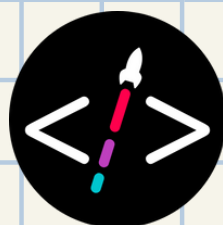
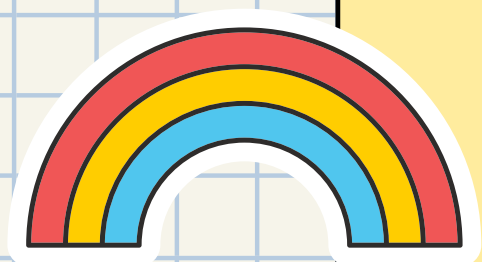
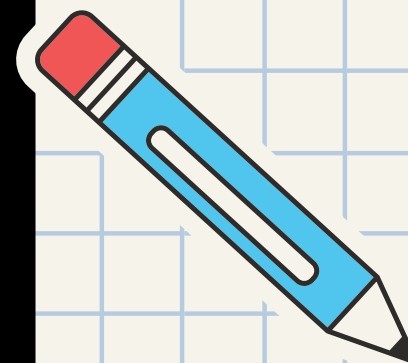
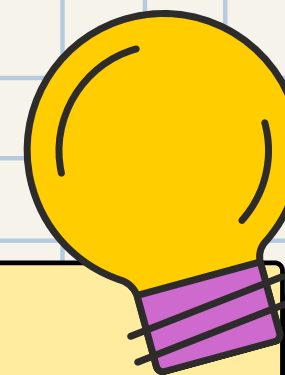


Conditional Statements



if-else statements

In computer programming, there may arise situations where you have to run a block of code among more than one alternatives.

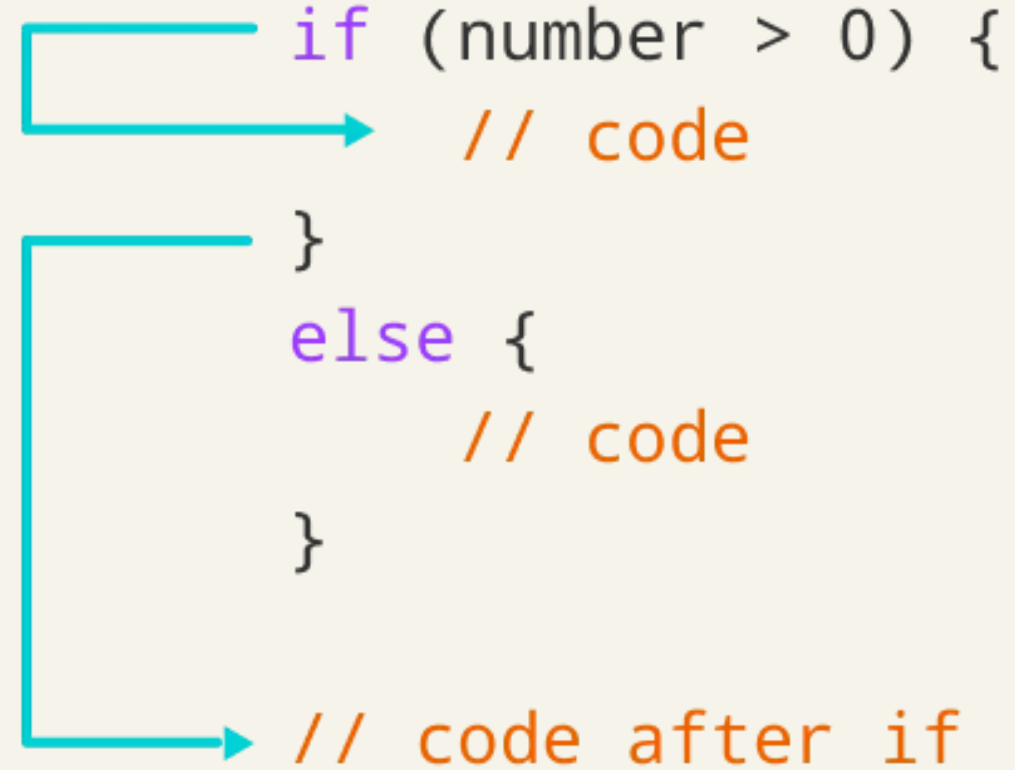
```
if (condition) {  
    // block of code if condition is true  
} else {  
    // block of code if condition is false  
}
```



if-else control flow

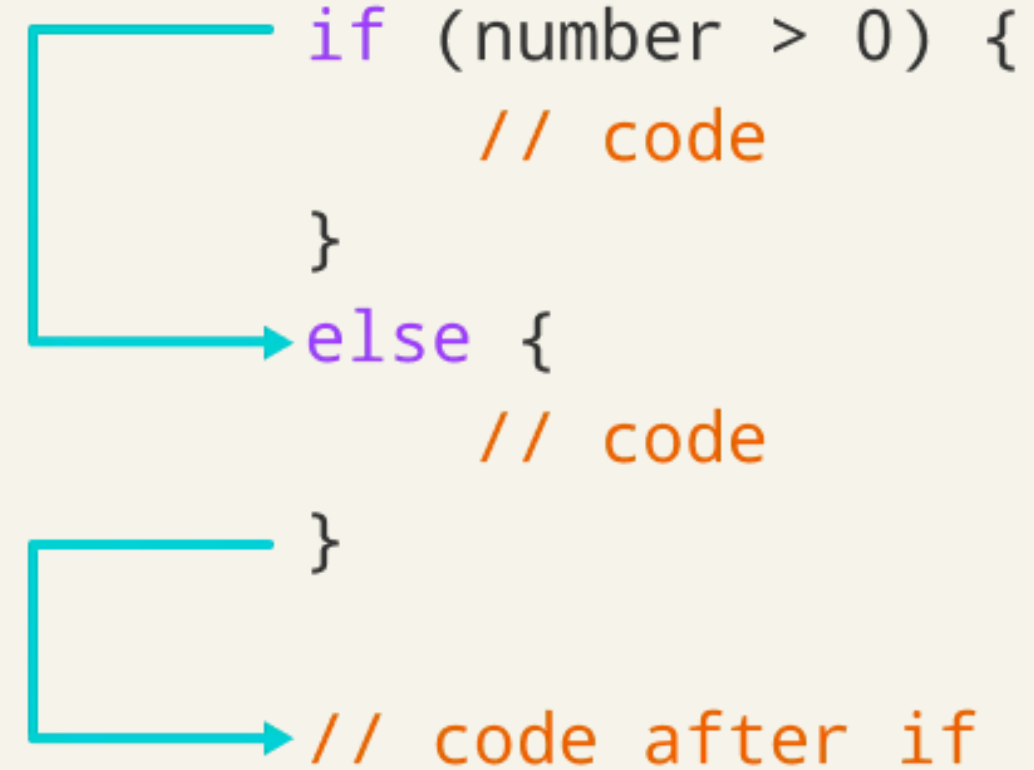
Condition is true

```
let number = 2;  
if (number > 0) {  
    // code  
}  
else {  
    // code  
}  
// code after if
```



Condition is false

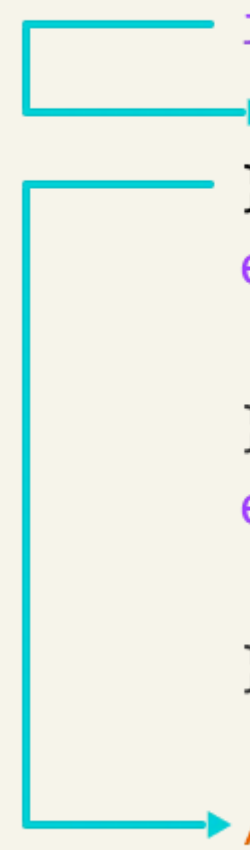
```
let number = -2;  
if (number > 0) {  
    // code  
}  
else {  
    // code  
}  
// code after if
```



if-else-else-if-else control flow


1st Condition is true

```
let number = 2;  
if (number > 0) {  
  // code  
}  
else if (number == 0){  
  // code  
}  
else {  
  //code  
}  
//code after if
```




2nd Condition is true

```
let number = 0;  
if (number > 0) {  
  // code  
}  
else if (number == 0){  
  // code  
}  
else {  
  //code  
}  
//code after if
```



All Conditions are false

```
let number = -2;  
if (number > 0) {  
  // code  
}  
else if (number == 0){  
  // code  
}  
else {  
  //code  
}  
//code after if
```



nested if-else statements

You can also use an if...else statement inside of an if...else statement. This is known as nested if...else statement.

```
if (condition1) {  
    if(condition2) {  
    } else {  
    }  
} else {  
    // block of code if condition is false  
}
```



Ternary Operator

A ternary operator evaluates a condition and executes a block of code based on the condition.

```
condition ? expression1 : expression2
```

The ternary operator evaluates the test condition.

- If the condition is true, expression1 is executed.
- If the condition is false, expression2 is executed.

