Programming JS

Margit Tennosaar

Loops

Loops

- Loops are used to repeat a block of code multiple times.
- Useful for iterating over collections, repeating tasks, and automating repetitive processes.
- JavaScript provides several types of loops to handle different situations.

The for Loop

Used when you know in advance how many times you want to execute a block of code.

```
for (initialization; condition; increment) {
  // Code to be executed on each iteration
}


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

The while Loop

Continues to execute a block of code as long as the specified condition is true.

```
while (condition) {
  // Code to be executed as long as the condition is true
}


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

The do...while Loop

Similar to the while loop, but ensures the code block is executed at least once.

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


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

The for...in Loop

Used to iterate over the properties of an object.

```
for (key in object) {
  // Code to be executed for each property
}


const person = { name: 'Alice', age: 25 };
for (let key in person) {
  console.log(`${key}: ${person[key]}`);
}
```

The for...of Loop

Creates a loop iterating over iterable objects like arrays and strings.

```
for (variable of iterable) {
  // Code to be executed for each element
}

let fruits = ['apple', 'banana', 'pear'];
for (let fruit of fruits) {
  console.log(fruit);
}
```

Loop control statements

break statement

```
for (let i = 0; i < 10; i++) {
  if (i === 5) {
    break;
  }
  console.log(i);
}
```

continue statement

```
for (let i = 0; i < 5; i++) {
  if (i === 2) {
    continue;
  }
  console.log(i);
}
```

Common mistakes to avoid

- Infinite Loops: Make sure your loop has a valid exit condition.
- Off-by-One Errors: Carefully check your loop conditions (< vs. <=).
- Misusing for...in with Arrays: Use for...of instead for arrays.
- Not Initializing Variables: Ensure loop variables are initialised correctly.

```
for (initialization; condition; increment) {
  // Code to be executed on each iteration
}
```

```
while (condition) {
  // Code to be executed as long as the condition is true
}
```

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

```
for (key in object) {
  // Code to be executed for each property
}
```

```
for (variable of iterable) {
  // Code to be executed for each element
}
```