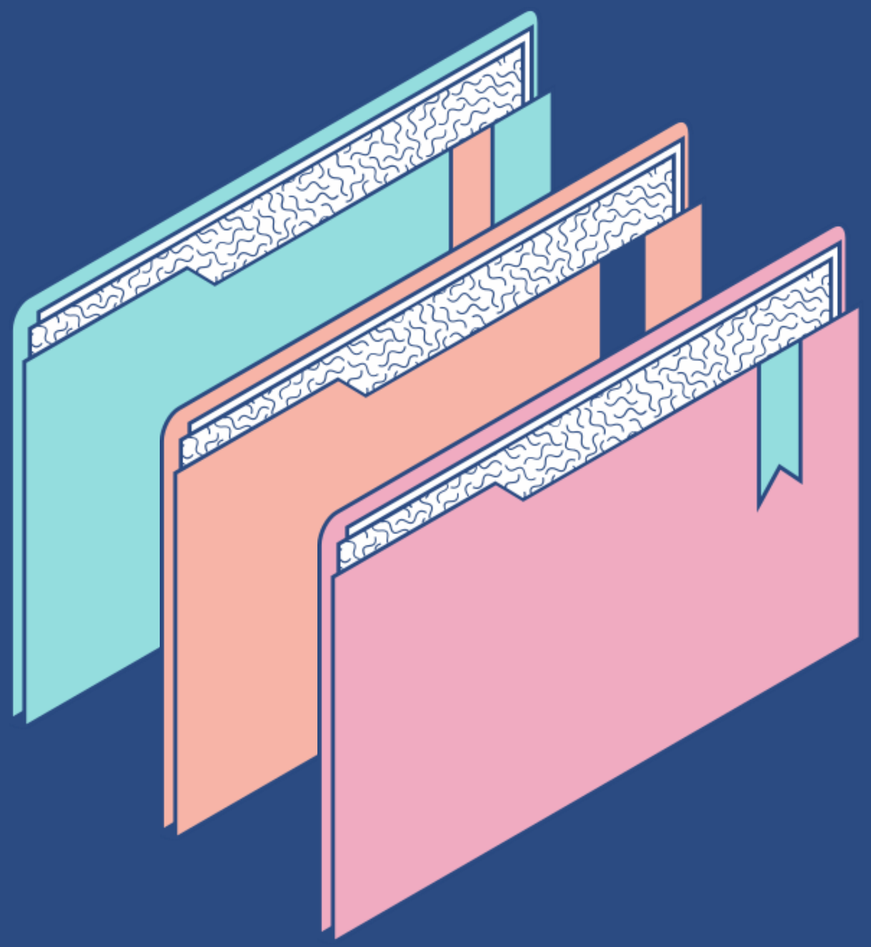
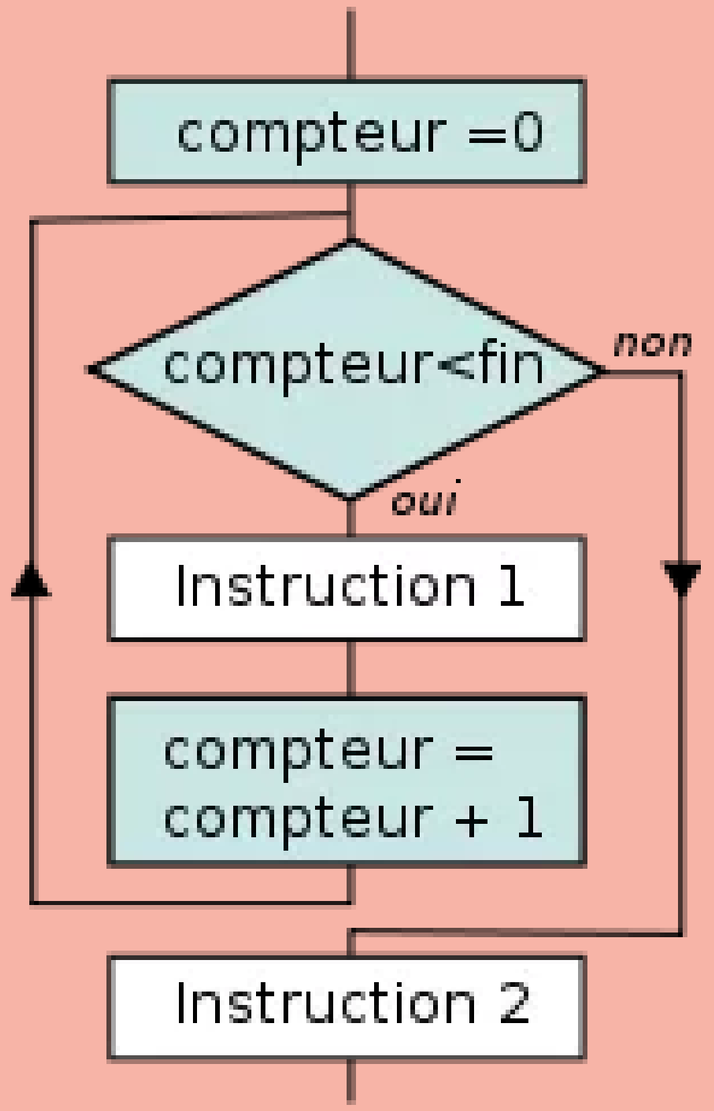
Javascript


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

SOMMAIRE



Structures de contrôle



Opérateurs de comparaison

Opérateur	Définition
==	Permet de tester l'égalité sur les valeurs
===	Permet de tester l'égalité en termes de valeurs et de types
!=	Permet de tester la différence en valeurs
<>	Permet également de tester la différence en valeurs
!==	Permet de tester la différence en valeurs ou en types
<	Permet de tester si une valeur est strictement inférieure à une autre
>	Permet de tester si une valeur est strictement supérieure à une autre
<=	Permet de tester si une valeur est inférieure ou égale à une autre
>=	Permet de tester si une valeur est supérieure ou égale à une autre

if...else...else if... else

```
let x = 0.5;

if(x > 1){
    alert('x contient une valeur strictement supérieure à 1');
}else if(x == 1){
    alert('x contient la valeur 1');
}else{
    alert('x contient une valeur strictement inférieure à 1');
}
```

Opérateurs logiques

Opérateur (nom)	Opérateur (symbole)	Description
AND (ET)	&&	Lorsqu'il est utilisé avec des valeurs booléennes, renvoie true si toutes les comparaisons sont évaluées à true ou false sinon
OR (OU)	||	Lorsqu'il est utilisé avec des valeurs booléennes, renvoie true si au moins l'une des comparaisons est évaluée à true ou false sinon
NO (NON)	!	Renvoie false si une comparaison est évaluée à true ou renvoie true dans le cas contraire

Opérateurs logiques

```
let x = 5;
let y = -20;

if(x >= 0 && x <= 10){
    document.getElementById('p1').innerHTML =
    'x contient une valeur comprise entre 0 et 10';
}

if(y < -10 || y > 10){
    document.getElementById('p2').innerHTML =
    'y contient une valeur plus petite que -10 ou plus grande que 10';
}

if(!(x <= 2)){
    document.getElementById('p3').innerHTML =
    'x contient une valeur strictement supérieure à 2';
}
```

Conditions ternaires

Les conditions ternaires permettent de simplifier l'écriture de conditions simples

```
let x = 15;

//Utilisation de l'opérateur ternaire
document.getElementById('p1').innerHTML =
x >= 10 ? 'x supérieur à 10' : 'x stric. inférieur à 10';

//Equivalent avec une structure if ... else
if(x >= 10){
    document.getElementById('p2').innerHTML = 'x supérieur à 10';
}else{
    document.getElementById('p2').innerHTML = 'x stric. inférieur à 10';
}
```

Switch … case

```
let x = 15;

switch(x){
    case 2:
        document.getElementById('p1').innerHTML = 'x stocke la valeur 2';
        break;
    case 5:
        document.getElementById('p1').innerHTML = 'x stocke la valeur 5';
        break;
    case 10:
        document.getElementById('p1').innerHTML = 'x stocke la valeur 10';
        break;
    case 15:
        document.getElementById('p1').innerHTML = 'x stocke la valeur 15';
        break;
    case 20:
        document.getElementById('p1').innerHTML = 'x stocke la valeur 20';
        break;
    default:
        document.getElementById('p1').innerHTML =
        'x ne stocke ni 2, ni 5, ni 10, ni 15 ni 20';
}
```

Boucles

Il existe 6 types de boucles différentes en JS :

- While
- Do … While
- For
- For … in
- For … of
- For await … of

```
let list = [4, 5, 6];

for (let i in list) {
   console.log(i); // "0", "1", "2",
}

for (let i of list) {
   console.log(i); // "4", "5", "6"
}
```

Opérateurs d'incrément

Exemple (opérateur + variable)	Résultat
`++x`	Pré-incrémentation : incrémente la valeur contenue dans la variable x, puis retourne la valeur incrémentée
`x++`	Post-incrémentation : retourne la valeur contenue dans x avant incrémentation, puis incrémente la valeur de $x
`--x`	Pré-décrémentation : décrémente la valeur contenue dans la variable x, puis retourne la valeur décrémentée
`x--`	Post-décrémentation : retourne la valeur contenue dans x avant décrémentation, puis décrémente la valeur de $x

Exercice d'application n°2 : Structures de contrôle

Ecrivez un script qui demande la quantité d'enfant.
Pour chaque enfant, le script demandera l'âge de l'enfant, et affichera sa catégorie :

- « Poussin » de 6 à 7 ans
- « Pupille » de 8 à 9 ans
- « Minime » de 10 à 11 ans
- « Cadet » de 12 ans à 17 ans

Le script affichera une erreur si l'âge saisi n'est pas dans la catégorie.

La demande de l'age pourra s'effectuer à l'aide de la méthode prompt()