Day 6 - Loops

Loops

In programming languages, we can use loops to carry out repetitive tasks. The below examples are commonly used loops in JS and other programming languages.

for Loop

Structure:

```
for(initialization; condition; increment/decrement) {
  // code goes here
}

// example
for(let i = 0; i <= 5; i++) {
  console.log(i)
}

// 0 1 2 3 4 5

// another example
for(let i = 0; i <= 5; i++){
  console.log(`${i} * ${i} = ${i * i}`)
}

// 0 * 0 = 0
// 1 * 1 = 1
// 2 * 2 = 4
// 3 * 3 = 9
// 4 * 4 = 16
// 5 * 5 = 25

// last example
const countries = ['Finland', 'Sweden', 'Denmark', 'Norway', 'Iceland']
const newArr = []
for(let i = 0; i < countries.length; i++){
  newArr.push(countries[i].toUpperCase())
}

// ["FINLAND", "SWEDEN", "DENMARK", "NORWAY", "ICELAND"]
```

Adding all elements in the array:

```
const numbers = [1, 2, 3, 4, 5]
let sum = 0;
for(let i = 0; i < numbers.length; i++) {
  sum = sum + numbers[i]
}

console.log(sum) // 15
```

Creating a new array based on the existing array:

```
const numbers = [1, 2, 3, 4, 5]
const newArr = []

for(let i = 0; i < numbers.length; i++) {
  newArr.push(numbers[i] ** 2)
}

console.log(newArr)  // [1, 4, 9, 16, 25]
```

```
const countries = ['Canada', 'Finland', 'Sweden']
const newArr = []

for (let i = 0; i < countries.length; i++) {
  newArr.push(countries[i].toUpperCase())
}

console.log(newArr);  // ['CANADA', 'FINLAND', 'SWEDEN']
```

while loop

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

// 0 1 2 3 4 5
```

do while loop

```
let i = 0
do {
  console.log(i)
  i++
} while (i <= 5)

// 0 1 2 3 4 5
```

for of loop

We use these loops for arrays. It is a handy way to iterate through an array if we are not interested in the index of each element in the array

Structure:

```
for (const element of arr) {
  // code goes here
}
```

```
const numbers = [1, 2, 3, 4, 5]

for (const num of numbers) {
  console.log(num)
}

// 1 2 3 4 5

for(const num of numbers) {
  console.log(num * num)
}

// 1 4 9 16 25

// adding all the numbers in the array
let sum = 0
for (const num of numbers) {
  sum = sum + num
}
console.log(sum) // 15

const webTechs = [
  'HTML',
  'CSS',
  'JavaScript',
  'React',
  'Redux',
  'Node',
  'MongoDB'
]

for (const tech of webTechs) {
  console.log(tech.toUpperCase())
}

// HTML CSS JAVASCRIPT REACT NODE MONGODB

for (const tech of webTechs) {
  console.log(tech[0]) // get only the first letter of each element,  H C J R N M
}
```

```
const countries = ['Finland', 'Sweden', 'Norway', 'Denmark', 'Iceland']
const newArr = []
for(const country of countries){
  newArr.push(country.toUpperCase())
}

console.log(newArr)  // ["FINLAND", "SWEDEN", "NORWAY", "DENMARK", "ICELAND"]
```

break

Used to interrupt a loop

```
for(let i = 0; i <= 5; i++) {
  if(i == 3) {
    break
  }
  console.log(i)
}

// 0 1 2
```

The above code stops if 3 found in the iteration process.

continue

We use the keyword *continue* to skip certain iterations

```
for(let i = 0; i <= 5; i++) {
  if(i == 3) {
    continue
  }
  console.log(i)
}

// 0 1 2 4 5
```