

Explications :

Pour chaque étape écrire une fonction comme_suit :

```
function etape_1(){  
    var image_list = ...;  
    ...  
}  
// Puis appeller la fonction dans la console "etape_1()"
```

étape 1

- récupérer les images dans un variable nommée *image_list*
- compter le nombre d'image (propriété "*length*")
- écrire une boucle sur la liste d'image et récupérer la valeur de l'attribut "src"

```
var image_list = ...;  
var len = ...;  
for (i = 0; i < len; i++){  
    ...  
}
```

étape 2

- créer une variable *section_list* qui récupère les éléments ayant la classe "section"
- compter le nombre de section
- boucler sur la liste de section et afficher le nombre d'enfants de chaque section dans la console
- boucler sur la liste de section et afficher le texte du titre de second niveau

étape 3

- Créer un événement sur le titre de chaque lien du menu et faire afficher le texte du lien dans la console
- Créer un événement sur le titre de chaque bloc dans la sidebar pour cacher/révéler l'élément "header-body"

étape 4

- Créer une variable contenant les éléments ayant la classe 'aside-block'
- afficher le titre de niveau 3 de chaque bloc trouvé
- créer un tableau comme suit :

```
var array = [];  
  
//pour chaque blocs trouvés, crée un objet littéral comme suit  
  
var o = {  
  "title" : "valeur de .block-header>h3",  
  'content': 'valeur de .block-body>p'  
}  
  
//faire afficher le tableau dans la console
```

étape 5

- compter le nombre de champs texte dans la formulaire
- pour chaque champs texte, ajouter une valeur par défaut.

étape 6

- récupérer la valeur du noeud texte de la première option de la liste déroulante
- sélectionner la première option de la liste

étape 7

- créer un événement “click” sur l’element dont l’attribut “id” est “add-field-btn”. A chaque “click” un champs texte doit être ajouté dans le premier “fieldset”.
- idem pour le bouton “add-checkbox-btn” pour ajouter une nouvelle case à coche

étape 8

- empêcher la soumission du formulaire et vérifier si les champs sont valides
- si le fomulaire est valide, récupérer toutes les valeurs tous les champs du formulaire au format json et les afficher dans la console;

étape 9

- créer un fichier “form_handler.php” qui va récupérer les données du formulaire et les renvoyer au format json.
- afficher l’objet json ainsi renvoyé dans la console
- Traiter le json renvoyé en tant qu’objet js, récupérer la valeur de la clé “field_1” et de la clé “description”, ajouter le fragment suivant dans l’element aside avec les données aisni récupérer :

```
<div class="block aside-block">
  <div class="block-header">
    <h3>{{ valeur field_1}}</h3>
  </div>
  <div class="block-body">
    {{ valeur filed_2 }}
  </div>
</div>
```

étape 10

- ajouter un événement “click” sur l’élément ayant la classe “slideshow” pour faire apparaître chaque slide l’un après l’autre afin de donner l’effet d’un slideshow.

remarques

Utiliser les attributs “data-” : javascript `el.setAttribute('data-index')`

Ainsi que les fonctions de typage et notamment *parseInt*.

Faire une boucle avec “for” ou : `[] .forEach.call(maliste, callback);` `addEventListener`

exemple de code à compléter

```
var o = {};  
o.ui = {};  
  
document.addEventListener("DOMContentLoaded", function() {  
    o.ui.slides = ...;  
    console.log(o.ui.slides);  
    [].forEach.call(o.ui.slides, function(el, i){  
        el.setAttribute('data-index', i);  
        el.????????????('click', function(e){  
            var i = parseInt(this.????????????('data-index'))+1;  
            if(i >= o.ui.slides.length){  
                i = 0;  
            }  
            //cacher image précédente  
            ??????.classList.toggle('slide-visible');  
            //révéler image courante  
            xxxx.classList.toggle('slide-visible');  
        });  
    })  
});
```

étape 11

Il va s’agir de refactoriser le code. Il faut utiliser le patron de conception “module” afin regrouper toutes les fonctions précédemment écrites dans un module.

Voir <http://addyosmani.com/resources/essentialjsdesignpatterns/book/#modulepatternjavascript>