

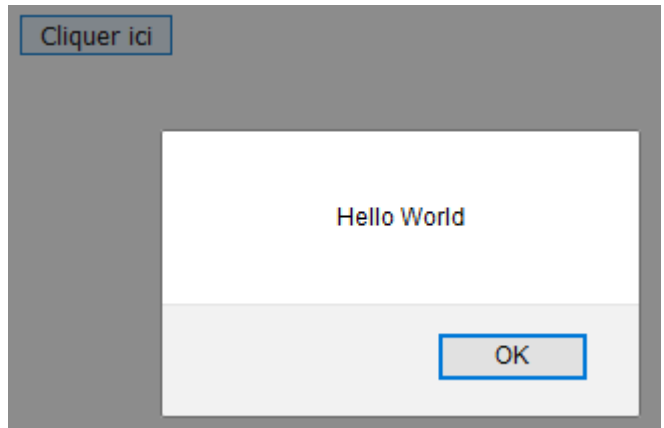
# Exercices gradués de Javascript

## Exercice 1 :

Faire en sorte que lorsqu'on clique sur un bouton d'un document HTML on fasse apparaître un boîte de dialogue modale contenant le texte « Hello World ». Faire appeler une fonction Javascript lors du click sur le bouton et programmer l'apparition de la boîte modale dans la fonction en question.

Voici le code HTML à copier pour commencer l'exercice et à modifier:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>Exercice 1</title>
<script type="text/javascript">
//code Javascript à écrire
</script>
</head>
<body>
<button>Cliquer ici</button>
</body>
</html>
```



## Exercice 2 :

Faire en sorte que lorsqu'on clique sur un bouton d'un document HTML on fasse apparaître le texte « Hello World » dans une balise span avec l'id « contenu ».

Voici le code HTML à copier pour commencer l'exercice et à modifier:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>Exercice 2</title>
<script type="text/javascript">
//code Javascript à écrire
</script>
</head>
<body>
<button>Cliquer ici</button>
<br/>Ici le contenu à afficher:<br/>
<span id="contenu">
<!--faire afficher Hello World ici-->
</span>
</body>
</html>
```

Cliquer ici  
Ici le contenu à afficher:  
Hello World

## Exercice 3 :

Faire en sorte qu'en cliquant sur un bouton la table de multiplication de 5 s'affiche dans une section div d'un document HTML. Ecrire le code Javascript dans un fichier séparé (il faut produire du texte au format HTML avec des retours à la ligne!). L'intérêt réside qu'en appelant ce fichier avec une extension .js l'éditeur de texte aléjoré Notepad++ (ou sblimeText ou équivalent) va permettre de faire de la coloration syntaxique du code Javascript et permettre l'aito-complétion de code. Voilà ce que doit donner l'exercice une fois fini :

Il faudra utiliser la concaténation de chaînes (utiliser le + est le plus pratique pour cela) et convertir les nombres en chaîne (la méthode toString() sur un objet de classe Number le permet)

Table de multiplication de 5 à la suite:

5x1=5  
5x2=10  
5x3=15  
5x4=20  
5x5=25  
5x6=30  
5x7=35  
5x8=40  
5x9=45  
5x10=50

Voici le code HTML à copier pour commencer l'exercice - rien à modifier dans ce code, écrire le fichier .js seulement:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>Exercice 3</title>
<script src="exercice3.js" type="text/javascript"></script>
<!--Ecrire le code Javascript dans un fichier séparé appelé "exercice3.js"-->
<!--Rien à écrire dans le document HTML-->
</head>
<body>
<button onclick="afficher()">Cliquer ici</button><br/><br/>
Table de multiplication de 5 à la suite:<br/>
<div id="table"></div>
</body>
</html>
```

#### Exercice 4 :

Ecrire une fonction produitTable(n) Javascript à qui on passe un entier n en paramètre et qui sort une chaîne de caractère contenant la table de multiplication de cet entier n au format HTML (avec retours à la ligne). Cette fonction renvoie donc un résultat.

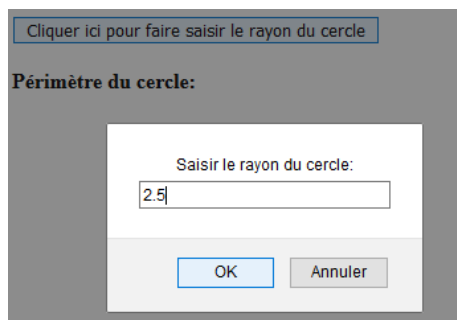
Appeler cette fonction dans une autre fonction afficheLesTables() qui sera elle-même appelée lors du click sur un bouton afin de faire afficher toutes les tables de multiplication de 1 à 10. Cette autre fonction ne renvoie rien (c'est donc une procédure) mais elle produit l'affichage ci-contre en appelant produitTable(n) uniquement.

Voici le code HTML à copier pour commencer l'exercice - rien à modifier dans ce code, écrire le fichier .js seulement:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>Exercice 4</title>
<script src="exercice4.js" type="text/javascript"></script>
<!--Ecrire le code Javascript dans un fichier séparé appelé "exercice4.js"-->
<!--Rien à écrire dans le document HTML-->
</head>
<body>
<button onclick="afficheLesTables()">Cliquer ici</button><br/><br/>
Tables de multiplication des entiers de 1 à 10:<br/><br/>
<div id="tables1a10"></div>
</body>
</html>
```

#### Exercice 5 :

Faire saisir un nombre décimal qui soit le rayon d'un cercle dans une boîte modale et faire afficher le périmètre du cercle ayant cette mesure de rayon dans une balise span d'identifiant « perimetre »



Cliquer ici

Tables de multiplication des entiers de 1 à 10:

Table de 1:

1x1=1  
1x2=2  
1x3=3  
1x4=4  
1x5=5  
1x6=6  
1x7=7  
1x8=8  
1x9=9  
1x10=10

Table de 2:

2x1=2  
2x2=4  
2x3=6  
2x4=8  
2x5=10  
2x6=12  
2x7=14  
2x8=16  
2x9=18  
2x10=20

Table de 3:

3x1=3

Cliquer ici pour faire saisir le rayon du cercle

Périmètre du cercle:

15.707963267948966

Note : on accède à la valeur de  $\pi$  par la propriété Math.PI

Rappel : parseFloat(chaine) permet de convertir en nombre un texte formaté au format nombre décimal.

prompt(message) permet de faire une saisie par boîte modale

Voici le code HTML à copier pour commencer l'exercice - rien à modifier dans ce code, écrire le fichier .js seulement:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>Exercice 5</title>
<script src="exercice5.js" type="text/javascript"></script>
<!--Ecrire le code Javascript dans un fichier séparé appelé "exercice5.js"-->
<!--Rien à écrire dans le document HTML-->
</head>
<body>
<button onclick="calculPerimetre()">Cliquer ici pour faire saisir le rayon du cercle</button><br/><br/>
<strong>Périmètre du cercle:</strong><br/>
<span id="perimetre"></span>
</body>
</html>
```

### Exercice 6 :

Ecrire une fonction Javascript encodeCesar(chaine) à qui on passe comme paramètre une chaîne de caractère et qui renvoie une chaîne contenant le codage César+1 (càd décalage de 1 de chacun des caractères de la page de code UTF16).

Créer une balise span d'identifiant « source » et une autre d'identifiant « cible »

Puis faire en sorte qu'avec un click sur un bouton on fasse saisir par une boite modale le texte à encoder et qu'on affiche ce texte dans « source » puis dans « cible » le codage César+1 du texte en appelant la fonction Javascript précédente par une fonction-procédure encodeCesar()

Contrainte : on veut que les caractères spéciaux suivants ne soient pas encodés par le codage César (Utiliser un tableau pour cela, contenant cette liste de caractères spéciaux) :

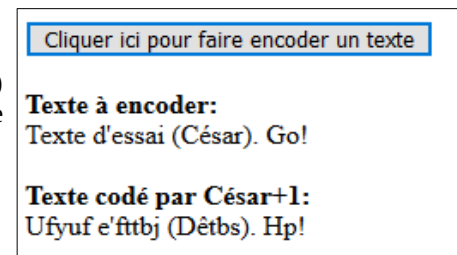
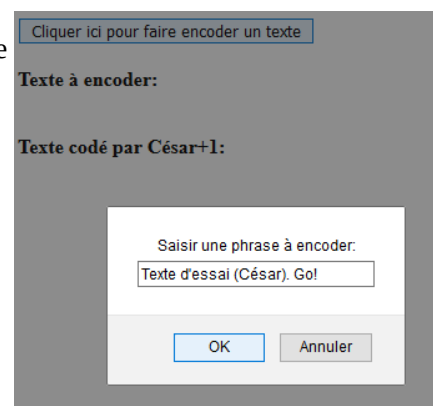
espace	,	;	(	)	.	?	!	'	"	:
--------	---	---	---	---	---	---	---	---	---	---

#### Rappels :

**chaine.charAt(i)** : caractère en position n°i dans la chaîne (entier de 0 à chaine.length-1)  
**chaine.charCodeAt(i)** : code UTF16 du caractère en position n°i dans la chaîne (entier de 0 à chaine.length-1)

**String.fromCharCode(code)** : caractère correspondant au code UTF16

**tableau.indexOf(element)** : indique la position de l'élément dans le tableau (-1 si pas trouvé)



Voici le code HTML à copier pour commencer l'exercice - rien à modifier dans ce code, écrire le fichier .js seulement:

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8"/>
<title>Exercice 6</title>
<script src="exercice6.js" type="text/javascript"></script>
<!--Ecrire le code Javascript dans un fichier séparé appelé "exercice6.js"-->
<!--Rien à écrire dans le document HTML-->
</head>
<body>
<button onclick="codageCesar()">Cliquer ici pour faire encoder un texte</button><br/><br/>
<strong>Texte à encoder:</strong><br/>
<span id="source"></span><br/><br/>
<strong>Texte codé par César+1:</strong><br/>
<span id="cible"></span>
</body>
</html>
```