112 Découverte tableaux de variables en JavaScript

1 Objectifs

Dans cette nouvelle exploration, vous allez appliquer les notions vues en cours.

Voici donc vos objectifs :

- Comprendre les tableaux dynamiques en JavaScript
- Déclarer et remplir et parcourir les tableaux
- Observer et raisonner (pour tirer des conclusions)
- Utiliser des tableaux multidimensionnels
- Utiliser cryptad : https://cryptpad.fr/sheet/#/2/sheet/edit/RkFY6wO8qXg-BD47gIyNw0qr/

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 trois fichiers :

- Créer un dossier "B1 Tables JavaScript"
- Créer un fichier "NOM_Prenom_TP_tables.html" (remplacez Nom et prénom par les vôtres)
- Copier le fichier"NOM_Prenom_style.css" de l'exploration précédente
- Créer un fichier "NOM_Prenom_TP_tables.js" (remplacez Nom et prénom évidemment)

Sauvegarder les fichiers.

2.2 Édition du fichier HTML

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

```
<!DOCTYPE html>
<html lang="fr">
	<head>
		<meta charset="UTF-8">
		<title>Tables</title>
		<meta name="viewport" content="width=device-width, initial-
scale=1.0">
		<link rel="stylesheet" type="text/css" href="styles.css" />
	</head>
	<body>
		<script src="NOM_Prenom_TP_tables.js"></script>
	</body>
</html>
```

Créez le fichier "**NOM_Prenom_TP_tables.js**" dans votre éditeur de texte.

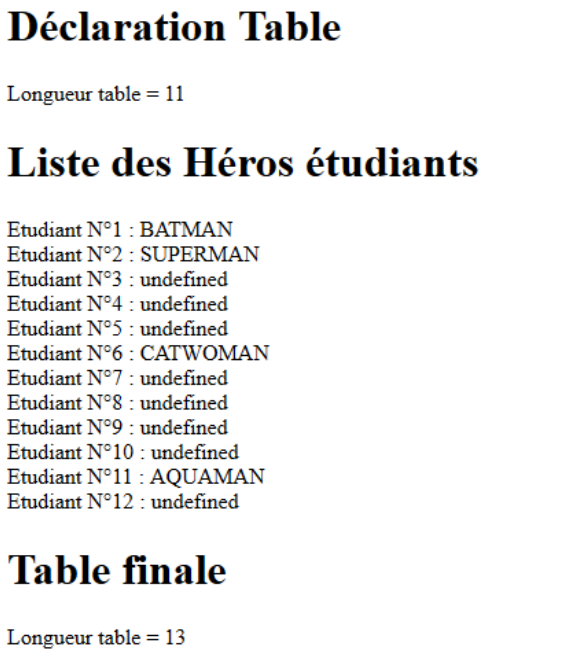
```
"use strict"     // force les déclarations de variables

// déclaration de fonction
function afficherTableau() {
      let res=""
      for (let idx=0; idx<12; idx++) {
            res = res+"Etudiant N°"+(idx+1)+": "+listeHeros[idx]+"<br>"
      }
      return res
}

// création d'un tableau de 10 cellules
document.write("<H1>Déclaration Table</H1>")
let listeHeros = new Array(11)
document.write("Longueur table = "+listeHeros.length)

// remplissage de certaines cellules
listeHeros[0] = "BATMAN"
listeHeros[1] = "SUPERMAN"
listeHeros[5] = "CATWOMAN"
listeHeros[10] = "AQUAMAN"
document.write("<H1>Liste des Héros étudiants</H1>")
document.write(afficherTableau())
listeHeros[12] = "PUSHMAN"
document.write("<H1>Table finale</H1>")
document.write("Longueur table = "+listeHeros.length)
```

Enregistrez vos deux documents et Affichez le document **NOM_Prenom_TP_tables.html** : vous devriez obtenir un affichage similaire :



Modifier votre avancement dans le tableau Cryptpad.

2.2.1 Questionnement

Observez la longueur de la table au début du script, puis à la fin du script.

Est-ce normal ? Pourquoi ?

2.2.2 Deuxième méthode de déclaration

Dans le script JavaScript, remplacez la ligne suivante :

```
let listeHeros = new Array(11)
let listeHeros = [11]
```

Il s'agit d'une autre méthode pour créer un tableau.

Observez la longueur de la table au début du script, puis à la fin du script.

Que pouvez-vous dire par rapport au fonctionnement précédent ?

2.2.3 Gestion automatique de longueur

Dans le script JavaScript, remplacez la ligne suivante :

```
for (let idx=0; idx<12; idx++) {
for (let idx=0; idx<listeHeros.length; idx++) {
```

Il s'agit d'une manière d'adapter la boucle à la longueur du tableau.

Jusqu'où affiche le programme maintenant ?

2.2.4 Troisième méthode de déclaration

Remplacez tout le code JavaScript par le suivant :

```
"use strict"     // force les déclarations de variables

// déclaration de fonction
function afficherTableau() {
     let res=""
     for (let idx=0; idx<listeHeros.length; idx++) {
          res = res + "Etudiant N°" + (idx+1) + " : " + listeHeros[idx] +
"<br>"
     }
     return res
}

// création d'un tableau de 10 cellules
document.write("<H1>Déclaration Table</H1>")
let listeHeros =
["BATMAN","SUPERMAN","","","","CATWOMAN","","","","","AQUAMAN"]

document.write("Longueur table = "+listeHeros.length)

document.write("<H1>Liste des Héros étudiants</H1>")
document.write(afficherTableau())
listeHeros[12] = "PUSHMAN"
document.write("<H1>Table finale</H1>")
document.write("Longueur table = "+listeHeros.length)
```

C'est la méthode la plus simple pour déclarer et remplir un tableau simultanément.

Que pouvez-vous dire par rapport au fonctionnement précédent ? Pourquoi les cellules ne sont plus indéfinies ?

2.2.5 Tests de performances

Il serait intéressant de savoir quelle instruction est la plus rapide. Pour cela, il faut créer un tableau suffisamment grand, le remplir avec des valeurs quelconques et afficher le temps écoulé. Cependant, le PC peut être très occupé à un instant T, ce qui rend la mesure seule, peu fiable.

Le protocole le plus efficace, consiste à remplir plusieurs fois le tableau, avec une boucle. Si la fonction de remplissage du tableau est la suivante :

```
function remplirTableau(nombre) {
      for (let idx=0; idx < nombre; idx++) {
            Tableau[idx] = "N°"+idx
      }
}
```

Pouvez-vous créer la boucle qui appelle cette fonction 10 fois ?

Voici la solution :

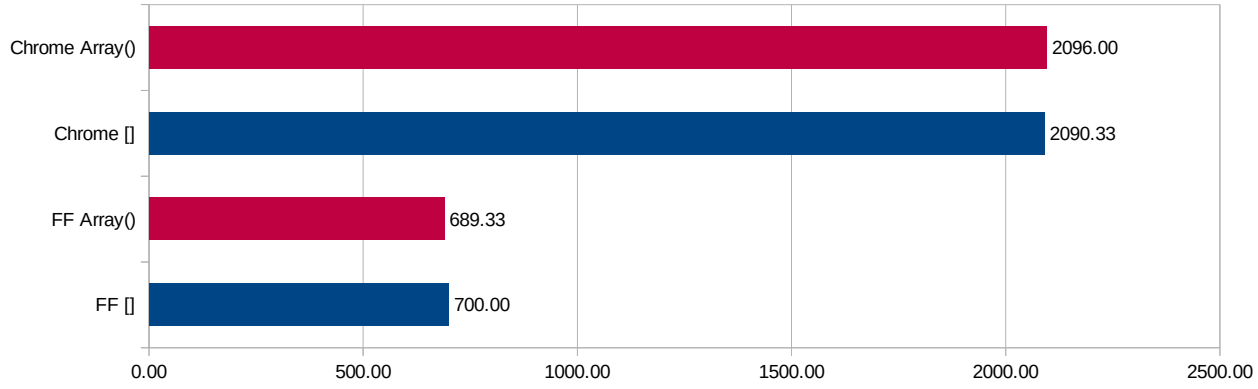
```
let max=10
for (let t=0; t < max; t++) {
      remplirTableau(nbre)
}
```

Pour chronométrer la fonction, il faut utiliser l'instruction `performance.now()` qui donne un résultat en milliseconde.

Exemple :

```
let t0=0
let t1=0
t0 = performance.now()
// ici les instructions ou fonctions à chronométrer
t1 = performance.now()
document.write((t1-t0) + " ms")
```

Sur mon PC, voici les résultats obtenus (traités avec un tableur) avec le code de la page suivante.



Rédigez le programme complet que vous appelerez "Test_performances_tableau.js", et validez votre solution avec la solution suivante (qui effectue la moyenne des différents essais) :

```
// déclaration de fonction
function remplirTableau(nombre) {
      for (let idx=0; idx < nombre; idx++) {
            tableau[idx] = "N°"+idx
      }
}

// création d'un tableau de 5.000.000 cellules
document.write("<H1>Déclaration Table</H1>")
let nbre = 5000000

// commentez cette ligne ou la suivante (pas les deux en même temps)
let tableau = [nbre]
// let tableau = new Array(nbre)

let t0=0
let t1=0
let moyenne=0
let max=10
for (let t=0; t < max; t++) {
      t0 = performance.now()
      remplirTableau(nbre)
      t1 = performance.now()
      moyenne=moyenne+(t1-t0)
      document.write("Durée = "+(t1-t0)+" ms<br>")
}
document.write("Moyenne = "+(moyenne/max)+" ms<br>")
```

Une seule ligne doit être commenté à la fois

N'oubliez pas d'appeler votre script dans le code HTML en insérant un nouveau script.

```
<script src="NOM_prenom_TP_tables.js"></script>
<script src="Test_performances_tableau.js"></script>
```

Testez votre script et notez la moyenne mesurée.

Puis, échangez les lignes à commenter (let tableau...), sauvegardez puis notez le nouveau résultat.

Si les tests sont plus long qu'une dizaine de secondes, diminuez la variable nbre.

Chrome est plus long que FireFox pour ce test. Les deux instructions sont équivalentes.

Modifier votre avancement dans le tableau Cryptpad.

3 Phase 2 – Création d'un tableau simple

Voici une petite mission simple : vous devez créer un tableau contenant le nom de chacun des étudiants de votre groupe (soit environ 15 cellules).

Pour les plus rapides, essayez de faire les parties avancées et experts. Si vous êtes en retard, ne faites que le programme de base.

3.1 Programme de base

Vous devez :

- Créer le tableau,
- Insérer manuellement les données (l'ordinateur ne peut deviner qui est là),
- Créer une fonction de lecture (avec une boucle)
- Utiliser la fonction avec un affichage

Modifier votre avancement dans le tableau Cryptpad.

3.2 Programme avancé

Pour aller plus loin, vous pouvez ajouter les éléments suivants :

- L'affichage doit se faire dans un tableau HTML (ajoutez les balises correspondantes)
- Les noms seront triés par ordre alphabétique (voir fonction JavaScript sort())

Modifier votre avancement dans le tableau Cryptpad.

3.3 Programme expert

Vous pouvez créer une fonction pour insérer les noms via une boite dialogue prompt() et une boucle avec une sortie grâce à :

- soit un champ vide
- soit une boite de dialogue confirm()

Pour ajouter des données dans un tableau, il faut utiliser :

- soit l'indice du tableau (si le tableau s'appelle etudiants, alors etudiants[x])
- soit par ajout automatique en fin de tableau (etudiants.push()

Modifier votre avancement dans le tableau Cryptpad.

3.4 SOLUTION

Ne regardez cette solution que lorsque vous pensez avoir réussi, ou au contraire, si vous êtes bloqué.

```
<!DOCTYPE html>
<html lang="fr">
<head>
	<meta charset="UTF-8">
	<title>Classe</title>
	<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
	<script>
		// déclaration des fonctions
		function remplirTableau() {
			let compteur = 0
			let nom = ""
			do {
				nom = prompt("Saisissez un nom.\nLaisser vide lorsque
vous avez fini.")
				if (nom != "") {
					monTableau[compteur] = nom
				}
				compteur++
			} while (nom != "")
		}

		function lireTableau() {
			let HTML = "<table border=1>"
			for (t=0; t<monTableau.length; t++) {
				HTML = HTML + "<tr><td>"+monTableau[t]+"</td></tr>"
			}
			HTML = HTML + "</table>"
			return HTML
		}

		// Exécution du programme
		document.write("<H1>Camarades de classe</H1>")
		let monTableau = []
		remplirTableau()
		let TexteHTML = lireTableau()
		document.write(TexteHTML)

	</script>
</body>
</html>
```