

A decorative wavy line in orange and white, running vertically along the left side of the slide.

# TypeScript

## Control Flow Statements

# Content

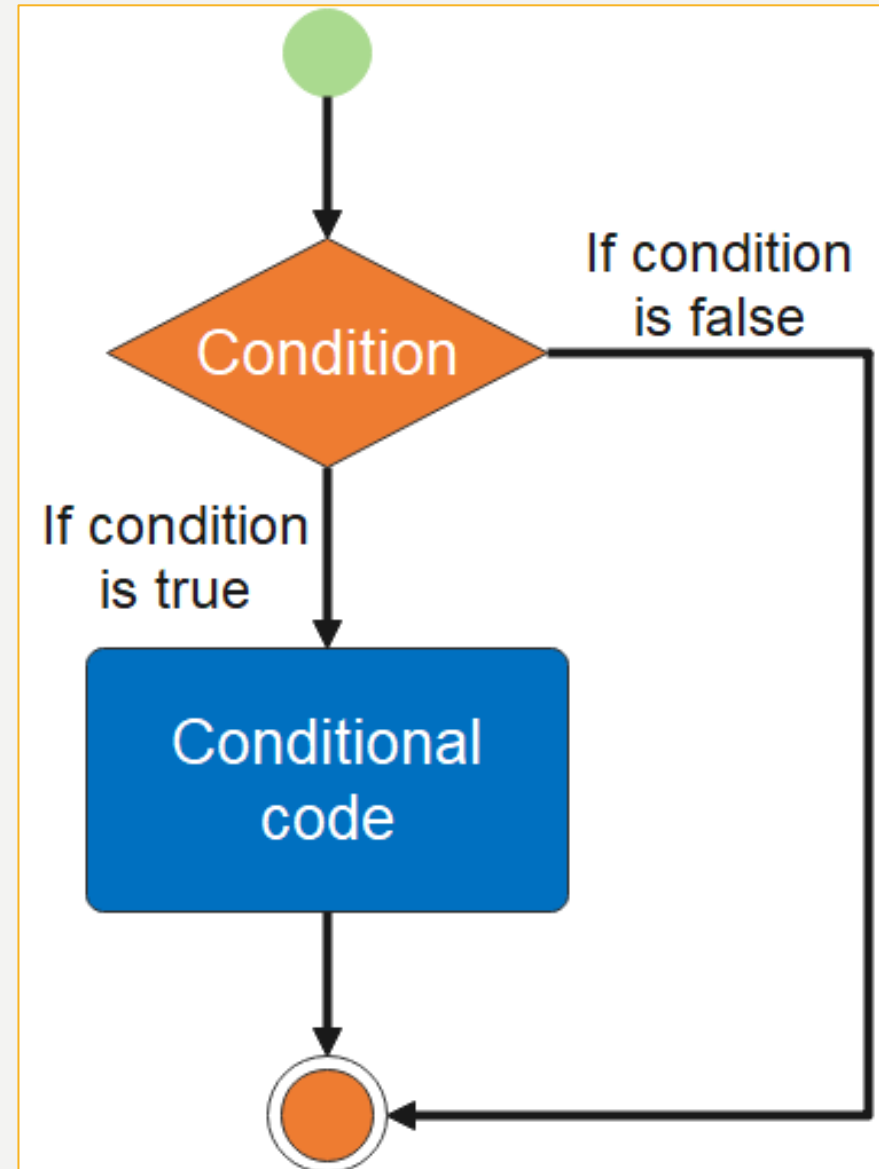
1. if ... else
2. switch ... case
3. for
4. while
5. do ... while
6. break
7. continue

# 1. if ... else

- JavaScript supports the following forms
  - `if` statement
  - `if...else` statement
  - `if...else if...` statement.
- Syntax

```
if(condition) {  
    // if-statement  
}
```

```
if(condition) {  
    // if-statements  
} else {  
    // else statements;  
}
```



```
const max = 100;
let counter = 100;

if (counter < max) {
  counter++;
}

console.log(counter); // 100
```

```
const max = 100;
let counter = 100;

if (counter < max) {
  counter++;
} else {
  counter = 1;
}

console.log(counter);
```

If you have a simple condition, you can use the ternary operator **?:** rather than the **if...else** statement to make code shorter

```
const max = 100;
let counter = 100;

counter < max ? counter++ : counter = 1;

console.log(counter);
```

```
let discount: number;
let itemCount = 11;

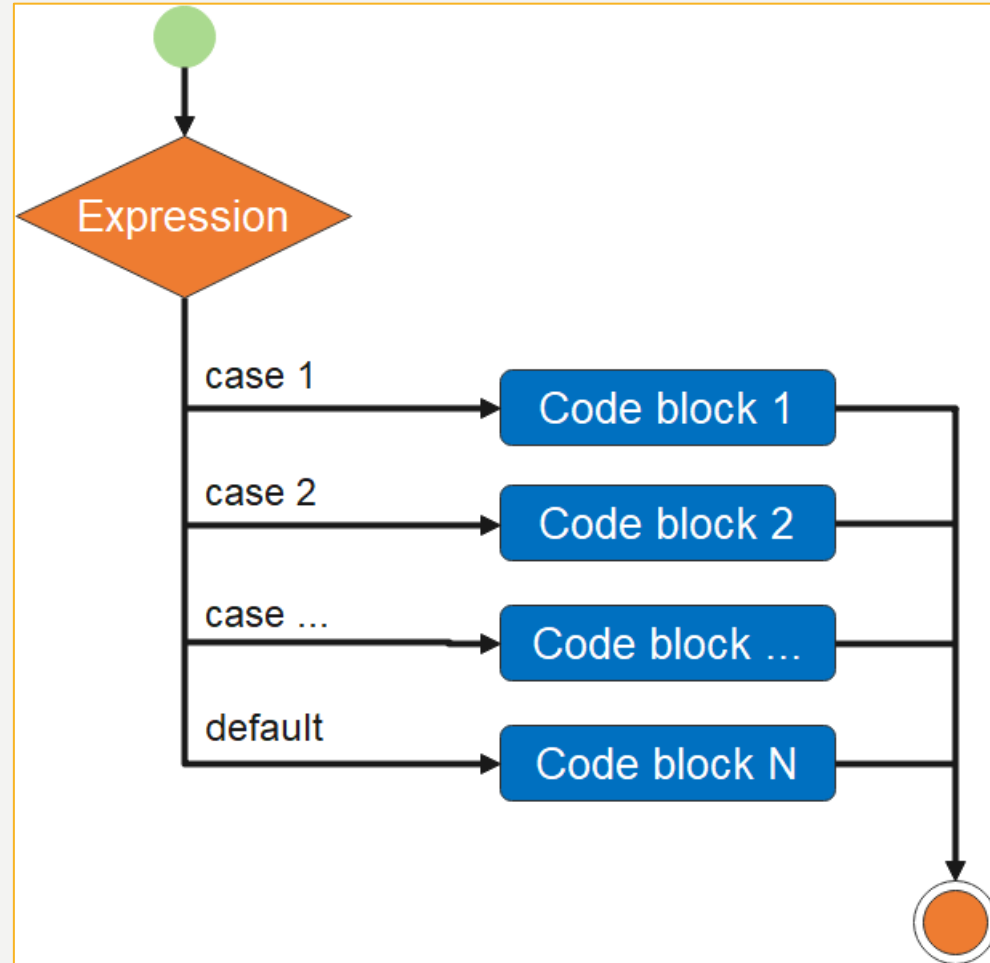
if (itemCount > 0 && itemCount <= 5) {
    discount = 5; // 5% discount
} else if (itemCount > 5 && itemCount <= 10) {
    discount = 10; // 10% discount
} else {
    discount = 15; // 15%
}

console.log(`You got ${discount}% discount.`)
```

## 2. switch ... case

- Use a **switch** statement which handles multiple **if...else if** statements.
- Syntax

```
switch (expression) {  
    case condition_1: statement(s)  
    break;  
  
    case condition_2: statement(s)  
    break;  
  
    // ...  
  
    case condition_3: statement(s)  
    break;  
  
    default: statement(s)  
}
```

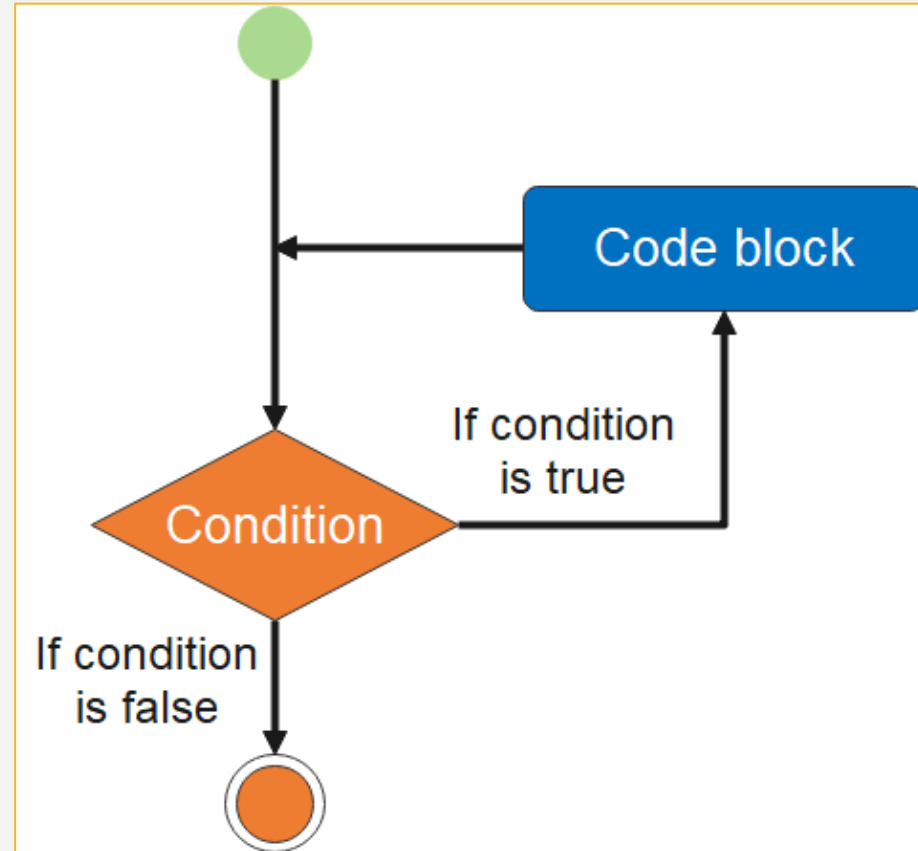


```
let targetId = 'btnDelete';

switch (targetId) {
  case 'btnUpdate':
    console.log('Update');
    break;
  case 'btnDelete':
    console.log('Delete');
    break;
  case 'btnNew':
    console.log('New');
    break;
}
```

# 3. for

- The **for** includes the following three important parts:
  - The loop initialization: initialize the counter to a starting value. It's executed before the loop begins.
  - The test statement: test if a given condition is true or not.
    - True: the code given inside will be executed.
    - False: stop the loop.
  - The iteration statement: increase or decrease the counter.





```
for(initialization; condition; expression) {  
    // statement  
}
```

```
for (let i = 0; i < 10; i++) {  
    console.log(i);  
}
```

```
let i = 0;  
for (; i < 10; i++) {  
    console.log(i);  
}
```

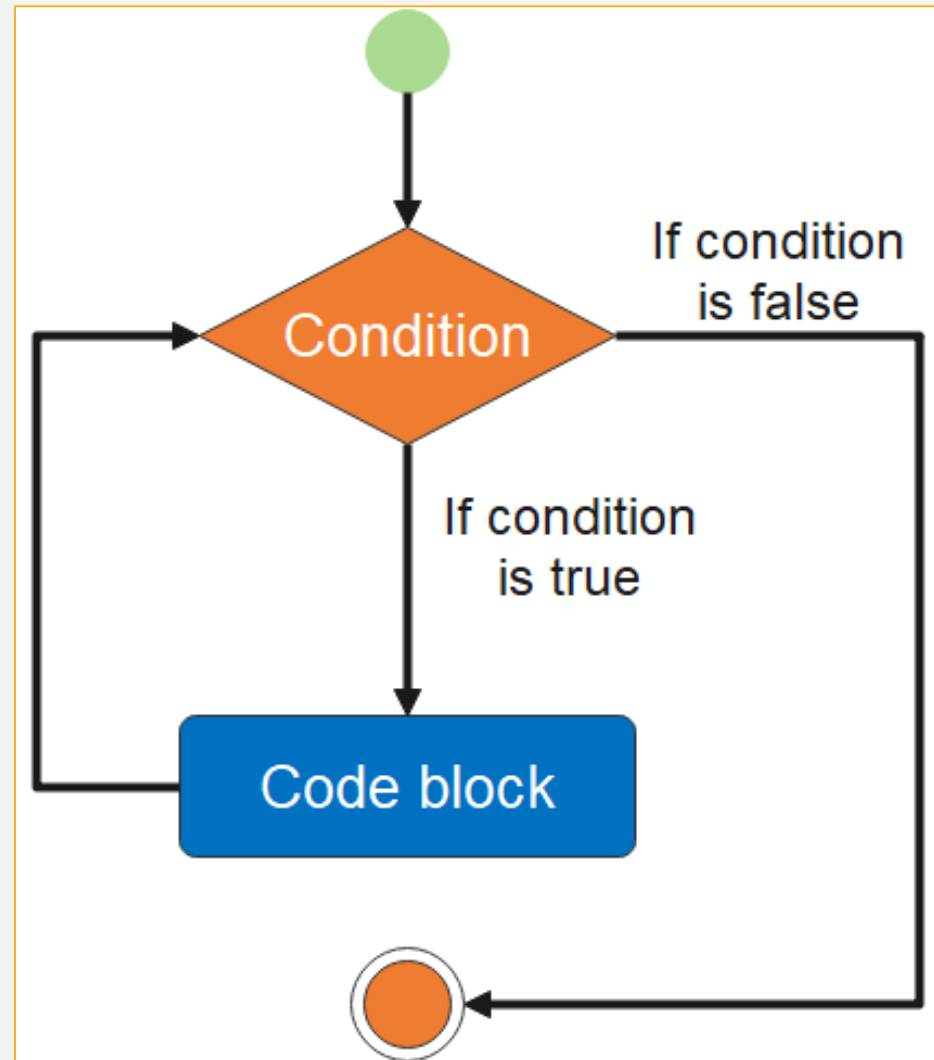
```
for (let i = 0; ; i++) {  
    console.log(i);  
    if (i > 9) break;  
}
```

```
let i = 0;  
for (; ;) {  
    console.log(i);  
    i++;  
    if (i > 9) break;  
}
```

## 4. while

- **while** loop is to execute a statement or code block repeatedly as long as an expression is true. Once the expression becomes false, the loop terminates.
- Syntax

```
while(condition) {  
    // do something  
}
```



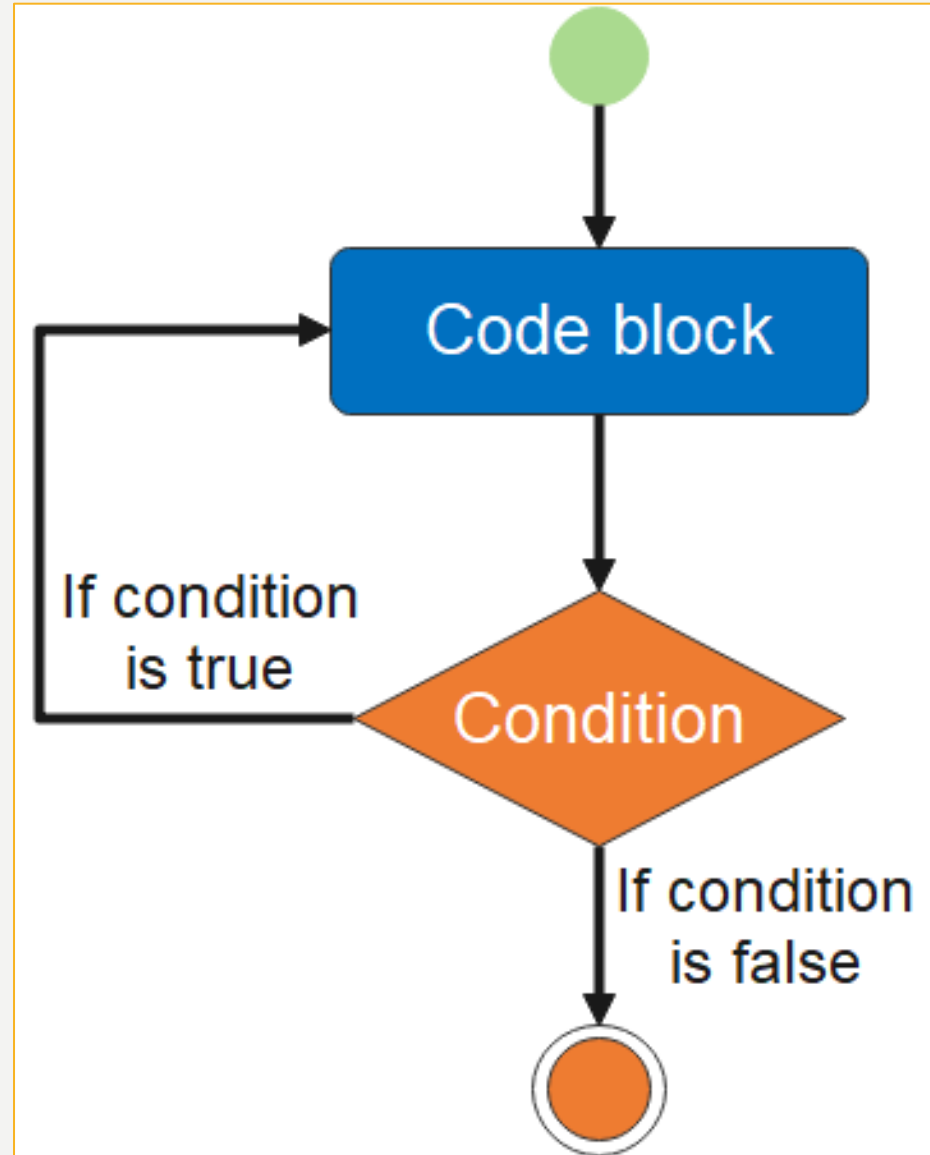
```
while(condition) {  
    // do something  
    // ...  
  
    if(anotherCondition)  
        break;  
}
```

```
let counter = 0;  
  
while (counter < 5) {  
    console.log(counter);  
    counter++;  
}
```

# 5. do...while

- The **do...while** loop is similar to the **while** loop except that the condition check happens at the end of the loop. This means that the loop will always be **executed at least once**, even if the condition is false.
- Syntax

```
do {  
    // do something  
} while(condition);
```



```
let i = 0;
```

```
do {
```

```
    console.log(i);
```

```
    i++
```

```
} while (i < 10);
```

## 6. break

- The break statement allows you to terminate a loop and pass the program control over the next statement after the loop.
- You can use the **break** statement inside the **switch...case**, **for**, **while**, and **do...while** statement

```
let products = [  
  { name: 'phone', price: 700 },  
  { name: 'tablet', price: 900 },  
  { name: 'laptop', price: 1200 }  
];  
  
for (var i = 0; i < products.length; i++) {  
  if (products[i].price == 900)  
    break;  
}
```

# 7. continue

- The continue statement is used to control **for**, **while**, or **do...while** loop.
- The **continue** statement skips to the end of the loop and continues the next iteration

```
for (let index = 0; index < 9; index++) {  
  
    // if index is odd, skip it  
    if (index % 2)  
        continue;  
  
    // the following code will be skipped for odd numbers  
    console.log(index);  
}
```



**THE END**