Mémo pseudo-code (complet)

Le **pseudo-code** est une étape intéressante dans la structuration de votre code JavaScript. Il s'agit en fait d'**écrire toutes nos instructions en français**, à l'intérieur de commentaires, puis de les traduire en code.

Exemple :

On demande à l'utilisateur son prénom et on affiche un message lui souhaitant la bienvenue en citant son prénom.

```
<script>
  // PSEUDO-CODE
  // ---------------------------------------------------------------------
  // Demande à l'utilisateur : "Quel est ton prénom ?"
  // Mets la réponse dans la boite : prenom
  // Ecris dans le body la concaténation suivante :
  // "Bonjour " + prenom + ", comment ça va ?"

  // CODE JAVASCRIPT
  // ---------------------------------------------------------------------

  // On demande à l'utilisateur de taper son prénom
  const prenom = window.prompt("Quel est ton prénom ?");

  // On écrit dans le document un message avec le prénom de l'utilisateur
  document.write("Bonjour " + prenom + ", comment ça va ?");
</script>
```

> **NB** : les commentaires dans le code JavaScript ne sont pas exactement du pseudo-code mais aident néanmoins à la bonne compréhension du code. N'hésitez pas à commenter votre code lors de son écriture en JavaScript.

Liste des pseudo-code pour l'examen

- window.alert("Attention, un monstre est dans l'école !");
- **Lance une alerte ...**

```
// Lance une fenêtre d'alerte avec comme message ...
```

- window.confirm("Veux-tu faire appel à un super-héro ?");
- **Demande à l'utilisateur de confirmer ...**

```
// Demande à l'utilisateur de confirmer ...
```

- window.prompt("A quel super-héro veux-tu faire appel ?");
- **Demande à l'utilisateur de répondre ...**

```
// Demande à l'utilisateur de répondre au message …
```

- document.write("Le schtroumpf grognon n’aime pas les super-héros");
- **Écris dans le body ...**

```
// Écris dans le body …
```

- document.title = "Le schtroumpf grr grr";
- **Modifie la propriété ...**

```
// Modifie la propriété ... de l'objet ... avec comme valeur ...
```

- let maVariable = "Bienvenue sur mon site"; const maVariable = "Je m'appelle Julien";
- **Mets la réponse dans la boîte ...**

```
// Mets xxx dans la boîte : xxx
```

- maVariable = Number(maVariable);
- **Convertis la boite en nombre**

```
// Convertis xxx en Number
```

- document.write("Je m'appelle" + prenom);
- **Fais la concaténation ...**

```
// Fais la concaténation : xxx + xxx ...
```

- let monOperation = (3 + 2) * 4
- **Fais le calcul suivant ...**

```
// Fais le calcul suivant : xxx * yyy / zzz
```

- if(condition) {
  // instructions exécutées quand la condition est vraie
  }
- **Si ...**

```
// Si ... alors ...
```

- if(condition) {
  // instructions exécutées quand la condition est vraie
  }
  else {
  // instructions exécutées quand la condition est fausse
  }
- **Si ... Sinon ...**

```
// Si ... alors...

// Sinon...
```

- if(condition1) {
  // instructions exécutées quand la condition1 est vraie
  }
  else if(condition2) {
  // instructions exécutées quand la condition2 est vrai
  }
- **Si ... Sinon si ...**

```
// Si ... alors ...

 // Sinon si ... alors ...
```

- while(condition) {
  // instructions à exécuter tant que la condition est vraie
  }
- Fais une boucle (« Répète (instructions) tant que (condition))

```
// Répète, tant que (condition),

    // les instructions suivantes :
```

- for(initialisation; condition; étape) {
  // instructions à répéter tant que la condition est vraie
  }
- **Fais une boucle en incrémentant ("Répète plusieurs fois …")**

```
// Répète,

    // pour un compteur i initialisé à ...

    // tant que le compteur i est ...

    // en incrémentant le compteur i

// les instructions suivantes : ...
```

- for(initialisation; condition; étape) {
  // instructions à répéter tant que la condition est vraie
  }
- **Fais une boucle en décrémentant ("Répète plusieurs fois …")**

```
// Répète,

    // pour un compteur i initialisé à ...

    // tant que le compteur i est ...

    // en décrémentant le compteur i

// les instructions suivantes : ...
```

- let XXX = [
  "Environnement",
  "HTML/CSS",
  "Javascript",
  "PHP"
  ];
- **Créé un tableau**

```
// Construis une armoire XXX
// Range XXX dans les tiroirs
```

- XXX[1] = "Belote";
- **Modifie un élément du tableau**

```
// Mets … dans le tiroir n°… de l'armoire XXX
```

- XXX[1] = "Belote";
- XXX.unshift("SGBD");
- XXX.push("Technique de l'image");
- **Mets la réponse dans l'armoire XXX**

```
// Mets … dans le tiroir n°… de l'armoire XXX

// Mets … dans le premier tiroir de l'armoire XXX

// Mets … dans le dernier tiroir de l'armoire XXX
```

- XXX.shift();
- XXX.pop();
- XXX.splice(y,x);
- **Supprime un tiroir de l'armoire XXX**

```
// Retire le premier tiroir de l'armoire XXX

// Retire le dernier tiroir de l'armoire XXX

// Retire x tiroirs de l'armoire XXX en commençant par le tiroir n°y
```

- XXX.indexOf("YYY");
- **Trouve l'indice d'un élément du tableau**

```
// Ouvre l'armoire XXX et trouve le n° du tiroir contenant YYY
```

- XXX.length;
- Récupère la taille du tableau

```
// La taille du tableau XXX
```

- for(let i = X; i < Y ou tailleDuTableau; i++) {
  // instructions à exécuter pour chaque élément du tableau
  }
- **Parcours le tableau XXX de A à Z**

```
// Ouvre les tiroirs n°X à n°Y de l'armoire XXX ...
// pour un compteur i initialisé à  ...
// tant que le compteur i est ...
// en incrémentant le compteur i
 // et exécute les instruction suivantes ...
```

- for(let i = tailleDuTableau -1 ou X; i >= 0 ou Y; i--) {
  // instructions à exécuter pour chaque élément du tableau
  (ex : document.write(XXX[i]);)
  }
- **Parcours le tableau XXX de Z à A**

```
// Ouvre les tiroirs n°X à n°Y de l'armoire XXX ...
// pour un compteur i initialisé à  ...
// tant que le compteur i est ...
// en décrémentant le compteur i
 // et exécute les instruction suivantes ...
```

Les opérateurs de comparaison

Opérateur	Signification
=== ou ==	Égal à
!== ou !=	Différent de
<	Inférieur à
<=	Inférieur ou égal à
>	Supérieur à
>=	Supérieur ou égal à

NB : si on utilise === , Javascript fait l'opération "égal à" en vérifiant le type de donnée ! En utilisant == , Javascript ne vérifie pas le type de donnée. C'est pareil pour !== et !=

Attention : vous devez savoir que toute valeur évaluée par le JavaScript dans un contexte booléen va être évaluée à `true` à l'exception des valeurs suivantes qui vont être évaluées à `false` :

- Le booléen `false` ;
- La valeur 0 ;
- Une chaine de caractères vide ;
- La valeur `null` ;
- La valeur `undefined` ;
- La valeur `NaN` (« Not a Number » = « n'est pas un nombre »).