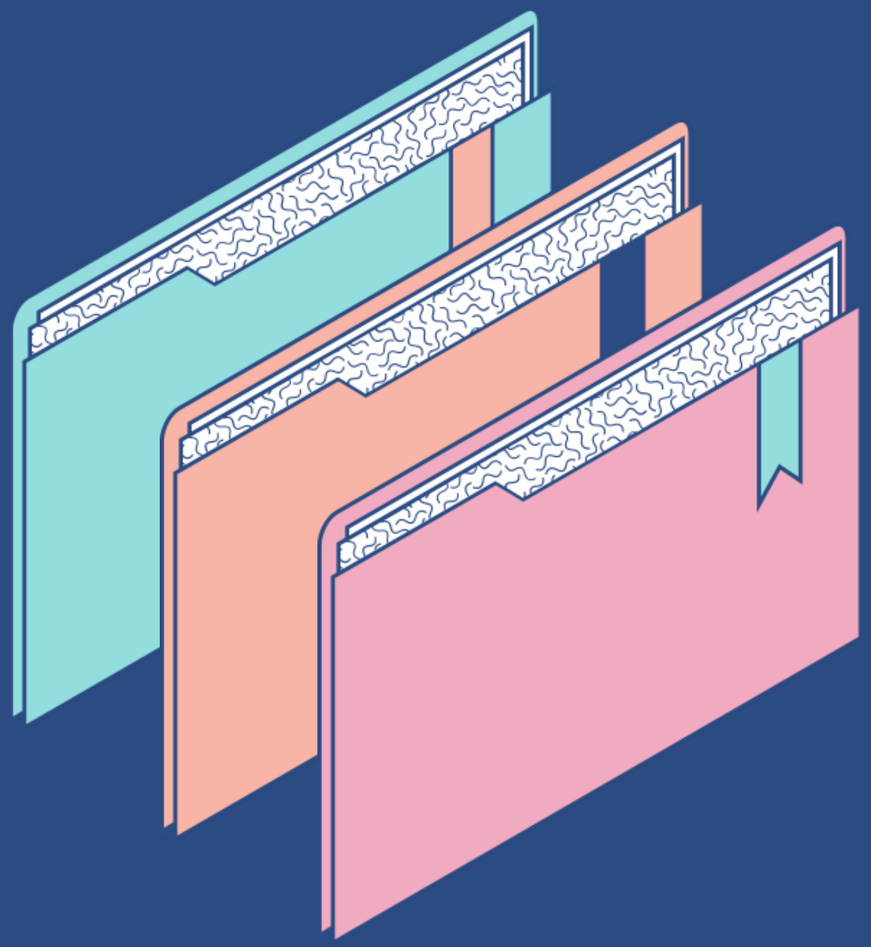
Javascript

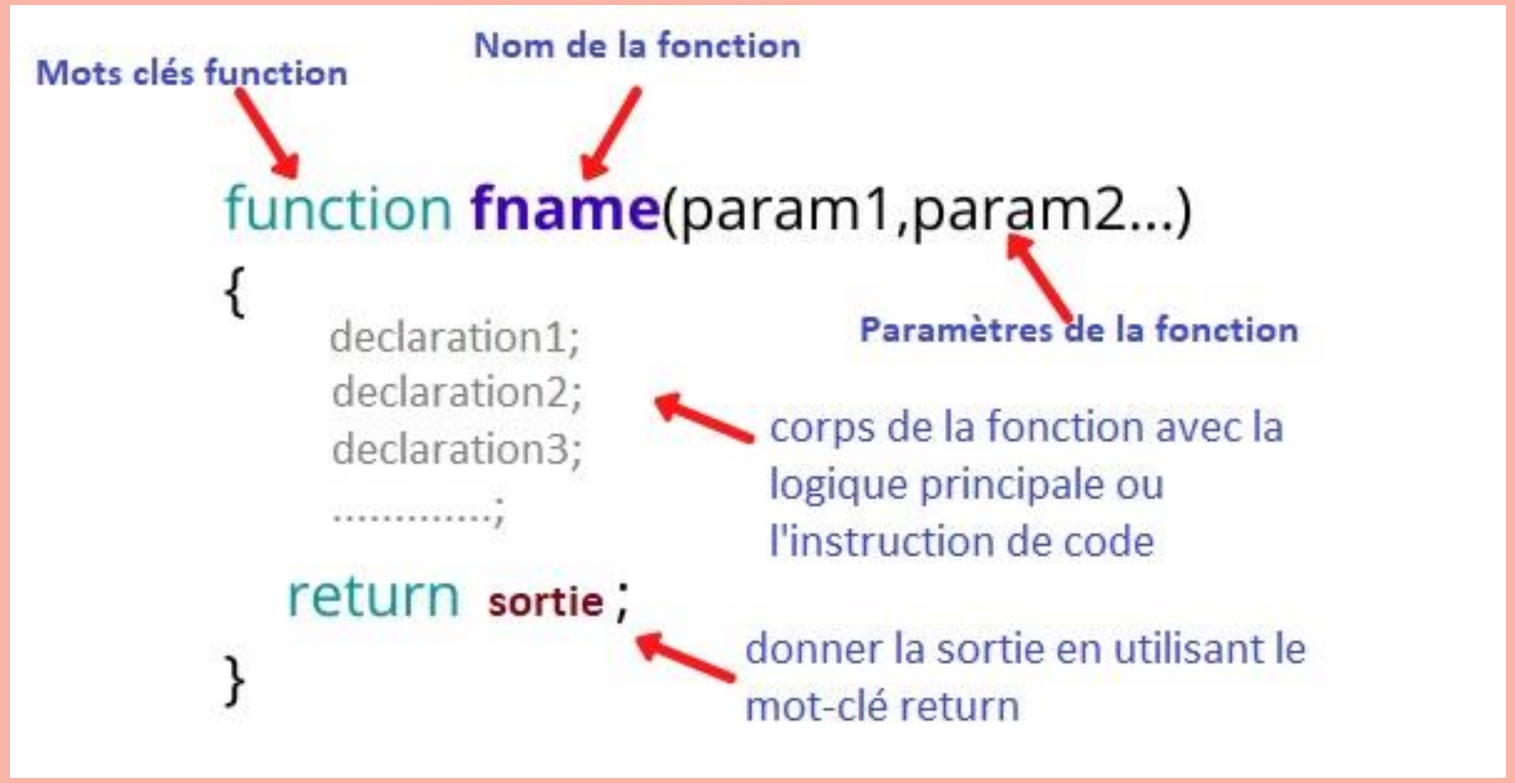

JavaScript
<script>
if(age > 19){
alert("Adult");
} else{
alert("Teenager");
}
</script>

SOMMAIRE



- Présentation du JS
- Variables
- Structures de contrôle
- Fonctions
- POO en JS
- Valeurs primitives
- Manipulation du BOM
- Manipulation du DOM
- Fonctions avancées
- Gestion des erreurs
- Stockage de données persistantes
- Canvas
- Asynchrone

Fonctions


Mots clés function
Nom de la fonction
function fname(param1,param2...)
{
declaration1;
declaration2;
declaration3;
.....;
return sortie ;
}
Paramètres de la fonction
corps de la fonction avec la logique principale ou l'instruction de code
donner la sortie en utilisant le mot-clé return

Fonctions JS

Comme dans de nombreux langages, il existe en JS des fonctions prédéfinies, et des fonction personnalisées.

Les fonctions prédéfinies peuvent être utilisées directement, et les fonctionnalisées doivent être écrites spécifiquement.

Fonctions prédéfinies

Pour utiliser les fonctions prédéfinies, il vous suffit d'appeler ces fonctions par leur nom, en leur fournissant les paramètres d'entrée éventuels.

```
push('cows');
```

https://waytolearnx.com/2019/09/liste-des-fonctions-javascript.html

Fonctions personnalisées

Les fonctions personnalisées permettent de créer du code réutilisable, plus maintenable et plus sécurisé.

```
/*On définit deux fonctions personnalisées.
 *La fonction aleatoire() se sert de la fonction (méthode) random().
 *La fonction multiplication() multiplie deux nombres entre eux.
 *On utilise une instruction return pour que nos fonctions, une fois appelées,
 *retournent le résultat de leur calcul afin qu'on puisse utiliser ce résultat*/
function aleatoire(){
    return Math.random() * 100;
}

function multiplication(nombre1, nombre2){
    //Attention : les "+" sont utilisés pour la concaténation !
    return nombre1 + ' * ' + nombre2 + ' = ' + (nombre1 * nombre2);
}

/*On appelle ou "invoque" ou encore "exécute" nos fonctions et on place les
 *résultats retournés dans les paragraphes p id='p1' et p id='p2' de notre
 *fichier HTML.
 *On fournit ici deux arguments à multiplication() pour que la fonction
 *s'exécute normalement. Ces arguments vont prendre la place des paramètres*/
document.getElementById('p1').innerHTML = aleatoire();
document.getElementById('p2').innerHTML = multiplication(5, 10);
```

Portée des variables

La portée d'une variable désigne l'espace dans lequel elle sera accessible.

Il existe 2 espaces principaux, l'espace global (ensemble de la page js)
et l'espace local (uniquement au sein de la fonction).

Portée des variables

```
//On déclare deux variables globales
let x = 5;
var y = 10;

//On définit une première fonction qui utilise les variables globales
function portee1(){
    document.getElementById('p1').innerHTML =
    'Depuis portee1() : <br>x = ' + x + '<br>y = ' + y;
}
//On définit une deuxième fonction qui définit des variables locales
function portee2(){
    let a = 1;
    var b = 2;
    document.getElementById('p2').innerHTML =
    'Depuis portee2() : <br> a = ' + a + '<br>b = ' + b;
}
//On définit une troisième fonction qui défnit également des variables locales
function portee3(){
    let x = 20;
    var y = 40;
    document.getElementById('p3').innerHTML =
    'Depuis portee3() : <br>x = ' + x + '<br>y = ' + y;
}

//On pense bien à exécuter nos fonctions !
portee1();
portee2();
portee3();

//On tente d'afficher des variables globales puis locales depuis l'espace global
document.getElementById('p4').innerHTML =
'Depuis l\'espace global : <br>x = ' + x + '<br>y = ' + y;

document.getElementById('p5').innerHTML =
'Depuis l\'espace global : <br>a = ' + a + '<br>b = ' + b;
```

Titre principal

Un paragraphe

Depuis portee1() :
x = 5
y = 10

Depuis portee2() :
a = 1
b = 2

Depuis portee3() :
x = 20
y = 40

Depuis l'espace global :
x = 5
y = 10

Portée des variables

```
function portee1(){
    let x = 1;
    var y = 2;
    if(true){
        let x = 5; //Variable différente
        var y = 10; //Même variable qu'au dessus
        document.getElementById('p1').innerHTML = 'x (dans if) = ' + x;
        document.getElementById('p2').innerHTML = 'y (dans if) = ' + y;
    }
    document.getElementById('p3').innerHTML = 'x (hors if) = ' + x;
    document.getElementById('p4').innerHTML = 'y (hors if) = ' + y;
}

portee1();
```

Titre principal

Un paragraphe

x (dans if) = 5

y (dans if) = 10

x (hors if) = 1

y (hors if) = 10

Les variables "let" se définissent dans leur bloc et sous blocs.
Les variables "var" se définissent dans l'ensemble de la fonction.

Retour des fonctions

La valeur de retour "return" d'une fonction est la valeur renvoyée par celle-ci lorsqu'elle a terminée son exécution.

Cette valeur pourra par la suite être utilisée dans le reste de l'application.

L'instruction return n'est pas obligatoire, mais si présente elle doit être présente à la fin de la fonction (stop le reste de l'éxécution)

Exercice d'application n°3 : Fonctions

Déplacez le code de l'exercice précédent au sein d'une fonction Exo2().

Créez une nouvelle fonction Exo3() au sein de votre fichier.

Dupliquez le code de l'exercice précédent dans cette nouvelle fonction, et modifiez le de sorte à demander le nombre d'enfant au sein d'une fonction getChild(), et que le calcul de la catégorie se fasse dans une fonction getCategorie().