



JavaScript Loops



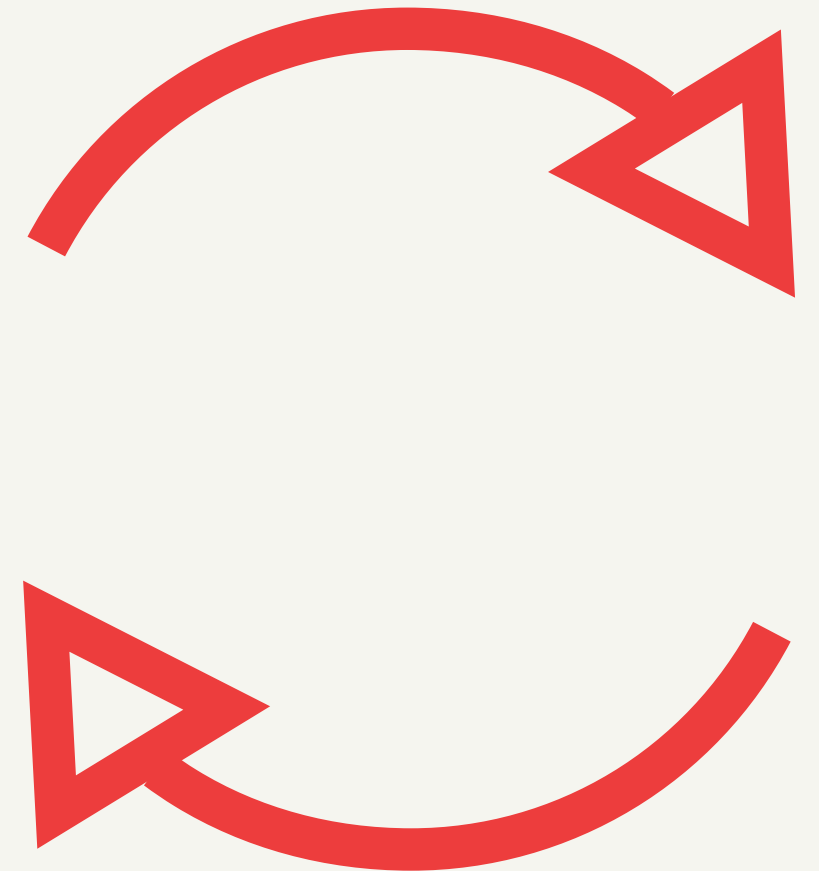
The For Loop

```
for (statement 1; statement 2; statement 3) {  
    // code block to be executed  
}
```

Statement 1 is executed (one time) before the execution of the code block.

Statement 2 defines the condition for executing the code block.

Statement 3 is executed (every time) after the code block has been executed.





Loops

```
var cars = ["BMW", "Volvo", "Saab", "Ford", "Fiat", "Audi"];  
var i;  
for (i = 0; i < cars.length; i++) {  
  
    text += cars[i] + "<br>";  
  
}
```

BMW
Volvo
Saab
Ford
Fiat
Audi



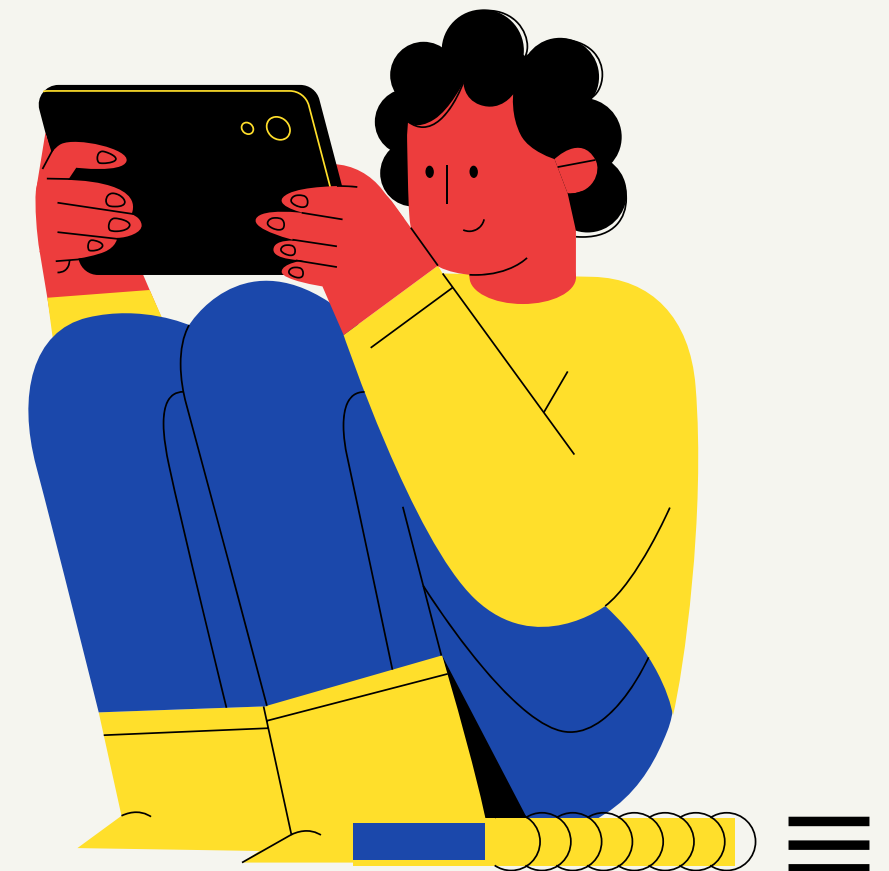


Loops

```
var cars = ["BMW", "Volvo", "Saab", "Ford"];  
var i = 2;  
var len = cars.length;  
var text = "";
```

```
for (; i < len; i++) {  
  
    text += cars[i] + "<br>";  
  
}
```

Saab
Ford





JavaScript For In

Object

```
for (variable in object) {
```

```
    //code
```

```
}
```

```
var person = {fname:"John", lname:"Doe", age:25};
```

```
var text = "";
```

```
var x;
```

```
for (x in person) {
```

```
    text += person[x];
```

John Doe 25

```
}
```



JavaScript For In



Array

```
for (variable in array) {
```

```
    //code
```

```
}
```

```
var numbers = [45, 4, 9, 16, 25];
```

```
var txt = "";
```

```
var x;
```

```
for (x in numbers) {
```

```
    txt += numbers[x] + " ";
```

```
}
```

```
document.getElementById("demo").innerHTML = txt;
```

```
//45 4 9 16 25
```





JavaScript For of

Array

```
var cars = ["BMW", "Volvo", "Mini"];
```

```
var x;
```

```
for (x of cars) {
```

```
    document.write(x + "<br >");
```

```
}
```

BMW
Volvo
Mini



JavaScript For of String

```
var txt = "JavaScript";
```

```
var x;
```

```
for (x of txt) {
```

```
    document.write(x + "<br >");
```

```
}
```

J
a
v
a
S
c
r
i
p
t



JavaScript While Loop



```
while (condition) {  
    // code block to be executed  
}
```

```
while (i < 10) {  
    text += "The number is " + i;  
    i++;  
}
```

The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9



JavaScript Do/While



```
do {  
    // code block to be executed  
} while (condition);
```

```
do {  
    text += "The number is " + i;  
    i++;  
} while (i < 10);
```

The number is 0
The number is 1
The number is 2
The number is 3
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9





JavaScript For/While

เปรียบเทียบ For & While

```
var cars = ["BMW", "Volvo", "Saab", "Ford"];
```

```
var i = 0;
```

```
var text = "";
```

```
for (;cars[i];){
```

```
    text += cars[i] + "<br>";
```

```
    i++;
```

```
}
```

```
var i = 0;
```

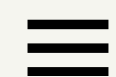
```
var text = "";
```

```
while (cars[i]) {
```

```
    text += cars[i] + "<br>";
```

```
    i++;
```

```
}
```





JavaScript Break and Continue





The Break Statement

```
for (i = 0; i < 10; i++) {  
    if (i === 3) { break; }  
    text += "The number is " + i + "<br>";  
}
```

The number is 0

The number is 1

The number is 2





The Continue Statement

```
for (i = 0; i < 10; i++) {  
    if (i === 3) { continue; }  
    text += "The number is " + i + "<br>";  
}
```

The number is 0
The number is 1
The number is 2
The number is 4
The number is 5
The number is 6
The number is 7
The number is 8
The number is 9





JavaScript Regular Expressions

JavaScript Regular Expressions

Syntax

`/pattern/modifiers;`

ตัวอย่าง

```
var patt = /w3schools/i;
```

i is a modifier

(modifies the search to be case-insensitive).



String Methods

search()

```
var str = "Ministry Of Justice";  
var n = str.search("Justice"); //n =12
```

search() with Regular Expression

```
var str = "Ministry Of Justice";  
var n = str.search(/justice/i); //n =12
```

```
var str = "Ministry Of Justice";  
var n = str.search(/justice/); //n = -1
```





String Methods

replace()

```
var str = "Justice Fund";  
var res = str.replace("Fund", "Care");  
  
//Justice Care
```

replace() using Regular Expression

```
var str = "Justice Fund";  
var res = str.replace(/fund/i, "Care");  
  
//Justice Care
```



Regular Expression Modifiers

- i **insensitive matching**
ค้นหาข้อความที่ตรงกันโดยไม่สนใจตัวอักษรตัวเล็กหรือใหญ่
- g **global match**
ค้นหาข้อความที่ตรงกันทั้งหมดแบบ case-sensitive
- m **multiline matching**
ค้นหาข้อความที่ตรงกันถึงแม้จะมีการเว้นบรรทัด





Regular Expression Modifiers

insensitive matching

ค้นหาข้อความที่ตรงกันโดยไม่สนใจตัวอักษรตัวเล็กหรือใหญ่ (เอาเฉพาะเจอครั้งแรก)

```
var str = "Ministry Of Justice Justice";
```

```
var patt1 = /justice/i;
```

```
var result = str.match(patt1);           //result = Justice
```





Regular Expression Modifiers

global match

ค้นหาข้อความที่ตรงกันทั้งหมดแบบ case-sensitive

```
var str = "Is this all there is?";
```

```
var patt1 = /is/g;
```

```
var result = str.match(patt1);
```

```
//result = is,is
```





Regular Expression Modifiers

multiline matching

ค้นหาข้อความที่ตรงกันถึงแม้จะมีการเว้นบรรทัด แบบ case-sensitive

```
var str = "\nls th\nis it?";
```

```
var patt1 = /^is/m;
```

```
var result = str.match(patt1);           //result = is
```





Regular Expression Patterns

Brackets

[abc] ค้นหาตัวอักษรที่ตรงกับตัวอักษรภายใน []

[0-9] ค้นหาตัวเลขที่อยู่ภายใน []

(x|y) ค้นหาทุกตัวอักษรที่ตรงกับภายในวงเล็บ
ซึ่งแบ่งโดย vertical bar |





Regular Expression Patterns

Brackets

[abc] ค้นหาตัวอักษรภายในวงเล็บ

```
var str = "Is this all there is?";
```

```
var patt1 = /[h]/g;
```

```
var result = str.match(patt1); // result = h,h
```





Regular Expression Patterns

Brackets

[0-9] ค้นหาตัวเลขภายในวงเล็บ

```
var str = "123456789";
```

```
var patt1 = /[1-4]/g;
```

```
var result = str.match(patt1); //result = 1,2,3,4
```





Regular Expression Patterns

(x|y) ค้นหาทุกคำที่อยู่ในวงเล็บซึ่งแบ่งโดย vertical bar |

```
var str = "re, green, red, green, gren, gr, blue, yellow";
```

```
var patt1 = /(red|green)/g;
```

```
var result = str.match(patt1);    //result = green,red,green
```





Regular Expression Patterns

Metacharacters

\d	ค้นหาตัวเลข
\s	ค้นหาช่องว่าง
\S	ค้นหาตัวอักษรใด ๆ
\b	ค้นหาตำแหน่งคำที่อยู่ภายในข้อความ ที่ขึ้นต้นด้วย \bWORD หรือลงท้ายด้วย WORD\b
.	ค้นหาตัวอักษรใด ๆ 1 ตัว





Regular Expression Patterns

Metacharacters

\d ค้นหาตัวเลข

```
var str = "Give 100%!";
```

```
var patt1 = /\d/g;
```

```
var result = str.match(patt1);  
var size = result.length;
```

//result = 1,0,0

//size = 3





Regular Expression Patterns

Metacharacters

\s ค้นหาช่องว่าง

```
var str = "Is this all there is?";
```

```
var patt1 = /\s/g;
```

```
var result = str.match(patt1);  
var size = result.length;
```

```
//result = , , ,  
//size = 4
```





Regular Expression Patterns

Metacharacters

\S ค้นหาตัวอักษรใด ๆ

```
var str = "Is this all there is?";
```

```
var patt1 = /\S/g;
```

```
var result = str.match(patt1);
```

```
//result = I,s,t,h,i,s,a,l,l,t,h,e,r,e,i,s,?
```





Regular Expression Patterns

Metacharacters

\b ค้นหาขึ้นต้น

```
var str = "HELLO,LOOK AT YOU!";
```

```
var patt1 = /\bLO/;
```

```
var result = str.search(patt1);
```

```
//result = 6
```

```
var str = "HELLO,LOOK AT YOU!";
```

```
var patt1 = /LO\b/;
```

```
var result = str.search(patt1);
```

```
//result = 3
```





Regular Expression Patterns

Metacharacters

- ค้นหาตัวอักษรใด ๆ 1 ตัว

```
var str = "That's hot!";
```

```
var patt1 = /h.t/g;
```

```
var result = str.match(patt1);
```

```
//result = hat,hot
```





Regular Expression Patterns

Quantifiers

n^+	ค้นหาตัวอักษรใดๆ อย่างน้อย 1 ตัวขึ้นไป
n^*	ค้นหาตัวอักษรใดๆ อย่างน้อย 0 ตัวขึ้นไป
$n^?$	ค้นหาตัวอักษรใดๆ อย่างน้อย 0 หรือ 1 ตัว





Regular Expression

test() Method

```
var str = "The best things in life are free";
```

```
var patt = /e/;
```

```
var res = patt.test(str);    //res = true
```





JavaScript Functions



JavaScript Function Syntax

```
function name(parameter1, parameter2, parameter3)
{
    // code to be executed
}
```



Function Invocation

การเรียกใช้

แบบที่ 1

เมื่อเกิดเหตุการณ์ เช่น ผู้ใช้งานคลิกปุ่ม

แบบที่ 2

เมื่อเรียกใช้จากชุดคำสั่ง JavaScript

แบบที่ 3

เรียกใช้แบบอัตโนมัติ



Function Return

การคืนค่า

```
var x = myFunction(4, 3);  
  
function myFunction(a, b)  
{  
    return a * b;  
}
```



ทำไม

ต้องมีฟังก์ชัน

เพื่อการเรียกใช้งานซ้ำหลายๆ ครั้ง

```
function toCelsius(fahrenheit)
{
```

```
    return (5/9) * (fahrenheit-32);
```

```
}
```

```
document.getElementById("demo").innerHTML =  
toCelsius(77);
```



นำไปใช้

```
var x = toCelsius(77);var text =  
"The temperature is " + x + " Celsius";
```

```
var text =  
"The temperature is " + toCelsius(77) + "  
Celsius";
```



Local Variables

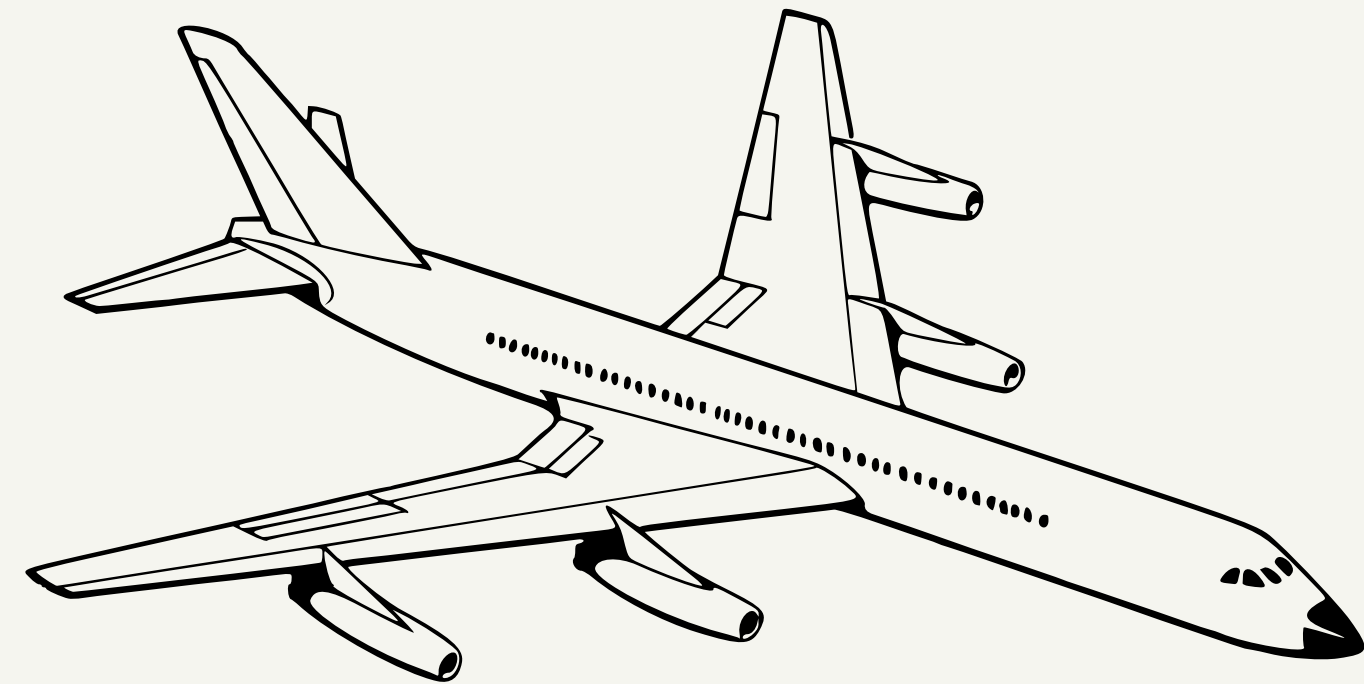
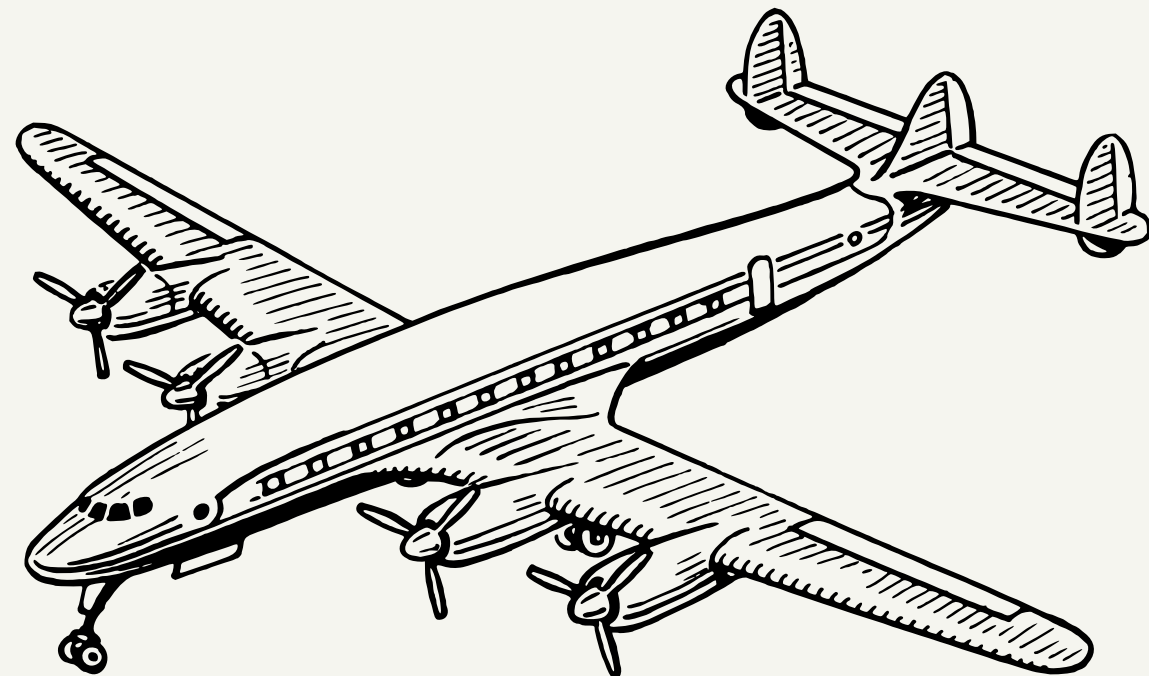
ตัวแปรภายใน

```
function myFunction()  
{  
    var carName = "Volvo";  
}
```



JS Object

```
airplane.company = boeing  
airplane.model = 777  
airplane.weight = 134,800 kg  
airplane.color = white
```



```
airplane.start()  
airplane.takeoff()  
airplane.landing()  
airplane.brake()
```





JS Object

```
var person = {  
  firstName: "John",  
  lastName : "Doe",  
  id        : 5566,  
  fullName : function() {  
    return this.firstName + " " + this.lastName;  
  }  
};
```

```
var result = person.fullName();
```

```
//result = John Doe
```





JavaScript Event





JavaScript Events



`<element event='some JavaScript'>`

`<element event="some JavaScript">`





JavaScript Events



```
<button  
  onclick="document.getElementById('demo').innerHTML = Date()"  
  The time is?  
</button>
```

```
<button onclick="this.innerHTML = Date()"  
  The time is?  
</button>
```

```
<button onclick="displayDate()"  
  The time is?  
</button>
```





JavaScript Events



```
<button  
  onclick="document.getElementById('demo').innerHTML = Date()"  
  The time is?  
</button>
```

```
<button onclick="this.innerHTML = Date()"  
  The time is?  
</button>
```

```
<button onclick="displayDate()"  
  The time is?  
</button>
```



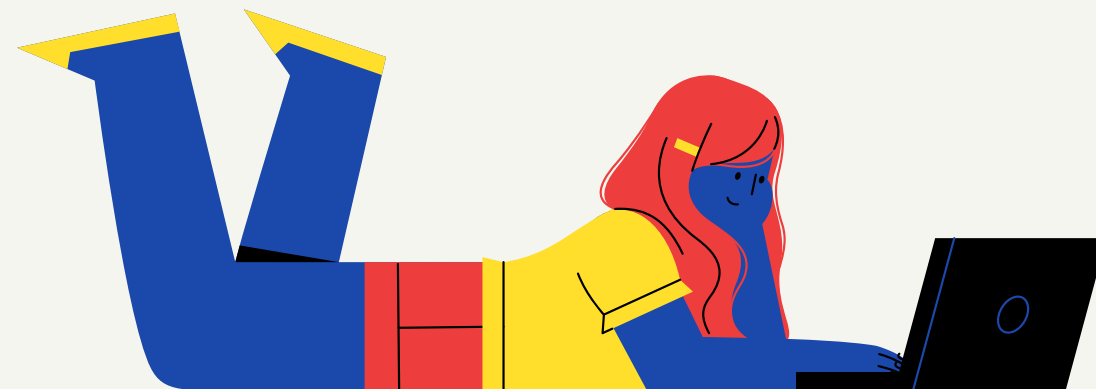


HTML Events

onchange เรียกคำสั่งเมื่ออีลีเมนต์ใดๆ มีการเปลี่ยนแปลง

onclick เรียกคำสั่งเมื่อมีการคลิก

onmouseover เรียกคำสั่งเมื่อเมาส์อยู่บนอีลีเมนต์





HTML Events

`onmouseout` เรียกคำสั่งเมื่อเมาส์เคลื่อนออกจากอีลีเมนต์

`onkeydown` เรียกคำสั่งเมื่อเมาส์มีการพิมพ์

`onload` เรียกคำสั่งหลังจากโหลดหน้าเพจเสร็จสิ้น





Math Object

`Math.PI;` `// returns 3.141592653589793`

`Math.round(4.7);` `// returns 5`

`Math.round(4.4);` `// returns 4`

`Math.pow(8, 2);` `// returns 64`

`Math.sqrt(64);` `// returns 8`

`Math.abs(-4.7);` `// returns 4.7`

`Math.ceil(4.4);` `// returns 5`

`Math.floor(4.7);` `// returns 4`





Math Object

```
Math.min(0, 150, 30, 20, -8, -200); // returns -200
```

```
Math.max(0, 150, 30, 20, -8, -200); // returns 150
```

```
Math.random(); // returns a random number
```

```
Math.floor(Math.random() * 10); // returns a random integer from 0 to 9
```

```
Math.floor(Math.random() * 11); // returns a random integer from 0 to 10
```



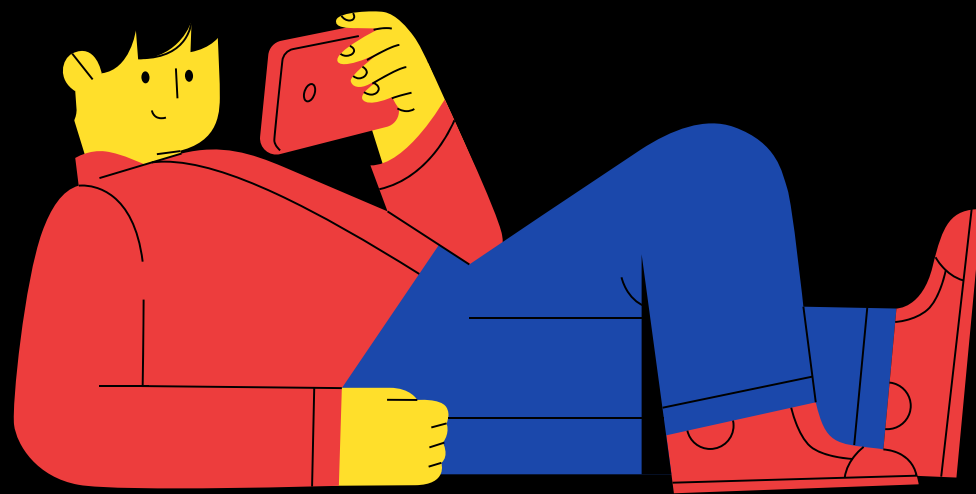


JavaScript Conditional



The if else Statement

```
if (condition) {  
    // block of code to be executed if the  
    condition is true  
} else {  
    // block of code to be executed if the  
    condition is false  
}
```



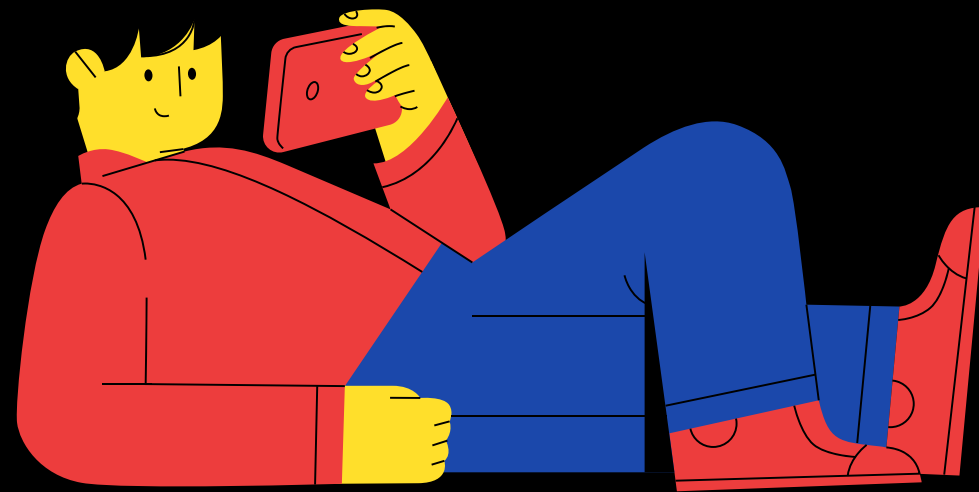
```
if (hour < 18) {  
    greeting = "Good day";  
}  
else {  
    greeting = "Good evening";  
}
```



The else if Statement



```
if (con1) {  
    // execute if con1 is true  
} else if (con2) {  
    // execute if con2 is true , con1 is false  
} else {  
    // execute if con1 and con2 is false  
}
```

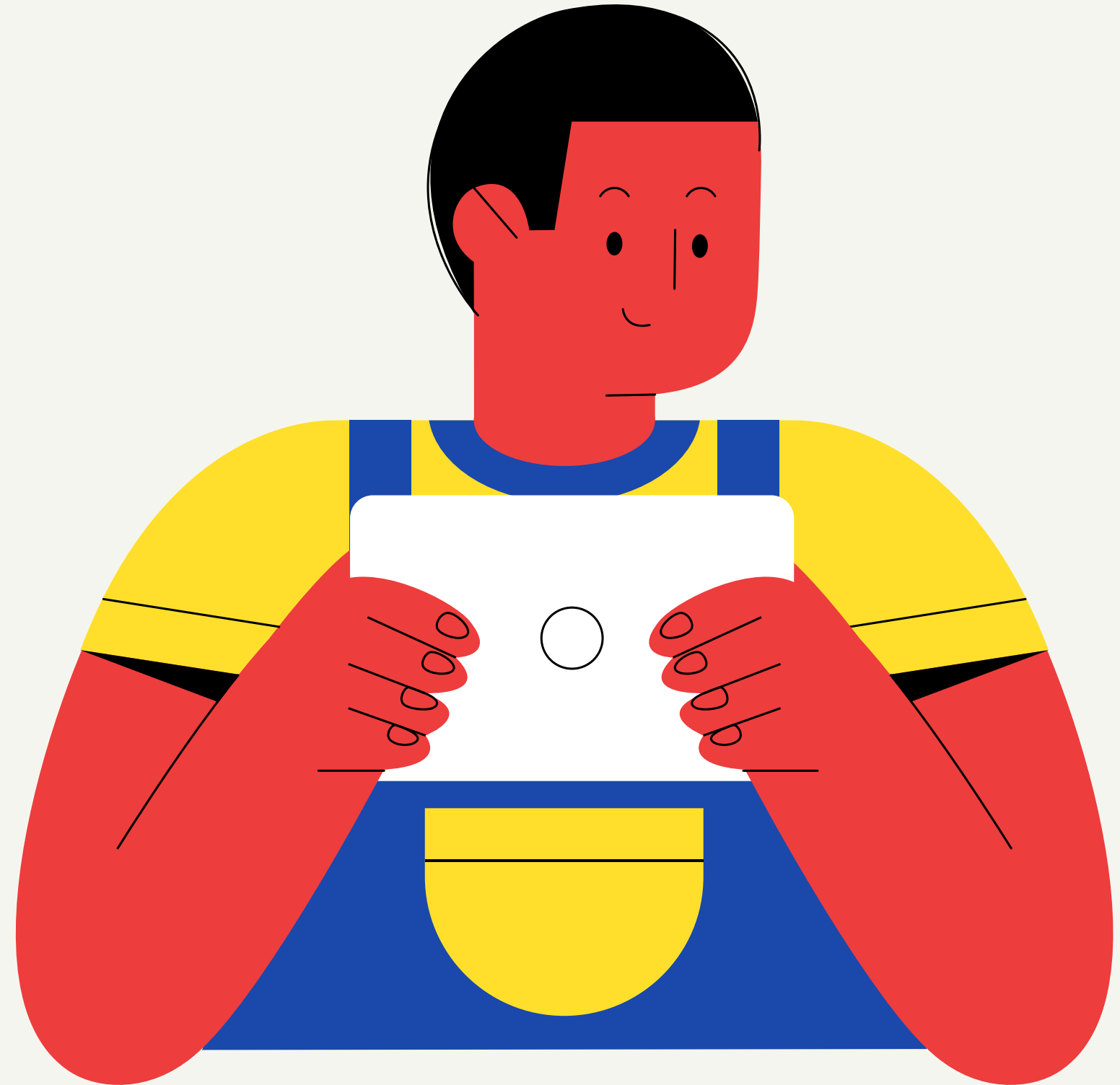


```
if (time < 10) {  
    greeting = "Good morning";  
} else if (time < 20) {  
    greeting = "Good day";  
} else {  
    greeting = "Good evening";  
}
```





JavaScript Switch



JavaScript Switch Statement

Syntax

```
switch(expression){  
    case x:  // code block  
        break;  
    case y:  // code block  
        break;  
    default: // code block  
}
```



JavaScript Switch Statement

```
switch (new Date().getDay()) {  
    case 1:  
        day = "Monday";  
        break;  
    case 3:  
        day = "Wednesday";  
        break;  
    case 5:  
        day = "Friday";  
        break;  
    default:  
        day = "Holiday";  
}
```



Thank you