

Comparaison PHP - JavaScript

Contenu

- **Comparaison PHP - JavaScript**

- Les variables
- Les opérateurs
- Les structures conditionnelles
- Les boucles
- Les tableaux
- Les fonctions
- Les classes et les objets

Les variables

Déclarer une variable

PHP

```
$nombre = 55;  
$chaine1 = 'Bonjour';  
$chaine2 = "Allo";  
$booleen = true;  
$variableNulle = null;
```

JavaScript

```
let nombre = 55;  
let chaine1 = 'Bonjour';  
let chaine2 = "Allo";  
let booleen = true;  
let variableNulle = null;
```

Accéder à une variable

PHP

```
$maVariable = 42;
```

```
/* Afficher la variable  
à l'écran */  
echo $maVariable;
```

JavaScript

```
let maVariable = 42;
```

```
/* Afficher la variable  
en console */  
console.log(maVariable);
```

Concaténer deux variables

PHP

```
$str = 'bao' . 'bab';
```

```
$prenom = 'Homer';
```

```
$nom = 'Simpson';
```

```
$nomComplet = "$prenom $nom";
```

JavaScript

```
let str = 'bao' + 'bab';
```

```
let prenom = 'Homer';
```

```
let nom = 'Simpson';
```

```
let nomComplet = `${prenom} ${nom}`;
```

Les opérateurs

Les opérateurs arithmétiques

Ce sont les mêmes!

PHP

```
$a = $b + $c;
```

```
$d = $e - $f;
```

```
$g = $h * $i;
```

```
$j = $k / $l;
```

```
$m = $n % $o;
```

JavaScript

```
let a = b + c;
```

```
let d = e - f;
```

```
let g = h * i;
```

```
let j = k / l;
```

```
let m = n % o;
```


Les opérateurs de comparaison

Ce sont les mêmes!

PHP

```
$a = $b == $c;  
$d = $e === $f;  
$g = $h != $i;  
$j = $k !== $l;  
$m = $n < $o;  
$p = $q <= $r;  
$s = $t > $u;  
$v = $w >= $x;
```

JavaScript

```
let a = b == c;  
let d = e === f;  
let g = h != i;  
let j = k !== l;  
let m = n < o;  
let p = q <= r;  
let s = t > u;  
let v = w >= x;
```

Les opérateurs logiques

Ce sont les mêmes!

PHP

```
$a = $b || $c;
```

```
$d = $e && $f;
```

```
$g = !$h;
```

JavaScript

```
let a = b || c;
```

```
let d = e && f;
```

```
let g = !i;
```

Les structures conditionnelles

if - else if - else

Même syntaxe!

PHP

```
if ($condition) {  
    // ...  
} else if ($autreCondition) {  
    // ...  
} else {  
    // ...  
}
```

JavaScript

```
if (condition) {  
    // ...  
} else if (autreCondition) {  
    // ...  
} else {  
    // ...  
}
```

Switch

Même syntaxe!

PHP

```
switch ($maVariable) {  
    case 'valeur1':  
        // ...  
        break;  
    case 'valeur2':  
        // ...  
        break;  
    default:  
        // ...  
}
```

JavaScript

```
switch (maVariable) {  
    case 'valeur1':  
        // ...  
        break;  
    case 'valeur2':  
        // ...  
        break;  
    default:  
        // ...  
}
```

Les boucles

La boucle While

Même syntaxe!

PHP

```
while ($condition) {  
    // ...  
}
```

JavaScript

```
while (condition) {  
    // ...  
}
```

La boucle For

PHP

```
for ($i = 0; $i < 10; $i++) {  
    // ...  
}
```

JavaScript

```
for (let i = 0; i < 10; i++) {  
    // ...  
}
```


La boucle For-Each

PHP

```
foreach ($monTableau as $element) {  
    // ...  
}
```

```
foreach ($monTableau as $id => $element)  
{  
    // ...  
}
```

JavaScript

```
monTableau.forEach((element) => {  
    // ...  
});
```

```
monTableau.forEach((element, id) => {  
    // ...  
});
```

Les tableaux

Déclarer un tableau numéroté

PHP

```
$monTableau = array(1, 3, 5, 7);
```

```
// Ou bien
```

```
$monTableau = [1, 3, 5, 7];
```

JavaScript

```
let monTableau = new Array(1, 3, 5, 7);
```

```
// Ou bien (plus courant)
```

```
let monTableau = [1, 3, 5, 7];
```

Accéder à un élément d'un tableau

Même syntaxe!

PHP

```
$element = $monTableau[8];
```

```
$monTableau[8] = 42;
```

JavaScript

```
let element = monTableau[8];
```

```
monTableau[8] = 42;
```

Ajouter un élément à la fin d'un tableau

PHP

```
$monTableau = [1, 3, 5, 7];
```

```
array_push($monTableau, 9);
```

```
array_push($monTableau, 11, 13);
```

JavaScript

```
let monTableau = [1, 3, 5, 7];
```

```
monTableau.push(9);
```

```
monTableau.push(11, 13);
```

Déterminer le nombre d'éléments dans un tableau

PHP

```
$monTableau = [1, 3, 5, 7];
```

```
$nombreElements = count($monTableau);
```

JavaScript

```
let monTableau = [1, 3, 5, 7];
```

```
let nombreElements = monTableau.length;
```

Les tableaux associatifs

- **Les tableaux associatifs sont un cas particulier en JavaScript**
- **Nous y reviendrons!**

Les fonctions

Déclarer une fonction

PHP

```
function additionner($x, $y) {  
    return $x + $y;  
}
```

JavaScript

```
function additionner(x, y) {  
    return x + y;  
}
```

```
// Syntaxe alternative  
let additionner = (x, y) => {  
    return x + y;  
};
```

```
// Ou encore  
let additionner = (x, y) => x + y;
```

Appeler une fonction

Même syntaxe!

PHP

```
$somme = additionner(3, 2);
```

JavaScript

```
let somme = additionner(3, 2);
```

Les classes et les objets

Définir une classe

PHP

```
class Chat {  
    private $nom;  
  
    function __construct($nom) {  
        $this->nom = $nom;  
    }  
  
    function miauler() {  
        echo $this->nom . ': Miaou!';  
    }  
}
```

JavaScript









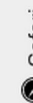







```
class Chat {  
    nom;  
  
    constructor(nom) {  
        this.nom = nom;  
    }  
  
    miauler() {  
        console.log(`${this.nom}: Miaou!`);  
    }  
}
```

Accessibilité des propriétés

- Par défaut, tout est public en JavaScript
- Une proposition expérimentale est en cours de déploiement pour ajouter des propriétés et méthodes privées (juin 2021)
 - On ajoute « # » devant leurs noms

Compatibilité des navigateurs

[Report problems with this compatibility data on GitHub](#)

													
	 Chrome	 Edge	 Firefox	 Internet Explorer	 Opera	 Safari	 WebView Android	 Chrome Android	 Firefox for Android	 Opera Android	 Safari on iOS	 Samsung Internet	 Node.js
Private class fields	74	79	90	No	62	14.1	74	74	90	53	14.5	No	12.0.0

https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Classes/Private_class_fields

Créer une instance d'une classe

Même syntaxe!

PHP

```
$monChat = new Chat('Arthur');
```

JavaScript

```
let monChat = new Chat('Arthur');
```

Appeler une méthode

PHP

```
$monChat = new Chat('Arthur');
```

```
$monChat->miauler();
```

JavaScript

```
let monChat = new Chat('Arthur');
```

```
monChat.miauler();
```

Les objets

- **En PHP, nous avons vu:**

- Qu'un objet a des propriétés
- Qu'une propriété a un nom et une valeur
- Qu'un tableau associatif a des clés et des valeurs

Les objets

- **Un objet est donc semblable à un tableau associatif**
 - Objet: Nom de propriété -> Valeur
 - Tableau associatif: Clé -> Valeur
- **En JavaScript, il n'y a pas de tableaux associatifs**
- **On utilise donc les objets pour répondre aux mêmes besoins**

Les objets

- **On peut créer un objet directement, sans l'instancier à partir d'une classe**

JavaScript

```
let etudiant = {  
  code: '201234567',  
  prenom: 'Alfred',  
  nom: 'Desrochers',  
};
```

Les objets

- **On accède aux attributs d'un objet soit avec le point (.), soit avec les crochets ([])**

JavaScript

```
let prenomEtudiant = etudiant.prenom;
```

```
let nomEtudiant = etudiant['nom'];
```

Les objets

- **Les crochets permettent d'utiliser la valeur d'une variable comme nom de propriété**

JavaScript

```
let cle = 'code';
```

```
let codeEtudiant = etudiant[cle];
```

Fin de la présentation

Des questions?



Photo par Emily Morter sur Unsplash