



Programming Hero

# Single-threaded Asynchronous vs Synchronous

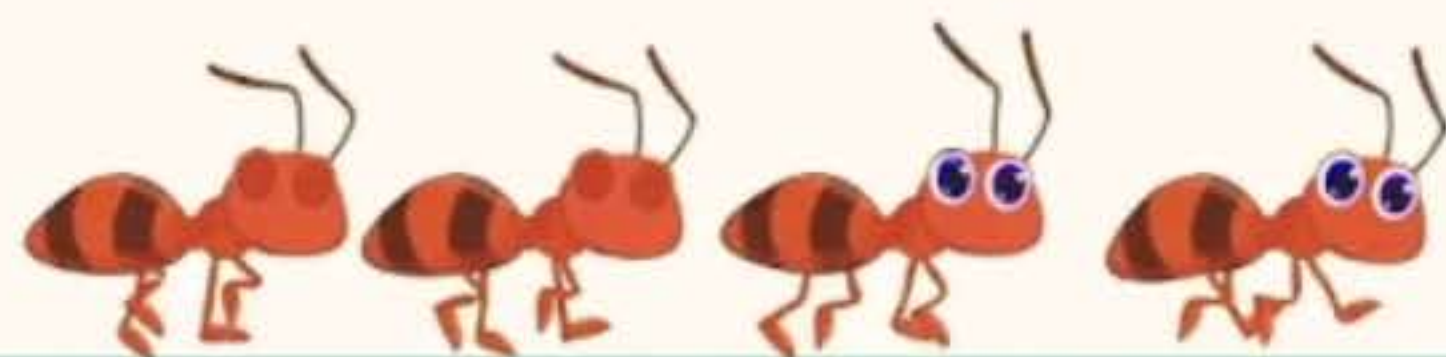




# | Single-threaded



# Single threaded



0:00:05

0:00:16

Ant Animation





# Single-threaded

PAGE 02

- ❖ **Single-threaded** means only one statement is executed at a time.
- ❖ JavaScript only has one call stack.
- ❖ JavaScript runs code line by line
- ❖ Must finish executing a piece of code before moving onto the next


```
function one() {  
  console.log('A');  
  two();  
  console.log('C');  
}  
  
function two() {  
  console.log('B');  
}  
  
one();
```

Call Stack

Console

'A'

one()

A man with short dark hair, wearing a blue t-shirt, is pointing his right index finger towards the 'Call Stack' area of the code editor. He is looking directly at the camera with a focused expression.



# Asynchronous vs Synchronous





# What do you mean by Synchronous?

---

- ❖ You'll see that until the first image is loaded completely
- ❖ the second image doesn't start loading.



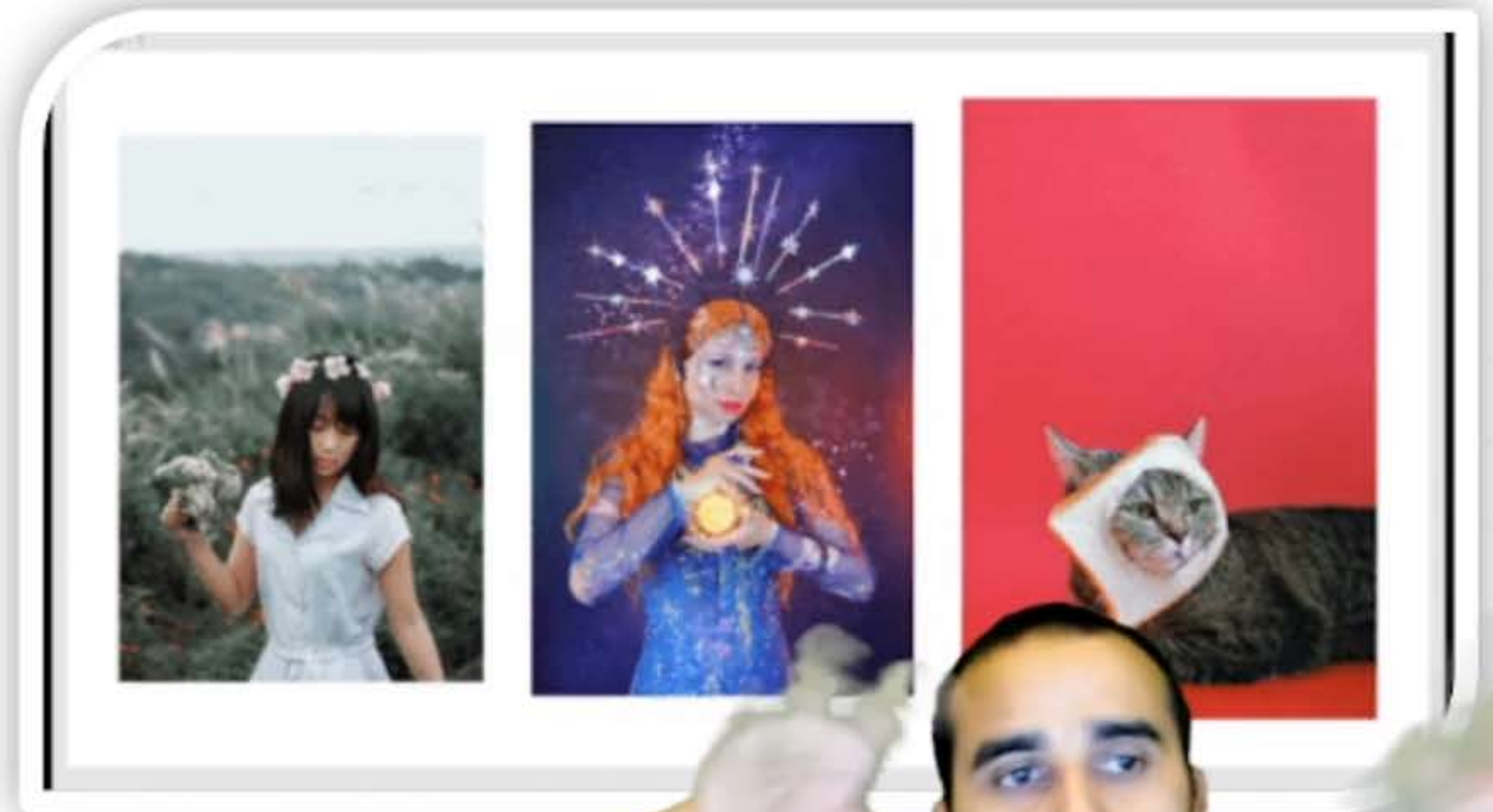
Hero



# What do you mean by Asynchronous?

---

- ❖ All the images are loading at their own pace.
- ❖ None of them is waiting for any of the others.



# Synchronous vs Asynchronous

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

Synchronous

Console

"I"

"eat"

"Ice Cream"

```
console.log("I");  
  
// This will be shown after 2 seconds  
  
setTimeout(()=>{  
  console.log("eat");  
},2000)  
  
console.log("Ice Cream")
```

Asynchronous

Console

"I"

"Ice Cream"

"eat"



# Synchronous vs Asynchronous

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

Synchronous

Console

"I"

"eat"

"Ice Cream"

```
console.log("I");  
  
// This will be shown after 2 seconds  
  
setTimeout(()=>{  
  console.log("eat");  
},2000)  
  
console.log("Ice Cream")
```

Asynchronous

Console

"I"

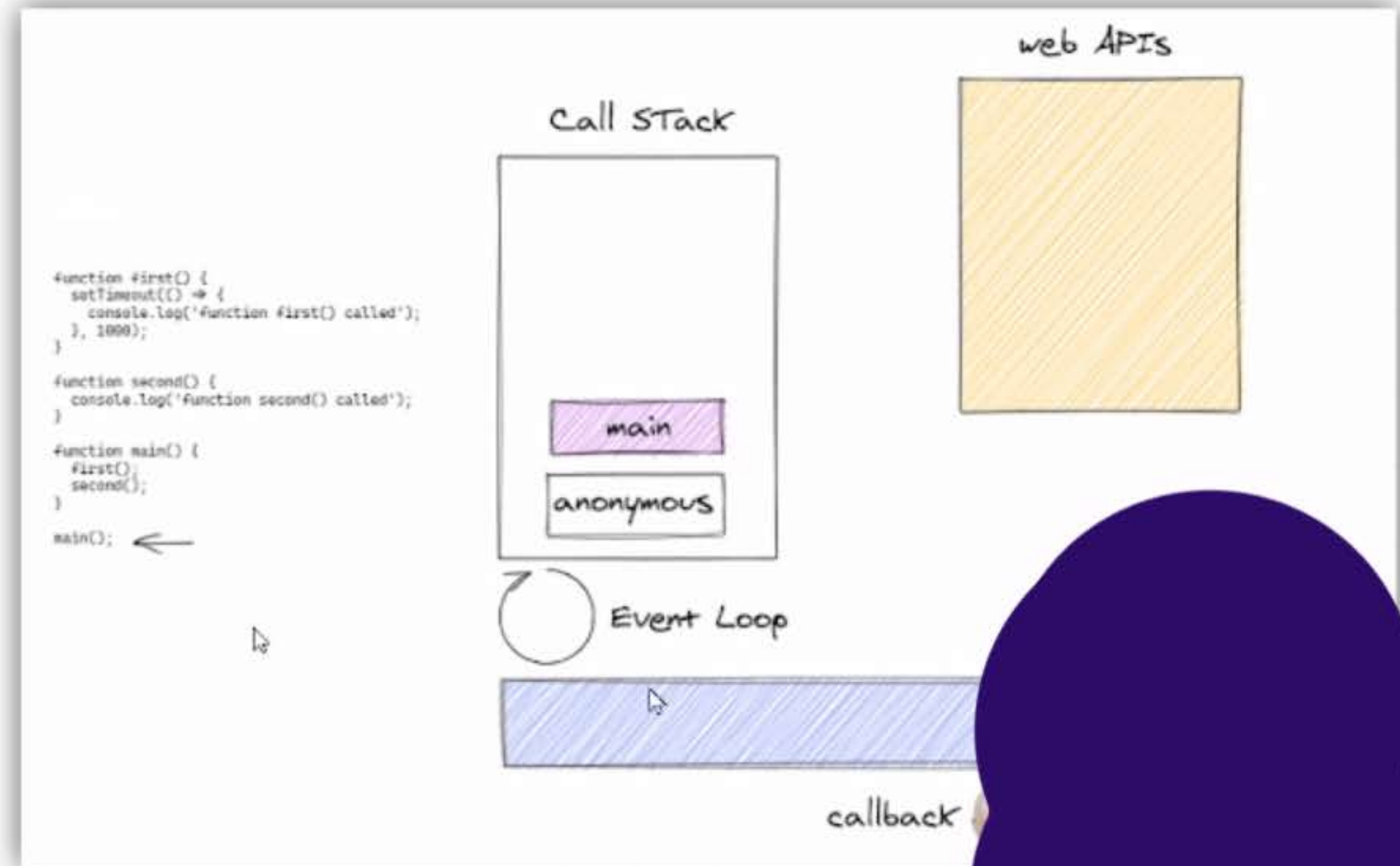
"Ice Cream"

"eat"



# How asynchronous JavaScript works?

- ❖ The **(anonymous)** function executes the script.
- ❖ **main()** calls **first()** and **first()** calls **setTimeout()**.
- ❖ **setTimeout()** pops off the call stack before **main()** can call **second()**.





**We will learn more about JS in  
our next videos**