

Exercice

donnez les valeurs des différentes variables après exécution de ce script

```
1 let variable1 = 5;  
2 let variable2 = variable1++;  
3 let variable3 = variable2++;  
4 const variable4 = ++variable3;  
5 console.log(variable1);  
6 console.log(variable2);  
7 console.log(variable3);  
8 console.log(variable4);
```

variable1 =

variable2 =

variable3 =

variable4 =

Exercice

Que retourne l'instruction suivante

```
console.log(0 || "Hello" && null || 42)
```

- ☐ 0
- ☐ "Hello"
- ☐ null
- ☐ 42
- ☐ false

Exercice

Quel est le résultat de ce code ?

```
1 function test() {  
2   if (true) {  
3     var a = 5;  
4     let b = 10;  
5   }  
6   console.log(a + b);  
}
```

```
7 }  
8 test();
```

Exercice

Complétez en utilisant la fonction map le code suivant pour qu'il affiche [4, 9, 16] :

NB : Le résultat sera donné sur une seule ligne

```
const nums = [2, 3, 4];  
const result =
```

Exercice

Quelle méthode permet de sélectionner tous les éléments avec la classe **error** ?

- ☐ document.querySelectorAll('#error')
- ☐ document.getElementsByTagName('error')
- ☐ document.getElementById('error')
- ☐ document.getElementsByClassName('error')
- ☐ document.querySelector('.error')

Exercice

Écrivez le code manquant pour ajouter un paragraphe avec le texte "Bonjour" dans une <div id="content"> :

```
const div =  
const p =  
  
div.
```

Exercice

Complétez le code suivant pour que le **résultat affiché soit 13**. Utilisez les classes suivante.

```
1 class MathUtils {  
2   static square(x) { return x * x; }  
3 }  
4  
5 class AdvancedMath extends MathUtils {  
6   static cube(x) { return x * x * x; }  
7 }  
  
console.log
```

Exercice

Que retourne ce code ?

```
1 const data = [{x: 2}, {x: 3}, {x: 4}];
2 const result = data.reduce((acc, curr) => ({x: acc.x + curr.x}));
3 console.log(result.x);
```

Exercice

Quel est le résultat de ce code ?

```
1 class Wallet {
2   #balance = 0;
3   add(amount) { this.#balance += amount; }
4   getBalance() { return this.#balance; }
5 }
6
7 const wallet = new Wallet();
8 wallet.add(100);
9 console.log(wallet.#balance);
```

Exercice

Que retourne le code suivant ?

```
1 console.log([...[... "Hello"].filter(c => c !== "l")])
```

Exercice

Écrivez le code javascript manquant pour créer un élément <div> avec une classe "alert". cette div sera ajouté à l'élément body.

Exercice

Que vaut le prédicat suivant :

Nom:

Pseudo:

Age:

```
if ($_POST["nom"] == "Fernand" || $_POST["pseudo"] == "Fernando")
```

Exercice

Que se passe-t-il si un fichier inclus via **include** contient une erreur de syntaxe dans le code à l'intérieur du fichier?

- ☐ Un E_WARNING est généré, le script continue
- ☐ Un E_ERROR est généré, le script s'arrête
- ☐ Aucune erreur n'est générée
- ☐ Le fichier est ignoré silencieusement

Exercice

Complétez ce code pour que le résultat affiché soit **11**

```
<?php
class Math {
    public function add( $a, $b ) {
        return array_sum( array( $a, $b ) );
    }
}

echo Math::add(2, 3**2);
```

?>

Exercice

Quelle fonction est recommandée pour éviter les attaques XSS lors de l'affichage de données utilisateur ?

Exercice

Quel est le résultat de ce code ?

```
1 <?php
2     $a = 5;
3     $b = &$a;
4     $b = 10;
5     $b = $a+1;
6     echo $a;
7 ?>
```

Exercice

Complétez le code pour vérifier si une clé existe dans un tableau multidimensionnel

```
<?php
    $data = [
        'user' => ['name' => 'Alice', 'age' => 25],
        'posts' => [['id' => 1], ['id' => 2]]
    ];

    $exists = array_key_exists('name', $data['user']) && array_key_exists(2,
array_column($data['posts'], 'id'));

    echo $exists;

?>
```

Exercice

Écrivez le résultat de ce code :

```
1 <?php
2
3 $users = [
4     'Aristide YAO',
5     'Mickaël Andrieu',
6     'Laurène Castor',
7 ];
8
9 $author = array_search('Laurène Castor', $users);
10 echo "Sauthor";
11 ?>
```

Exercice

Quel est le résultat de ce code ?

```
1 function sum(...$numbers) {  
2     return array_sum($numbers) / count($numbers);  
3 }  
4 echo sum(2, 4, 6, 8);
```

- ☐ Erreur
- ☐ 20
- ☐ 5
- ☐ 4
- ☐ 10

Exercice

Quelle fonction utiliser pour obtenir un message d'erreur lisible après un `json_decode()` échoué ?

- ☐ `json_error()`
- ☐ `json_get_error()`
- ☐ `json_last_error_msg()`
- ☐ `json_last_error()`
- ☐ `json_error_msg()`

Exercice

Donnez le résultat d'exécution de ce code :

```
1 <?php  
2     function toto(&...$nums) {  
3         foreach ($nums as &$n) $n++;  
4     }  
5     $a = 1; $b = 2;  
6     toto($a, $b);  
7     echo "$a, $b";  
8 ?>
```

Exercice

Que retourne ce code ?

```
1 <?php  
2 function test(...$vals) {  
3     return match(count($vals)) {  
4         1 => 'un',
```

```

5         2 => 'deux',
6         default => 'beaucoup'
7     };
8 }
9 echo test(2);
10 ?>

```

Exercice

```

<?php
class Config {
    // ...

}

Config::$env = 'dev';
echo Config::$env; //Affiche dev
?>

```

Exercice

Complétez pour écrire dans un fichier JSON en mode ajout exclusif:

```

$data = ['new' => 'entry'];

file_put_contents('log.json', $data, FILE_APPEND);

```

Exercice

Que se passe-t-il si vous utilisez credentials: 'omit' avec mode: 'cors' ?

- ☐ La requête échoue systématiquement.
- ☐ La requête est envoyée, mais sans cookies ni en-têtes d'authentification.
- ☐ Le navigateur force same-origin à la place.
- ☐ Aucune des réponses.

Exercice

Complétez les phrases avec les termes appropriés.

L'API Fetch retourne une qui se résout avec un objet même en cas d'erreur HTTP (comme 404).

Pour interrompre une requête fetch(), on utilise et .

Le mode dans une requête Fetch empêche les requêtes cross-origin, tandis que les autorise sous certaines conditions.

La méthode de l'objet Promise permet d'exécuter plusieurs promesses en parallèle et rejette immédiatement si l'une échoue.

Une promesse dans l'état `Pending` signifie que l'opération est en cours, tandis que `Fulfilled` indique qu'elle a réussi.

Exercice

Quel comportement aura `fetch()` si le serveur renvoie une redirection 302 ?

- ☐ Elle suit la redirection automatiquement si `redirect: 'follow'`
- ☐ Elle rejette la promesse avec une erreur.
- ☐ Elle bloque la requête jusqu'à confirmation de l'utilisateur.
- ☐ Aucune des réponses.

Exercice

Pourquoi `Promise.all()` n'est-elle pas idéale pour charger 100 requêtes en parallèle ?

Exercice

Pourquoi `mode: 'same-origin'` n'est-il pas équivalent à `no-cors` ?

Exercice

Quelle est la différence entre `credentials: 'include'` et `credentials: 'same-origin'` dans une requête Fetch ?