

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 boîte : 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

- Lance une alerte ...

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

- Demande à l'utilisateur de confirmer ...

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

- Demande à l'utilisateur de répondre ...

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

- Écris dans le body ...

```
// Écris dans le body ...
```

- **Modifie la propriété ...**

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

- **Mets la réponse dans la boîte ...**

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

- **Convertis la boîte en nombre**

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

- **Fais la concaténation ...**

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

- **Fais le calcul suivant ...**

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

- **Si ...**

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

- **Si ... Sinon ...**

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

```
// Sinon...
```

- **Si ... Sinon si ...**

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

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

- **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 : ...
```

- **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 : ...
```

- **Crée un tableau**

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

- **Modifie un élément du tableau**

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

- **Mets la réponse dans l'armoire XXX**

```
// Mets ... dans le tiroir n°... de l'armoire XXX  
  
// Mets XXX dans le premier tiroir de l'armoire XXX  
  
// Mets XXX dans le dernier tiroir de l'armoire XXX
```

- **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 en commençant par le tiroir n° ...
```

- **Trouve l'indice d'un élément du tableau**

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

- **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 ...
```

- **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 ...
```