


A small yellow square containing the letters 'JS' in a bold, dark grey, sans-serif font.

Web Developer Scripts Clients

A large yellow square containing the letters 'JS' in a bold, dark grey, sans-serif font.

03 – Les structures conditionnelles



Les conditions

Afin de nous permettre d'exécuter du code seulement si une (ou plusieurs) condition est remplie, nous devons aborder la notion de structures conditionnelles mais également les opérateurs de comparaison et logiques.



Les opérateurs de comparaison

Il existe 8 opérateurs de comparaison :

==	Égalité des valeurs
>	Supériorité stricte
>=	Supériorité
<	Infériorité stricte
<=	Infériorité
!=	Différence des valeurs
===	Égalité des valeurs et des types
!==	Différence des valeurs ou des types

Le résultat d'une comparaison est un booléen



Les opérateurs de comparaison

Test du résultat d'une comparaison :

```
let result_compar;
```

```
result_compar = 3 > 4;
```

```
console.log(` ${result_compar} ${typeof result_compar} `);
```

```
result_compar = 3 < 4;
```

```
console.log(` ${result_compar} ${typeof result_compar} `);
```



Les opérateurs de comparaison

Différence entre `==` et `===`

```
let val1 = 2;  
let val2 = '2';  
  
let result_compar;  
  
result_compar = (val1 == val2);  
console.log(` ${result_compar} ${typeof result_compar} `);  
  
result_compar = (val1 === val2);  
console.log(` ${result_compar} ${typeof result_compar} `);
```



Les opérateurs de comparaison

Le `==` réalise une conversion de type, et seules les valeurs sont alors comparées : égalité faible.

Le `===` ne réalise pas de conversion de type, les valeurs et les types sont alors comparés : égalité stricte.



La sélection

La première structure conditionnelle est la sélection.

Elle nous permettra de n'exécuter un bloc d'instructions que « si » une (ou plusieurs) condition est vraie.

```
if(condition(s)){  
    instruction(s)  
}
```



La sélection

Exemple

Affichage d'un message si un défaut est présent :

```
let default = 1;  
...  
if(default === 1){  
    console.log('Défaut détecté');  
}
```




La sélection

Exemple

Sélection du mode sombre d'une page :

```
const isDarkMode = confirm("Activer le mode sombre ?");  
if (isDarkMode) {  
    document.body.style.background = "black";  
}
```



Les opérateurs logiques

Il est possible de combiner plusieurs conditions grâce aux opérateurs logiques.

Il existe 3 opérateurs logiques :

&&	ET
	OU
!	NON



Les opérateurs logiques

Table de vérité du ET :

A	B	A && B
0	0	0
0	1	0
1	0	0
1	1	1



Les opérateurs logiques

Table de vérité du OU :

A	B	$A \parallel B$
0	0	0
0	1	1
1	0	1
1	1	1



Les opérateurs logiques

Table de vérité du NON :

A	!A
0	1
1	0



Les opérateurs logiques

Les opérations suivantes constituent des courts-circuits :

- `true || « quelque chose »` est court-circuité en `true`, ce qui signifie que « quelque chose » n'est pas évalué
- `false && « quelque chose »` est court-circuité en `false`, ce qui signifie que « quelque chose » n'est pas évalué



L'alternative

La deuxième structure conditionnelle est l'alternative.

Elle nous permettra d'exécuter un bloc d'instructions « si » une (ou plusieurs) condition est vraie, un second bloc « sinon » :

```
if(condition(s)){  
    instruction(s) 1  
}  
else{  
    instruction(s) 2  
}
```



L'alternative

Exemple

Affichage d'un message selon l'âge :

```
const MAJORITE = 18;
let age = Number(prompt('Quel est votre âge ?'));

if (age >= MAJORITE) {
  console.log('Vous êtes majeur(e)');
}
else {
  console.log('Vous êtes mineur(e)');
}
```




L'alternative

Il est possible d'imbriquer les structures conditionnelles et/ou de les juxtaposer :

```
if(condition(s) 1){  
    if(condition(s) 2){ // Imbrication  
        instruction(s) 1  
    }  
    else{  
        instruction(s) 2  
    }  
}  
else if(condition(s) 3){ // Juxtaposition  
    instruction(s) 3  
}  
else {  
    instruction(s) 4  
}
```



L'alternative

L'alternative peut être formalisée par l'opérateur ternaire :

```
condition(s) ? instruction(s) 1 : instruction(s) 2
```

Est équivalent à :

```
if(condition(s)){  
    instruction(s) 1  
}  
else {  
    instruction(s) 2  
}
```



L'alternative

Exemple

Affichage d'un message selon l'âge :

```
const MAJORITE = 18;  
let age = Number(prompt('Quel est votre âge ?'));  
  
age >= MAJORITE ? console.log('Vous êtes majeur(e)') :  
console.log('Vous êtes mineur(e));
```



L'alternative

Exemple

Affichage d'un message si l'utilisateur est connecté :

```
const connecte = true;  
console.log(`Utilisateur ${connecte ? "connecté" : "déconnecté"}`);
```



Le choix multiple

La troisième structure conditionnelle est le choix multiple.

Elle nous permettra de sélectionner un bloc d'instructions en fonction de la valeur et du type d'une variable parmi une liste de valeurs (égalité stricte) :

```
switch (variable) {  
  case valeur_1:  
    instruction(s) 1  
    [break;]  
  ...  
  default:  
    instruction(s)  
}
```



Le choix multiple

Exemple

Affichage du nom du mois :

```
let mois = parseInt(prompt('Quel mois choisissiez vous ?'));

switch (mois) {
  case 1:
    console.log('Vous avez choisi janvier');
    break;
  case 2:
    console.log('Vous avez choisi février');
    break;
  ...
  default:
    console.log('Vous avez choisi un mois qui n\'existe pas');
}
```



Le choix multiple

Il est également possible d'utiliser le choix multiple en associant un bloc d'instructions à plusieurs valeurs :

```
let mois = parseInt(prompt('Quel mois choisissez vous ?'));

switch (mois) {
  case 1:
  case 3:
  case 5:
  case 7:
  case 8:
  case 10:
  case 12:
    console.log('Mois de 31 jours');
    break;
  case 4:
  case 6:
  case 9:
  case 11:
    console.log('Mois de 30 jours');
    break;
  case 2:
    console.log('Mois de 28 ou 29 jours');
    break;
  default:
    console.log('Vous avez choisi un mois qui n\'existe pas');
}
```



Le choix multiple

Si on désire utiliser les « case » avec des conditions et non plus des valeurs, il faut passer la valeur `true` au `switch` :

```
switch (true) {  
  case conditions_1:  
    instruction(s) 1  
    [break;]  
  ...  
  default:  
    instruction(s)  
}
```

NB : À éviter et privilégier des `if-else` pour la lisibilité



Le choix multiple

Exemple :

```
switch (true) {  
  case points >= 18:  
    console.log('La plus grande distinction');  
    break;  
  case points >= 16:  
    console.log('Grande distinction');  
    break;  
  ...  
  default:  
    ...  
}
```



La confirmation

Nous pouvons maintenant utiliser un troisième type de dialogue, la confirmation, avec l'instruction `confirm` :

```
let choix = confirm('Voulez-vous continuer ?');  
  
if(choix === true){  
  ...  
}  
else {  
  ...  
}
```



Bonnes pratiques

- Utiliser `===` et `!==` (éviter `==` et `!=`)
- Écrire des conditions claires et simples
 - Éviter l'évidence : `if (a === true)`
 - Préférer la concision : `if (a)`
- Bien indenter les blocs `{ ... }` pour la lisibilité
- Préférer `if / else if / else` pour la clarté
- Utiliser le ternaire uniquement pour des expressions courtes
- Dans un `switch`, ne pas oublier `break` après chaque `case`
- Tester les conditions avec `console.log()` pour vérifier les valeurs
- Regrouper les cas similaires dans un `switch` pour éviter les répétitions



Exercices

3.1.1 Écrire un algorithme qui détermine si un nombre entier lu au clavier est nul ou pas. L'algorithme affichera le message approprié parmi les suivants: "Le nombre est nul", "Le nombre n'est pas nul".

3.1.2 Écrire un algorithme qui détermine si un nombre entier non nul lu au clavier est positif ou négatif. L'algorithme affichera le message approprié parmi les suivants: "Le nombre est positif", "Le nombre est négatif".

3.1.3 Écrire un algorithme qui détermine si un nombre entier non négatif lu au clavier est pair ou impair. L'algorithme affichera le message approprié parmi les suivants: "Le nombre est pair", "Le nombre est impair".

3.1.4 Écrire un algorithme qui détermine si un nombre entier lu au clavier est égal à 30, 40 ou 50. L'algorithme affichera le message approprié parmi les suivants: "Valeur correcte", "Valeur incorrecte".



Exercices

3.1.5 Écrire un algorithme qui détermine si un nombre entier lu au clavier est nul, positif ou négatif. L'algorithme affichera le message approprié parmi les suivants: "Nombre positif", "Nombre négatif", "Nombre nul".

3.1.6 Écrire un algorithme qui recherche et affiche le maximum parmi 5 nombres entiers lus au clavier. La recherche du maximum se fera au fur et à mesure des entrées successives. Il n'est donc pas question d'utiliser 5 données différentes pour mémoriser les valeurs des 5 nombres, l'algorithme n'utilisera pour ce faire qu'une seule variable.



Exercices

3.1.7 Écrire un algorithme qui chiffre un caractère lu au clavier d'après la règle suivante :

- Chaque lettre est remplacée par sa suivante dans l'alphabet
- Z sera remplacé, par A, z par a (une majuscule reste majuscule, une minuscule reste minuscule)
- Chaque chiffre est remplacé par son suivant (9 sera remplacé par 0)
- Tout autre caractère sera remplacé par le caractère *
- L'algorithme affichera le caractère crypté
- Obtenir la valeur Unicode d'un caractère :

```
let str = 'pomme';
```

```
let code = str.charCodeAt(0); // premier caractère de la variable str
```

- Obtenir un caractère à partir d'une valeur Unicode :

```
let caractere = String.fromCharCode(code+1); // retourne 'q'
```



Exercices

3.1.8 Écrire un algorithme qui calcule le montant d'une facture comportant une seule ligne de facture, connaissant le prix unitaire de l'article et le nombre d'articles achetés. La TVA est uniformément de 21%. Une remise sera calculée sur le montant HTVA d'après la règle suivante :

- 100 articles et plus: 10%
- 1000 articles et plus:
 - 20% si le montant HTVA est d'au moins 1000€
 - 15% sinon



Exercices

L'algorithme affichera le nombre d'articles, le prix unitaire, le montant HTVA sans remise, la remise, le montant HTVA avec remise, le taux de TVA, le montant de la TVA et le montant TVAC. Exemple :

FACTURE.

Nombre d'articles :

Prix unitaire :

Montant HTVA avant remise :

Remise :

Montant HTVA avec remise :

TAUX TVA (21%) :

Montant TVA :

Montant TVAC :



Exercices

3.1.9 Écrire un algorithme qui lit au clavier le millésime d'une année et détermine si c'est une année bissextile.

3.1.10 Écrire un algorithme qui calcule le prix d'une consommation d'eau selon le tarif suivant :

- les 30 premiers m^3 sont gratuits
- les volumes allant de 31 à 500 m^3 : 0,5€/m³
- les volumes allant de 501 à 1000 m^3 : 0,4€/m³
- les volumes suivants : 0,25€/m³
- avec un forfait de 15€ et une TVA de 21%