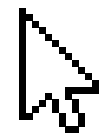




# Computer Science : JavaScript

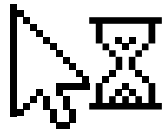


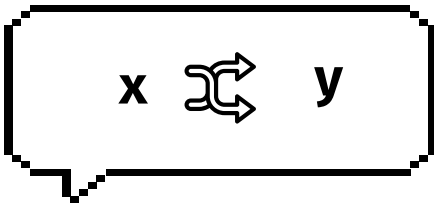
**Week 7: Loops**



# Loops

**Loops** defined as code that repeats instructions until a certain condition is met, known as a *stopping condition*. This makes repeating tasks simple.





# 01

## 'for' Loops

Repeat a set of instructions when the number of iterations are known



# Step 1

for ()



Write *for* keyword with  
parentheses



# Step 2

```
for (let num = 0; num < 4; num++)
```



Inside Parentheses:

1. **Initialization**: Declare iterator variable
2. **Stopping Condition**: the condition the loop runs against. If the condition is true the block will continue to run until the condition is false
3. **Iteration statement**: expression that updates iterator variable value

*Separate by semicolons*



# Step 3

```
for (let num = 0; num < 4; num++) {  
  console.log (num);  
}
```

// Output: 0 1 2 3



Inside Block:

Your block code will execute for each value until condition is false



# Practice

Create a for loop that will output numbers 4 to 12.





# Practice

Create a for loop that will count down from 6 to 0.

Hint: use greater than or equal to sign



# Loop an Array ('for' version)

```
const food = ['burger', 'steak', 'pasta']
for (let i = 0; i < food.length; i++) {
  console.log (food[i]);
}
```

// Output: 'burger' 'steak' 'pasta'



Inside Parentheses:

- Set *i* to 0 - *array index start at 0*
- `.length` - we want our *i* to always be one less than the length of the array.
- The block will return each element in the array by its index *i*



# Practice

Create a for loop for an array assigned to the variable `sports` with 4 strings.





# Nested Loop

```
let num1 = [23, 5, 7]
let num2 = [122, 7, 8]

for (let i=0; i < num1.length; i++) {
  for (let j=0; j < num2.length; j++) {
    if (num1 [i] === num2[j]) {
      console.log ('They share' + num1[i])
    }
  }
}
```



When a for loop runs within another for loop.

Each element of the outer loop will run through all the elements of the inner loop one at a time.



# Practice

## Setup:

Variable 1: `myGuests` assign to an array of 4 names

Variable 2: `yourGuests` assign to an array of 3 names  
and two of the names are the same as your first list

Variable 3: `sameGuest` assign to an empty array

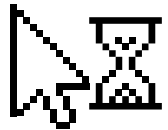
## Task:

You will create a nested for loop (outer loop `myGuest`  
and inner loop `yourGuest`). If the elements of the  
arrays are the same it will push the elements into the  
empty array `sameGuest`



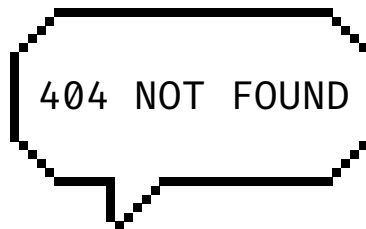
# Note: Returns

If you are using return in loops please be mindful of its function. return tends to terminate loops or functions and just return the value it wants. So it may end your function early.





# 02



## while Loop

Repeats a set of instruction when the numbers  
of iterations are unknown



# Step 1

```
let counter = 1;
```



Declare the variable(s)



## Step 2

```
let counter = 1;  
while (counter < 4)
```



Write the *while* keyword with parentheses  
(): Stopping condition should be inside the  
parentheses



# Step 3

```
let counter = 1;
while (counter < 4) {
  console.log (counter);
  counter++;
};
```

// Output: 1 2 3



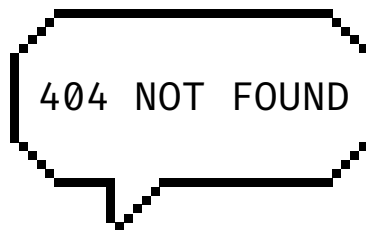
Inside Block:

Your block code will first execute the task then will the next increments until the condition is false.

**Note:** If you don't increment then the value will remain equal to 1 and the code will continuously be true so it will run forever called an *Infinite Loop*, which we want to avoid.



# 03



## do... while Loop

It will do a task once and continue to do it until the condtion is met



# Difference between *while* and *do... while...*

`while`: will only perform a task if it is true  
and will not run if it is false

`do... while...`: will run at least once whether it  
is true or false





# Step 1

```
let eggs = 3;  
let eggsAdded = 0;
```



Declare the variable(s)



# Step 2

```
let eggs = 3;  
let eggsAdded = 0;  
do {  
  eggsAdded++  
  console.log (eggsAdded)  
}
```



Write the *do* keyword with block  
{}: task you want to perform



# Step 3

```
let eggs = 3;  
let eggsAdded = 0;  
do {  
  eggsAdded++;  
  console.log (eggsAdded);  
} while (eggsAdded < eggs);
```



Write *while* keyword with parentheses

Inside Parentheses: Write stopping condition



# Practice

Change the sample codes variable values so the condition is false





IDENTIFIED

03

Break



# Break a loop

You can use *break* to break out of a loop if it meets a specific condition.

Ex:

```
for(let i= 0; i< 100; i++) {  
  console.log ('check')  
  if (i>3) { //I set my condition  
    break;  
  }  
} //outputs check check check
```

