legend>`.
- Dans des divisions `<div>` sans contenu sémantique sinon.

Exemple (Ensemble de champ)

```
<fieldset>
  <legend>Mensurations</legend>
  <input type="text" name="taille">
  <input type="text" name="poids">
</fieldset>
```


Document License

Étiquettes

- La plupart des champs sont naturellement accompagnés d'une étiquette (`<label>`).
- On peut la placer où on veut, en général juste à gauche ou à droite du champ.
- Son attribut `for` référence l'attribut `id` du champ correspondant.
- Dans les navigateurs graphiques, un clic sur l'étiquette d'un champ permet en général de sélectionner le champ.

Example (Étiquette)

```
<label for="taille">Taille :</label>
<input type="text" name="taille" id="taille">
```

Plan

Créer et héberger un site Web

HTTP

Formulaires HTML

Généralités

Les champs de saisie

Langages côté serveur

Un exemple : PHP

Document License

Champs de saisie

- La balise `<input>` a une utilisation très vaste dans les formulaires. Elle représente un champ de saisie.
- L'attribut `type` détermine le type (texte, mot de passe, liste, etc.) du champ.
- L'attribut `name` (nom du paramètre de la requête HTTP) est obligatoire (sauf pour les types `"reset"` et `"submit"`); il permet de préciser au serveur à quelle saisie on fait référence.

Exemple (Zone de texte pour écrire un commentaire)

```
<input type="text" name="Commentaire">
```

Saisie d'une ligne de texte

- `type ="text"` est utilisé pour la saisie d'un texte dont la taille est inférieure à une ligne.
- L'attribut `value` permet de préciser la valeur par défaut (facultatif).
- La taille maximale de la chaîne de caractères à saisir peut être spécifiée à l'aide de l'attribut `maxlength` (facultatif).

Exemple

```
<input type="text" name="prenom" value="Jordy" maxlength="50">
```

Saisie d'un mot de passe

- `type ="password"` est utilisé pour la saisie d'un texte dont les caractères sont remplacés par des astérisques: c'est généralement utilisé pour la saisie des mots de passe. Le mot de passe est quand même transmis en clair au serveur!
- L'attribut `value` permet de préciser la valeur par défaut (facultatif).
- La taille maximale de la chaîne de caractères à saisir peut être spécifiée à l'aide de l'attribut `maxlength` (facultatif).

Exemple

```
<input type="password" name="pwd" value="12345678">
```

Choix multiples parmi une liste

- `type ="checkbox"` permet de choisir plusieurs éléments parmi une liste de possibilités.
- Cela se matérialise sous forme de cases à cocher.
- La valeur retournée est obligatoirement précisée à l'aide de l'attribut `value`.
- L'attribut `checked ="checked"` permet de cocher la case par défaut.
- Certains langages côté serveurs imposent que les noms de champs de formulaire à choix multiples se terminent par [].

Exemple

```
<input type="checkbox" name="pub[]" value="site"
  checked="checked" id="pub-site">
<label for="pub-site">Recevoir des offres de notre site</label>

<input type="checkbox" name="pub[]" checked="checked"
  value="externe" id="pub-externe">
<label for="pub-externe">Recevoir des offres externes</label>
```

Choix unique parmi une liste

- `type ="radio"` permet de choisir un seul élément parmi une liste de possibilités.
- Cela se matérialise sous forme de boutons radio.
- La valeur retournée est obligatoirement précisée à l'aide de l'attribut `value`.
- L'attribut `checked ="checked"` permet de préciser la valeur par défaut.

Exemple

```
Recevoir de la pub:
<input type="radio" name="pub" value="oui" id="pub-oui"
  checked="checked">
<label for="pub-oui">oui</label>

<input type="radio" name="pub" value="non" id="pub-non">
<label for="pub-non">non</label>
```


Document License

Fichiers joints

- `type ="file"` permet de joindre au formulaire un fichier.
- À cause de la taille de la requête due au téléchargement (upload) du fichier, il faut impérativement utiliser la méthode POST et l'encodage `multipart/form-data`.

Exemple

```
<label for="fichier">Fichier:</label>
<input type="file" name="fichier" id="fichier">
```

Champs cachés

- `type ="hidden"` permet de cacher des champs au client mais leur contenu est envoyé avec le formulaire.
- Ceci permet de préciser des informations, en utilisant l'attribut `value`, concernant l'interaction client/serveur.
- C'est à utiliser avec précaution car cela peut être à l'origine de problèmes de sécurité assez graves: ne pas oublier que le client peut éditer la page à la main pour changer la valeur de ces champs!

Exemple

```
<input type="hidden" name="monnaie_utilisee" value="EUR">
<input type="hidden" name="customerCB" value="c2415-345-8563">
```

Ré-initialisation d'un formulaire

- `type ="reset"` permet de réinitialiser le formulaire en affectant aux différents champs leur valeur par défaut.
- L'attribut `value` permet de changer le texte du bouton correspondant.

Exemple

```
<input type="reset" value="Tout effacer">
```


Document License

Soumettre le formulaire

- `type ="submit"` permet de soumettre le formulaire.
- Le client envoie le contenu du formulaire à l'adresse précisée par l'attribut `action` de la balise `<form>` .
- L'attribut `value` permet de changer le texte du bouton correspondant.

Exemple

```
<input type="submit" value="Envoyer">
```

Saisie de plusieurs lignes de texte

- Pour les saisies multiligne, on utilise la balise `<textarea>` .
- Le texte délimité par cette balise permet d'initialiser la valeur par défaut du champ.
- La balise fermante est obligatoire même si le champ est vide.
- Les attributs `rows` et `cols` (obligatoires) permettent de spécifier la taille en lignes et colonnes de la fenêtre de saisie.

Exemple

```
<textarea name="bio" cols="40" rows="5">
  Fille de Josiane Balasko,
  Marilou Berry fait ses premiers pas au cinéma à 8 ans...
</textarea>
```


Document License

Menus de sélection

- La balise `<select>` permet d'ajouter une liste de sélection:
  - L'attribut facultatif `size` permet de préciser le nombre de choix apparaissant sur la page Web. Par défaut, ce nombre est initialisé à 1.
  - L'attribut `multiple ="multiple"` permet d'autoriser des réponses multiples. Dans ce cas, pour PHP, on donnera toujours un nom se terminant par `[]`.
- Les choix du menu sont indiqués à l'aide de la balise `<option>` :
  - L'attribut `selected ="selected"` permet de spécifier le(s) choix par défaut.
  - L'attribut `value` permet de spécifier la valeur associée au choix.

Exemple

```
<select name="age">
  <option value="20">Moins de 20 ans</option>
  <option value="35" selected="selected">21 à 35 ans</option>
  <option value="50">36 à 50 ans</option>
  <option value="51">Plus de 51 ans</option>
</select>
```

Plan

Créer et héberger un site Web

HTTP

Formulaires HTML

Langages côté serveur

Un exemple : PHP

Document License

Rôle des programmes côté serveur

- Le Web, ce n'est pas qu'un ensemble de documents HTML statiques !
- Les programmes côté serveur permettent :
  - de traiter des soumissions de formulaire ;
  - d'afficher de manière uniforme l'ensemble des pages d'un site ;
  - de proposer des applications interactives ;
  - de permettre à l'utilisateur d'ajouter ou modifier du contenu ;
  - etc.

Licence de droits d'usage

Langages de programmation

- CGI (Common Gateway Interface) : interface normalisée permettant de faire communiquer le serveur Web avec un programme s'exécutant sur le serveur ;
- CGI permet d'utiliser n'importe quel langage de programmation (compilés comme C, C++, Java, ou interprétés comme Perl, Python, Ruby, etc.) pour écrire des langages côté serveur.
- Mais certains langage sont plus adaptés au développement Web :
  - comportent des fonctions spécialement dédiées à HTTP, HTML, etc. ;
  - s'intègrent de manière plus efficace et plus pratique avec le serveur Web (moyennant des extensions logicielles) ;
  - ont une syntaxe spécialement conçue, qui mêle code HTML envoyé tel quel et instructions de programmation interprétées.
- Quelle que soit la technologie utilisée, le code du programme n'est pas accessible par le navigateur Web, seulement le résultat de son exécution.

Licence de droits d'usage

Langages côté serveur

PHP : un des langages les plus populaires, s'intègre très facilement avec Apache (libre)

ASP et ASP.NET : destiné à être utilisé avec IIS (Microsoft, commercial)

ColdFusion (Adobe, commercial)

JSP (Java Server Pages) : permet de mêler instructions Java et code HTML ; nécessite un serveur d'applications Java (p. ex., Tomcat) en plus d'Apache (Sun, gratuit voire libre)

Servlets Java : véritables programmes Java, plutôt pour les applications complexes côté serveur avec peu d'interaction côté client ; nécessite un serveur d'applications Java en plus d'Apache (Sun, gratuit voire libre)

Licence de droits d'usage

Frameworks d'applications Web

- Les langages présentés ci-avant restent assez basiques et généralistes.
- N'encouragent pas forcément une organisation propre d'un site Web.
- Framework : ensemble d'un langage de programmation, d'une bibliothèque de fonctions, d'outils externes, de bonnes pratiques à suivre. . .
- Permet d'abstraire la création d'une page Web.
- Suit en général le modèle MVC (voir transparent suivant).
- Inclut parfois la génération de code JavaScript côté client pour créer directement une application Web fortement dynamique (p. ex., validation de formulaire) ; intégration Ajax également.
- Fortement recommandé pour créer des applications complexes. . . mais également complexe à maîtriser !

Licence de droits d'usage

Modèle MVC

- Principe de génie logiciel, utilisé dans d'autres domaines
- Particulièrement adapté au cas des applications Web !
- Séparation propre entre :

  Modèle : données manipulées par l'application et fonctions de manipulation de ces données ; réutilisable pour d'autres applications Web

  Vue : présentation des données ; facilement échangeable pour changer l'apparence et la structure du site

  Contrôleur : contrôle la manière dont l'utilisateur interagit, au travers de la vue, avec les données du modèle

Licence de droits d'usage

Frameworks côté serveur les plus populaires

ASP.NET : DotNetNuke

ColdFusion : Model-Glue, Fusebox

Java : Struts, Spring, JavaServer Faces, Google Web Toolkit

Perl : Catalyst

PHP : CakePHP, CodeIgniter, Symphony, Zend

Python : Django

Ruby : Ruby on Rails (a eu beaucoup d'influence !)

Smalltalk : Seaside

. . . et beaucoup d'autres !

Licence de droits d'usage

CMS

- CMS (Content Management System)
- Permettent de créer des sites Web sans aucun développement.
- Fonctionnalités :
  - édition simplifiée de page (syntaxe wiki ou bbcode, ou contrôle JavaScript texte enrichi) ;
  - ajout de contenu externe (images, documents annexes, etc.) ;
  - gestion d'utilisateurs, contrôle d'accès, etc. ;
  - modules de gestion de forums, de blogs ;
  - thèmes graphiques prêt à l'emploi ;
  - contrôle de version.
- Suivant les CMS, extensions pouvant être nombreuses.
- Certains CMS spécialisés : blogs (Dotclear, Movable Type, TypePad), commerce électronique (PrestaShop, Magento), forums (phpBB, MyBB), etc.

Licence de droits d'usage

CMS les plus utilisés

Nombre de téléchargements hebdomadaires :

WordPress	PHP	433 767
Joomla !	PHP	189 429
Drupal	PHP	62 500
Umbraco	.NET/XSLT	5 670
eZ Publish	PHP	5 612
CMS Made Simple	PHP	4 903
SilverStripe	PHP	2 500
e107	PHP	2 242
Xoops	PHP	1 209
TikiWiki	PHP	373
phpWebSite	PHP	347
Typo3	PHP	100
Alfresco	Java	57
DotNetNuke	ASP.NET	?
Jahia	Java	?
Liferay	Java	?
modx	PHP	?
OpenCMS	Java	?
Plone	Python	?
TextPattern	PHP	?

Licence de droits d'usage

CMS et programmation Web

Même en utilisant un CMS, il est utile de connaître les technologies de base du Web (HTML, CSS, JavaScript, langage côté serveur) :

- pour créer ses propres styles CSS (quasi-indispensable) ;
- pour développer des applications complexes, propres au site ;
- pour développer ou adapter des petites extensions ;
- pour comprendre ce qui se passe en cas de problème ;
- pour s'assurer que les pages respectent certaines conditions (validité W3C, accessibilité).

Licence de droits d'usage

CMS et Wikis

Frontière peu nette entre les deux... Les Wikis mettent l'accent sur :

- a priori, tout est éditable par un grand nombre d'utilisateurs ; pas de notion de « propriété »d'une page ou d'un document.
- insistance sur le contrôle de versions

Nombreux wikis : MediaWiki, TWiki, Dokuwiki...

Licence de droits d'usage

Choisir un CMS

Point auxquels faire attention :

- la licence (la plupart des CMS populaires sont libres, mais pas tous)
- le langage sous-jacent, important s'il y a besoin de modifier/personnaliser le logiciel, et également important pour savoir comment le déployer
- la disponibilité de fonctionnalités spécifiques au site : blogs, forums, support vidéo, etc. Il est possible d'utiliser des composants externes pour chacune des tâches, mais plus simple de rester sur un seul système.

Licence de droits d'usage

Inconvénients potentiels

- Plus lent qu'un site classique ; fonctionnalités de cache permettant d'accélèrer, mais peut poser problème dans certains contextes.
- Failles de sécurité, code sur lequel on n'a pas de contrôle.
- Logiciels orphelins.
- Migration vers un autre système potentiellement coûteuse.
- Non adapté à tous les usages.

Licence de droits d'usage

Repérer une technologie côté serveur

Impossible d'avoir accès au code source, donc comment savoir avec quelle technologie côté serveur un site a été réalisé ?

- URL : (`.php` pour PHP, `.jsp` pour JSP, `.asp` pour ASP, `/servlet/` pour une servlet...). Tout cela est configurable, mais bien souvent la configuration par défaut est utilisée.
- Ça peut être écrit en toutes lettres sur le site (souvent pour un CMS)
- Regarder les commentaires dans le code HTML, CSS...
- Organisation des fichiers et répertoires, librairies JavaScript ou CSS chargées, etc.
- Cookies, en particulier identifiants de session

Licence de droits d'usage

Plan

Créer et héberger un site Web

HTTP

Formulaires HTML

Langages côté serveur

Un exemple : PHP
- Le langage PHP
- PHP et HTTP
- Sessions et autentification

Licence de droits d'usage

Plan

Créer et héberger un site Web

HTTP

Formulaires HTML

Langages côté serveur

Un exemple : PHP

Le langage PHP

PHP et HTTP

Sessions et autentification

Licence de droits d'usage

PHP et HTML

- Script PHP : document HTML (par exemple), dans lequel est incorporé du code PHP.
- Le code PHP est à l'intérieur d'une pseudo-balise `<?php ... ?>` (ou `<? ... ?>` , ou `<?= ... ?>` qui est un raccourci pour `<? echo ... ?>`).

Exemple

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

<html>
  ...
  <body>
  <h1><?php echo 2+2; ?></h1>
</body>
</html>
```


Licence de droits d'usage

Syntaxe

- Syntaxe similaire à (et inspirée de) C, Java :
  - instructions séparées de points-virgules
  - instructions de base : `if` , `while` , `for` ...
  - opérateurs de base : `=` , `==` , `++` , `<=` ...
  - mêmes commentaires : `//` , `/* */` , mais `#` fonctionne également
  - littéraux similaires (mais voir plus loin l'interpolation)
  - etc.
- Mais des différences importantes :
  - nouvelles instructions (e.g., `foreach`), nouveaux opérateurs (`.` est la concaténation de chaîne)
  - variables non typées, non déclarées, de portée (par défaut) la fonction toute entière
  - variables toujours préfixées d'un signe « $ »

Licence de droits d'usage

Interpolation

Les chaînes de caractère en PHP peuvent être :

- entourées par des apostrophes : `'Hello world'`. Elles se comportent comme les chaînes de caractère Java, mais les séquences d'échappement (`\n` ...) ne sont pas disponibles.
- entourées par des guillemets doubles : `"Hello\n$world"`. Les séquences d'echappement sont autorisées, et les noms de variables sont interpolés (remplacés par leur valeur) En cas de valeur complexe, utiliser la syntaxe `${var}` :

  `"La valeur est {$matrix[$i][$j]}"`

Licence de droits d'usage

- Un tableau peut contenir des valeurs de types différents : entier, chaîne de caractères, etc.

Exemple

```
$tab[0] = "1er element";
$tab[1] = "2e element";
$tab[2] = 120;
```

Licence de droits d'usage

- En PHP, l'affectation d'un indice à un nouvel élément est automatique : il correspond à la première cellule vide.

Exemple

```
$tab2[] = "1er element";
$tab2[] = "2e element";
$tab2[] = 120;
```

- L'initialisation d'un tableau peut également se faire à l'aide de la fonction `array`

Exemple

```
$tab3 = array("1er element","2e element",120);
```


Licence de droits d'usage

- Les indices numériques sont remplacés par des chaînes de caractères appelées clefs.
- Pour un tableau donné, toutes les clefs doivent être différentes.
- La fonction `array` peut également être utilisée pour intialiser le tableau.

Exemple

```
$prenom["Belmondo"] = "Jean-Paul";
$prenom["Delon"] = "Alain";
$prenom["Deneuve"] = "Catherine";
$prenom = array("Balasko" => "Josiane", "Bourvil" => "");
```

Licence de droits d'usage

- Le parcours d'un tableau associatif est plus complexe que celui des tableaux indicés : on ne peut pas se baser sur l'ordre des indices pour effectuer une boucle simple.
- On peut utiliser un curseur sur le tableau : sorte de flèche indiquant l'élément du tableau actuellement visé.
- Les fonctions `next` et `prev` permettent de déplacer le curseur initialement positionné sur le premier élément du tableau.
- Les fonctions `key` et `current` renvoient respectivement la clef et la valeur de l'élément courant.

Licence de droits d'usage

Exemple (Afficher toutes les personnes de la liste prenom)

```
echo "Nom=".key($prenom).
     " Prenom=".current($prenom)."\n";

while(next($prenom)) {
  echo "Nom=".key($prenom)." Prenom=".current($prenom);
}
```


Licence de droits d'usage

- Il est cependant plus facile de parcourir un tableau associatif à l'aide de l'instruction `foreach`.

Exemple

```
foreach ($prenom as $cle => $valeur) {
  echo "Nom=$cle Prenom=$valeur\n";
}
```

- Cette instruction peut également être utilisée pour des tableaux indicés.

Exemple

```
foreach ($tableau as $valeur) {
  ...
}
```


Licence de droits d'usage

- Pour faciliter la visibilité d'un script PHP et pour rendre automatique l'exécution de certaines tâches répétitives, on peut définir des fonctions.
- La construction `list ( ... )` peut être utilisée pour récupérer dans plusieurs variables différentes une valeur de retour qui est un tableau.

Licence de droits d'usage

Exemple

```
function Kenshin() {
  return array ("Kenshin Le Vagabond", "Nobuhiro Watsuki", 28);
}

function Addition($x,$y) {
  $somme = $x+$y;
  return $somme;
}

$z=Addition($x,$y);
list ($a,$b,$c)=Kenshin();
```


Licence de droits d'usage

Mais aussi...

- Langage très (trop ?) riche
- Programmation orientées objet : `class`, `$object->field`... souvent utilisée par les frameworks PHP
- Très large bibliothèque de fonction, avec de nombreuses bibliothèques externes disponible
- Astuce : `http://php.net/fonction` donne la documentation de la fonction PHP *fonction*

Licence de droits d'usage

Validation

- Un script PHP avec du code PHP incorporé au sein de code HTML n'est pas un document XHTML valide !
- Ce qu'il faut valider, c'est une (des) page(s) HTML produites par le script.
- Possibilité d'indiquer une URL au validateur du W3C. Malheureusement pas utilisable quand l'URL est privée. Possibilité dans ce cas de sauvegarder le fichier HTML produit, et de l'envoyer au validateur du W3C comme fichier local.
- Il n'y a pas de « validateur » PHP, mais les erreurs de syntaxe causeront des erreurs à l'exécution du script.
- Rien ne change pour les CSS, on peut les valider à part.

Licence de droits d'usage

Plan

Créer et héberger un site Web

HTTP

Formulaires HTML

Langages côté serveur

Un exemple : PHP

- Le langage PHP
- PHP et HTTP
- Sessions et autentification

Licence de droits d'usage

$_GET et $_POST

- Les paramètres HTTP peuvent être récupérées en PHP grâce au tableaux associatifs `$_GET` et `$_POST`.
- Les valeurs de ce tableau peuvent être des variables simples ou des tableaux indicés : ces derniers sont les paramètres à choix multiples dont on a suffixé le nom de `[]` dans le code HTML.

Exemple

```
echo "<p>Votre login est: " . $_POST["login"] . "</p>";
echo "<p>Vous avez coché les genres: ";
for($i=1;$i<=count($_POST['genre']);$i=$i+1) {
  echo $_POST['genre'][$i] . " ";
}
echo "</p>";
```


Licence de droits d'usage

Les informations relatives aux fichiers transférés sont disponibles dans un tableau associatif $_FILES :

- les clés sont les noms des champs de formulaire d'où provient le fichier
- las valeurs sont des ensembles de propriétés (décrits comme des tableaux associatifs) décrivant le fichier reçu par le serveur auxquelles s'ajoute la propriété *error* qui permet de savoir si le transfert s'est bien déroulé

Exemple (dans un fichier *FormTransfert.html*)

```
<form enctype="multipart/form-data"
  action="TransfertFichier.php" method="post">
  ...
  <div>
    <label for="maPhoto">Choisissez un fichier :</label>
    <input type="file" name="maPhoto" id="maPhoto" />
  </div>
  ...
</form>
```


Licence de droits d'usage

name	est le nom du fichier sur la machine du client
tmp_name	est le nom du fichier temporaire sur la machine du serveur
size	est la taille du fichier, en octets
type	est le type MIME du fichier, par exemple « image/gif »

Exemple (dans le fichier *TransfertFichier.php*)

```
$fichier=$_FILES['maPhoto'];
echo "Nom fichier client: ".$fichier['name']."<br />";
echo "Nom fichier serveur: ".$fichier['tmp_name']."<br />";
echo "Taille du fichier: ".$fichier['size']."<br />";
echo "Type du fichier: ".$fichier['type']."<br />";
```


Licence de droits d'usage

name	est le nom du fichier sur la machine du client
tmp_name	est le nom du fichier temporaire sur la machine du serveur
size	est la taille du fichier, en octets
type	est le type MIME du fichier, par exemple « image/gif »

Exemple (dans le fichier *TransfertFichier.php*)

```
$fichier=$_FILES['maPhoto'];
echo "Nom fichier client: ".$fichier['name']."<br />";
echo "Nom fichier serveur: ".$fichier['tmp_name']."<br />";
echo "Taille du fichier: ".$fichier['size']."<br />";
echo "Type du fichier: ".$fichier['type']."<br />";
```


Licence de droits d'usage

UPLOAD_ERR_OK pas d'erreur, le transfert s'est bien passé

UPLOAD_ERR_INI_SIZE le fichier transmis dépasse la taille maximale autorisée

UPLOAD_ERR_PARTIAL le fichier est transféré seulement partiellement

UPLOAD_ERR_NO_FILE aucun fichier n'a été transféré

Exemple

```
$codeErreur = $_FILES['maPhoto']['error'];

if($codeErreur!=UPLOAD_ERR_OK) {
  echo "<p>Erreur lors du transfert du fichier.</p>";
}
```


Licence de droits d'usage

- La fonction PHP `copy($source,$destination)` permet de copier le fichier *source* vers *destination*. Important parce que le fichier temporaire pourra être détruit à la fin du script !
  - Attention : le programme doit avoir les droits d'accès et d'écriture sur les répertoires dans lesquels les fichiers sont copiés.
- La fonction `md5(ch)` permet de générer une nouvelle chaîne de caractères à partir de *ch*. Il est à peu près impossible d'obtenir deux valeurs identiques pour des chaînes différentes ce qui permet de considérer cette fonction comme un cryptage de chaîne. On peut en particulier utiliser md5 pour générer un nom de fichier quand on n'a rien de mieux (par exemple, un identifiant).

Exemple

```
// Copie du fichier dans le répertoire PHOTOS
copy($_FILES['toto']['tmp_name'],"./PHOTOS/$id.jpg");
```


Licence de droits d'usage

Plan

Créer et héberger un site Web

HTTP

Formulaires HTML

Langages côté serveur

Un exemple : PHP

- Le langage PHP
- PHP et HTTP
- Sessions et autentification

Licence de droits d'usage

session : ensemble d'informations conservées tout au long d'une interaction avec les différentes pages d'un site Web

authentification : mécanisme permettant d'associer un identifiant (login) à un utilisateur d'un site Web, de manière à permettre de la personnalisation du contenu, ou de la gestion de droits sur une application Web ; habituellement, l'authentification est faite grâce à un mot de passe.

Licence de droits d'usage

- Méthode d'autentification HTTP simple disponible au niveau du serveur Web, mais impose une modification de la configuration de celui-ci ; un peu lourd, difficilement connectable à un SGBD.
- Pas de gestion de session à proprement parler en HTTP. Alternatives :
  - Paramètres HTTP cachés dans l'URL (méthode GET). Mécanisme un peu lourd, puisque tous les liens doivent être changés pour incorporer ces paramètres.
  - Cookies.

Licence de droits d'usage

- Informations, sous la forme de clés/valeurs, qu'un serveur Web demande à un client Web de conserver et de retransmettre à chaque requête HTTP.
- `setcookie($name,$value)` : demande client de stocker un cookie de nom $name et de valeur $value.
- `$_COOKIE` est un tableau associatif des cookies que le client a envoyé au serveur Web.

Licence de droits d'usage

PHP fournit une abstraction de la gestion de session (utilisant un cookie, mais sans avoir à le gérer soi-même).

session_start() ouvre une session en cours, ou crée une nouvelle session s'il n'y a pas de session ouverte ; ceci est à placer au tout début du script PHP, ou en tous cas avant que quoi que ce soit n'ait été écrit dans la page, de manière à pouvoir modifier les en-têtes de la réponse HTTP.

session_destroy() termine la session en cours.

session_id() fournit un identifiant de la session en cours.

$_SESSION contient l'ensemble des paramètres de session (tableau associatif clé/valeur), disponibles dans les différentes pages Web de la même session.

Licence de droits d'usage

- Un formulaire demande login et mot de passe.
- Un script de traitement de ce formulaire, contrôle que le login et le mot de passe sont corrects (par exemple à l'aide d'une table MySQL) :
  - Si c'est le cas, crée une session PHP (`session_start();`), y ajoute un paramètre nommé par exemple valid_user (`$_SESSION['valid_user']=1;`) et redirige vers une autre page.
  - Sinon, redirige vers la page de formulaire.
- Les autres pages (pages auxquelles les utilisateurs authentifiés et seulement eux ont accès) commençent par un `session_start();` et contrôlent si l'utilisateur est identifié (`if($_SESSION['valid_user']==1) { ... }`) et sinon redirigent vers la page de formulaire
- Une page de déconnexion appelle `session_destroy();`

Licence de droits d'usage

Licence de droits d'usage

Contexte public } avec modifications

Par le téléchargement ou la consultation de ce document, l'utilisateur accepte la licence d'utilisation qui y est attachée, telle que détaillée dans les dispositions suivantes, et s'engage à la respecter intégralement.

La licence confère à l'utilisateur un droit d'usage sur le document consulté ou téléchargé, totalement ou en partie, dans les conditions définies ci-après et à l'exclusion expresse de toute utilisation commerciale.
Le droit d'usage défini par la licence autorise un usage à destination de tout public qui comprend :
– le droit de reproduire tout ou partie du document sur support informatique ou papier,
– le droit de diffuser tout ou partie du document au public sur support papier ou informatique, y compris par la mise à la disposition du public sur un réseau numérique,
– le droit de modifier la forme ou la présentation du document,
– le droit d'intégrer tout ou partie du document dans un document composite et de le diffuser dans ce nouveau document, à condition que :
    – L'auteur soit informé.

Les mentions relatives à la source du document et/ou à son auteur doivent être conservées dans leur intégralité.

Le droit d'usage défini par la licence est personnel et non exclusif.
Tout autre usage que ceux prévus par la licence est soumis à autorisation préalable et expresse de l'auteur : sitepedago@telecom-paristech.fr