

## JavaScript Révisions

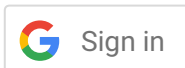
### Bonjour, Demo User #809

Ces exercices vous permettent de mesurer votre compréhension et maîtrise des concepts vus en cours de JavaScript.

Vos résultats sont calculés localement, quand vous passez d'une question à l'autre. Chaque QCM est noté sur 1 point. Chaque exercice de code, sur 3 points.

Il est recommandé d'utiliser la console JavaScript pour tester vos solutions.

À vous de jouer !



## Les bases - QCM

### Question 1 (1 points)

Comment ouvrir la console JavaScript dans Google Chrome ?

- ☐ En tapant "console"
- ☐ En appelant le prof
- ☐ En demandant gentiment à Siri
- ☒ En pressant Cmd-Alt-J ou Ctrl-Shift-J

### Question 2 (1 points)

Que retourne `typeof` quand il est appliqué sur `"bonjour"` ?

- ☒ "string"
- ☐ string
- ☐ "object"
- ☐ undefined

**Question 3 (1 points)**

Types de valeurs en JavaScript. Quel est l'intrus ?

- ☐ string
- ☐ boolean
- ☒ decimal
- ☐ number

**Question 4 (1 points)**

Comment créer une variable en JavaScript ?

- ☐ maVariable;
- ☒ var maVariable;
- ☐ x = 0;
- ☐ maVariable = 'bonjour';

**Question 5 (1 points)**

Quel est le type de cette variable:

```
var maVariable = 3.2;
```

- ☐ string
- ☒ number
- ☐ decimal
- ☐ boolean

**Question 6 (1 points)**

Quel est le type de cette variable:

```
var maVariable = '3.2';
```

- ☒ string
- ☐ number
- ☐ decimal
- ☐ boolean

### Question 7 (1 points)

Comment afficher la valeur d'une variable appelée `maVariable` depuis la console ?

- ☒ `maVariable;`
- ☐ `var maVariable;`
- ☐ `maVariable?`
- ☐ `show maVariable`

### Question 8 (1 points)

Comment changer la valeur d'une variable existante ? (déjà créée)

- ☐ `var maVariable = 4;`
- ☒ `maVariable = 4;`
- ☐ `maVariable(4);`
- ☐ `4 = maVariable;`

### Question 9 (1 points)

Si j'ai créé une variable dont la valeur est un nombre, que se passera-t-il si je lui affecte ensuite une chaîne de caractères ?

- ☐ erreur, car le type est différent.
- ☐ erreur, car on ne peut pas changer la valeur d'une variable.
- ☒ la valeur de la variable va être remplacée par la chaîne de caractères.
- ☐ les deux valeurs vont être concaténées.

### Question 10 (1 points)

En respectant les conventions indiquées en cours, quelle affectation faut-il exécuter pour que `J'ai tout compris !` s'affiche à l'écran ?

```
alert(message);
```

- ☐ `message = "J'ai tout compris !";`
- ☐ `message = 'J"ai tout compris !';`
- ☒ `message = 'J\'ai tout compris !';`
- ☐ `message = J'ai tout compris !`

## Conditions - QCM

### Question 11 (1 points)

Que vaut cette expression de comparaison de valeur ?

```
3.2 === '3.2'
```

- ☐ true
- ☒ false
- ☐ undefined
- ☐ c'est une affectation

### Question 12 (1 points)

Quelle section va être exécutée, si on exécute le code suivant ?

```
var nb = 2;  
if (nb === 1) {  
    // A  
} else {  
    // B  
}
```

- ☐ A
- ☒ B
- ☐ A et B
- ☐ aucune

### Question 13 (1 points)

Quelle section de va être exécutée, si on exécute le code suivant ?

```
var nb = 2;  
if (nb === 2) {  
    // A  
} else if (nb > 1) {  
    // B  
} else {  
    // C  
}
```

- ☒ A
- ☐ B
- ☐ A et B
- ☐ A, B et C

### Question 14 (1 points)

À quoi ressemblerait l'arbre de décision correspondant à ce code:

```
var reponse = prompt('as-tu faim ?')
if (reponse === 'oui') {
  var reponse2 = prompt('aimes-tu les burgers ?');
  if (reponse2 === 'oui') {
    alert('alors je t\'en offre un !');
  } else {
    alert('dommage !');
  }
} else {
  alert('désolé');
}
```

- ☐ une boîte et deux branches
- ☐ deux boîtes de même niveau
- ☒ une boîte de niveau 1, et une boîte de niveau 2
- ☐ une boîte et trois branches

### Question 15 (1 points)

Pourquoi faut-il éviter d'utiliser les opérateurs `==` et `!=` ?

- ☐ car il vaut mieux utiliser une affectation `=`
- ☐ car ils sont trop stricts
- ☒ car ils sont trop laxistes
- ☐ var `===` et `!==` sont plus lisibles

## Conditions - Code

### Question 16 (3 points)

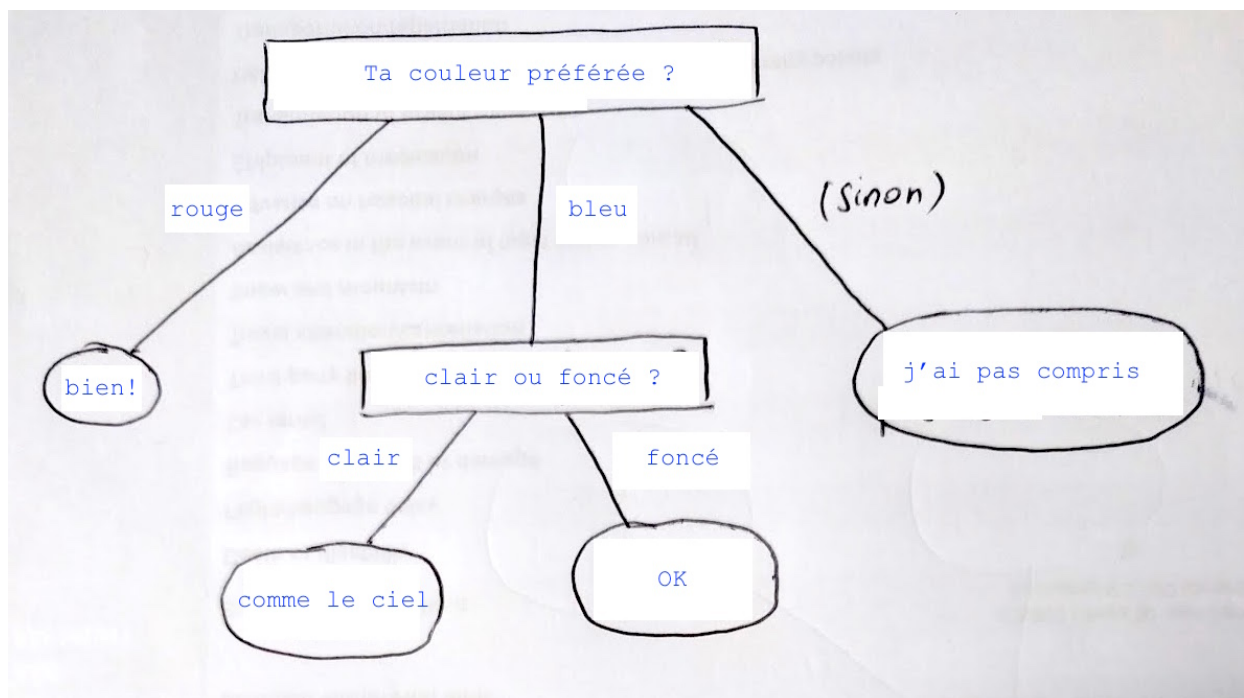
Implémenter une condition qui affecte 'egal' à une variable `resultat` seulement si une autre variable `nombre` vaut strictement 4. Indenter correctement.

Saisissez votre code Javascript ici

```
if (nombre === 4) {  
    resultat = 'egal';  
}
```

### Question 17 (3 points)

Implémenter un chat-bot à partir de l'arbre de décision suivant:



Comme vu et pratiqué en cours:

- Les questions posées par l'ordinateur sont représentées par des rectangles, et sont à implémenter à l'aide de la fonction `prompt()`.
- Les réponses comprises par l'ordinateur sont écrites à côté de chaque branche, et sont à implémenter à l'aide de conditions.

- Les messages à afficher par l'ordinateur sont représentés par des cercles, et sont à implémenter à l'aide de la fonction `alert()`.

Vous serez noté(e) sur:

- le respect **à la lettre** du texte des questions et des réponses (espaces, accents, et majuscules/minuscules compris).
- le respect des règles d'indentation et autres conventions vues en cours. (ex: 2 espaces par niveau d'indentation)
- le bon fonctionnement de votre code, sans erreurs, depuis la console JavaScript de Google Chrome, pour chacun des cas illustrés dans l'arbre de décision.

Saisissez votre code Javascript ici

```
var reponse = prompt('Ta couleur préférée ?');
if (reponse === 'rouge') {
    alert('bien!');
} else if (reponse === 'bleu') {
    var reponse2 = prompt('clair ou foncé ?');
    if (reponse2 === 'clair') {
        alert('comme le ciel');
    } else if (reponse2 === 'foncé') {
        alert('OK');
    }
} else {
    alert('j\'ai pas compris');
}
```

---

## Algo / Boucles - QCM

### Question 18 (1 points)

Combien de fois les instructions vont-elles être exécutées ?

```
for ( var i = 0; i < 4; i++ ) {
    // instructions
}
```



- ☐ 0 fois
- ☐ 1 fois
- ☐ 3 fois
- ☒ 4 fois

### Question 19 (1 points)

Combien de fois les instructions vont-elles être exécutées ?

```
for ( var i = 3; i >= 1; i-- ) {  
  // instructions  
}
```

- ☐ 0 fois
- ☐ 1 fois
- ☒ 3 fois
- ☐ 4 fois

## Algo / Boucles - Code

### Question 20 (3 points)

Implémenter un programme de moins de 4 lignes qui affiche 50 fois 'Bonjour!' dans la console. Respecter les conventions et règles d'indentation vues en cours.

Saisissez votre code Javascript ici

```
for ( var i = 0; i < 50; i++ ) {  
  console.log('Bonjour!');  
}
```

# Fonctions - QCM

## Question 21 (1 points)

```
function maFonction(param) {  
  return param + 2;  
}
```

Ceci est:

- ☐ un appel de fonction
- ☒ une définition de fonction
- ☐ une affectation de fonction
- ☐ une fonction qui ne fonctionne pas

## Question 22 (1 points)

```
maFonction(4);
```

Ceci est:

- ☒ un appel de fonction
- ☐ une définition de fonction
- ☐ une affectation de fonction
- ☐ une fonction qui ne fonctionne pas

## Question 23 (1 points)

```
// cette fonction concatène un zéro à la fin de la valeur passée en paramètre  
function maFonction(param) {  
  return param + '0';  
}
```

Comment savoir si cette fonction fonctionne bien ? (c.a.d. sans bug)

- ☐ il suffit de la copier-coller dans la console
- ☐ il faut taper maFonction dans la console
- ☒ vérifier que le test passe: `maFonction(1) === '10'`;
- ☐ vérifier que `maFonction(1)` renvoie bien `true`

### Question 24 (1 points)

Supposons que nous avons défini une fonction `doubler()` qui retourne le double du nombre passé en paramètre, lors de son appel.

Que se passe-t-il si on exécute l'instruction suivante:

```
var maVariable = doubler(3);
```

- ☒ le résultat va être affecté à `maVariable`
- ☐ le résultat va s'afficher dans la console
- ☐ `maVariable` contient la définition de la fonction
- ☐ `maVariable` contient l'appel de la fonction

## Fonctions - Code

### Question 25 (3 points)

Définir une fonction `soustraire` qui retourne le résultat de la soustraction `a - b`, `a` et `b` étant des paramètres de cette fonction.

Respecter les conventions et règles d'indentation vues en cours.

Saisissez votre code Javascript ici

```
function soustraire(a, b) {  
  return a - b;  
}
```

---

### Question 26 (3 points)

Définir une fonction `somme` qui retourne la somme des trois nombres passés en paramètres.

Exemple d'appel: `somme(1, 3, -2);` doit retourner `2`.

Saisissez votre code Javascript ici

```
function somme(a, b, c) {  
  return a + b + c;  
}
```

---

### Question 27 (3 points)

Définir une fonction `repetier` qui affiche `n` fois `'Bonjour!'` dans la console, puis qui retourne `n`, `n` étant un paramètre de cette fonction.

Respecter les conventions et règles d'indentation vues en cours.

Saisissez votre code Javascript ici

```
function repeter(n) {  
  for (var i = 0; i < n; i++) {  
    console.log('Bonjour!');  
  }  
  return n;  
}
```

---

### Question 28 (3 points)

Définir une fonction `sommeIntervalle` qui retourne la somme de tous les nombres entiers compris entre les nombres `premier` et `dernier` (compris) passés en paramètres.

Exemple d'appel: `sommeIntervalle(2, 5);` doit retourner `14` (résultat de `2 + 3 + 4 + 5`).

Saisissez votre code Javascript ici

```
function sommeIntervalle(premier, dernier) {  
    var somme = 0;  
    for (var i = premier; i <= dernier; i++) {  
        somme = somme + i;  
    }  
    return somme;  
}
```

---

## Tableaux - Code

### Question 29 (3 points)

Créez une variable `nombres` de type tableau et contenant les nombres `1`, `2` et `3`.

Saisissez votre code Javascript ici

```
var nombres = [1, 2, 3];
```

---

### Question 30 (3 points)

Vous disposez d'une variable `fruits` contenant un tableau de chaînes de caractères.

Saisissez le code JavaScript pour créer une variable `troisieme` et lui affecter la valeur du 3ème élément de ce tableau.

Saisissez votre code Javascript ici

```
var troisieme = fruits[2];
```

---

### Question 31 (3 points)

Vous disposez d'une variable `fruits` contenant un tableau de chaînes de caractères.

Saisissez le code JavaScript permettant de retirer le dernier élément de ce tableau, et d'afficher la valeur de cet élément dans la console.

Saisissez votre code Javascript ici

```
console.log(fruits.pop());
```

---

### Question 32 (3 points)

Définir une fonction `tableauContient` qui prend deux paramètres:

- `tableau` : un tableau de chaînes de caractères
- `chaîne` : une chaîne de caractères

...et retourne:

- `false` si la valeur `chaîne` n'a pas été trouvée dans le tableau `tableau`,
- ou le premier indice (à partir de 0) auquel a été trouvé la valeur `chaîne` dans le tableau `tableau`.

Exemples d'appels:

- `tableauContient(['a', 'b', 'c'], 'b');` doit retourner `1`.
- `tableauContient(['a', 'b', 'c'], 'd');` doit retourner `false`.

Saisissez votre code Javascript ici

```
function tableauContient(tableau, chaîne) {  
    var indice = tableau.indexOf(chaîne);  
    if (indice === -1) {  
        return false;  
    } else {  
        return indice;  
    }  
}
```

---

## Rendu de copie

Cliquez sur le bouton ci-dessous pour obtenir votre note totale, ainsi que la solution de chaque exercice.

✓ RENDRE LA COPIE

