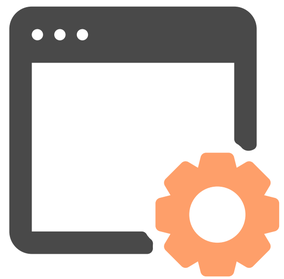
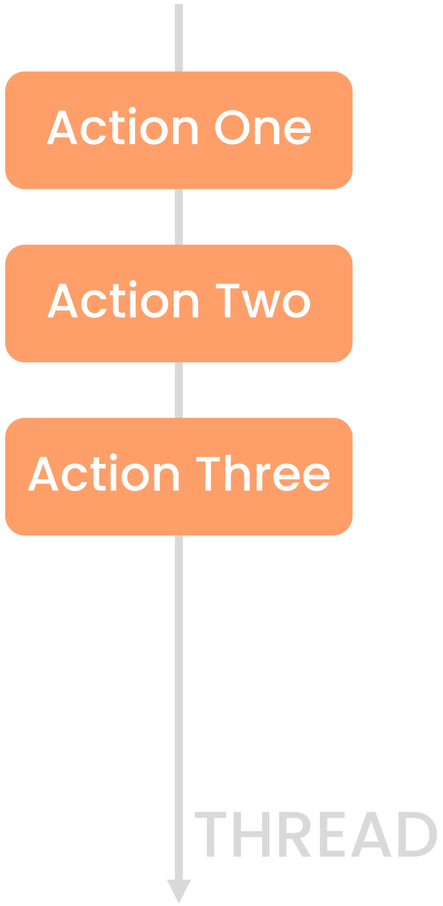
# Asynchronous Control Flow

M1W4 – October 18 Cohort



# Single Threaded

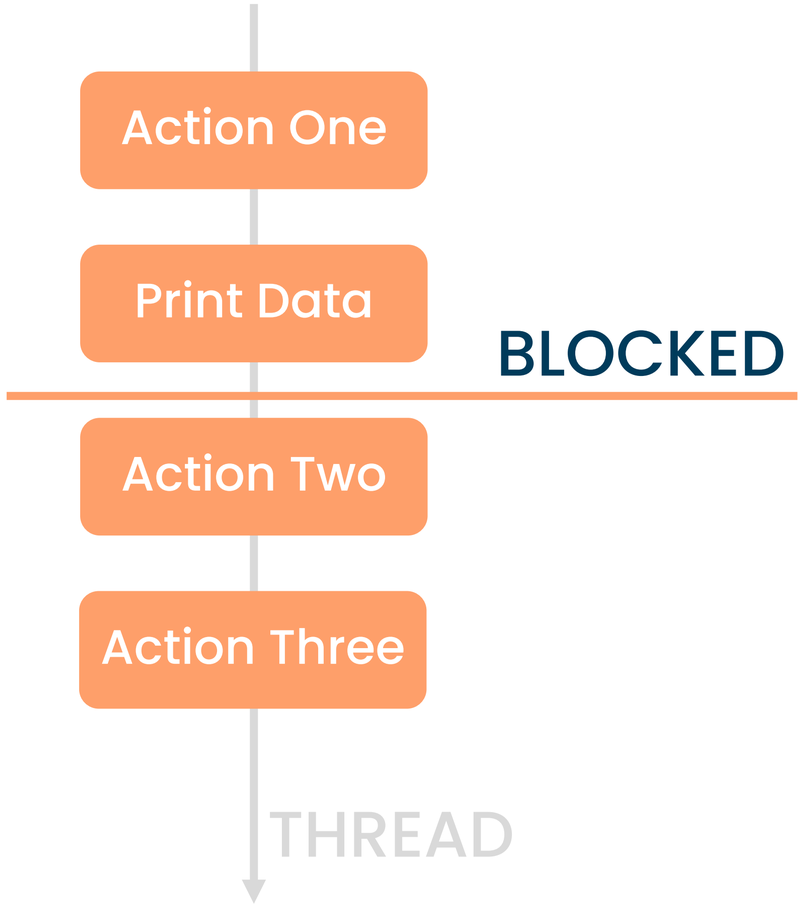
Each code is executed one after the other




```
1   function actionOne() {
2     console.log("(1) Action One!");
3   }
4
5   function actionTwo() {
6     console.log("(2) Action Two!");
7   }
8
9   function actionThree() {
10    console.log("(3) Action Three!");
11  }
12
13  actionOne();
14  actionTwo();
15  actionThree();
16
17
```

# Single Threaded

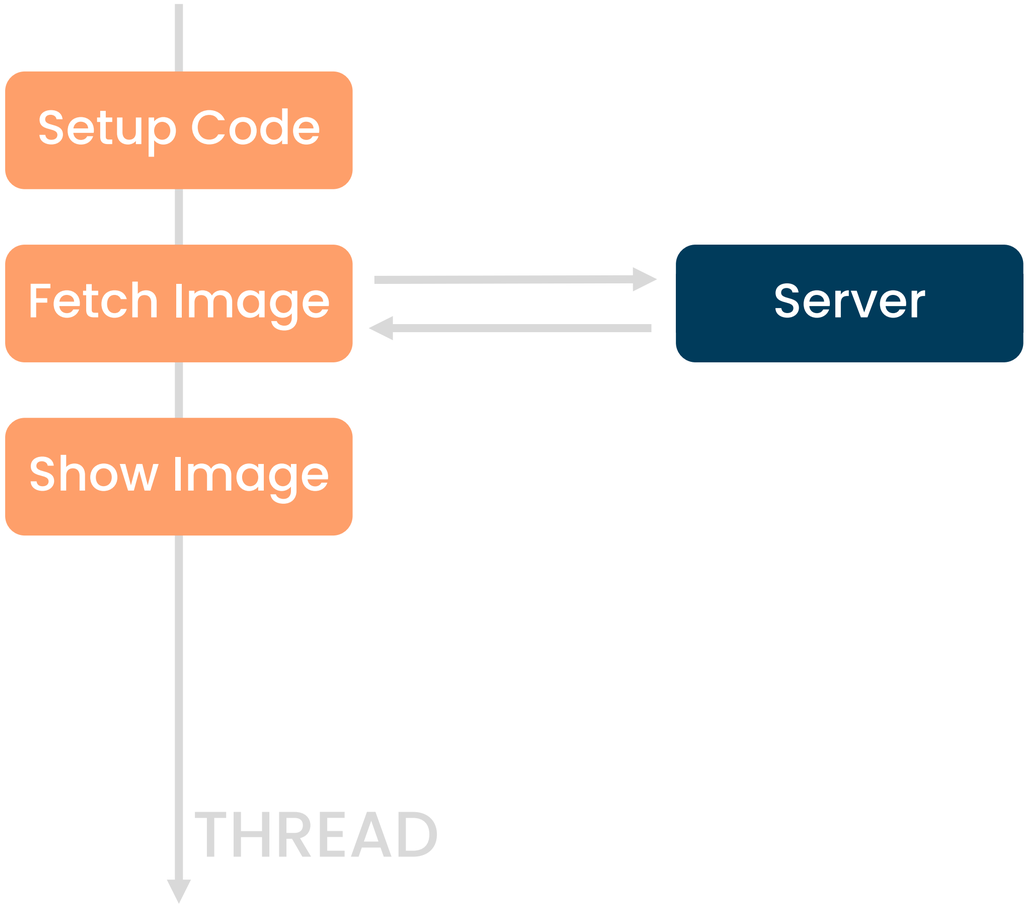
Each code is executed one after the other




```
1  function actionOne() {
2    console.log("(1) Action One!");
3  }
4
5  function printData() {
6    const stop = 1000;
7    for (let i = 0; i < stop; i++) {
8      const date = new Date();
9      console.log(date);
10   }
11 }
12
13 function actionTwo() {
14   console.log("(2) Action Two!");
15 }
16
17 function actionThree() {
18   console.log("(3) Action Three!");
19 }
20
21 actionOne();
22 printData();
23 actionTwo();
24 actionThree();
25
26
```

# Callbacks

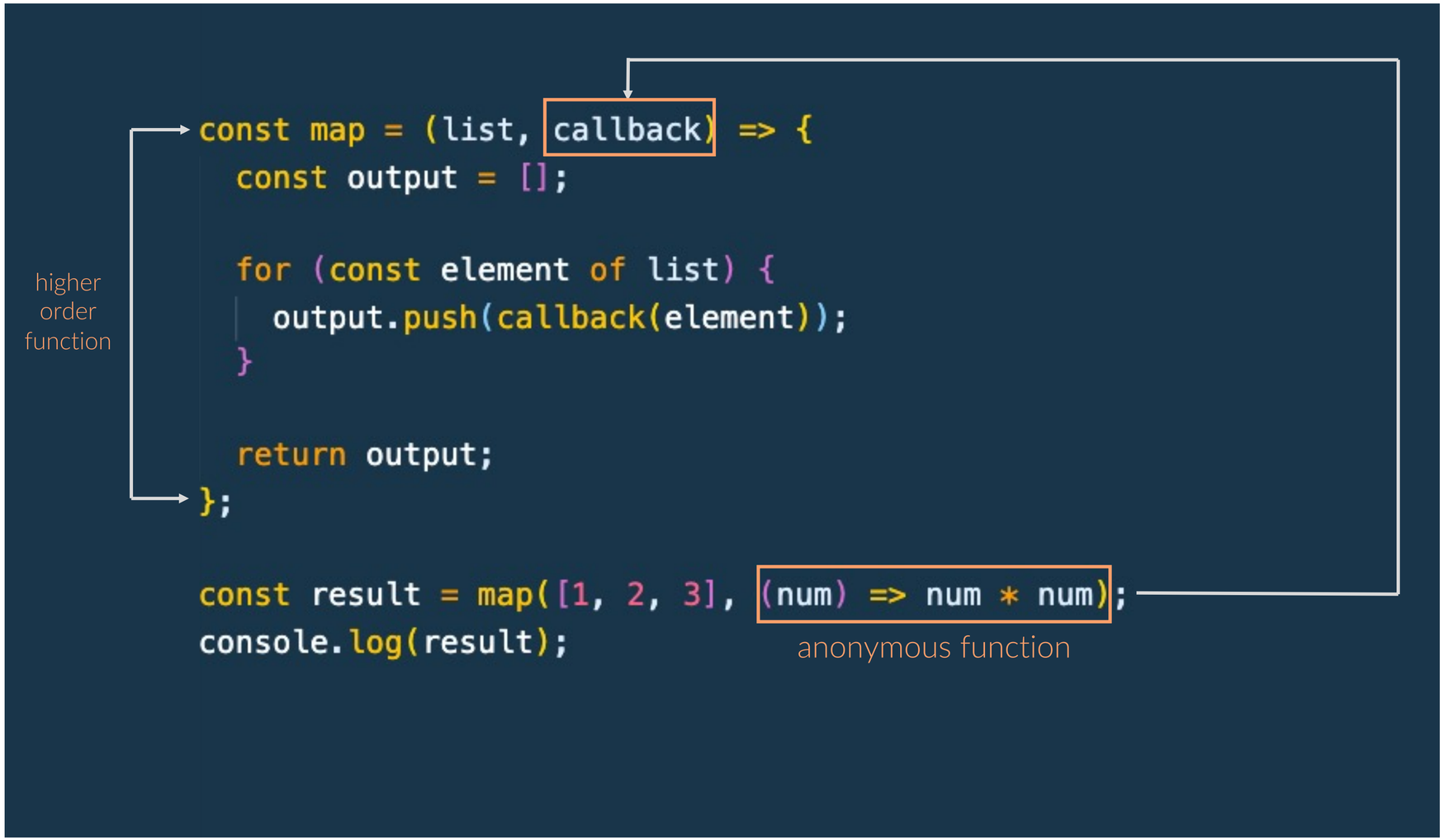
How do we return values from Asynchronous Operations?




```
1  import fetch from "node-fetch";
2
3  const url = "https://dog.ceo/api/breeds/image/random";
4
5  const response = fetch(url)
6    .then((response) => {
7      return response.json();
8    })
9    .then((data) => {
10     console.log("data", data);
11   });
12
13 const showImage = response.data;
14 console.log("showImage", showImage);
15
16
17
```

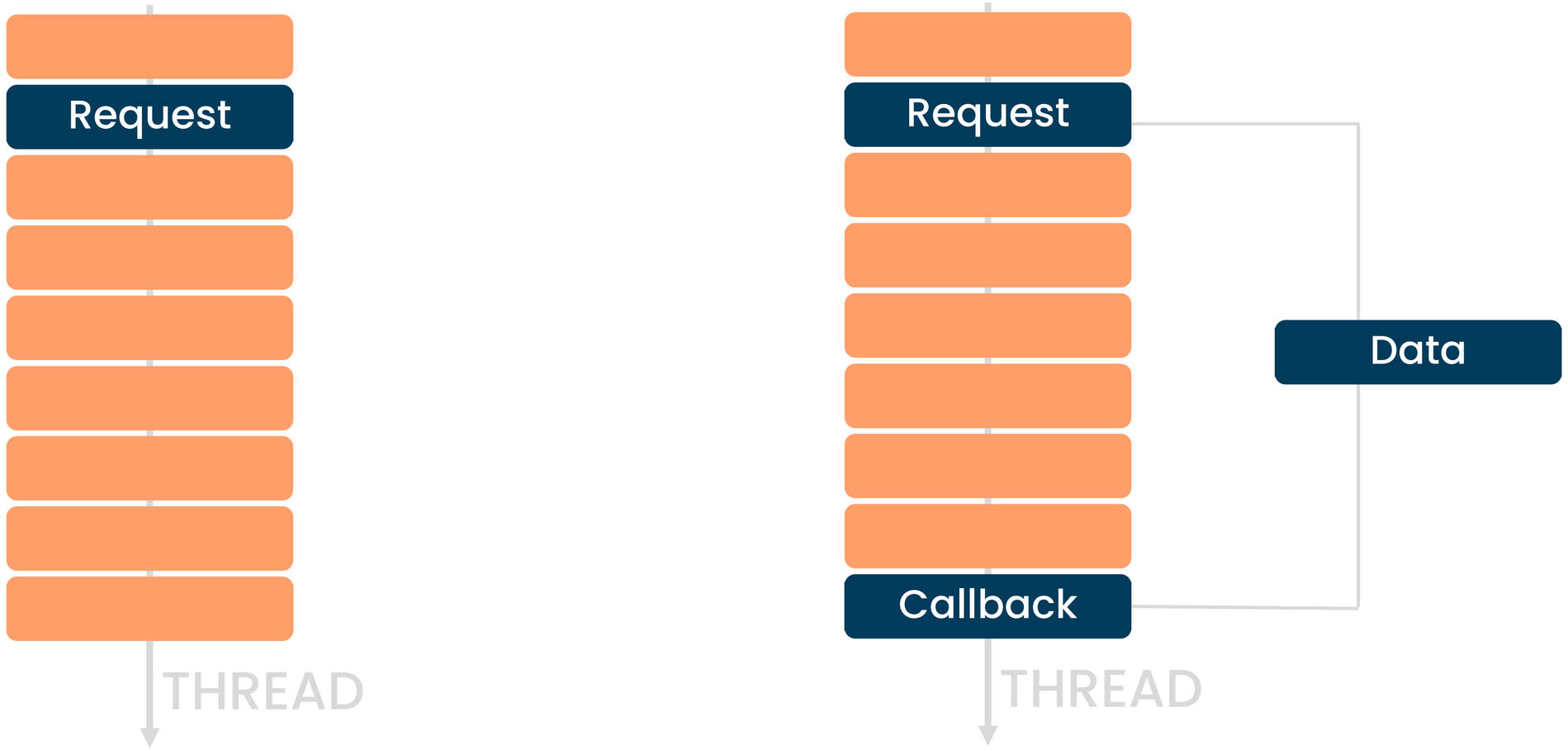
# Callbacks

How do we return values from Asynchronous Operations?



# Blocking vs Non-Blocking

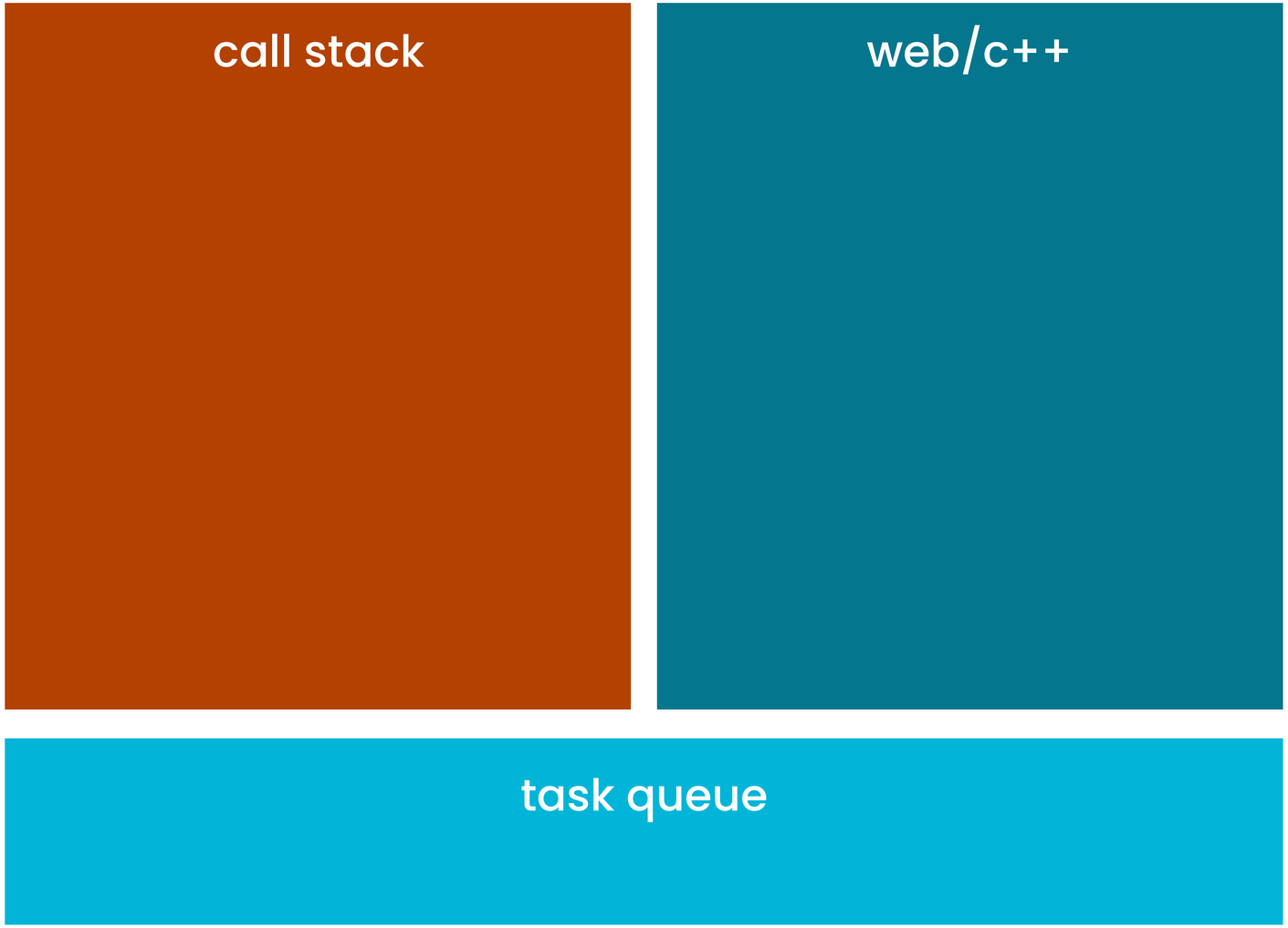
Synchronous vs Asynchronous Operations



# Asynchronous Workflow

What's happening under the hood?

```
const addition = (num1, num2) => {
 return num1 + num2;
};

const sayHi = (name) => {
 return `Hi ${name}`;
};

addition(3, 8);

setTimeout(() => {
 console.log("I'm Asynchronous");
}, 1000);

sayHi("Maui");

getData((data) => {
 console.log(data);
});
```



# Asynchronous Workflow

What's happening under the hood?

```
const addition = (num1, num2) => {
 return num1 + num2;
};
```

call stack

web/c++

task queue

# Asynchronous Workflow

What's happening under the hood?

```
const addition = (num1, num2) => {
 return num1 + num2;
};

const sayHi = (name) => {
 return `Hi ${name}`;
};
```

call stack

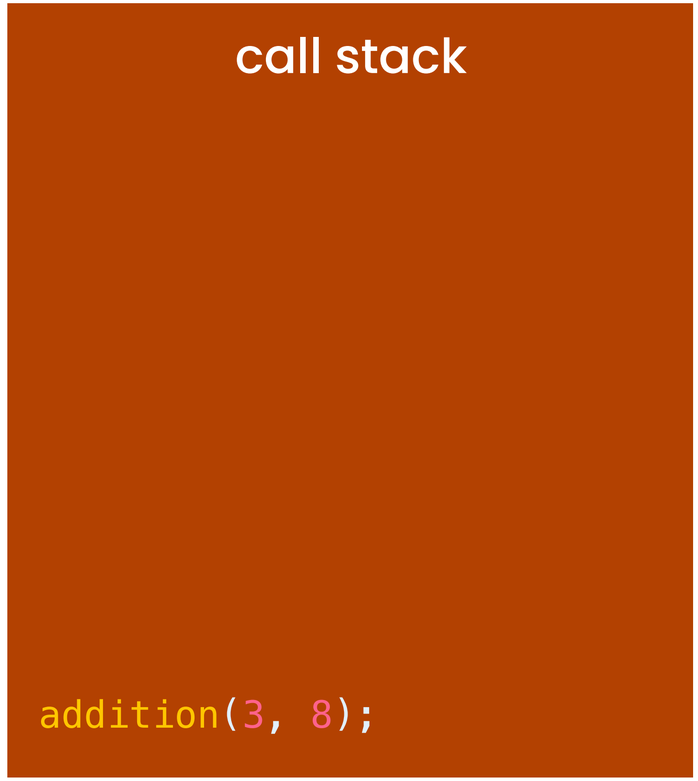
web/c++

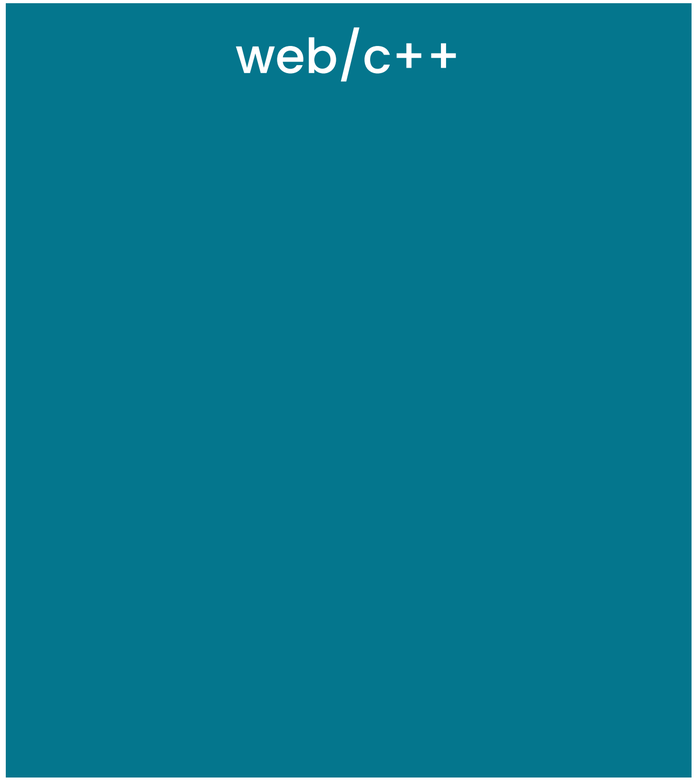
task queue

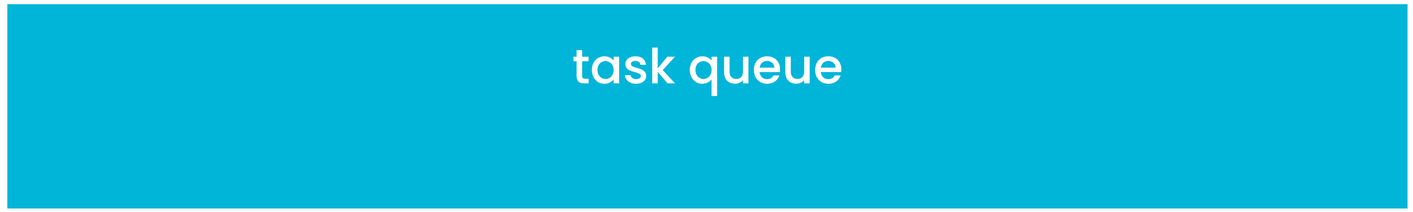
# Asynchronous Workflow

What's happening under the hood?

```
const addition = (num1, num2) => {
 return num1 + num2;
};

const sayHi = (name) => {
 return `Hi ${name}`;
};

addition(3, 8);
```



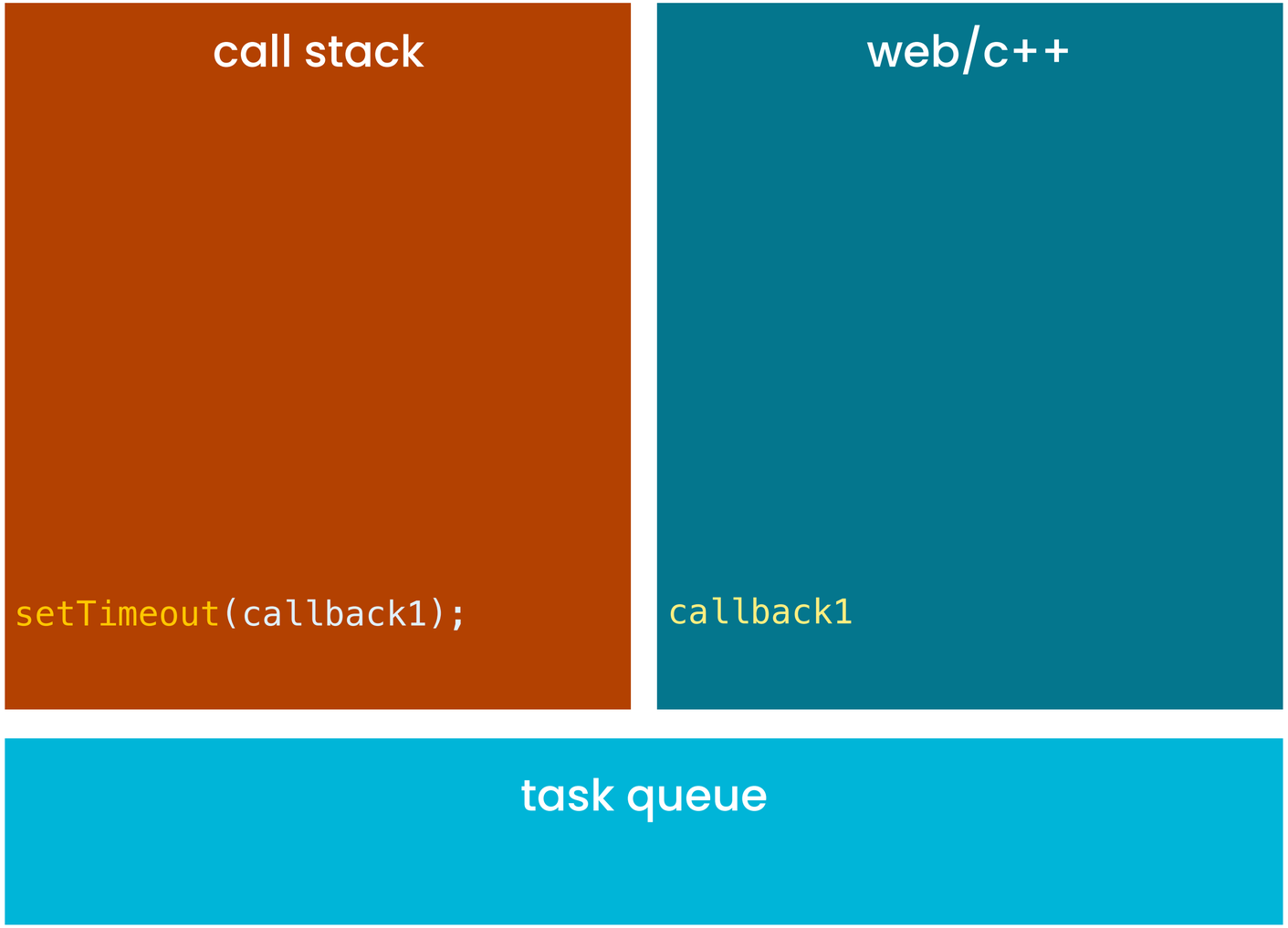



# Asynchronous Workflow

What's happening under the hood?

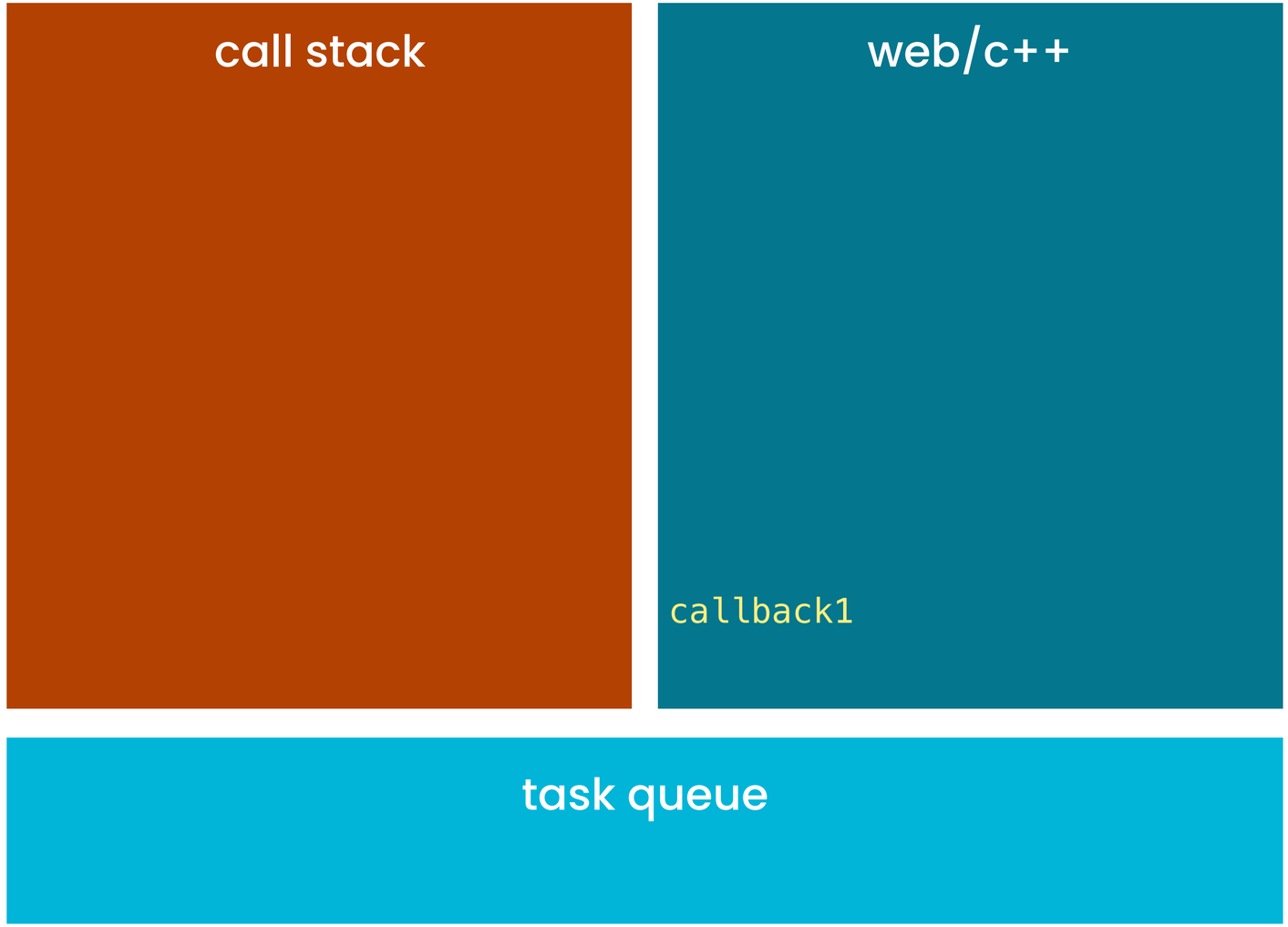
```
const addition = (num1, num2) => {
 return num1 + num2;
};

const sayHi = (name) => {
 return `Hi ${name}`;
};

addition(3, 8);

setTimeout(() => {
 console.log("I'm Asynchronous");
}, 1000);
```



# Asynchronous Workflow

What's happening under the hood?

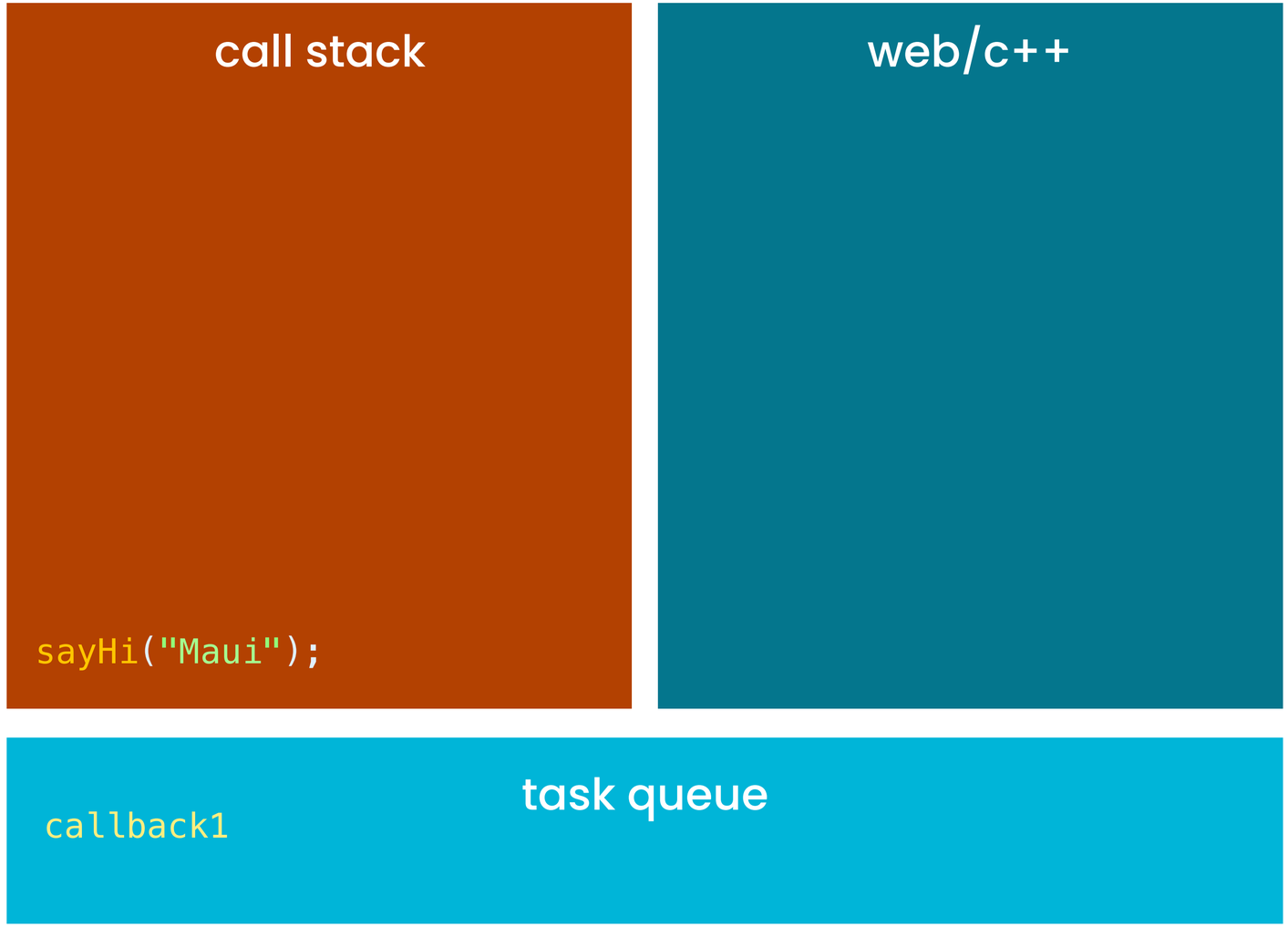
```
const addition = (num1, num2) => {
 return num1 + num2;
};

const sayHi = (name) => {
 return `Hi ${name}`;
};

addition(3, 8);

setTimeout(() => {
 console.log("I'm Asynchronous");
}, 1000);
```



# Asynchronous Workflow

What's happening under the hood?

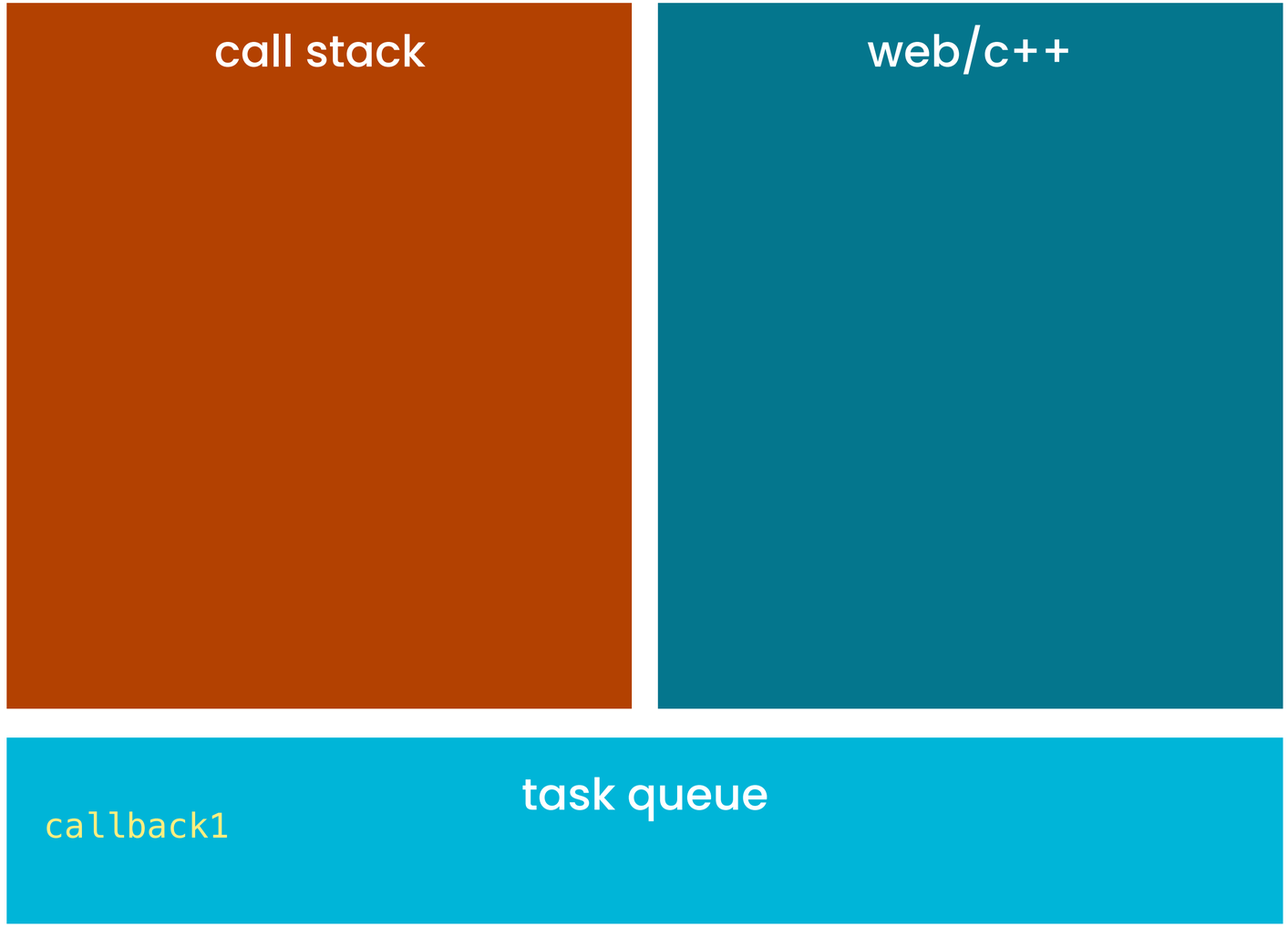
```
const addition = (num1, num2) => {
 return num1 + num2;
};

const sayHi = (name) => {
 return `Hi ${name}`;
};

addition(3, 8);

setTimeout(() => {
 console.log("I'm Asynchronous");
}, 1000);

sayHi("Maui");
```



# Asynchronous Workflow

What's happening under the hood?

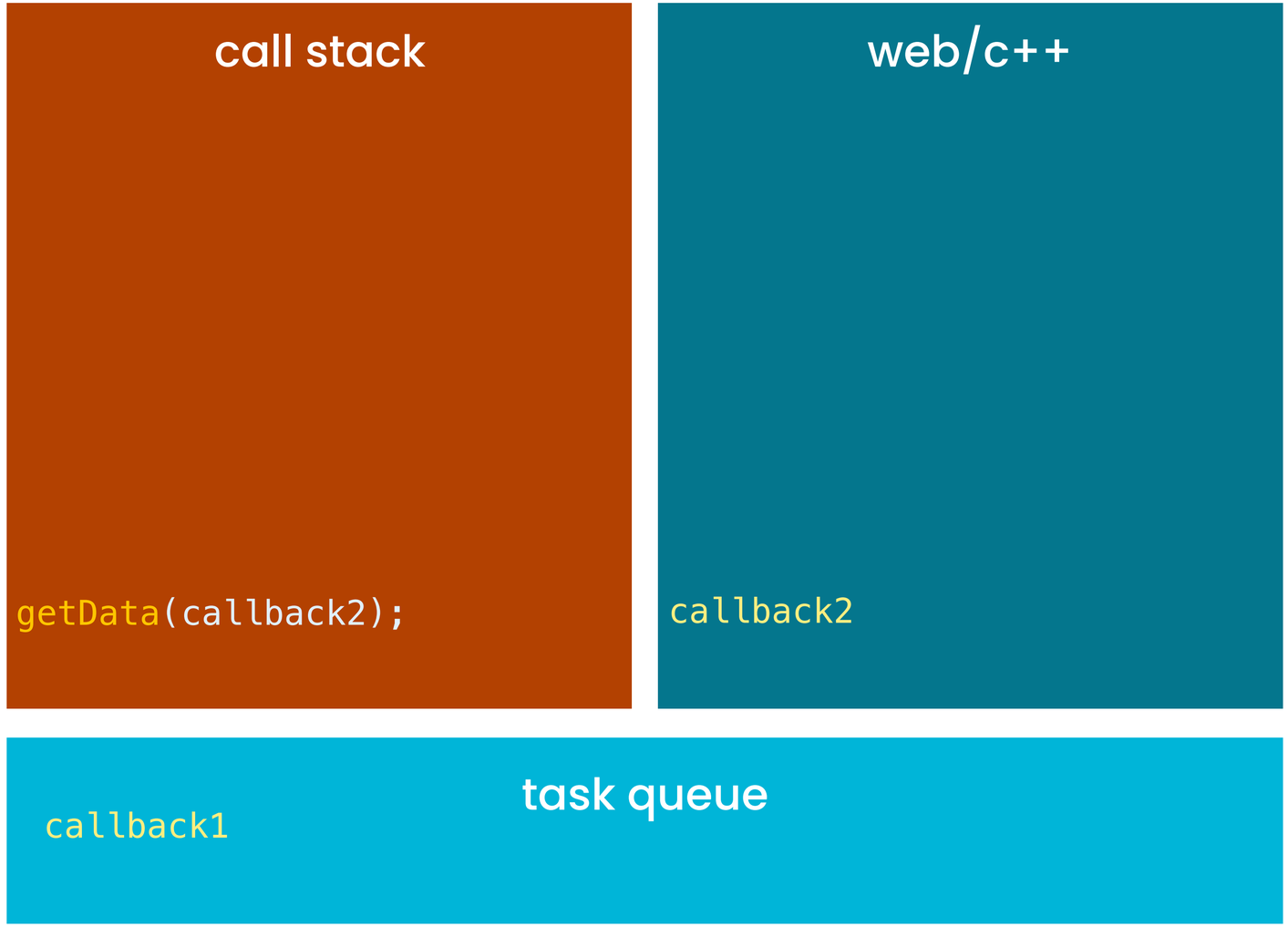
```
const addition = (num1, num2) => {
 return num1 + num2;
};

const sayHi = (name) => {
 return `Hi ${name}`;
};

addition(3, 8);

setTimeout(() => {
 console.log("I'm Asynchronous");
}, 1000);

sayHi("Maui");
```



# Asynchronous Workflow

What's happening under the hood?

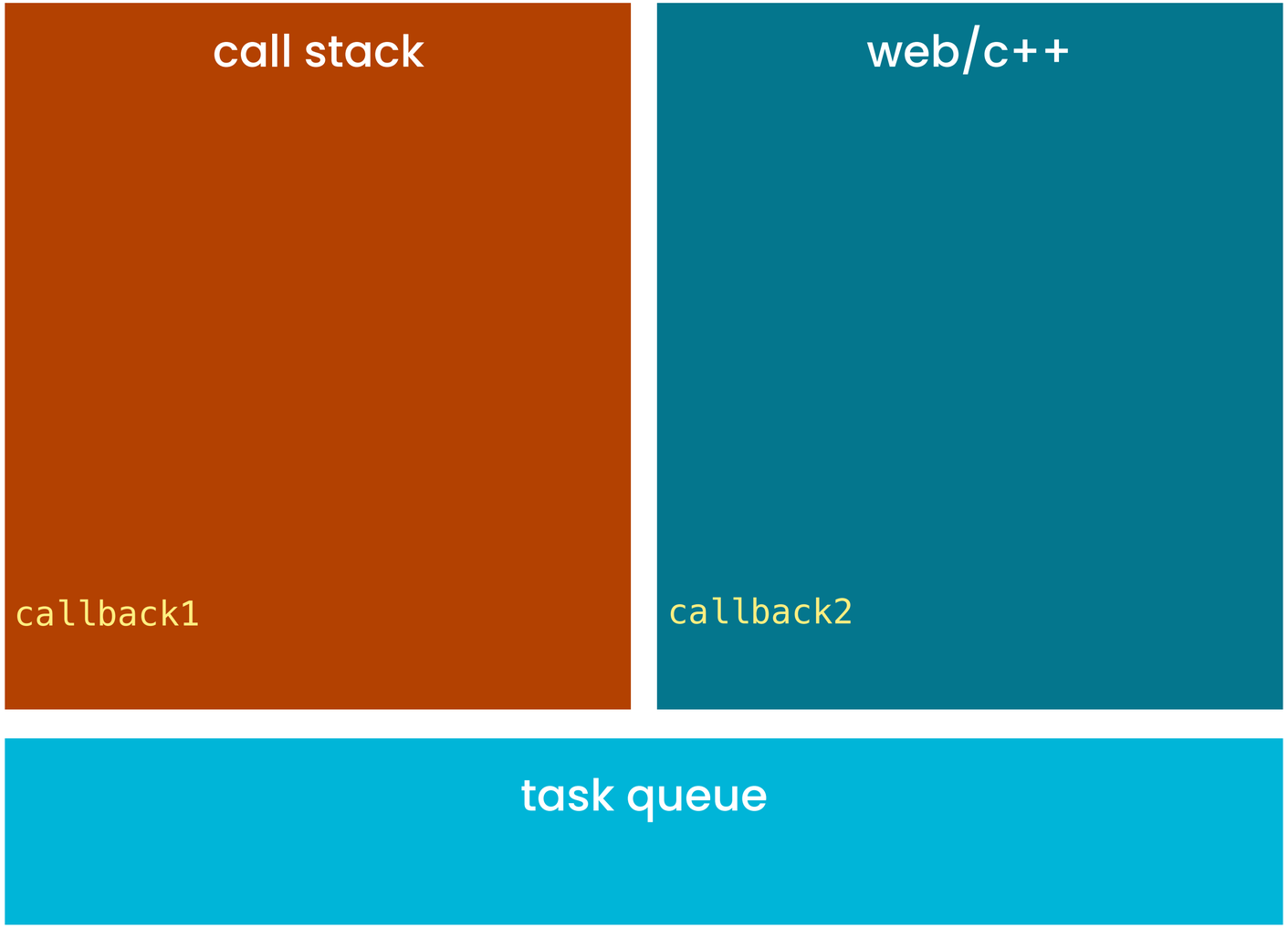
```
const addition = (num1, num2) => {
 return num1 + num2;
};

const sayHi = (name) => {
 return `Hi ${name}`;
};

addition(3, 8);

setTimeout(() => {
 console.log("I'm Asynchronous");
}, 1000);

sayHi("Maui");

getData((data) => {
 console.log(data);
});
```



# Asynchronous Workflow

What's happening under the hood?

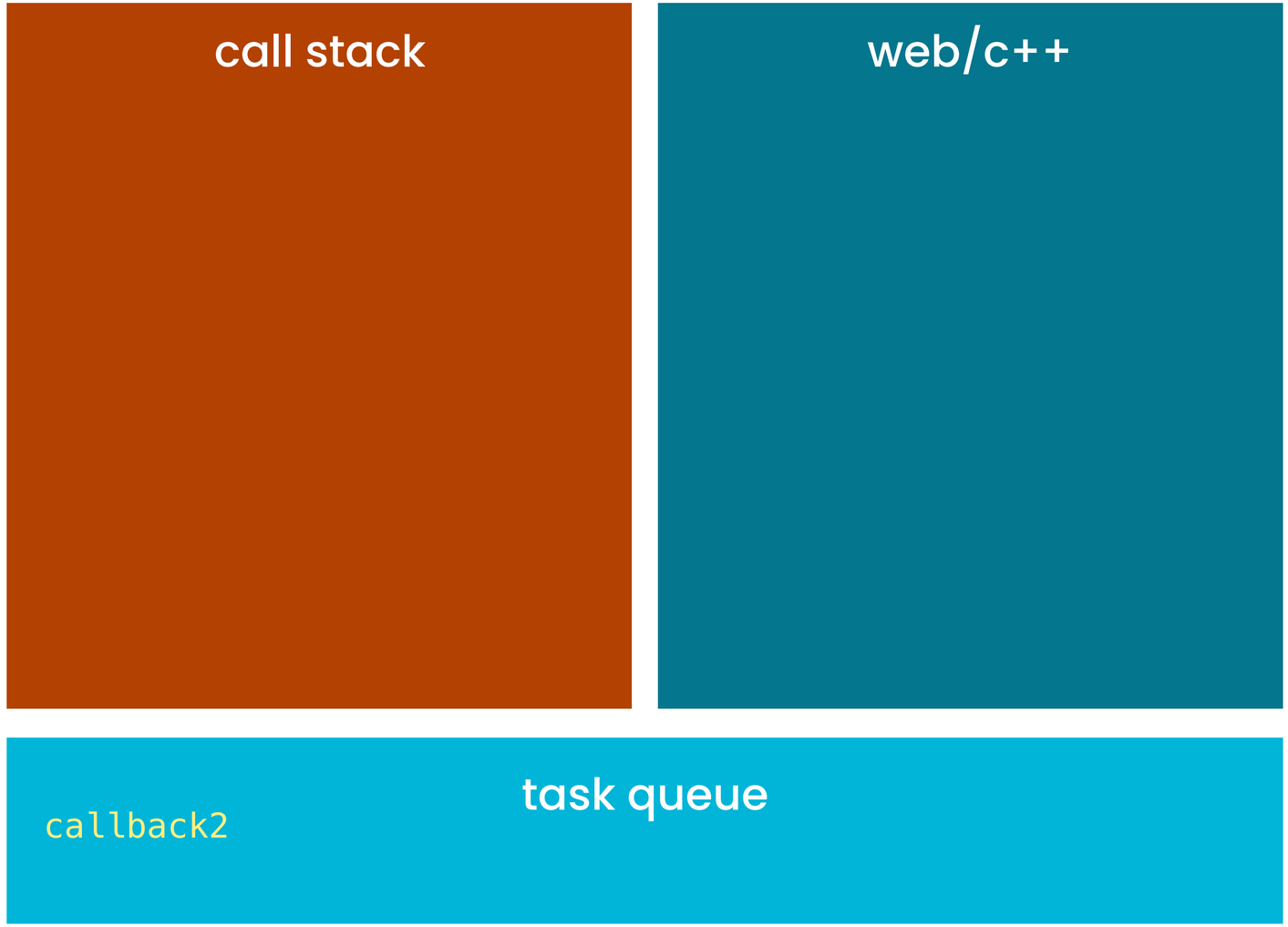
```
const addition = (num1, num2) => {
 return num1 + num2;
};

const sayHi = (name) => {
 return `Hi ${name}`;
};

addition(3, 8);

setTimeout(() => {
 console.log("I'm Asynchronous");
}, 1000);

sayHi("Maui");

getData((data) => {
 console.log(data);
});
```



# Asynchronous Workflow

What's happening under the hood?

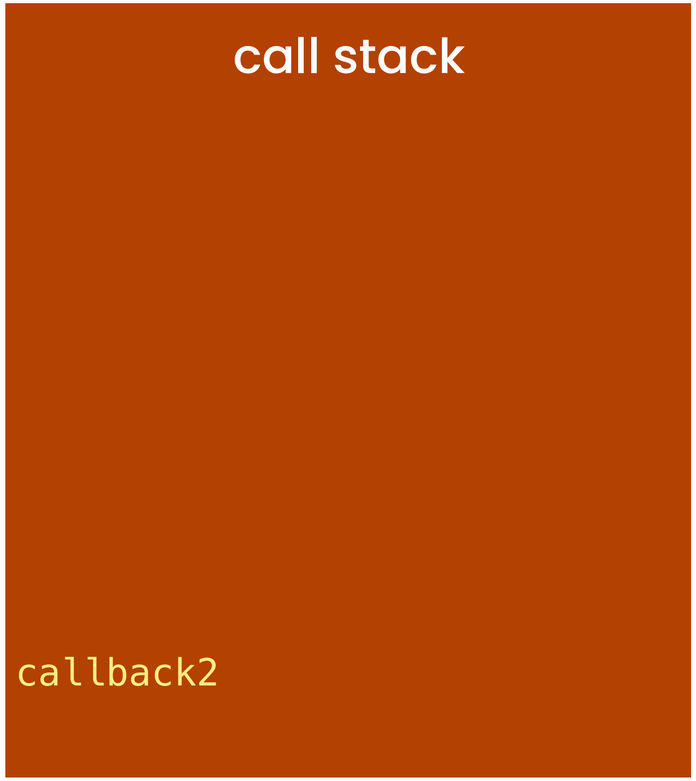
```
const addition = (num1, num2) => {
 return num1 + num2;
};

const sayHi = (name) => {
 return `Hi ${name}`;
};

addition(3, 8);

setTimeout(() => {
 console.log("I'm Asynchronous");
}, 1000);

sayHi("Maui");

getData((data) => {
 console.log(data);
});
```

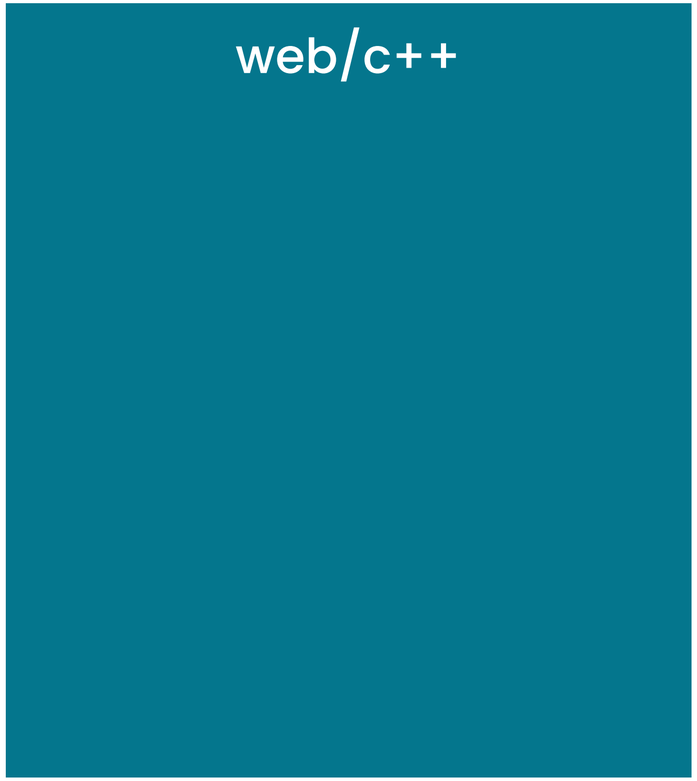


# Asynchronous Workflow

What's happening under the hood?

```
const addition = (num1, num2) => {
 return num1 + num2;
};

const sayHi = (name) => {
 return `Hi ${name}`;
};

addition(3, 8);

setTimeout(() => {
 console.log("I'm Asynchronous");
}, 1000);

sayHi("Maui");

getData((data) => {
 console.log(data);
});
```




# Asynchronous Workflow

What's happening under the hood?

```
const addition = (num1, num2) => {
 return num1 + num2;
};

const sayHi = (name) => {
 return `Hi ${name}`;
};

addition(3, 8);

setTimeout(() => {
 console.log("I'm Asynchronous");
}, 1000);

sayHi("Maui");

getData((data) => {
 console.log(data);
});
```