JAVASCRIPT CONCEPTS: SCOPE AND ASYNC/AWAIT

LOGISTICS

PLAN FOR TODAY

- Checking in
- Turning in MP1
- Scope and async/await
- Work time!

CHECKING IN

- What’s up?
- Questions in Discord?
- Attendance at office hours?

Check-in survey: https://forms.gle/4P5cVzejdHEeiNco6

TURNING IN MP1

The writeup should be on your portfolio page, like MP0! Remember:

- In-progress screenshots
- A gif of your extension in action
- Include Kudos to fellow classmates and online resources

I encourage you to publish your extension on the Chrome Web store

JAVASCRIPT CONCEPTS: SCOPE

SCOPE

If a variable is defined inside a code block (curly braces: `{...}`), it will only be visible inside that block.

```
{
  // do some job with local variables that should not be seen outside
  let message = "Hello"; // only visible in this block
  alert(message); // Hello
}

alert(message);
```

What do you think will happen?

SCOPE

This rule holds for variables defined inside loops and conditionals:

```
if (true) {
  let phrase = "Hello!";

  alert(phrase);
}

alert(phrase);
```

What do you think will happen?

SCOPE

```
if (true) {
  let x = 1;
}

console.log(x);
```

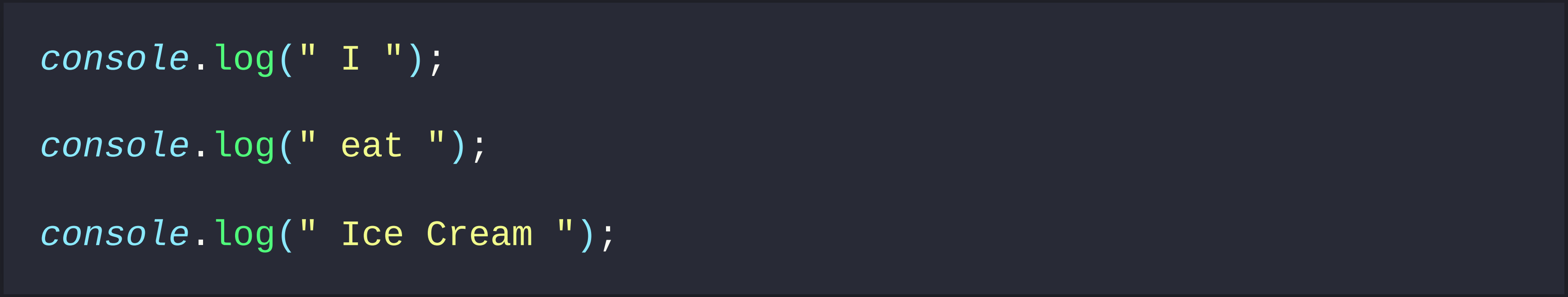
What do you think will happen?

JAVASCRIPT CONCEPTS: ASYNC/AWAIT

ASYNC/AWAIT

- Many tasks you might want to do in your code will take time to complete
- However, JavaScript does not wait for code to complete by default!
- This is because we do not want our applications to become unresponsive

SYNCHRONOUS CODE



```
console.log(" I ");

console.log(" eat ");

console.log(" Ice Cream ");
```

ASYNCHRONOUS CODE

```
console.log("I");

setTimeout(() => {
  console.log("eat");
}, 2000);

console.log("Ice Cream");
```

What do you think will happen?

EXAMPLE

- We can use async/await to write our code so it looks like it is syncronous

Example: background-colorizer extension

WORK TIME!