



W1 - Piscine PHP

W-WEB-024

Rush 1

Premier Rush

Rush 1

repository name: rush1

language: HTML, CSS, JS, PHP



- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.

COMPÉTENCES à ACQUÉRIR

- Création d'un formulaire d'inscription / connexion
- Vérification et sécurisation des user inputs en Javascript et en PHP
- Les basiques de l'authentification

Bienvenue dans ce premier Rush de piscine ! :)

Le temps de ce Rush nous allons mettre de côté les notions de PHP que vous avez pu apprendre jusqu'à aujourd'hui.

Ce rush va vous permettre d'aborder le HTML5, CSS3 et Javascript, langages informatiques que vous devrez maîtriser pour mener à bien la plupart de vos projets de cette année.

Les Rushs étant beaucoup plus libres que les exercices, n'hésitez pas à ajouter des fonctionnalités. Ces fonctionnalités pourront être récompensées par des points bonus. Mais n'oubliez pas qu'avant de penser aux bonus, votre projet devra implémenter les fonctionnalités de base !

CONSIDÉRATIONS

Pour ce rush, vous êtes libre de travailler comme vous l'entendez, à distance, en home office, ou en groupe au même endroit ; en fonction des consignes données par votre équipe pédagogique. Dans tous les cas, vous **DEVEZ** mettre en place une organisation de groupe et une gestion de projet.

DÉROULÉ DE LA SOUTENANCE

- Vous devez être acteur de votre soutenance !
- Vous devez commencer la soutenance en expliquant ce que vous avez mis en place pour réaliser le projet (rétro-planning, découpage, difficultés rencontrées, tâches non terminées, etc) avec un support à l'appui (git, trello, gantt, uml, doc, pdf, etc.) [5 min]
- Vous devez ensuite présenter toutes les fonctionnalités que vous aurez implémenté en suivant le fil conducteur du sujet. [15 min]



Tout ce qui n'est pas montré en soutenance n'est pas noté.

- Enfin il pourra vous être demandé de préciser certains aspects du projet ou d'effectuer un recode de tout ou partie de celui-ci. [5 min]
- Il vous sera enfin fait un retour sur la qualité globale de votre rendu et sur le déroulé de la soutenance. [5 min]



Afin de devenir de bons développeurs il vous faut être rigoureux et méthodique. Cela s'applique aussi à la qualité du code que vous allez produire. Pour cela, vous devez respecter des normes de code ! Pour le PHP, par exemple, vous devez respecter la norme **PSR-12**.



Vous pouvez vérifier la norme de vos projets PHP avec PHP_CodeSniffer : https://github.com/squizlabs/PHP_CodeSniffer



Vous pouvez vérifier la norme de vos pages HTML sur le site du W3C : <https://validator.w3.org/>



Toute erreur de norme entraînera un malus conséquent sur votre note finale.



Toute erreur d'un script PHP, ou JS, sera sanctionné par des points malus.

ETAPE 1

Rendu : rush1_step1/

Restrictions :

- Aucun attribut de style dans le HTML, si besoin, il faut un fichier CSS à part.
- Aucun code JavaScript dans le HTML, si besoin, il faut un fichier JS à part.
- **Tout doit être valide W3C !**

Réalisez une page HTML nommée "my_form.html" qui sera **valide W3C**. Elle comportera un formulaire de type POST où l'action se fera sur cette même page.

Les champs du formulaire seront :

- **sexe** de type radio, avec comme valeurs Homme, Femme OU Autre
- **civilite** de type select, avec comme valeurs M. ou Mme.
- **nom** de type texte, avec au moins 2 caractères alphabétiques
- **email** de type email
- **telephone** de type tel
- **website** de type url
- **date de naissance** de type date, au format dd/mm/yyyy,
- **hobbies** de type checkbox, avec comme valeurs possibles : Jeux video, Cinema, Lecture, Sport et Informatique
- **token** de type hidden, avec pour valeur my first website
- **validation** de type submit

Chaque champ aura son label associé.



ETAPE 2

Rendu : rush1_step2/

Restrictions :

- Aucun attribut de style dans le HTML, si besoin, il faut un fichier CSS à part.
- Aucun code JavaScript dans le HTML, si besoin, il faut un fichier JS à part.
- **Tout doit être valide W3C !**

Reprenez les pages créées dans la première étape et intégrez une feuille de style nommée `my_style.css` qui sera **valide W3C**.

Vous devrez faire en sorte que :

- La taille de la police (hors titres) ait une taille de 12 pixels
- Un champ sur 3 doit avoir une bordure de 2 pixels en pointillés de couleur bleue
- Tous les labels seront alignés avec une longueur de 150 pixels
- Chaque champ aura son label à sa gauche et sera cliquable (pour mettre le focus)
- Quand le curseur passe au-dessus d'un bouton ou d'un champ, une animation se produit
- Le bouton de validation aura une ombre verticale noire et des bordures arrondies de 5 pixels



ETAPE 3

Rendu : rush1_step3/

Restrictions :

- Aucun attribut de style dans le HTML, si besoin, il faut un fichier CSS à part.
- Aucun code JavaScript dans le HTML, si besoin, il faut un fichier JS à part.
- **Tout doit être valide W3C !**

Reprenez les pages créées dans les étapes précédentes et intégrez un script Javascript nommé `my_script.js`. Ce-dernier comportera une fonction de validation de votre formulaire. Elle sera appelée à chaque fois que l'on cliquera sur le bouton de validation, et autorisera ou non l'envoi du formulaire.

De plus, ce script devra :

- Colorer le fond des champs non remplis en rouge et afficher les erreurs de façon précise.

BONUS 1

Rendu : rush1_bonus/

Restrictions :

- Aucun attribut de style dans le HTML, si besoin, il faut un fichier CSS à part.
- Aucun code JavaScript dans le HTML, si besoin, il faut un fichier JS à part.
- Tout doit être valide W3C !

Réalisez une page HTML5 nommée `my_navigator.php` qui sera **valide W3C**. Cette page devra être accessible une fois l'utilisateur connecté. Vous devez donc implémenter un formulaire de connexion vérifiant le login/mail et le mot de passe de l'utilisateur voulant se connecter.

Vérifiez sans recharger la page (`#ajax`) que tous les champs sont remplis correctement et que l'utilisateur qui essaye de se connecter existe bien. Autrement, signalez une erreur à l'aide de Javascript/CSS, toujours sans actualiser la page.

Vous devez faire en sorte qu'il existe au moins 3 utilisateurs qui puissent se connecter durant votre sou-tenance.

BONUS 2

Rendu : rush1_bonus/

Restrictions :

- Aucun attribut de style dans le HTML, si besoin, il faut un fichier CSS à part.
- Aucun code JavaScript dans le HTML, si besoin, il faut un fichier JS à part.
- **Tout doit être valide W3C !**

Reprenez les pages créées dans les deux étapes précédentes. Vous devrez faire en sorte qu'il soit possible de créer un compte utilisateur lorsque l'on n'est pas connecté. Ce formulaire d'inscription devra implémenter les champs login/mail, password/confirmation.

Si le formulaire est soumis alors qu'un champ n'est pas ou mal rempli, vous devez le signaler à l'utilisateur à l'aide de Javascript et CSS sans recharger la page.

Le login/mail de l'utilisateur connecté devra apparaître dans la page accompagné d'un lien de déconnexion. Il faut aussi un lien pour "supprimer le compte".



BONUS 3

Rendu : rush1_bonus/

Restrictions :

- Aucun attribut de style dans le HTML, si besoin, il faut un fichier CSS à part.
- Aucun code JavaScript dans le HTML, si besoin, il faut un fichier JS à part.
- **Tout doit être valide W3C !**

Idées de Bonus :

- Possibilité d'enregistrement des données
- Vérification des doublons
- Possibilité d'édition des données
- Pré-remplissage des différents champs
- Possibilité d'uploader un avatar dans un champ `avatar` de type file
- Raccourcis/code clavier pour effectuer des test unitaires (debug)
- Une interface utilisateur pratique et bien pensée
- Possibilité de récupérer son mot de passe par mail/question secrète en cas d'oubli
- Possibilité de modifier son mot de passe une fois connecté
- Possibilité d'uploader un avatar et de l'afficher une fois connecté
- Possibilité d'envoyer des messages aux autres utilisateurs
- Sécurisation des fichiers pour chaque utilisateur
- Possibilité de dossiers partagés entre plusieurs utilisateurs
- Possibilité de gestion des droits par utilisateur/groupe
- Une interface utilisateur pratique et bien pensée
- Un historique des actions effectuées
- Full AJAX
- ...