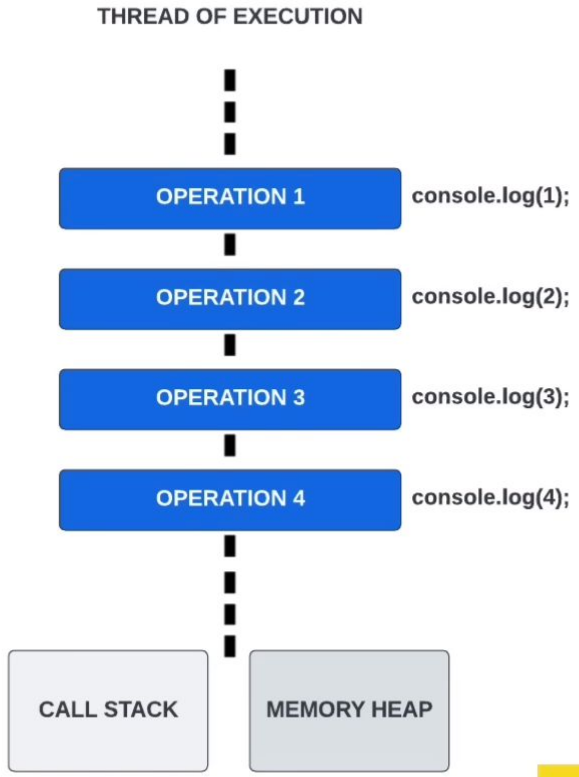
The Event Loops

Javascript: Behind The Scenes

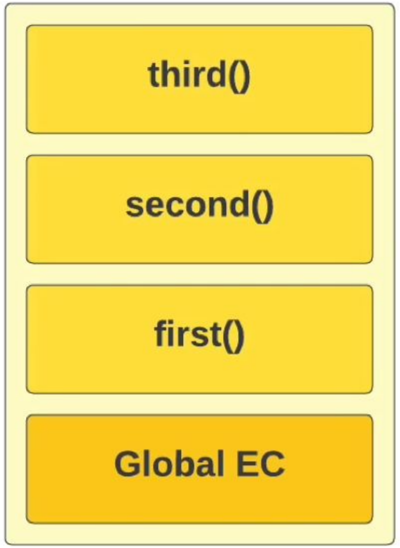
Thread Of Execution

- ✓ JavaScript is a **single-threaded** language
- ✓ Single sequential flow of control
- ✓ JavaScript is a **synchronous language** with asynchronous capabilities
- ✓ A thread has a **call stack** & **memory**

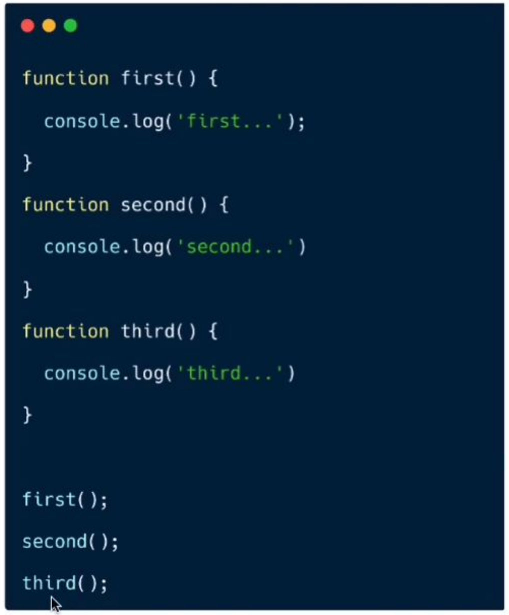


The Call Stack

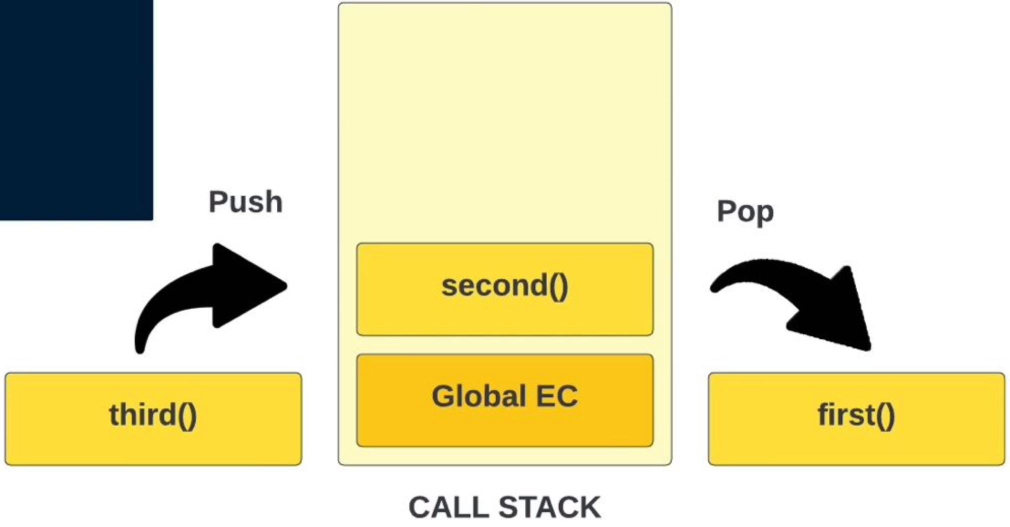
- ✓ Stack of functions to be executed
- ✓ Manages **execution contexts**
- ✓ Stacks are LIFO **last in first out**



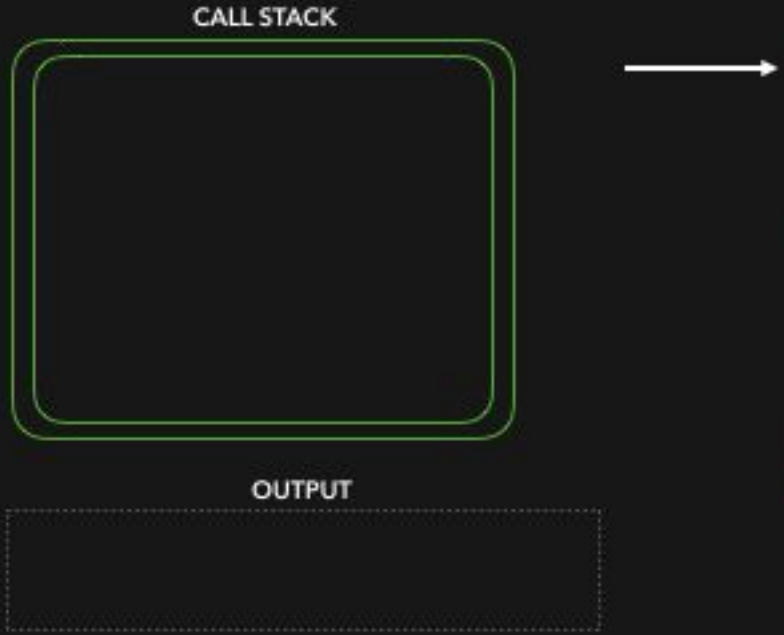
CALL STACK



```
function first() {
  console.log('first...');
}
function second() {
  console.log('second...')
}
function third() {
  console.log('third...')
}


first();
second();
third();
```

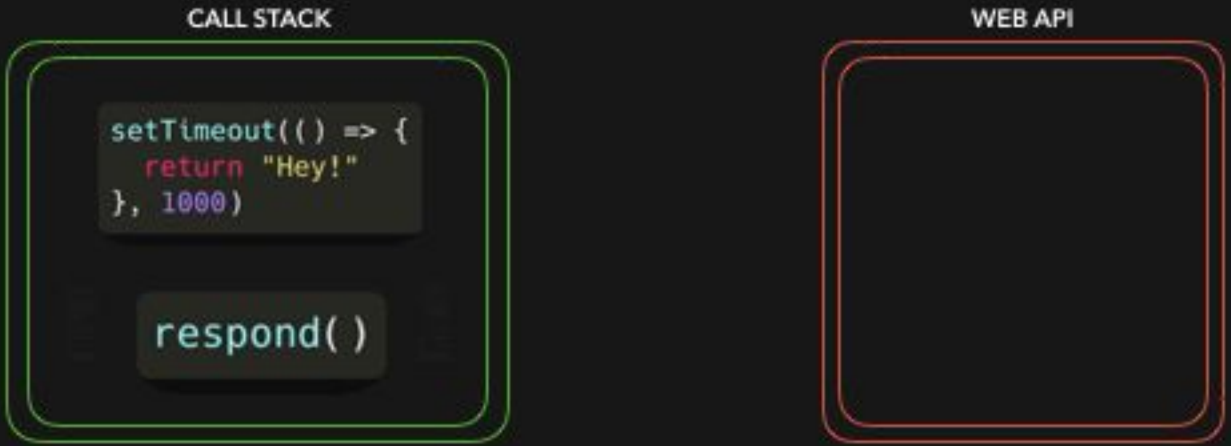


1 || Functions get **pushed to** the call stack when they're **invoked** and **popped off** when they **return a value**



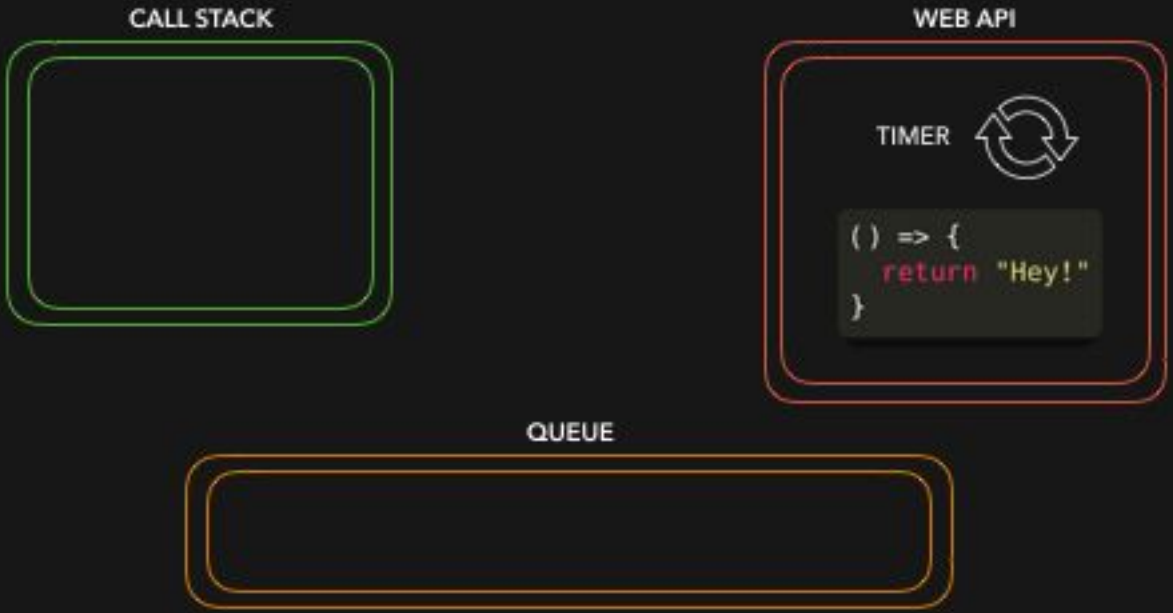
```
function greet() {
  return "Hello!"
}

function respond() {
  return setTimeout(() => {
    return "Hey!"
  }, 1000))
}

greet()
respond()
```

2 || `setTimeout` is provided to you by the *browser*, the **Web API** takes care of the callback we pass to it.



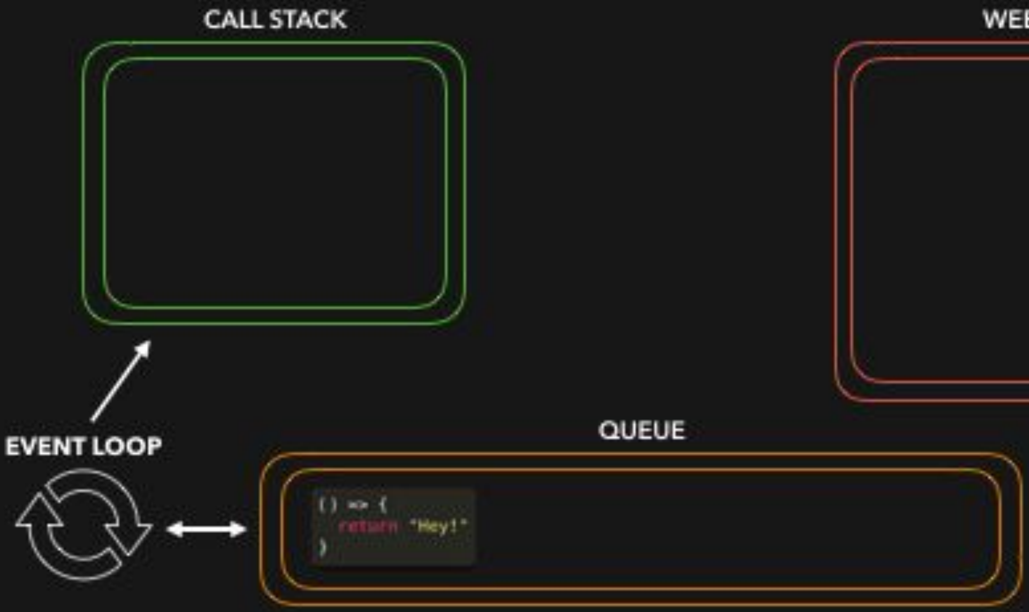
```
setTimeout(() => {
  return "Hey!"
}, 1000)
```

3 || When the timer has finished (1000ms in this case), the callback gets passed to the **callback queue**



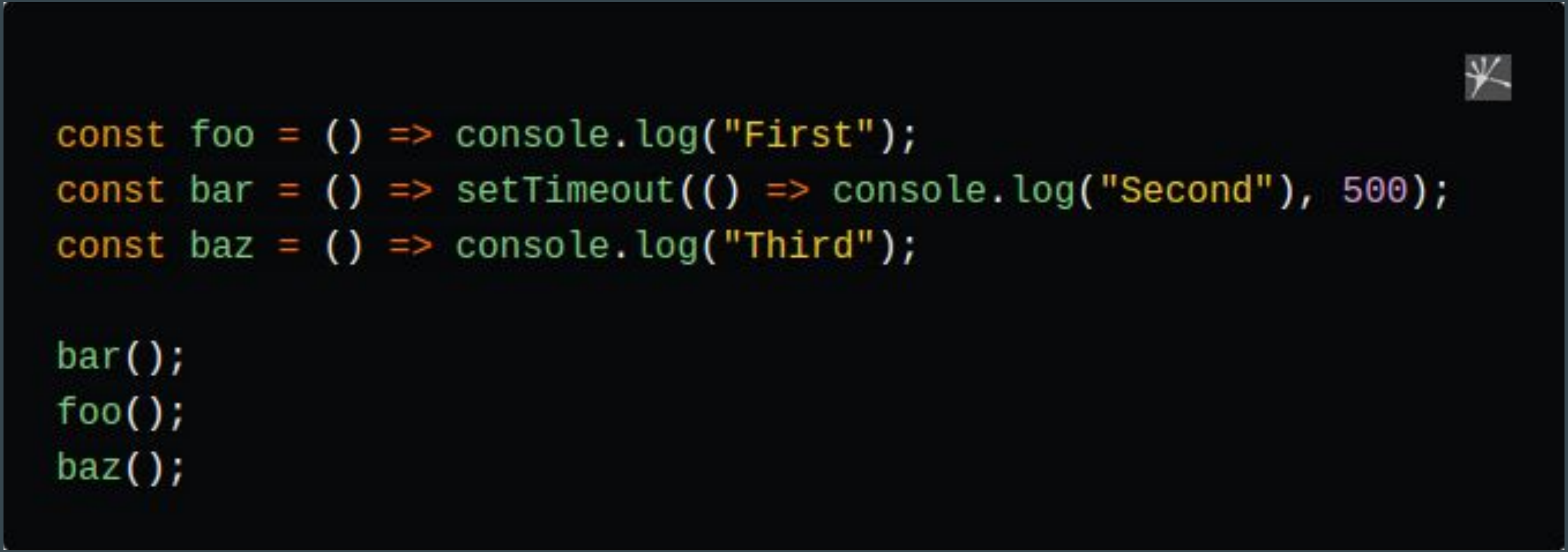
```
() => {
  return "Hey!"
}
```

4 || The **event loop** looks at the **callback queue** and the **call stack**. If the call stack is <u>empty</u>, it pushes the first item in the queue onto the stack.



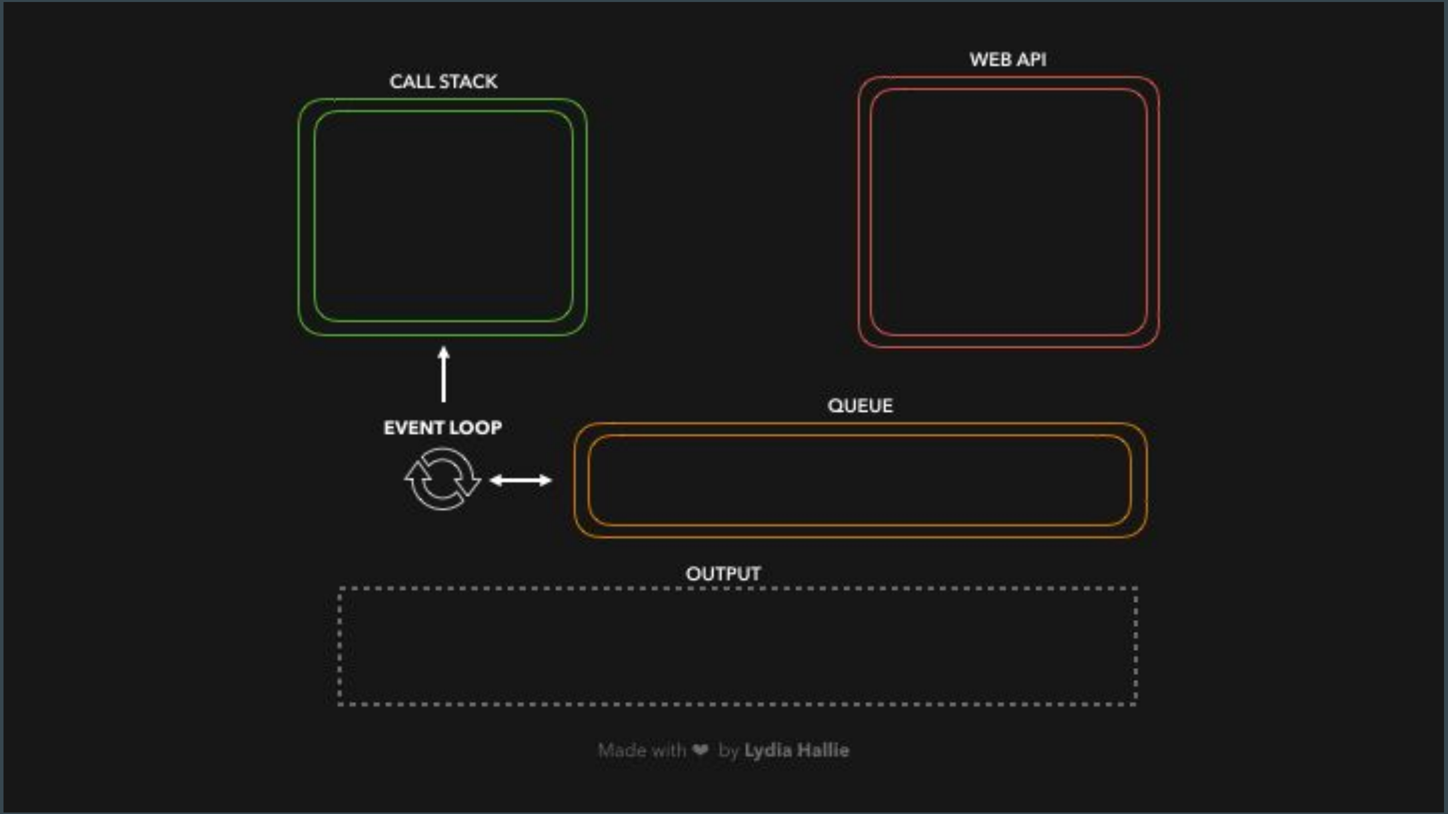
```
() => {
  return "Hey!"
}
```

Mini Quiz



```javascript
const foo = () => console.log("First");
const bar = () => setTimeout(() => console.log("Second"), 500);
const baz = () => console.log("Third");

bar();
foo();
baz();
```

Answer

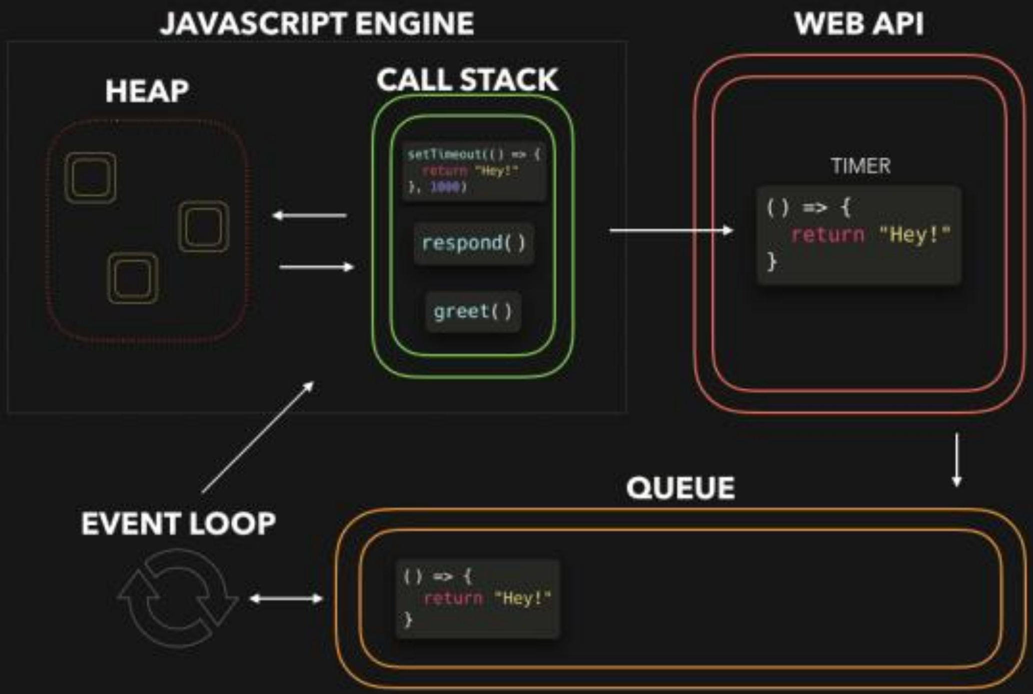


Output:
> First

>Third

>Second

Complete illustration



```
setTimeout(() => {
  return "Hey!"
}, 1000)
respond()
greet()
```

```
() => {
  return "Hey!"
}
```

```
function greet() {
  return "Hello!"
}

function respond() {
  return setTimeout(() => {
    return "Hey!"
  }, 1000))
}

greet()
respond()
```

Thank You

•••

Any Questions

Resources

https://www.youtube.com/watch?v=28AXSTCpsyU&t=131s&ab_channel=TraversyMedia

https://dev.to/lydiahallie/javascript-visualized-event-loop-3dif

http://latentflip.com/loupe