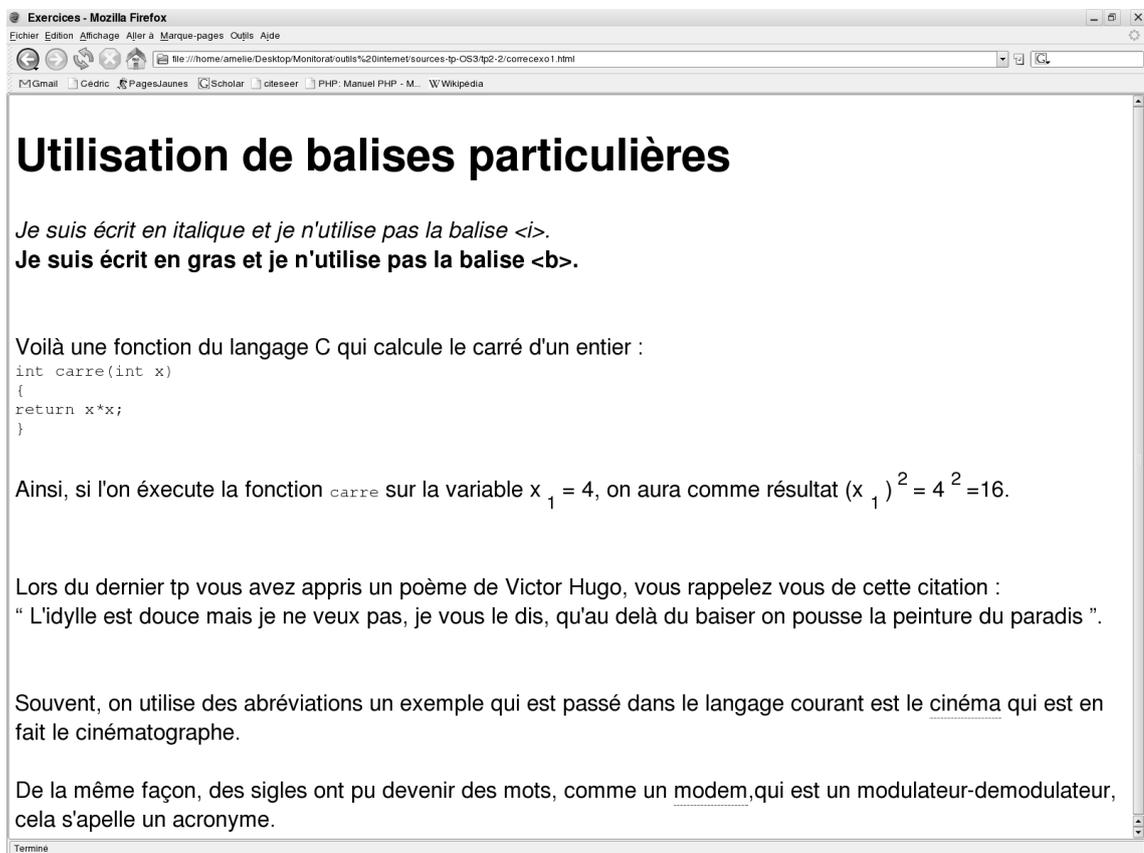


# TP 3 - Les formulaires Internet et Outils (IO2)

## 1 Utilisation de balises particulières

**Exercice 1 — *Les balises particulières*** Reproduisez la page web suivante. N'oubliez pas d'utiliser les balises `<abbr>` et `<acronym>` lorsque cela est nécessaire.



## 2 Les liens internes

On a vu que l'élément `a` pouvait avoir comme attribut `href="url"` lorsqu'on voulait afficher un lien vers une page externe située à l'adresse `url`. Mais on peut en fait être plus précis que ça et indiquer où exactement dans la page, le navigateur doit se positionner. Il faut pour cela avoir défini un *point d'ancrage* (d'où le nom de la balise) pour l'endroit que l'on veut pointer. Cela se fait grâce à l'attribut `name="nom"` de l'élément `a` à l'endroit voulu dans la page située à l'adresse `url`. Ensuite, on peut y faire référence avec l'attribut `href="url#nom"` (ne pas oublier le #). Lorsque l'ancre est située sur la même page que le pointeur, on omet l'url.

**Exercice 2 — Les liens internes** Récupérez le fichier <http://www.pps.jussieu.fr/~tasson/enseignement/I02/tp3/liens.html> Faites en sorte que le sommaire soit « cliquable » et renvoie vers le début des chapitres correspondants. De la même façon, faites en sorte que l'on puisse revenir au sommaire, par un simple clic, à la fin de chaque chapitre.

### 3 Les formulaires

XHTML permet la création de formulaires grâce à un certain nombre d'éléments. Les formulaires occupent une place un peu à part dans une page car ils constituent une composante interactive de la page. Un formulaire permet à un internaute de remplir des champs textuels, de cocher des cases, etc... mais surtout d'envoyer ces informations au serveur qui héberge la page. Ces informations peuvent ensuite être traitées pour, par exemple, afficher une page correspondant aux choix faits par l'internaute.

Nous allons pour l'instant nous intéresser simplement à la déclaration de formulaires. L'aspect traitement de l'information sera abordé la semaine prochaine.

**Exercice 3 — Les attributs des formulaires XHTML** Afin de bien assimiler les éléments utiles pour un formulaire, nous allons créer une page XHTML qui en résume les définitions. Ces définitions sont celles d'une grande partie des attributs de déclaration d'un formulaire, seulement, 9 erreurs se sont glissées dans cette liste. Présentez sous forme d'une liste de définition les attributs de déclaration d'un formulaire en corrigeant les erreurs du texte de la façon suivante, les éléments faux doivent être barrés et les corrections soulignées.

Liste des éléments utiles pour un formulaire :

`form` : Pour déclarer un formulaire.

C'est l'élément qui englobe tous les autres. Il a trois attributs principaux :

- `action` : mode de transmission des paramètres
- `method` : référence du programme qui doit être exécutée sur le serveur
- `enctype` : type d'encodage des données du formulaires, de type GET ou POST

*Les balises de saisies :*

`input` : Pour définir un champ.

Il peut contenir plusieurs attributs dont l'attribut `type` qui peut avoir les valeurs suivantes :

- `text` : Pour définir une plage de texte.
- `password` : Pour obtenir un mot de passe (le texte sera caché mais non crypté).
- `button` : Crée des cases à cocher.
- `checkbox` : Crée aussi des cases à cocher mais un seul choix sera possible.
- `radio` : Crée un bouton.
- `file` : Pour pouvoir donner le chemin d'accès d'un fichier sur l'ordinateur de l'internaute.
- `hidden` : Pour interdire l'accès à ce champs-là.
- `reset` : Pour réinitialiser le formulaire
- `submit` : Pour envoyer les informations (nous y reviendrons un peu plus tard).

D'autre part, il peut (mais ne doit pas dans le cas des types `checkbox` et `radio`) contenir l'attribut `value="nom"` dont la signification dépend de la valeur de `type` :

- Si le type est `button`, `reset` ou `submit`, alors `nom` est le texte inscrit sur le bouton.
- Si le type est `password`, `text` ou `hidden`, alors `nom` est la valeur qui sera transmise au moment de l'envoi des données.
- Si le type est `checkbox` ou `radio`, alors `nom` est le texte inscrit par défaut.

`select` : Définit un menu déroulant. Le choix sera proposé via l'élément `option` en utilisant l'attribut `value="choix"`

Fin de la liste.

**Exercice 4 — Déclaration de formulaire** Les éléments `input` et `select` peuvent de plus contenir l'attribut `name="nom"` où `nom` sera plus tard utilisé pour traiter les données. Pour prendre de bonnes habitudes, dans la suite du TP vous ajouterez systématiquement un attribut `name` à tous les éléments destinés à être traités.

Créez une page web intitulée *Mon formulaire*, dont la référence du programme qui doit être exécutée sur le serveur est la chaîne de caractère «nom» et qui propose les champs suivants :

- Nom
- Prénom
- Mot de passe
- Confirmation du mot de passe
- Genre : choix masculin ou féminin
- Votre âge : choix dans un menu déroulant proposant des âges de 18 à 25 ans.
- Vous aimez : choix non exhaustif entre thé et café

Vous n'oubliez pas le bouton d'envoi ni celui de réinitialisation.

Présentez ce formulaire sous forme d'un tableau de deux colonnes, dans la première il y aura les noms des champs et dans la deuxième les champs. faites en sorte que les bordures du tableau soient invisibles.

Récupérez le fichier <http://www.pps.jussieu.fr/~tasson/enseignement/I02/tp3/form.php>. Que se passe t-il quand on appuie sur le bouton d'envoi ?