

# Javascript

## TD1 – XHTML

### Exercice 1 : Ranger les oeufs

Dans cet exercice :

- utiliser la fonction `prompt()` pour demander une saisie à l'utilisateur  
<https://developer.mozilla.org/fr/docs/Web/API/Window/prompt>
- utiliser la fonction `console.log()` pour afficher les résultats  
<https://developer.mozilla.org/fr/docs/Web/API/Console/log>

On veut ranger des œufs dans des boîtes de 12 et connaître le nombre de boîtes pleines.

L'algorithme doit demander à l'utilisateur de saisir un nombre d'œufs.

Si le nombre est inférieur à 12, le programme affiche : « Pas assez d'œufs pour faire une boîte ».

Si le nombre est supérieur à 12, il indique le nombre de boîtes pleines et le nombre d'œufs restants.

Par exemple, si le nombre d'œufs est 27, l'algorithme répondra : « 2 boîtes et 3 œufs restants »

### Exercice 2 : Ciblage en JS

Récupérer la base de code située dans le dossier takeoff.zip

Question 1 : Afficher dans la console :

- L'élément ayant l'id « title »
- Les éléments ayant la classe « elegant »
- Les éléments h2
- Les éléments a ayant la classe « more »
- L'élément a situé dans le footer

Question 2 : afficher dans la console

- le texte du h1
- le code html du dernier paragraphe du dernier article
- le nombre d'éléments ayant la classe elegant

Question 3 : récupérer, dans une constante « star », le second li du menu ul. Afficher dans la console :

- son texte

- son l'id
- son parent direct
- le nombre d'éléments enfant
- le contenu html de chacun de ses enfants
- l'id de l'élément précédent
- l'id de l'élément suivant

### Exercice 3 : Jours d'ouverture

Dans cet exercice, on se référera à la documentation suivante :

[https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global\\_Objects/Date](https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Global_Objects/Date)

Créer un fichier html et un fichier js associé. Le fichier html contiendra ces deux éléments

```
<p id="day"></p>
<div id="message"></div>
```

Première partie : Ecrire le code JS qui affiche la date du jour dans la balise p#day

Conseil : chercher et utiliser dans la documentation les fonctions :

- toString()
- toLocaleDateString()

Seconde Partie : Ecrire le code JS qui insère le message « Nous sommes ouvert » ou le message « nous sommes fermé », en fonction du jour de la semaine, dans la balise div#message. Les jours de fermeture sont le samedi et le dimanche

Conseil : utiliser la fonction getDay()

### Exercice 4 : Thème jour/nuit

Créer un fichier html avec un titre « Le jour et la nuit ». Associer un fichier css. Centrer les éléments sur la page.

Ecrire le code Javascript permettant d'afficher la page en mode « jour » ou en mode « nuit », en fonction de l'heure.

Précisions : le mode jour est actif de 8h à 18h uniquement. En mode nuit, les textes sont en clair sur un fond foncé.

Conseils : En Javascript, créer un programme qui ajoute la classe « dark » à l'élément body en fonction de l'heure. Compléter le css, pour que le style de la page soit modifié : écriture claire et fond foncé.

### Exercice 5 : Résultats du loto

Afficher, sur une page html, un nombre entier aléatoire, compris entre 0 et 99. Vous pouvez utiliser la fonction `Math.random()`

### Exercice 6 : Formulaire de login

**Objectif** : Créer un formulaire de login avec un positionnement aléatoire des boutons de saisie des mots de passe.

Connexion à votre compte

Identifiant (10 chiffres)

Mot de passe (6 chiffres)

○ ○ ○ ○ ○ ○

4 7 9 3 0

6 8 2 1 5

Se connecter

**Conseil** : voici un code simple pour récupérer les valeurs aléatoires d'un tableau

```
const numbers = [0,1,2,3,4,5,6,7,8,9]
while(numbers.length){
  const random = Math.floor(Math.random() * numbers.length);
  const number = numbers.splice(random, 1)[0];
  console.log(number);
}
```

**A suivre** : au clic sur un bouton, modification de l'apparence des puces et remplissage du mot de passe...