Conditionals and Loops

Sprint 4

Comparison Operators

Perform logical comparison operations:

- greater than (`>`)
- greater than or equal to (`>=`)
- less than (`<`)
- less than or equal to (`<=`)
- equality (`==`) and strict equality (`===`)
- inequality (`!=`) and strict inequality (`!==`)

Equality vs. Strict Equality

```
// regular equality doesn't check type
console.log("42" == 42);  // true

// strict equality returns false if types are different
console.log("42" === 42);  // false
```

Conditional Statements

Run different code in different situations.

The `if` block:

```
if (condition) {
  // this code is run only when the condition is true
}
```

Logical Operators

Perform Boolean algebra logical operations:

- logical AND (`&&`) — returns `true` if both operands are `true`
- logical OR (`||`) — returns `true` if one or more operands is `true`
- logical NOT (`!`) — returns `true` if its operand is `false`

```
const A = true;
const B = false;

A && A // true, since both operands are true
A && B // false, since B is false

A || B // true, since A is true
B || B // false, since both operands are false

!A     // false, since A is true
!B     // true, since B is false
```

Loops: `while` and `for`

The `while` loop allows you to repeatedly execute a block of code:

```
// syntax
while (condition) {
   // this code runs while condition is true
}

// example
let i = 0;          // initialize a counter
while (i < 3) {     // if i < 3, execute the loop
  console.log(i);   // log i to the console
  i++;              // add 1 to i
}

// logs 0, 1, and 2 to the console,
// on separate lines
```

The `for` loop allows you to specify the number of iterations:

```
// syntax
for (initialization; condition; increment) {
  // this code runs while condition is true
}

// example: initialization and incrementing
// are built in to the for loop
for (let i = 0; i < 3; i++) {
  console.log(i);  // log i to the console
}

// logs 0, 1, and 2 to the console,
// on separate lines
```