**HTML : mode d’emploi**

***Ce document est une prise de notes sur les MOOCs « Tutoriel HTML complet» et «Envoyer et extraire les données des formulaires » disponibles aux adresses suivantes :*** [**https://tutorialehtml.com/fr/html-tutoriel-html-complet/**](https://tutorialehtml.com/fr/html-tutoriel-html-complet/) **et** [**https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Envoyer\_et\_extraire\_les\_donn%C3%A9es\_des\_formulaires**](https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Envoyer_et_extraire_les_donn%C3%A9es_des_formulaires)**.**

HTML : langage de balises

Pour **ouvrir un fichier html** qui est sur notre ordinateur, on **copie-colle le path** sur un navigateur web comme chrome

Mots entourés de < et > = **balises**: spécifie les régions d’un doc html

Base d’un code html :

<html>

<body>

Le contenu de la page internet ira ici

</body>

</html>

* **<html>** : indique au navigateur le début d’un code html
* **<head>** : se met immédiatement après html, sert à indiquer au navigateur le titre de la page, son contenu et d’autres infos : c’est ce titre de page qui apparaît dans la nom de l’onglet sur Chrome ; on peut mettre du Javascript ou du CSS dans le head
* **<title>** : se situe entre les deux balises head : donne le nom de la page
* **<body>** : indique où le contenu de la page internet commence
* **<h2>** : se situe après <body> : titre qui apparaît dans la page, avant le contenu : plus gros que le caractère du contenu : il existe les balises <h1>, …, <h6> pour des tailles différentes de textes avec <h1> le plus gros caractère
* **<p>** : ouverture d’un paragraphe, qui se finit avec </p> : saute automatiquement une ligne au début et à la fin du paragraphe

Il faut toujours indiquer au navigateur l’arrêt d’une fonction **: on ferme la balise** <balise> avec un / donc avec la balise **</balise>.** Bien respecter l’**ordre** d’ouverture et de fermeture des balises.

Defs :

* **balises**: spécifie les régions d’un doc html, commande que la navigateur exécute, a 3 parties : ouverture, contenu, fermeture
* **élément**: balise complète, qui possède une ouverture <tag> et une fermeture </tag>
* **attribut**: change et personnalise la valeur d’un élément en html

Remarque : toutes les balises sont écrites en petites lettres

Attention : certaines balises n’ont pas de contenu donc ne respectent pas le modèle ouverture-contenu-fermeture, et combinent les deux balises d’ouverture et de fermeture

Exemple : **<br/>** pour le **saut de ligne**

Les **attributs** personnalisent les balises : ils servent par exemple à changer la couleur de la police, ou encore de changer la taille d’une image. Ils servent à mettre en forme un site internet.

Exemple d’attributs génériques, qui s’utilisent avec quasiment toutes les balises :

* **class** et **id**: étudiés plus tard ; associés à p par ex

<p id="italicparagraph">Paragraphe type 1, italique </p>

* **name**: si on donne un nom à un élément, l’élément devient une variable de script pour Javascript, ASP et PHP. S’utilise beaucoup dans les champs de texte interactif comme input :

<input type="text" name="TextField" /> : pas d’effet sur l’apparence de la zone de texte

* **title**: ajoute un titre pop-up à chaque contenu de l’élément : la visualisation apparaît quand on arrête notre souris au-dessus du texte : s’utilise avec la balise <h2> notamment
* **align**: aligne un titre au à droite, gauche ou au centre de la page selon la valeur de l’attribut

Les balises ont des attributs standards (par défaut).

A partir de paragraphe, je saute tout jusqu’à ce qui m’intéresse : les formulaires.

Un **formulaire** est un ensemble de champs de texte

**Champs de texte** : permettent à l’utilisateur d’entrer du texte, ils ont quelques attributs (la balise est **form**)

* type : type de champ de texte, comme texte, envoi ou mot de passe
* name : donne un nom au champ pour s’y référer ultérieurement
* size : taille du champ
* maxlength : valeur max du nombre de caractères pouvant être tapés

(données encryptées = sécurisées)

On veut ajouter le **bouton d’envoi**, qui a l’attribut Send ou Submit et qui doit être en dernier dans le formulaire.

Destination pour les données entrées dans le formulaire ainsi que le mode de transfert doivent être spécifiés à travers les arguments suivants :

* **method**: on utilisera la méthode post, qui envoie le formulaire avec l’info insérée sans envoyer données de l’utilisateur
* **action**: URL où l’information est envoyée

Exemple :

<form method="post" action="mailto:votreemail@email.com">.

Name: <input type="text" size="10" maxlength="40" name="name" /><br />

Password: <input type="password" size="10" maxlength="10" name="password" /><br />

<input type="submit" value="Send" />

</form>

Ici, le **action=« mailto:thibault.latrace@gmail.com »** fait que quand on valide le formulaire, ça ouvre notre boîte de mail avec comme destinataire déjà le [thibault.latrace@gmail.com](mailto:thibault.latrace@gmail.com).

Tuto pour gérer les données une fois qu’elle ont été envoyées par l’utilisateur : <https://developer.mozilla.org/fr/docs/Web/Guide/HTML/Formulaires/Envoyer_et_extraire_les_donn%C3%A9es_des_formulaires>.

Le **action** définit où envoyer l’information saisie par l’utilisateur quand le formulaire est envoyé.

On peut les envoyer à une **URL précise** comme http://foo.com : **<form action="http://foo.com">** ou à une **URL relative** /somewhere\_else : **<form action="/somewhere\_else">**.

S’il n’y a pas d’attribut action, les données sont envoyées à la même page que le formulaire.

On peut utilisé une URL https (=http sécurisé) : les données sont alors chiffrées avec le reste de la requête, même si le form est sur une page http. Mais si formulaire est sur page sécurisée et que URL de action est non sécurisée, navigateur affiche une alerte de sécurité.