

TP 8 – PHP et pages dynamiques

Internet et Outils (IO2)

1 Changement de classes au vol

Les feuilles de style CSS ont normalement un effet *statique* sur les documents XHTML. Mais si l'on combine PHP et CSS, on peut obtenir des effets de style *dynamiques*, en particulier via des modifications des attributs de `class`

Reprenez l'exemple du TP5 `tp5_css.html` ou si vous l'avez perdu http://www.pps.jussieu.fr/~tasson/enseignement/IO2/TP8/tp8_exo1.php et son sommaire "cliquable". Placez le dans votre répertoire `public_html`. Le but des questions suivantes est d'arriver à ce qu'un 'click' sur un élément du sommaire :

- rende rouge cet élément du sommaire
- rend rouge le titre et les écrits de la section correspondante
- continue à entraîner un déplacement vers la section correspondante.

Exercice 1 (Rappels de CSS) — Organisez la page de manière à avoir le sommaire dans une "boîte" en haut, qui reste en haut et le texte dans une "boîte" au milieu avec une barre de déroulement sur le côté.

Quand on clique sur un des liens, on doit toujours voir le sommaire. Astuce : utilisez `position : fixed`.

Centrez le tout, encadrez, et rapprochez-vous autant que faire ce peut du résultat de l'image 1. (Sans y passer la séance entière non plus...)

Exercice 2 (Modification des balises `<a href ...>`) — Modifiez la première balise de la manière suivante :

` Un texte ` devient ` Un texte `. Ouvrez la page résultante dans votre navigateur.

1. Que se passe-t-il au niveau de la page ?
2. Que se passe-t-il au niveau de l'url ?
3. A votre avis, comment aller récupérer la valeur "1" en php ? (Trouvez la réponse en établissant une analogie avec les formulaires).

Exercice 3 (Application) — Pour prouver à votre moniteur préféré que vous avez bien compris de quoi il retournait à la question précédente :

Modifiez les quatre balises `` du sommaire puis faites apparaître sous le sommaire (et donc dans la "boîte" du haut) une ligne : "Vous naviguez actuellement dans la partie numéro X" . Où "X" est égal à un si on est dans la première partie, à 2 si on est dans la deuxième, etc ... Par défaut "X=1"

Exercice 4 (Fonction PHP) — Ecrire une fonction PHP `Rouge($i)` qui renvoie la chaîne de caractères : `class= "red"` si on navigue actuellement dans la i-ème partie et rien sinon.

Exercice 5 (Fin de l'exercice) — Commencez par changer les pseudo-classes de l'élément `a` de manière à ce que les liens s'affichent par défaut toujours en noir. En utilisant la fonction ainsi définie, modifiez votre fichier `.php` et votre `.css` de manière à faire afficher en rouge la partie courante, son titre, et son emplacement dans le sommaire.



FIG. 1 – Résultat exercice 1

2 Styles à la carte

Cette partie est un peu plus difficile car moins détaillée et reprend les thèmes abordés dans la partie 1. Si vous êtes en retard et préférez réviser les formulaires, il peut être intéressant de sauter cette partie de manière temporaire pour aller directement faire la partie 3. Vous y reviendrez si vous avez le temps

Cet exercice vous montre que les fichiers css peuvent radicalement changer l'allure d'une page web. Vous êtes tous familiers avec les "skins" ? Dans cet exercice on vous propose de voir comment les utiliser. Le fichier de départ `tp8_exo2.php` et les trois fichiers `.css` ont été préparés pour vous et vous attendent sous les noms de `exo2style0.css` `exo2style1.css` `exo2style3.css` à l'adresse suivante : [www.pps.jussieu.fr/ tasson/enseignement/TP8](http://www.pps.jussieu.fr/tasson/enseignement/TP8). Téléchargez les et enregistrez les où bon vous semble.

Exercice 6 (Changement de skin) — Utilisez les connaissances acquises dans la partie 1 et vos petites cellules grises pour modifier le fichier php : le but est que lorsque vous cliquez sur l'un des liens vers le fichier css, la page se mette à utiliser le fichier de style `.css` correspondant.

Exercice 7 (Finitions) — N'oubliez pas de modifier la ligne "Le style courant est ..." pour qu'elle affiche effectivement le style courant. Remarque, il faut que par défaut le style courant affiché soit `exo2style0.css`.

3 Etes-vous forts en orthographe ?

Le but de cette section est de faire une page (et quand je dis UNE je pèse mes mots) sur laquelle on trouve trois exercices d'orthographe. Il y a bien sûr comme auparavant un sommaire cliquable qui reste en haut dans un cadre, et les exercices qui apparaissent dans un cadre en dessous. Chaque exercice d'orthographe correspond à un formulaire. Quand on valide ce formulaire, le score (ie le nombre de réponses correctes) doit d'afficher. Les questions suivantes ont pour but de vous guider pas à pas à ce résultat.

Contrairement aux exercices précédents, vous créez cette page de A à Z. (En revanche il est vivement recommandé de réutiliser la feuille de style de la partie 1).

Exercice 8 (Squelette HTML) — Sans vous préoccuper pour l'instant du PHP, créez la page avec le sommaire cliquable et les 3 formulaires comme sur les images suivantes. Vous donnerez comme nom aux balises qui correspondent à des réponses `name = "q1", name = "q2"...` etc jusqu'à `name = "q20"` pour la balise `select` qui correspond au verbe `geindre`

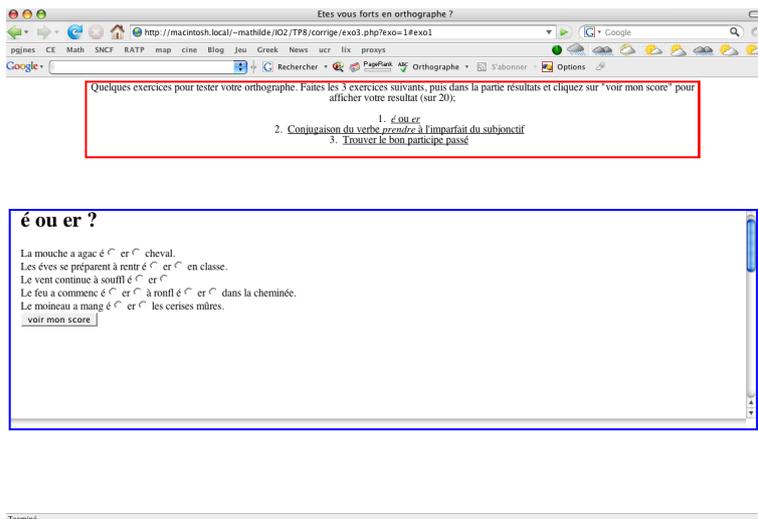


FIG. 2 – Exercice 3 : premier exercice d'orthographe "é ou er"

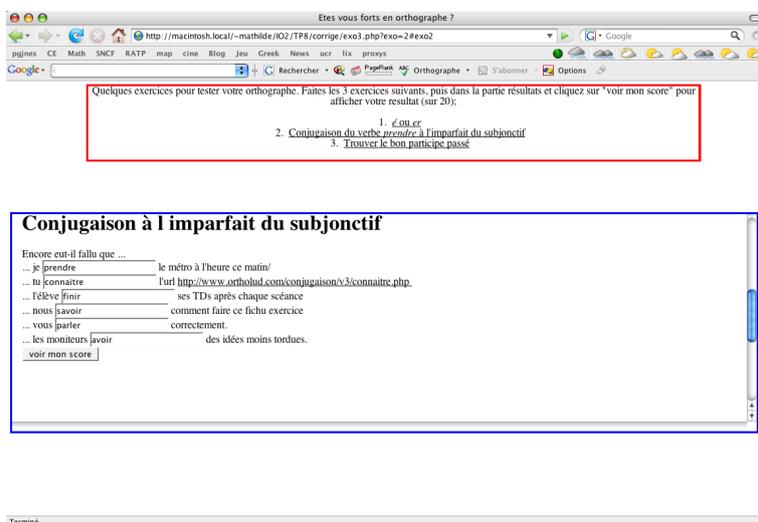


FIG. 3 – Exercice 3 : deuxième exercice d'orthographe "imparfait du subjonctif"

du dernier exercice. Les champs `action` des formulaires auront pour l'instant pour valeur `action="exo3.php"`, c'est à dire qu'ils rappelleront la page elle même. Pour le dernier exercice faites preuve d'imagination pour mettre dans chaque menu déroulant 3 reponses fausses et une reponse juste.

Exercice 9 (Avant de comptabiliser les résultats...) — Pour nous simplifier la vie au moment de comptabiliser les points, nous allons utiliser des boucles `for`. Il s'agit donc d'accéder facilement pour chaque question à la "bonne réponse". Je vous propose pour cela, quand cela est possible (c'est à dire dans les formulaires 1 et 3) de donner `value="1"` aux bonnes réponses et `value="0"` aux mauvaises. Quel est l'inconvénient de cette méthode cependant ?

Pour le deuxième exercice on ne peut clairement pas procéder ainsi. Créez donc une variable-tableau PHP `$resultat2` dans laquelle vous stockerez les bonnes reponses aux questions q7 à q12.

Exercice 10 (Comptabilisation des points) — Pour chaque partie, écrivez un petit script

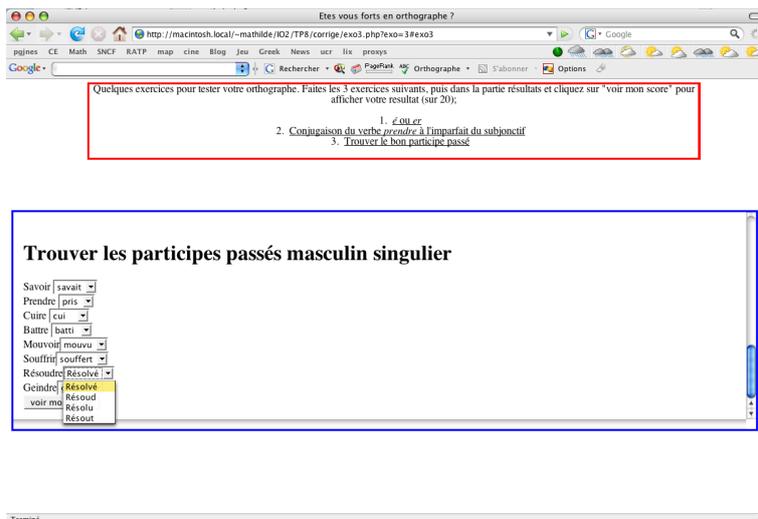


FIG. 4 – Exercice 3 : troisième exercice d'orthographe "passés simples"

Trouver les participes passés masculin singulier

Savoir | savait |
 Prendre | pris |
 Cuire | cui |
 Battre | batti |
 Mouvoir | mouvu |
 Souffrir | souffert |
 Résoudre | Résolu |
 Geindre | geigné |
 voir mon score | votre score est de 8 sur 8

FIG. 5 – Affichage du score

PHP, qui permette de compter les points pour la question. Utilisez impérativement une boucle `for` à chaque fois. Stockez ces points dans une variable `$score`.

Exercice 11 (Affichage des points) — On veut, sans créer de nouvelle page, que pour chaque exercice le script s'exécute si l'on a appuyé sur la touche "voir mon score" (pensez à `isset`), et que le résultat s'affiche sous l'exercice. (Comme sur la figure 5). Que se passe-t-il pour les exercices 2 et 3? Remédiez au problème en allant modifier légèrement la partie `action=...` des formulaires. Note : il est normal que le formulaire se remette sur ses valeurs par défaut.