

TP Formulaire Paiement

Première logique avec les structures de contrôle

Choix des articles

- Ours en peluche



15 euros

Quantity:

2

- Voiture push and go (10 Euros)



10 euros

Quantity:

5

Panier

- Ours en peluche $15 \times 2 = 30$
- Voiture push and go $10 \times 5 = 50$
- Epée en mousse $8 \times 3 = 24$

Total = 104 euros

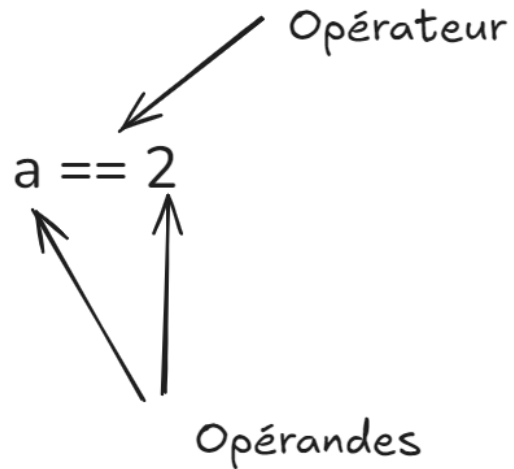
TTC = 124 euros

Mode de paiement

☐ Visa Card ☐ Master Card ☒ Paypal

Opérateurs de comparaison

Un opérateur logique retourne uniquement true ou false. (opérateur booléen)



Opérateurs d'égalité

L'opérateur d'égalité compare l'équivalence de deux opérandes.

- `==` (égal à) : Vérifier si deux valeurs sont égales.
- `!=` (différent de) : Vérifier si deux valeurs sont différentes.

Egalité strict

L'opérateur d'égalité stricte (`===`) vérifie si ses deux opérandes sont égales et renvoie un booléen correspondant au résultat. À la différence de l'opérateur d'égalité, l'opérateur d'égalité stricte considère toujours des opérandes de types différents comme étant différents.

```
console.log(1 === 1);  
// Expected output: true  
  
console.log('hello' === 'hello');  
// Expected output: true  
  
console.log('1' === 1);  
// Expected output: false  
  
console.log(0 === false);  
// Expected output: false  
  
/*----- Alors que : -- */  
  
console.log(1 == 1);  
// Expected output: true  
  
console.log('hello' == 'hello');  
// Expected output: true  
  
console.log('1' == 1);  
// Expected output: true  
  
console.log(0 == false);  
// Expected output: true
```

Opérateurs de relation

Un opérateur relationnel est utilisé pour établir une relation entre deux opérandes, comme "plus grand que" (>) ou "moins que" (<).

- > (plus grand que)
- < (moins que)
- >= (plus grand ou égal)
- <= (moins que ou égal)

```
document.write(1 > 2); // false
document.write('<br>');
document.write(1 >= 1); // true
document.write('<br>');
document.write(1 < 2); // true
document.write('<br>');
document.write(1 <= 2); // true
document.write('<br>');
document.write(1 == 1); // true
document.write('<br>');
document.write(1 != 2); // true
```

If / else

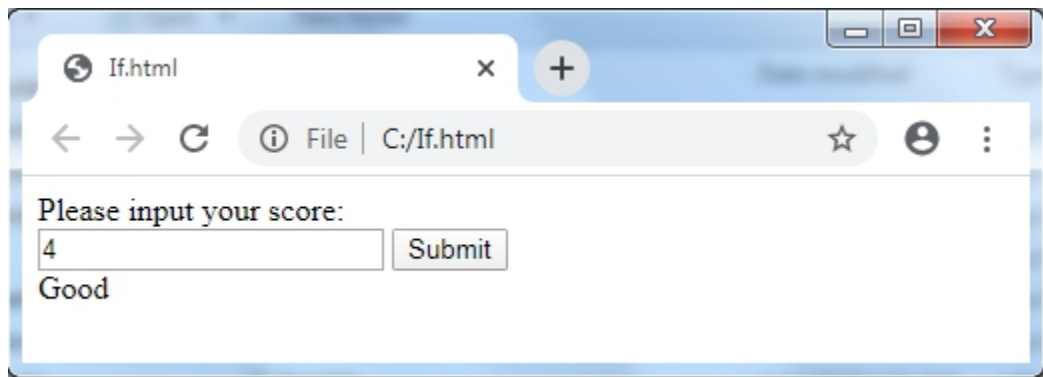
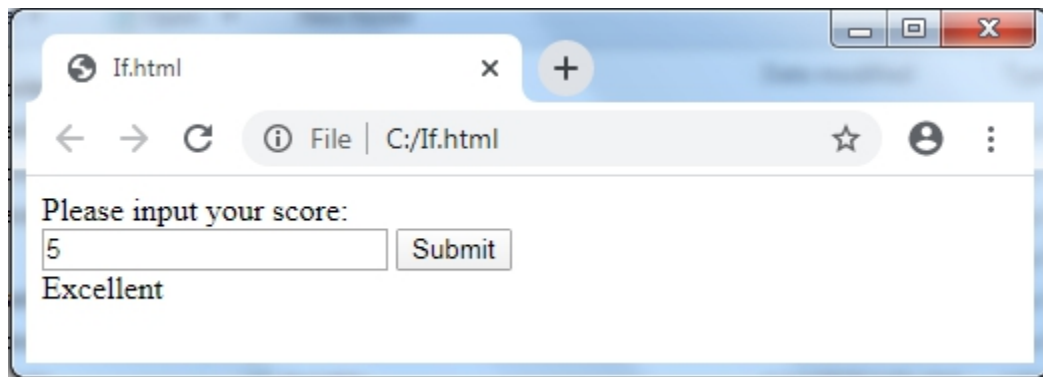
Syntaxe

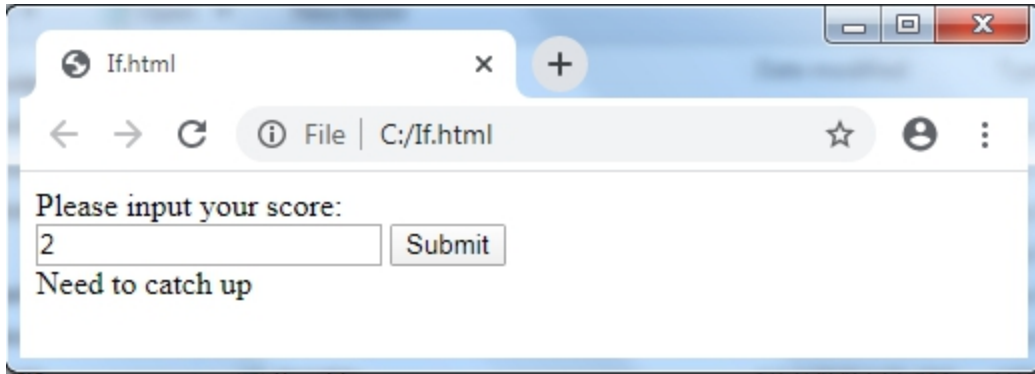
1. if (expression){}
2. if (expression) { } else { }
3. if (expression) { } else if { }

Application

Ecrire un script qui suit l'algorithme suivant :

```
if score == 5 : 'Excellent'  
else if score == 4 : 'Good'  
else : 'Need to catch up'
```





Solution :

if.html

```
<p>Please input your score : </p>
<input type="text" id="score" value="" />
<input type="button" value="Submit" onClick="doSubmit()" />
<br />
<span id="message"></span>
<script type="text/javascript">
  function doSubmit() {
    let elementScore= document.getElementById("score");
    let elementMessage= document.getElementById("message");
    let score = parseInt(elementScore.value);
    if (score == 5) {
      elementMessage.innerHTML = "Excellent";
    } else if (score == 4) {
      elementMessage.innerHTML = "Good";
    } else {
      elementMessage.innerHTML = "Need to catch up";
    }
  }
</script>
```

Explication :

`innerHTML` permet d'afficher et de modifier le contenu HTML d'un élément.

```
let content = element.innerHTML; // Accède au contenu
```

```
element.innerHTML = "new content"; // Modifie le contenu
```

Boîte de confirmation `confirm()`

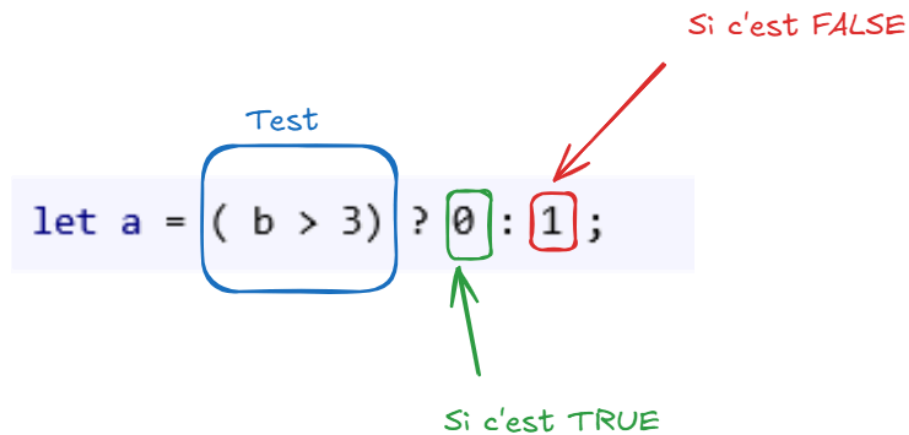
`confirm()` : Affiche un message avec deux options, "OK" et "Annuler". Elle est souvent utilisée pour demander une confirmation à l'utilisateur. Cette boîte renvoie `true` si l'utilisateur clique sur "OK", et `false` sur "Annuler".

```
let response = confirm("Voulez-vous continuer ?");
console.log(response);

if (response) {
    // L'utilisateur a cliqué sur OK
    document.write("On continue le shopping");
} else {
    // L'utilisateur a cliqué sur Annuler
    document.write("Merci au revoir");
}
```

Opérateurs ternaires

Cet opérateur permet d'affecter une valeur à une variable en fonction du résultat à un test. Sa syntaxe est la suivante :



```
let a = ( b > 3 ) ? 0 : 1 ;
```

ce qui est équivalent à

```
let a;  
if( b > 3 ) {  
  a = 0;  
} else {  
  a = 1;  
}
```

La ternaire fait un teste et une affectation

Exemple :

```
let message = confirm("Voulez-vous continuer ?")?'On continue':'Au Revoir';  
console.log(response);
```


Opérateurs Logiques

Opérateurs Logiques: et &&, ou ||, not !

l'opérateur logique AND &&

&& renvoie **true** si les deux côtés de l'opération sont **true** , sinon **false**

```
document.write( 1 > 2 && 3 > 4 ); // false AND true = false
document.write( 2 > 1 && 3 > 4 ); // true AND false = false
document.write( 2 > 1 && 4 > 3 ); // true AND true = true
```

L'opérateur logique OR ||

|| Le résultat est **false** lorsque les deux côtés de l'opération sont **false** , sinon **true** ;

```
document.write( 2 > 1 || 3 > 4 ) // true OR false = true
document.write( 1 > 2 || 4 > 3 ) // false OR true = true
document.write( 1 > 2 || 3 > 4 ) // false OR false = false
```

L'opérateur logique NOT !

! si retourne **true** , le résultat est **false** , sinon est **true**

```
document.write( ! (2 > 1) ) // NOT( true ) = false
document.write( ! (3 > 4) ) // NOT( false ) = true
```

Exemple de taxe sur la paie :

```
Tax = salary * tax rate
level:
500 -- 2000 $ : tax rate 10%
2000--5000 $ : tax rate 15%
5000-- 20000 $ : tax rate 20%
More than 20000$ : tax rate 30%
```


En javascript :

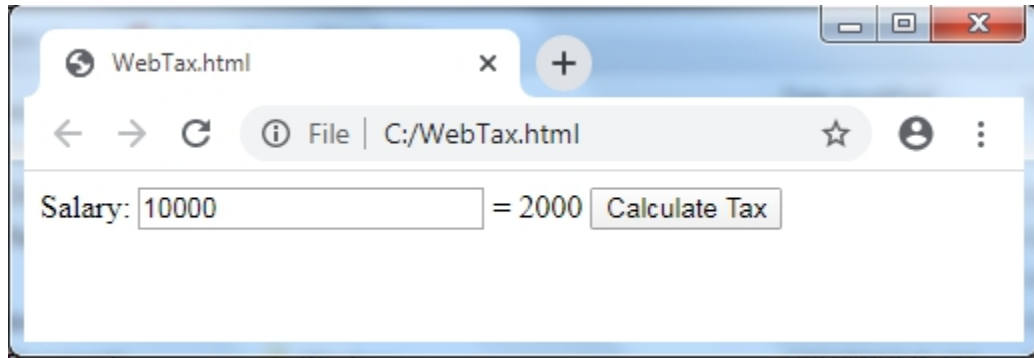
```
<script type="text/javascript" >
var salary = 10000 ;
var tax = 0 ;
if ( salary >= 500 && salary < 2000 ) {
tax = salary * 0.1 ;
} else if ( salary >= 2000 && salary < 5000 ) {
tax = salary * 0.15 ;
} else if ( salary >= 5000 && salary < 20000 ) {
tax = salary * 0.2 ;
} else {
tax = salary * 0.3 ;
}
document.write( "tax amount=" + tax);
</script>
```

Résultat :

tax amount = 2000

Application

 Exercice : Créez un fichier: **webTax.html**. Écrire un script qui permet de saisir le salaire, puis au clic du bouton “Calculate tax”, affiche la taxe à payer selon la tranche de salaire.



Switch

Algorithme :

```
Début ;  
Entrez un nombre n  
  
Si n = 1 Alors "Pay by Visa Card"  
Si n = 2 Alors "Pay by Master Card"  
Si n = 3 Alors "Pay by Paypal"  
Sinon dans tous les autres cas " Pay by face to face "  
Fin;
```

break : terminer le code à exécuter, puis sauter à l'extérieur.

continue: terminer le code à exécuter, puis continuer le code suivant.

```
<script>  
var num = 1;  
switch (num) {  
  case 1:  
    document.write('Pay by Visa Card');  
    break; // terminer le code à exécuter, puis sauter à l'extérieur.  
  case 2:  
    document.write('Pay by Master Card');  
    break;  
  case 3:  
    document.write('Pay by Paypal');  
    break;  
  default:  
    document.write('Pay by face to face');  
}  
</script>
```

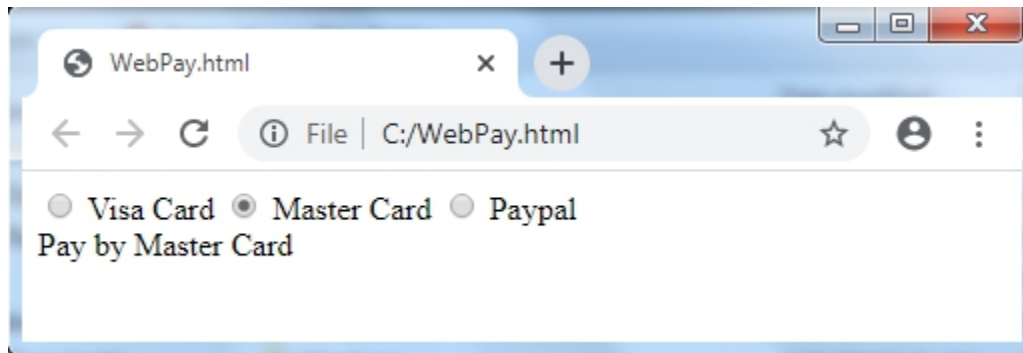
Résultat:

Pay by Visa Card

Application

📁 Créez un fichier: **WebPay.html**

this : pointe sur l'élément courant



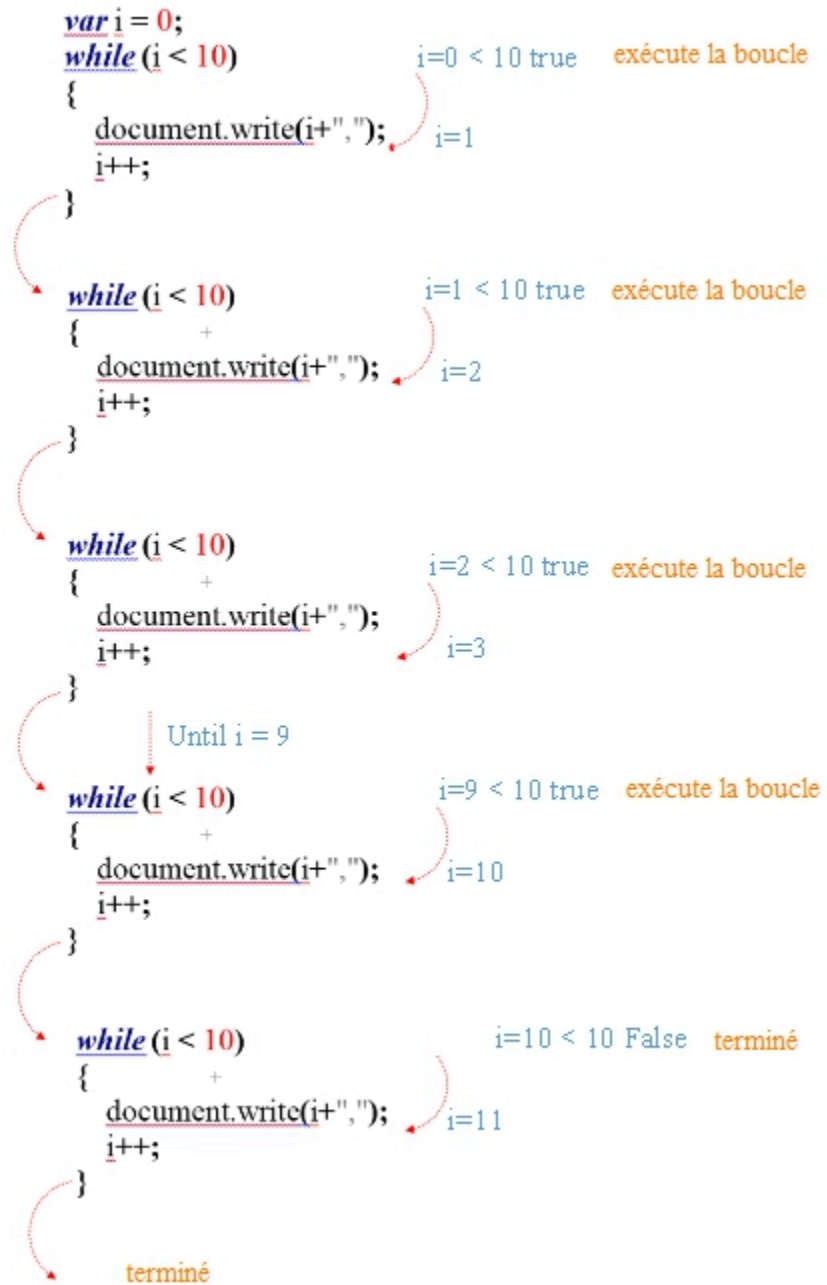
```
<input type="radio" name="card" value="1" onclick="doPay(this)" /> Visa Card
<input type="radio" name="card" value="2" onclick="doPay(this)" /> Master Card
<input type="radio" name="card" value="3" onclick="doPay(this)" /> Paypal
<br />
<span id="result"></span>
<script type="text/javascript">
  const elementResult = document.getElementById("result");

  function doPay(element) {
    let num = parseInt(element.value);
    switch (num) {
      case 1:
        elementResult.innerHTML = "Pay by Visa Card";
        break;
      case 2:
        elementResult.innerHTML = "Pay by Master Card";
        break;
      case 3:
        elementResult.innerHTML = "Pay by Paypal";
        break;
      default:
        elementResult.innerHTML = "Pay by face to face";
    }
  }
</script>
```

Boucle While

```
while (test){  
    // corps de la boucle  
}
```

Si le test est true, on exécute le corps de la boucle, sinon quitte la boucle



1. Créez un fichier: WhileLoop.html

```
<script>
let i = 0 ;
while ( i < 10 )
{
    document.write( i+ "," );
    i++;
}
</script>
```

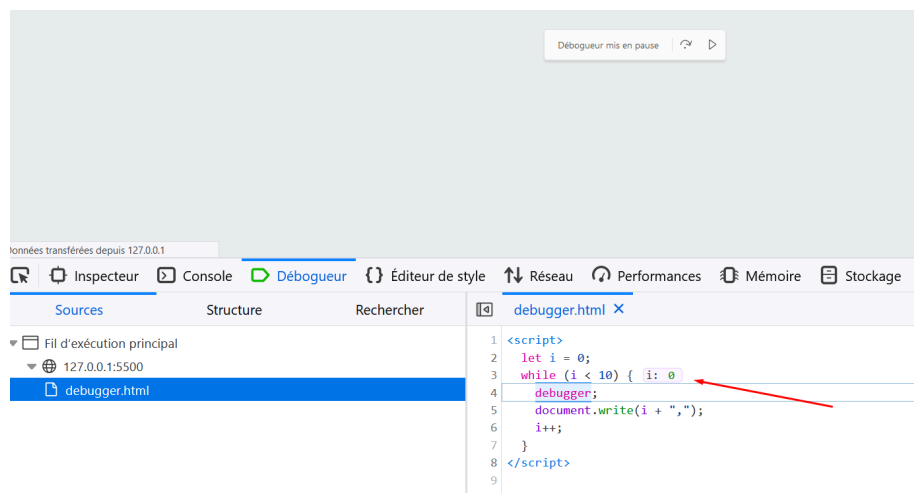
Résultat :

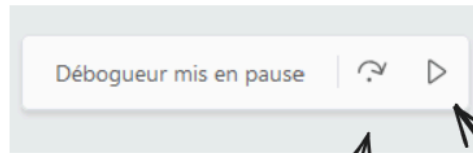
0,1,2,3,4,5,6,7,8,9,

C'est le bon moment pour tester le debugger;

```
<script>
let i = 0;
while (i < 10) {
    debugger;
    document.write(i + "," );
    i++;
}
</script>
```

Aller à l'onglet débogueur de votre navigateur






Pas à pas

Jusqu'au prochain
Break Point

Application :

 Exercice : Saisie répétée avec conditions multiples

Créer un programme qui demande à l'utilisateur de saisir une valeur jusqu'à ce que cette valeur corresponde à l'une des options prédéfinies.

Les valeurs acceptables sont "rouge", "vert" et "bleu"

Continuez à demander tant que la valeur saisie n'est pas l'une des trois valeurs acceptables. N'oubliez pas de convertir la saisie en minuscules ou en majuscules pour faciliter la comparaison.

Une fois qu'une valeur acceptable est saisie, affichez un message de confirmation et terminez le programme.

Exemple d'exécution :

```
Entrez une couleur (rouge, vert ou bleu) : jaune
Couleur non reconnue. Veuillez réessayer.
Entrez une couleur (rouge, vert ou bleu) : violet
Couleur non reconnue. Veuillez réessayer.
Entrez une couleur (rouge, vert ou bleu) : vert
Vous avez choisi une couleur valide : vert. Merci !
```

Bonus :

- Ajoutez un compteur pour afficher le nombre de tentatives à la fin.
- Permettez à l'utilisateur de quitter le programme à tout moment en saisissant "q" ou "quit"

1. Créez un fichier: ForLoop.html

```
<script>
for (var i = 0; i < 10; i++) {
    document.write(i + ',');
}
</script>
```

Résultat :

0,1,2,3,4,5,6,7,8,9,

Application



Exercice :

Réalisez un script qui permet de générer une liste d'articles aléatoire.

Liste des articles générés :

1. Article 1 - Prix : 45.67 €
2. Article 2 - Prix : 12.30 €
3. Article 3 - Prix : 78.90 €
4. Article 4 - Prix : 3.45 €
5. Article 5 - Prix : 99.99 €
6. Article 6 - Prix : 50.00 €
etc ...

Règle supplémentaire.

- Les articles se nomment simplement avec leur numéro. Exemple Article 1, Article 2 ...
- Générez un nombre aléatoire d'articles (entre 5 et 15).
- Générez un prix aléatoire entre 1 et 100 (vous pouvez utiliser des nombres décimaux pour plus de réalisme).
- Affichez la liste des articles générés avec leurs noms et leurs prix.

Incrémentation et décrémentation

📁 Créez un fichier: **incrementation.html**

Observez :

```
let d = 4;
console.log(d++); // 4 : imprime d puis incrémente
console.log(d); // 5 : imprime d
console.log(++d); // 6 : incrémente puis imprime d
console.log(d); // 6 : imprime d
console.log(d--); // 6 : imprime d puis décrémente 1
console.log(d); // 5 : imprime d
console.log(--d); // 4 : décrémente 1 puis imprime d
console.log(d); // 4 : imprime d
```

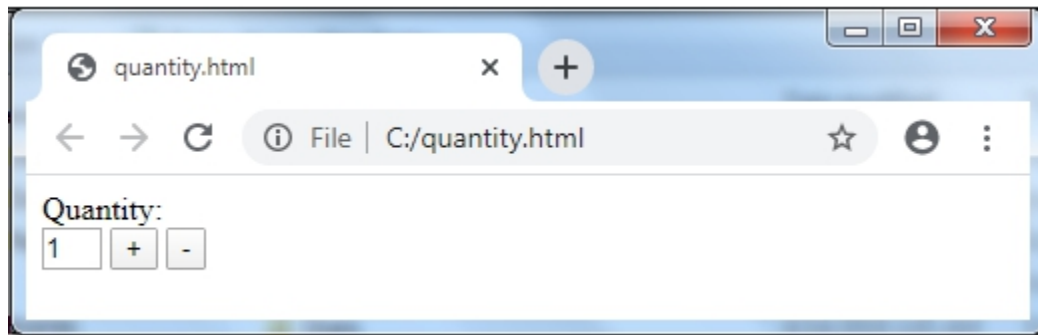
Observez :

```
let result = 10;
result = result + 1;
console.log(result); // = 11
//
result = 10;
result++;
console.log(result); // = 11
//
result = 10;
result += 1;
console.log(result); // = 11
```

Exercice

✏️ Créer un **additionneur** et un **soustracteur** de quantité: **quantity.html**

Cliquez sur le bouton + pour augmenter la quantité de 1, et cliquez sur le bouton - pour diminuer la quantité de 1.



Exercice de synthèse

Simulation d'une boutique en ligne interactive

Créer un programme qui simule une boutique en ligne interactive, permettant à l'utilisateur de sélectionner des articles, gérer un panier, et finaliser une commande

Partie 1. Génération et affichage des articles

- Placer 3 articles dans le HTML
- Pour chaque article, on affiche un nom, un prix aléatoire, et quantités initiales (mettez 0 pour toutes les quantités au départ).
- Utiliser un générateur pour ajouter le prix de façon aléatoire.

Partie 2. Gestion des quantités

- Pour chaque article, ajoutez des boutons "+" et "-" pour incrémenter ou décrémenter la quantité.
- Assurez-vous que la quantité ne puisse pas être négative.

Partie 3. Panier en temps réel

- Affichez une section "Panier" qui se met à jour dynamiquement.
- Pour chaque article sélectionné, montrez : nom, quantité, prix unitaire, et sous-total.
- Calculez et affichez le sous-total du panier.
- Ajoutez une taxe (par exemple 20%) et calculez le total TTC.

Partie 4. Choix du mode de paiement

- Proposez différents modes de paiement (par exemple : Carte bancaire, PayPal, Virement bancaire).
- Utilisez une structure `switch/case` (ou son équivalent selon le langage) pour gérer les différentes options de paiement.

Partie 5. Confirmation de commande

- Ajoutez un bouton "Confirmer la commande".
- Lors de la confirmation, utilisez une boucle `while` pour vérifier si tous les champs nécessaires sont remplis (mode de paiement sélectionné, panier non vide, etc.).