

1 OBJECTIFS

Dans cette exploration, contrairement à d'habitude, vous serez en autonomie.

Voici donc vos objectifs :

- Être capable de suivre des consignes écrites (lire correctement)
- Comprendre les éléments employés
- Écrire des codes JavaScript qui respectent un cahier des charges
- Savoir déboguer vos codes (messages d'erreurs, etc.)
- Utiliser le cryptpad <https://cryptpad.fr/sheet/#/2/sheet/edit/DaW5m5cuziYbS9LsSlzey+05/>

2 PHASE 1 – PRÉPARATION

2.1 CRÉATION DU DOSSIER ET DES FICHIERS

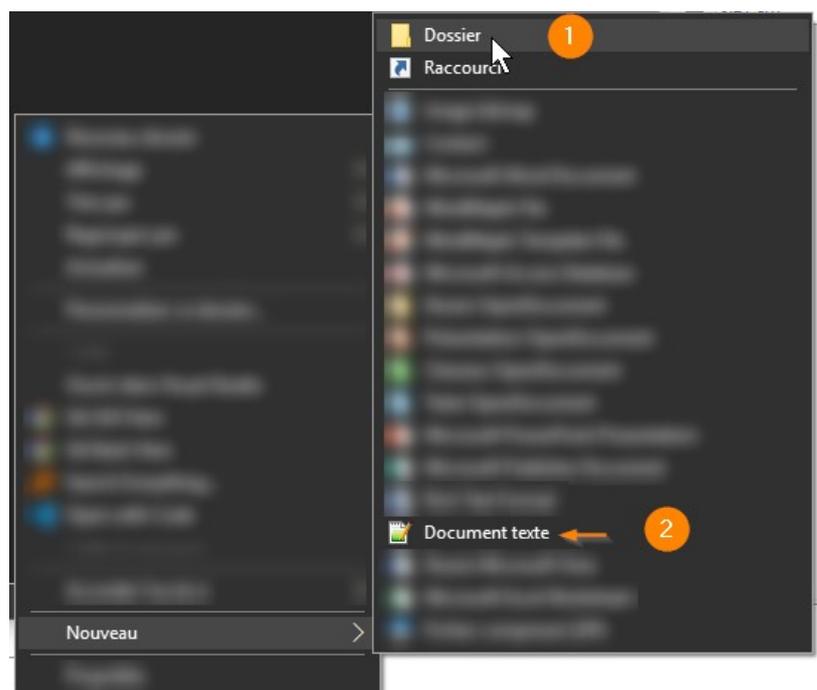
Pour commencer cette exploration, vous allez préparer votre environnement en création un dossier et deux fichiers :

- Créer un dossier "JavaScript Boucles"
- Créer un fichier "NOM_Prenom_TP_boucles.html" (remplacez Nom et prénom par les vôtres)
- Créer un fichier "NOM_Prenom_style.css" (remplacer Nom et Prénom par les vôtres)

Démarche : Ouvrir l'explorateur de fichier, dans votre dossier de travail, faire un clic droit et dans le menu qui apparaît, sélectionner la bonne action.

Dans le cas de la création du document texte, il faut ensuite renommer le fichier **et** son extension (un avertissement s'affiche qu'il convient d'ignorer).

Attention : évitez les accents dans les nommages de fichiers.





2.2 ÉDITION DU FICHIER CSS

Nous allons créer un style CSS pour afficher nos tableaux et nos cellules de façon claire. Nous en profiterons pour éditer le style des balises H1 pour commencer à apprendre CSS.

Ouvrir le fichier "**NOM_Prenom_style.css**" dans votre éditeur de texte.

Coller les instructions suivantes :

```
table {  
    border:1px solid black;  
}  
  
td {  
    border:1px solid red;  
}  
  
.couleur {  
    color: OrangeRed;  
}
```

Pour la balise <table> nous activons une bordure de 1 pixel, ligne pleine et de couleur noire.

Pour la balise <td> nous activons une bordure de 1 pixel, ligne pleine mais de couleur rouge.

Enfin, nous créons une classe dont l'écriture cela une couleur Orange-Rouge.

Sauvegarder le fichier.

2.3 ÉDITION DU FICHIER HTML

Ouvrir le fichier "**NOM_Prenom_TP_boucles.html**" dans votre éditeur de texte.

Coller votre modèle de page HTML habituel, il doit notamment contenir (aux bons endroits) :

- <title>NOM boucles Javascript</title>
- <meta charset="UTF-8">
- <link rel="stylesheet" type="text/css" href="NOM_Prenom_style.css" />
- <script language="JavaScript"> et la balise fermante </script>

Dans le bloc BODY ajouter le code suivant :

```
<body>  
    <h1>A table !</h1>  
    <table>  
        <tr><td>Cellule 1</td><td>Cellule 2</td></tr>  
        <tr><td>Cellule 3</td><td>Cellule 4</td></tr>  
    </table>  
    <h1>Table en bois de Javascript</h1>  
  
    <script language="JavaScript">
```



Sauvegarder votre fichier et tester l'affichage.

[i](#) Modifier votre avancement dans le tableau Cryptpad.

3 PHASE 2 – CRÉATION D'UN TABLEAU SIMPLE

Vous allez maintenant créer un tableau, mais en JavaScript.

3.1 TABLEAU À UNE DIMENSION

À l'intérieur des balises `<script>` il faut insérer une boucle pour générer des cellules automatiquement.

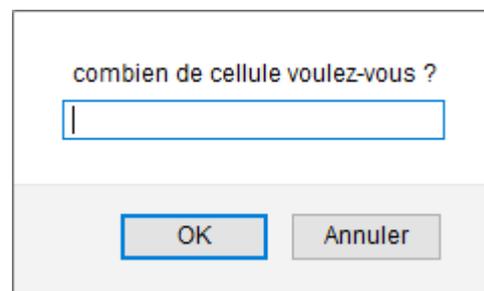
Voici le code à produire :

```
// exécution du code
let nbCellule = window.prompt("combien de cellule voulez-vous ?")
let tableau='<table><tr>'
// boucle while décroissante
do {
    tableau = tableau + '<td>' + "Cellule n°" + nbCellule + '</td>'
    nbCellule--
} while(nbCellule > 0)
tableau = tableau + "</tr></table>"
document.write(tableau)
```

Notez la présence d'une fenêtre de dialogue (bloquante) qui demande le nombre de cellule.

Cette boîte est appelée par l'instruction **window.prompt()**

On récupère la réponse dans une variable, ici *nbCellule*.



La ligne de code **nbCellule--** remplace le code **nbCellule = nbCellule - 1**. Vous rencontrerez souvent ce genre de code simplifié (notamment **variable++** qui est équivalent à **variable = variable + 1**)

Si tout fonctionne correctement, vous obtiendrez l'affichage suivant (avec une valeur de 8) :



A table !

Cellule 1	Cellule 2
Cellule 3	Cellule 4

Table en bois de Javascript

Cellule n°8	Cellule n°7	Cellule n°6	Cellule n°5	Cellule n°4	Cellule n°3	Cellule n°2	Cellule n°1
-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------

i Modifier votre avancement dans le tableau Cryptpad.

3.2 UTILISATION D'UNE FONCTION

Comme nous aurons besoin de créer des cellules facilement, le mieux est de programmer une fonction. Elle accepte un nombre de cellule à créer et elle renvoie une ligne.

```
<tr>
  <td> contenu </td>   <td> contenu </td>   <td> contenu </td>   <td> contenu </td>
</tr>
```

Dans un premier temps, essayez de réfléchir au code que vous pourriez écrire, puis dans un deuxième temps, comparez-le avec le code suivant (à ajouter au fichier HTML, sous la balise <script>) :

```
// déclaration de fonctions
function DessinerCellules(nombre) {
  let Resultat=""
  for (cellule = 0; cellule < nombre; cellule++) {
    Resultat = Resultat + "<td>Cellule n°" + cellule + "</td>"
  }
  return Resultat
}
```

il suffit maintenant de remplacer la boucle existante (entre les lignes en gris très clair ci-dessous)...

```
let tableau='<table><tr>'
// boucle while décroissante
do {
  tableau = tableau + '<td>' + "Cellule n°" + nbCellule + '</td>'
  nbCellule--
} while (nbCellule > 0)
tableau = tableau + "</tr></table>"
```

... par



```
let tableau='<table><tr>'
// appel de la fonction
tableau = tableau + DessinerCellules(nbCellule)
tableau = tableau + "</tr></table>"
```

Sauvegardez votre fichier et rafraîchissez le navigateur. Si tout fonctionne correctement, vous obtiendrez l'affichage suivant (avec une valeur de 13) :

Table en bois de Javascript

Cellule n°0	Cellule n°1	Cellule n°2	Cellule n°3	Cellule n°4	Cellule n°5	Cellule n°6	Cellule n°7	Cellule n°8	Cellule n°9	Cellule n°10	Cellule n°11	Cellule n°12
-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	-------------	--------------	--------------	--------------

[i Modifier votre avancement dans le tableau Cryptpad.](#)

3.3 COLORISATION D'UNE CELLULE SUR DEUX

L'objectif de ce paragraphe est de montrer comment introduire un changement simple : une cellule sur deux aura une couleur différente.

Il faut introduire une condition, mais il existe plusieurs solutions. Réfléchissez à la solution que vous pourriez écrire ; comment faire pour basculer une valeur à chaque passage dans la boucle ?

Solutions :

- La première solution est mathématique : en utilisant une division entière, il est possible de vérifier si la division décimale par 2 est égale à la division entière par 2... quand les quotients sont égaux, c'est que le nombre est divisible par 2.

Dividende	263	15	Diviseur
	113	17	Quotient
Reste	8		

- La seconde solution est également mathématique : plutôt que d'obtenir le quotient, il suffit de vérifier que le reste de la division est égale à 0. On peut aussi utiliser une fonction modulo pour cela.
- La troisième solution est logique : il suffit d'utiliser une bascule de type variable `!=variable` et si la variable est égale à 0 appliquer une couleur, dans l'autre cas, ne rien faire. Cette solution reste plus limitée que la précédente.

Il existe peut-être d'autres solutions que nous n'aborderons pas. Pour le moment, il est nécessaire d'utiliser une classe CSS pour permettre ce changement : c'est la classe 'couleur' déjà présente dans le fichier CSS.

Dans la fonction `DessinerCellules(nombre)`, utilisez l'une des solutions suivantes :



3.3.1 modulo à zéro

```
for (cellule = 0; cellule < nombre; cellule++) {
  if (cellule % 2) {
    Resultat = Resultat + "<td>case n°" + cellule + "</td>"
  } else {
    Resultat = Resultat + "<td class='couleur'>case n°" + cellule + "</td>"
  }
}
```

3.3.2 bascule variable drapeau (levé ou baissé)

```
let drapeau=0
for (cellule = 0; cellule < nombre; cellule++) {
  drapeau = !drapeau
  if (drapeau) {
    Resultat = Resultat + "<td>case n°" + cellule + "</td>"
  } else {
    Resultat = Resultat + "<td class='couleur'>case n°" + cellule + "</td>"
  }
}
```

Notez la déclaration de la variable 'drapeau' en dehors (et avant) la boucle for.

Notez également qu'avec cette négation, la variable drapeau ne peut prendre que deux états (vrai ou faux), ce qui ne conviendra pas si on veut changer de couleur une fois sur 10, par exemple.

3.3.3 Vérification

Vous devriez avoir un affichage similaire à ceci :

A table !

Cellule 1	Cellule 2
Cellule 3	Cellule 4

Table en bois de Javascript

case n°0	case n°1	case n°2	case n°3	case n°4	case n°5	case n°6
----------	----------	----------	----------	----------	----------	----------

[Modifier votre avancement dans le tableau Cryptpad.](#)



Travail supplémentaire : essayez d'afficher une case sur cinq avec une couleur différente.

case n°0 case n°1 case n°2 case n°3 case n°4 case n°5 case n°6 case n°7 case n°8 case n°9 case n°10 case n°11 case n°12 case n°13 case n°14 case n°15

i Modifier votre avancement dans le tableau Cryptpad.

4 PHASE 3 – CRÉATION D'UN TABLEAU À DEUX DIMENSIONS

Le procédé est simple, il suffit de créer une boucle autour de la fonction dessinerCellules().

4.1 TABLEAU À DEUX DIMENSIONS

Faites une boucle de 10 lignes en modifiant votre code, sans regarder la solution.

L'affichage devrait ressembler à ceci (avec un nombre de cellule à 6) :

case n°0	case n°1	case n°2	case n°3	case n°4	case n°5
case n°0	case n°1	case n°2	case n°3	case n°4	case n°5
case n°0	case n°1	case n°2	case n°3	case n°4	case n°5
case n°0	case n°1	case n°2	case n°3	case n°4	case n°5
case n°0	case n°1	case n°2	case n°3	case n°4	case n°5
case n°0	case n°1	case n°2	case n°3	case n°4	case n°5
case n°0	case n°1	case n°2	case n°3	case n°4	case n°5
case n°0	case n°1	case n°2	case n°3	case n°4	case n°5
case n°0	case n°1	case n°2	case n°3	case n°4	case n°5
case n°0	case n°1	case n°2	case n°3	case n°4	case n°5

Lorsque vous pensez avoir terminé, comparez votre code, au code ci-dessous.

```
let tableau='<table>'
for (t=0; t < 10; t++) {
    tableau = tableau + '<tr>' + DessinerCellules(nbCellule) + '</tr>'
}
tableau = tableau + "</table>"
```

i Modifier votre avancement dans le tableau Cryptpad (seul).

i Modifier votre avancement dans le tableau Cryptpad (avec solution).



4.2 TABLE DE MULTIPLICATION

Il serait pratique de créer une table en deux dimensions, qui soit utile (et dont le contenu ne soit pas figé).

Il faut donc ajouter :

- un titre (avec balise H1) sous la précédente table
- une table de multiplication 10 colonnes x 12 lignes (pourquoi se limiter 10 x 10 = 100 ? allons jusqu'à 120)
- une fonction avec deux boucles imbriquées

4.2.1 création de la fonction multiplication

La première difficulté est d'afficher la table verticalement, alors que nous serons obligés (format de tableau HTML se construisant par ligne) de faire notre boucle intérieure en ligne.

Table de 1	Table de 2
1 x 1 = 1	2 x 1 = 2
1 x 2 = 2	2 x 2 = 4
1 x 3 = 3	2 x 3 = 6
...	

L'idée est donc de créer une fonction qui accepte en paramètre la ligne. Si le numéro de la ligne est 0, on en profitera pour écrire le titre "Table de X" avec X variant de 1 à 10

Voici la fonction, à placer juste sous la précédente fonction :

```
function multiplication(ligne) {
  let resultat = ""
  let texte = ""
  for (y=1; y < 11; y++) {
    if (ligne == 0) {
      // ligne 0 : on affiche le titre
      resultat = resultat + '<td><strong>'
      resultat = resultat + "Table de " + y + '</strong></td>'
    } else {
      // autres lignes : on affiche le calcul
      texte = y + " x " + ligne + " = " + ligne*y
      resultat = resultat + '<td>' + texte + '</td>'
    }
  }
  return resultat
}
```



On peut tester la fonction en ajoutant après le dernier `document.write()` le code suivant :

```
document.write(tableau)
document.write('<table><tr>'+multiplication(5)+'</tr></table>')
```

le résultat doit s'afficher coller à la table précédente :

1 x 5 = 5	2 x 5 = 10	3 x 5 = 15	4 x 5 = 20	5 x 5 = 25	6 x 5 = 30	7 x 5 = 35	8 x 5 = 40	9 x 5 = 45	10 x 5 = 50
-----------	------------	------------	------------	------------	------------	------------	------------	------------	-------------

i Modifier votre avancement dans le tableau Cryptpad (seul).

4.2.2 Création de la table complète

Il suffit maintenant de remplacer la ligne `document.write()` ajouté juste au-dessus par une boucle de 0 à 12 (0 permettra d'afficher la ligne de titre).

Réfléchissez et proposer un code avant de découvrir le code de la solution.

Solution :

```
document.write("<h1>Table de multiplication</h1>")
tableau='<table>'
for (t=0; t < 13; t++) {
    tableau = tableau + '<tr>' + multiplication(t) + '</tr>'
}
tableau = tableau + '</table>'
document.write(tableau)
```

Le résultat devrait correspondre à ceci :

Table de multiplication

Table de 1	Table de 2	Table de 3	Table de 4	Table de 5	Table de 6	Table de 7	Table de 8	Table de 9	Table de 10
1 x 1 = 1	2 x 1 = 2	3 x 1 = 3	4 x 1 = 4	5 x 1 = 5	6 x 1 = 6	7 x 1 = 7	8 x 1 = 8	9 x 1 = 9	10 x 1 = 10
1 x 2 = 2	2 x 2 = 4	3 x 2 = 6	4 x 2 = 8	5 x 2 = 10	6 x 2 = 12	7 x 2 = 14	8 x 2 = 16	9 x 2 = 18	10 x 2 = 20
1 x 3 = 3	2 x 3 = 6	3 x 3 = 9	4 x 3 = 12	5 x 3 = 15	6 x 3 = 18	7 x 3 = 21	8 x 3 = 24	9 x 3 = 27	10 x 3 = 30
1 x 4 = 4	2 x 4 = 8	3 x 4 = 12	4 x 4 = 16	5 x 4 = 20	6 x 4 = 24	7 x 4 = 28	8 x 4 = 32	9 x 4 = 36	10 x 4 = 40
1 x 5 = 5	2 x 5 = 10	3 x 5 = 15	4 x 5 = 20	5 x 5 = 25	6 x 5 = 30	7 x 5 = 35	8 x 5 = 40	9 x 5 = 45	10 x 5 = 50
1 x 6 = 6	2 x 6 = 12	3 x 6 = 18	4 x 6 = 24	5 x 6 = 30	6 x 6 = 36	7 x 6 = 42	8 x 6 = 48	9 x 6 = 54	10 x 6 = 60
1 x 7 = 7	2 x 7 = 14	3 x 7 = 21	4 x 7 = 28	5 x 7 = 35	6 x 7 = 42	7 x 7 = 49	8 x 7 = 56	9 x 7 = 63	10 x 7 = 70
1 x 8 = 8	2 x 8 = 16	3 x 8 = 24	4 x 8 = 32	5 x 8 = 40	6 x 8 = 48	7 x 8 = 56	8 x 8 = 64	9 x 8 = 72	10 x 8 = 80
1 x 9 = 9	2 x 9 = 18	3 x 9 = 27	4 x 9 = 36	5 x 9 = 45	6 x 9 = 54	7 x 9 = 63	8 x 9 = 72	9 x 9 = 81	10 x 9 = 90
1 x 10 = 10	2 x 10 = 20	3 x 10 = 30	4 x 10 = 40	5 x 10 = 50	6 x 10 = 60	7 x 10 = 70	8 x 10 = 80	9 x 10 = 90	10 x 10 = 100
1 x 11 = 11	2 x 11 = 22	3 x 11 = 33	4 x 11 = 44	5 x 11 = 55	6 x 11 = 66	7 x 11 = 77	8 x 11 = 88	9 x 11 = 99	10 x 11 = 110
1 x 12 = 12	2 x 12 = 24	3 x 12 = 36	4 x 12 = 48	5 x 12 = 60	6 x 12 = 72	7 x 12 = 84	8 x 12 = 96	9 x 12 = 108	10 x 12 = 120

i Modifier votre avancement dans le tableau Cryptpad.