

Secteur Tertiaire Informatique
Filière « Etude et développement »

Séquence « Développer des pages Web »

Programmation objet en JavaScript

Apprentissage

Mise en pratique

Evaluation

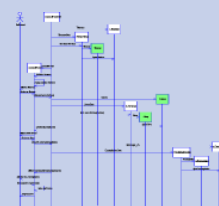
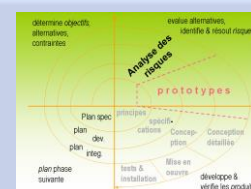
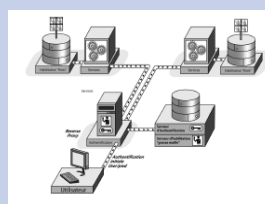


TABLE DES MATIERES

| | |
|---|---|
| Table des matières | 3 |
| 1. Formulaire de saisie | 5 |
| 1.1 Design de la page | 5 |
| 1.2 Programmation des contrôles de saisie avec JQuery | 6 |
| 2. Fonctions | 8 |
| 2.1 Tableaux et objets | 8 |
| 2.2 Classe, Objet et Prototype | 8 |
| 2.2.1 Pour commencer | 8 |
| 2.2.2 Manipulation d'objets instanciés | 9 |
| 2.2.3 Manipulation d'objet littéral | 9 |
| 2.2.4 Manipulation du prototype d'une classe | 9 |

Préambule

Ces exercices permettant de mettre en pratique la programmation objet en JavaScript.

Objectifs

Les deux premiers entraînements concernent la mise en œuvre de la programmation moderne JavaScript/JQuery. Le dernier est un exercice de synthèse à réaliser de 3 manières différentes pour bien illustrer le procédural et l'orienté objet en JavaScript.

Méthodologie

Réalisez les exercices au fur et à mesure de l'étude du support d'apprentissage [A-programmation-objet-javascript.pdf](#) et reportez-vous à ce document pour trouver les syntaxes et comprendre les principes mis en œuvre.

1. FORMULAIRE DE SAISIE

A partir des extraits de code donnés dans le chapitre 1 du support d'apprentissage `A-programmation-objet-javascript.pdf`, réalisez cette mini-application en suivant les consignes ci-dessous.

1.1 DESIGN DE LA PAGE

- Réalisez tout d'abord la présentation HTML du formulaire sans vous préoccuper du code JavaScript (phase de maquettage) ;
- Le formulaire n'a pas forcément d'attribut `action` renseigné (il se rappellera alors lui-même automatiquement) ; prévoyez les habituels boutons `submit` et `reset` ;
- La saisie se fera systématiquement dans des zones de texte ;
- Il peut être judicieux de prévoir des styles CSS (ou même une feuille de styles séparée) pour pouvoir gérer l'affichage ou le masquage de la zone de message (`<div>` identifiée 'erreur' dans l'exemple) ; faites simple pour le rendu graphique : le but n'est pas ici de s'entraîner sur les styles CSS et la présentation standard HTML peut suffire ;

Exemple de design :

Petit formulaire

Pseudonyme

Mot de passe

Confirmation

E-mail

Exemple d'affichage en cas d'erreur :

Petit formulaire

Vous n'avez pas rempli correctement les champs du formulaire !


Pseudonyme

Mot de passe

Confirmation

E-mail

Message de la page Web

 Mots de passe différents
Email obligatoire

OK

Mise en pratique : Programmation objet en JavaScript

Afpa © 2016 – Section Tertiaire Informatique – Filière « Etude et développement »

1.2 PROGRAMMATION DES CONTROLES DE SAISIE AVEC JQUERY

Réalisez cette version de contrôle de saisie du formulaire par JavaScript en utilisant JQuery : testez le contenu à chaque frappe au clavier pour affichez la bordure en vert quand tout va bien et en rouge en cas d'erreur. Bien entendu, un contrôle global sera aussi réalisé lors de la soumission du formulaire.

- Insérez une balise `<script>` en fin de partie `body` de manière à appeler 2 scripts JavaScript externes, l'un pour le framework JQuery, l'autre pour votre propre script ;
- Implémentez une fonction anonyme par la méthode JQuery `.ready()` de l'objet `document` de manière à récupérer par la fonction JQuery `$()` les références des différentes zones de saisie dans des variables afin de leur implémenter les fonctions anonymes de contrôle dynamique en cours de frappe pour changer la couleur de bordure (vert = OK, rouge = erreur - utilisez la méthode JQuery `.css()`) :
 - Testez à la volée que le deuxième mot de passe est bien identique au premier ;
 - Ajoutez une même vérification à la volée pour toutes les zones de saisie : 5 caractères au minimum ;
- Implémentez par JQuery une fonction anonyme, associée à l'événement `click` du bouton `submit`, qui réalisera tous les tests en série comme pour la première version ; en cas d'erreur, il faudra neutraliser l'action HTML par défaut de ce bouton ;
- Modularisez le code JavaScript en réalisant une sous-fonction, nommée, assurant le contrôle élémentaire d'une zone de saisie passée en paramètre de manière à changer sa couleur de bordure et à faire apparaître la zone d'erreur si nécessaire ; cette sous-fonction sera appelée en série par la fonction de contrôle général associée au bouton d'envoi ;
- Terminez en écrivant une nouvelle fonction associée à l'événement `click` du bouton `reset` de manière à rétablir les couleurs de bordure des zones de saisie et faire disparaître la zone d'erreur (l'effacement des zones de saisie sera laissé aux bons soins de HTML grâce à l'action par défaut de ce bouton `reset`) ;
- Effectuez tests et mise au point en usant et abusant du débogueur du navigateur.

Exemples de rendus graphiques :

Petit formulaire

Pseudonyme

Mot de passe

Confirmation

E-mail

en cours de frappe

Petit formulaire

Vous n'avez pas rempli correctement les champs du formulaire !

Pseudonyme

Mot de passe

Confirmation

E-mail

après validation

2. FONCTIONS

Il s'agit simplement ici de mettre en œuvre quelques techniques JavaScript/JQuery au sein de scripts qui n'afficheront rien en page HTML mais produiront des messages sur la console de débogage du navigateur. Il sera donc nécessaire d'activer le débogueur pour voir le résultat des traitements.

2.1 TABLEAUX ET OBJETS

On veut tout d'abord représenter en JavaScript notre héros préféré, 'Bob Morane', de 3 manières différentes :

- Ecrivez une page HTML vide, support du script, appelant en fin de `body` le script JQuery et celui de l'exercice ;
- Implémentez par JQuery une fonction anonyme quand le document est prêt pour créer les 3 structures et en afficher le résultat en console de débogage :
 - Créez un tableau ordinaire indexé par instantiation d'un objet JavaScript `Array`, et contenant simplement 2 valeurs, le nom et le prénom ; afficher le nom dans la console ;
 - Créez un tableau associatif contenant 2 postes 'nom' et 'prenom' avec les valeurs correspondantes ; afficher le nom dans la console ;
 - Créez un objet littéral doté de 2 propriétés 'nom' et 'prenom' affectées des valeurs correspondantes ; afficher le nom dans la console ;
 - Testez en essayant les différentes variantes de notation permettant d'accéder à une valeur d'un tableau/objet JavaScript ; placez des points d'arrêt sur chaque affichage de manière à consulter finement les variables dans le débogueur ;
- Recopiez le code de cette première version et conservez l'original en commentaire ; transformez la construction globale de manière à oublier l'exécution automatique JQuery en réalisant une fonction anonyme auto-exécutable ; testez ;

2.2 CLASSE, OBJET ET PROTOTYPE

2.2.1 Pour commencer

Ecrivez un script permettant de définir une classe JavaScript 'Employe' comportant :

- Une propriété 'nom'
- Une propriété 'prenom'
- Une propriété 'branche'
- Un attribut privé 'salaire'
- Un constructeur d'initialisation qui affecte ces 4 informations
- Un getter sur le 'salaire' (`getSalaire()`)
- Une méthode calculant le salaire annuel (`salaireAnnuel()`)
- Un setter sur le salaire (`setSalaire()`) qui contrôle que le nouveau salaire est bien supérieur au salaire actuel ; en cas d'erreur, il s'agit de lever une exception JavaScript à l'aide de l'instruction `throw` ;

Mise en pratique : Programmation objet en JavaScript

Afpa © 2016 – Section Tertiaire Informatique – Filière « Etude et développement »

Ecrivez un script exploitant cette Classe ; ce script sera enrichi au fur et à mesure de l'évolution de cet exercice. Dans un premier temps, écrivez simplement les instructions JavaScript permettant d'instancier deux objets `Employe`, par exemple `Pierre` et `Paul` en vous inspirant des exemples données dans le support d'apprentissage. Les messages de trace seront simplement affichés en console du débogueur.

Ecrivez une page HTML de support, qui n'affiche rien de particulier mais qui charge les 2 scripts.

Testez en navigateur, débogueur activé pour vérifier votre construction et vos syntaxes ; placez des points d'arrêt pour explorer vos objets.

2.2.2 Manipulation d'objets instanciés

Enrichissez le script permettant d'exploiter votre classe `Employe`.

Les deux objets '`Pierre`' et '`Paul`' sont déjà instanciés. Affichez par de simples `console.log()` leurs `nom`, `prenom`, `branche` et `salaire` ; pour cette dernière information, appelez aussi bien l'attribut que le *getter*. Testez

Ajoutez à l'objet `Pierre` une propriété '`chef`' dont la valeur sera la référence de l'objet `Paul` ; affichez le nom du chef de `Pierre` et le nom du chef de `Paul`. Testez.

Ajoutez une modification du salaire de `Pierre`, en proposant d'abord un salaire inférieur au salaire initial puis un salaire supérieur. Protégez le code en captant les exceptions éventuellement levées. Testez

2.2.3 Manipulation d'objet littéral

Enrichissez encore le script permettant d'exploiter votre classe `Employe`.

Ajoutez la création d'un objet littéral '`Jacques`' comportant 4 propriétés similaires à celles de la classe `Employe`. Affichez les valeurs des propriétés de `Jacques`. Affectez son salaire par un salaire inférieur au salaire initial. Ajoutez à l'objet `Paul` une propriété `chef` dont la valeur est la référence de cet objet `Jacques`. Affichez le nom du chef de `Pierre` et le nom du chef du chef de `Paul`. Testez.

2.2.4 Manipulation du prototype d'une classe

Enrichissez à nouveau le script permettant d'exploiter votre classe `Employe`.

Ajoutez au prototype de la classe `Employe` une nouvelle propriété `sonChef` initialisée à la valeur `null` par défaut. Affectez à `Pierre` la valeur de cette propriété de manière à définir `Paul` comme 'son chef'. De même, affectez le chef de `Paul` comme étant `Jacques`. Affichez les noms de ces nouveaux chefs pour les objets `Pierre`, `Paul` et `Jacques`. Placez un point d'arrêt au premier affichage de manière à pouvoir bien observer les données dans le débogueur du navigateur. Testez à fond et explorez les structures de données dans le débogueur.

CREDITS

ŒUVRE COLLECTIVE DE l'AFPA

Sous le pilotage de la DIIP et du centre d'ingénierie sectoriel Tertiaire-Services

Equipe de conception (IF, formateur, mediatiseur)

B. Hézard – formateur

Stéphane Thomy - formateur

Ch. Perrachon – Ingénieure de formation>

Date de mise à jour : 27/4/16

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droits ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconque. »

Mise en pratique : Programmation objet en JavaScript

Afpa © 2016 – Section Tertiaire Informatique – Filière « Etude et développement »