

## Les conditions et les boucles en JavaScript

### Instruction **if...else** en JavaScript



#### Instructions conditionnelles

Les instructions conditionnelles sont utilisées pour exécuter du code en fonction de différentes conditions.

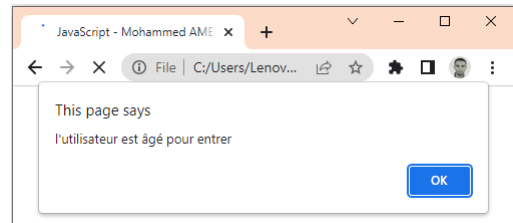
L'instruction **if** exécute un bloc de codes si la condition donnée est vraie.

```
if(condition){  
    // bloc de codes à exécuter lorsque la condition est vraie  
}
```

L'instruction **else** exécute un bloc de codes si la condition donnée est fausse.

```
if (condition) {  
    // bloc de codes à exécuter lorsque la condition est vraie  
} else {  
    // bloc de codes à exécuter lorsque la condition est fausse  
}
```

```
<script>  
    var age = 26;  
    if (age > 18) {  
        alert("l'utilisateur est âgé pour entrer ");  
    } else {  
        alert("l'utilisateur n'a pas l'âge requis pour entrer ");  
    }  
</script>
```

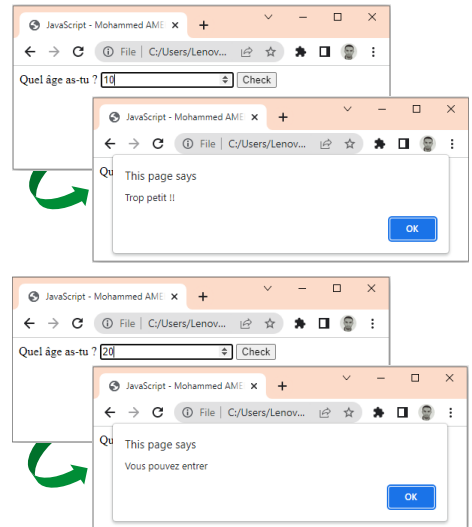


# Instruction **if...else** en JavaScript



## Instructions conditionnelles

```
<form>
  <label for="age">Quel âge as-tu ?</label>
  <input type="number" id="age" min="1">
  <button type="button" id="submit">Check</button>
</form>
<script>
  document.getElementById("submit").addEventListener("click", function() {
    var age = document.getElementById("age").value;
    if (age > 18) {
      alert("Vous pouvez entrer");
    } else {
      alert("Trop petit !!")
    }
  });
</script>
```



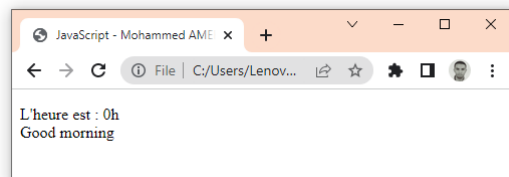
# Instruction **if...else** en JavaScript



## Instructions conditionnelles

Utilisez **else if** pour spécifier une nouvelle condition à tester, si la première condition est fausse.

```
<p id="demo"></p>
<script>
  const heure = new Date().getHours();
  let salutation;
  if (heure < 12) {
    salutation = "Good morning";
  } else if (heure < 18) {
    salutation = "Good afternoon";
  } else if (heure < 22) {
    salutation = "Good evening";
  } else {
    salutation = "Good night";
  }
  document.getElementById("demo").innerHTML =
    "L'heure est : " + heure + "h <br>->" + salutation;
</script>
```



## Instruction **switch** en JavaScript

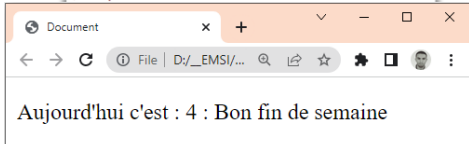


En programmation, il arrive que l'on ait besoin de plusieurs instructions conditionnelles « if ».

Au lieu d'écrire plusieurs instructions if, nous pouvons raccourcir le code avec l'instruction « **switch** ».

- L'expression de switch est évaluée une fois.
- La valeur de l'expression est comparée aux valeurs de chaque cas.
- S'il y a correspondance, le bloc de code associé doit être exécuté.
- S'il n'y a pas de correspondance, le bloc de code par défaut est exécuté.
- **break** ; Lorsque JS atteint un break mot-clé, il sort du bloc. Cela arrêtera l'exécution à l'intérieur du bloc de switch.
- **default** ; spécifie le code à exécuter s'il n'y a pas de correspondance de casse.

```
<p id="demo"></p>
<script>
let msg;
var jour = new Date().getDay()
switch (jour) {
  case 1:
    msg = "Bon début de semaine";
    break;
  case 2:
    msg = "Bon début de semaine";
    break;
  case 3:
  case 4:
  case 5:
    msg = "Bon fin de semaine";
    break;
  default:
    msg = "C'est de Week-end";
}
document.getElementById("demo").innerHTML =
  "Aujourd'hui c'est : " + jour + " : " + msg;
</script>
```

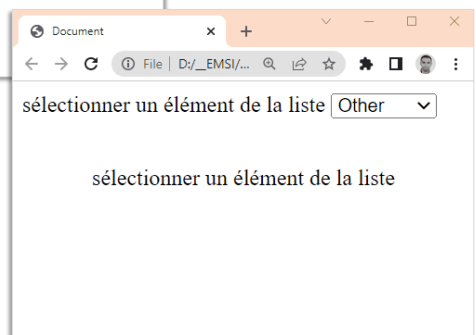


## Instruction **switch** en JavaScript



```
function checkFruit() {
  var fruit = document.getElementById('fruits').value;
  switch (fruit) {
    case "pomme":
      var msg = "10 DH";
      document.getElementById('msg').innerHTML = msg;
      document.getElementById('image').style.display = 'inline-block';
      document.getElementById('image').src = 'pomme.png';
      break;
    case "frais":
      var msg = "15 DH";
      document.getElementById('msg').innerHTML = msg;
      document.getElementById('image').style.display = 'inline-block';
      document.getElementById('image').src = 'frais.png';
      break;
    case "orange":
      var msg = "7 DH";
      document.getElementById('msg').innerHTML = msg;
      document.getElementById('image').style.display = 'inline-block';
      document.getElementById('image').src = 'orange.png';
      break;
    case "avocado":
      var msg = "25 DH";
      document.getElementById('msg').innerHTML = msg;
      document.getElementById('image').style.display = 'inline-block';
      document.getElementById('image').src = 'avoca.png';
      break;
    default:
      document.getElementById('image').style.display = 'none';
      document.getElementById('msg').innerHTML =
        "sélectionner un élément de la liste";
      break;
  }
}
```

```
<body>
<label for="fruits">sélectionner un élément de la liste</label>
<select id="fruits" onchange="checkFruit()">
  <option value="other">Other</option>
  <option value="pomme">Pomme</option>
  <option value="frais">Frais</option>
  <option value="orange">Orange</option>
  <option value="avocado">Avocado</option>
</select>
<br>
<p>
  <img src="" height="100" id="image">
  <br><span id="msg"></span>
</p>
<script src="script.js"></script>
</body>
```



## Les boucles en JavaScript {for}

- Lorsque nous programmons, il arrive que nous ayons besoin d'exécuter la même instruction à plusieurs fois.
- une boucle for exécute les instructions répétitivement tant que la condition indiquée est vérifiée.

```
var somme=0;  
for( var i=1 ; i<=10 ; i++ ){  
    somme += i;  
}
```



Créez une boucle for allant de 0 à 9. Et affichez le résultat dans la console de vos navigateurs.

Syntaxe:

```
for(initialisation ; condition ; opération ){  
    ... instructions ...  
}
```

- Initialisation: définit une variable avant le début de la boucle
- Condition: définit la condition d'exécution de la boucle
- Opération: incrémentation ou décrémentation par exemple

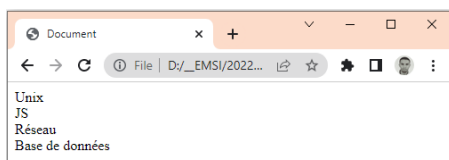
## Les boucles en JavaScript {for ... of}

- L'instruction for ... of effectue un bouclage sur des objets itérables.
- Les objets itérables les plus basiques sont les tableaux et les chaînes de caractères.
- Syntaxe :

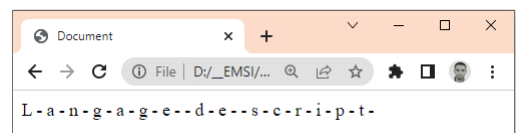
```
for(variable of iterable){  
    ... instructions ...  
}
```

Exemple :

```
var matieres = ["Unix", "JS", "Réseau", "Base de données"];  
for (m of matieres) {  
    document.write(m + "<br>");  
}
```



```
var str = "Langage de script";  
for (l of str) {  
    document.write(l + " - ");  
}
```



## Les boucles en JavaScript {while & do...while}



Tant qu'une condition donnée est vraie, la boucle **while** exécute les instructions.

Syntaxe :

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

Exemple :

```
function demander(){  
    var nb=0;  
    while(nb<=100){  
        nb = prompt("Entrez un nombre > 100");  
    }  
    alert("Merci !");  
}
```

Copiez la fonction illustrée dans l'exemple dans un fichier JS. Ensuite, appelez-la dans un fichier HTML et décrivez son fonctionnement.

La boucle **do...while** exécute les instructions tant que la condition donnée est vraie.

Syntaxe :

```
do{  
    ... instructions ...  
} while(condition)
```



Refaire l'exemple précédent à l'aide de la boucle **do...while**.  
Qu'elle est la différence entre la boucle **while** et la boucle **do...while** ?