Take 2

JavaScript 102

101 review

More Javascript

Evaluation / "Running" the program

- How does the computer run the instructions of a program?
- It 'evaluates' them. Evaluating a statement starts like this:

1. Take all variables from the statement and replace them with their value.
2. Do any calculations. Anything in parentheses comes first.

```
1  let name = "Gregory";
2  let age = 35;
3  console.log(name);
4  console.log(age + 3);
```

```
1  // During evaluation, the
2  // program is transformed
3  // by the computer.
4  let name = "Gregory";
5  let age = 35;
6  console.log("Gregory");
7  console.log(38);
```

```
let name = "George";
let age = 35;
let year = 2024;

// Exercise 1
console.log(name);

// Exercise 2
let nextYear = year + 1;
console.log(nextYear);

// Exercise 3
console.log(year - age);
```

`let` and `const`

- Some variables change, some don't
- We know we can declare variables with the keyword `let`:
- `let name = "Greg";`
- Variables declared with `let` can be changed ('reassigned'):
- `let name = "Kaia"; name = "Maia";`

`let` and `const`

- We can also declare variables with the keyword `const`:
- `const gregsBirthYear=1991;`
- Variables declared with `const` cannot be reassigned:
- `const year = 2024; year = 2025;`

- If you try anyway, it will produce an error:
- `Uncaught TypeError: Assignment to constant variable.`

Reassigning `const`

```
1  let temperature = 20;
2  temperature = 18; // Ok
3  temperature = 15; // Ok
4
5  const temperature = 20;
6  // Produces "Uncaught TypeError:
7  // Assignment to constant variable."
8  temperature = 18;
```

Work through the HTML and CSS articles on MDN

POP QUIZ

Find the errors

```
1  lot javascript = "cool";
2  CONST pizza = "tasty";
3
4  console.log(age);
5  let age;
6  console.log(age);
7  age = 21;
8  console.log(age);
```

More JavaScript

Conditionals (`if` / `else`)

- Sometimes an instruction should only be done **if** a certain condition is true
- Let's make a program to tell the user if they are old enough to enter a music venue. Anyone 18 or older, is old enough to enter.

```
let age = 19;
console.log("You are old enough to enter.")
```

- What happens if we change the age to 16?
- Now the user is no longer old enough to enter the venue.
- Do we have to re-write the whole program?

- We can use `if` and `else` to build **conditional** statements.

```
let age = 19;

if(age >= 18) {
  console.log("You are old enough to enter.")
} else {
  console.log("You are too young to enter.")
}
```


if
(
age >= 18
)
keyword `if`
open parenthesis
condition
close parenthesis

```
let age = 19;

if(age >= 18) {
  // Only executed if condition is `true`
  console.log("You are old enough to enter the venue.")

} else {
  // Only executed if condition is `false`
  console.log("You are too young to enter the venue.")

}
```

What is `true`, what is `false`?

- We can compare things in Javascript using `==`, `<`, `<=`, `>`, and `>=`
- `1 == 1` → this is true
- `1 == 2` → this is false
- `false == false` → this is true
- `8 < 1000` → this is true
- `8 > 1000` → this is false

- Remember: conditions, just like other statements, get their variables replaced when they're executed:

```
let age = 19;
if(age >= 18) ...
```

- After replacement, this code becomes:

```
let age = 19;
if(19 >= 18) ...
```

```
let age = 19;
if(true) ...
```

Boolean algebra

- To do calculations with `true` and `false`, we can't use `+` or `-`.
- We need boolean operators! `&&` and `||` are the most common.

`&&`, pronounced AND

```
1  true && true == true
2  true && false == false
3  false && true == false
4  false && false == false
```

If both operands are `true`, then the result is `true`. Otherwise it's `false`.

`||`, pronounced OR

```
1  true || true == true
2  true || false == true
3  false || true == true
4  false || false == false
```

If either operands is `true`, then the result is `true`. Otherwise it's `false`.

A little more complex

```
(true && true) || false == ?
(true && false) || false == ?
(false && false) || true == ?
```

Curly Braces in JS

- `{` and `}` are used to tell Javascript when a block of code starts and ends

```
let name = "Roger";

console.log(`Good morning, ${name}`);

if(name == "Santa") {
  // Only executed if condition is `true`
  console.log("Hey it's Santa Claus!!")
}
// No 'else' block
```

```
let name = "Roger";

console.log(`Good morning, ${name}`)

if(name == "Santa") {
  // Only executed if condition is `true`
  console.log("Hey it's Santa Claus!!");
} else {
  console.log("Hey you're not Santa!");
}
```

Recap