level

ECMAScript 6 & Beyond the web

How JavaScript works?

JS

How JavaScript **really** works?

Is JavaScript single or multi threaded?

JS

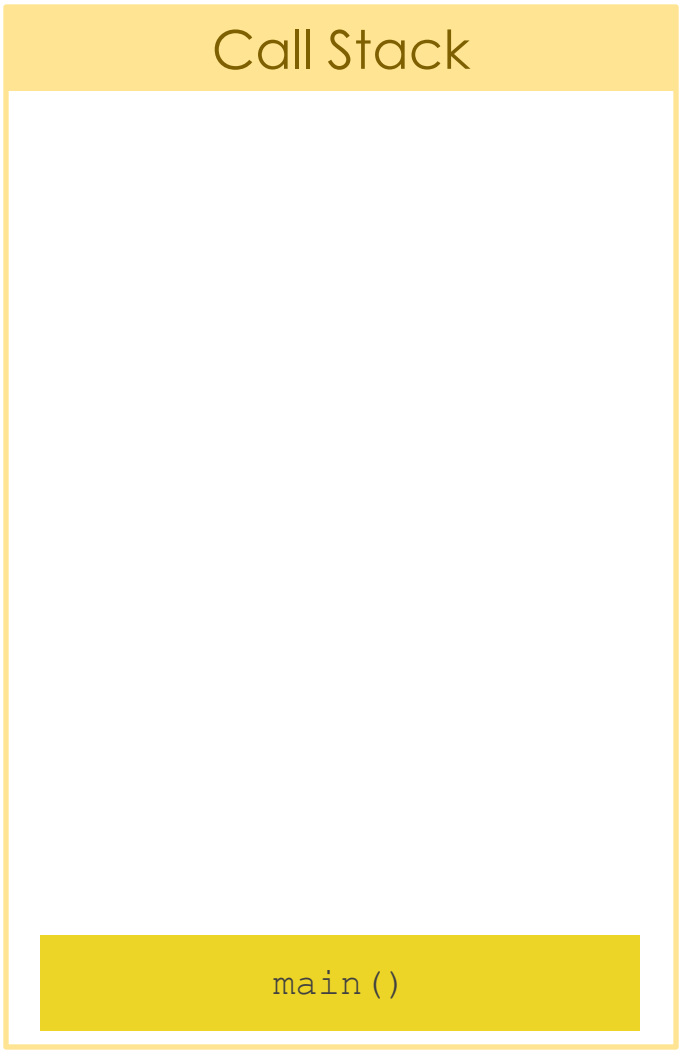
Call Stack

```
function sayHello(){
    console.log("Hello, OS");
}

function sayHi(){
    sayHello();
    console.log("Hi, OS");
}

sayHi();
```

Call Stack

Call Stack

```
function sayHello(){
    console.log("Hello, OS");
}

function sayHi(){
    sayHello();
    console.log("Hi, OS");
}

sayHi();
```



Call Stack

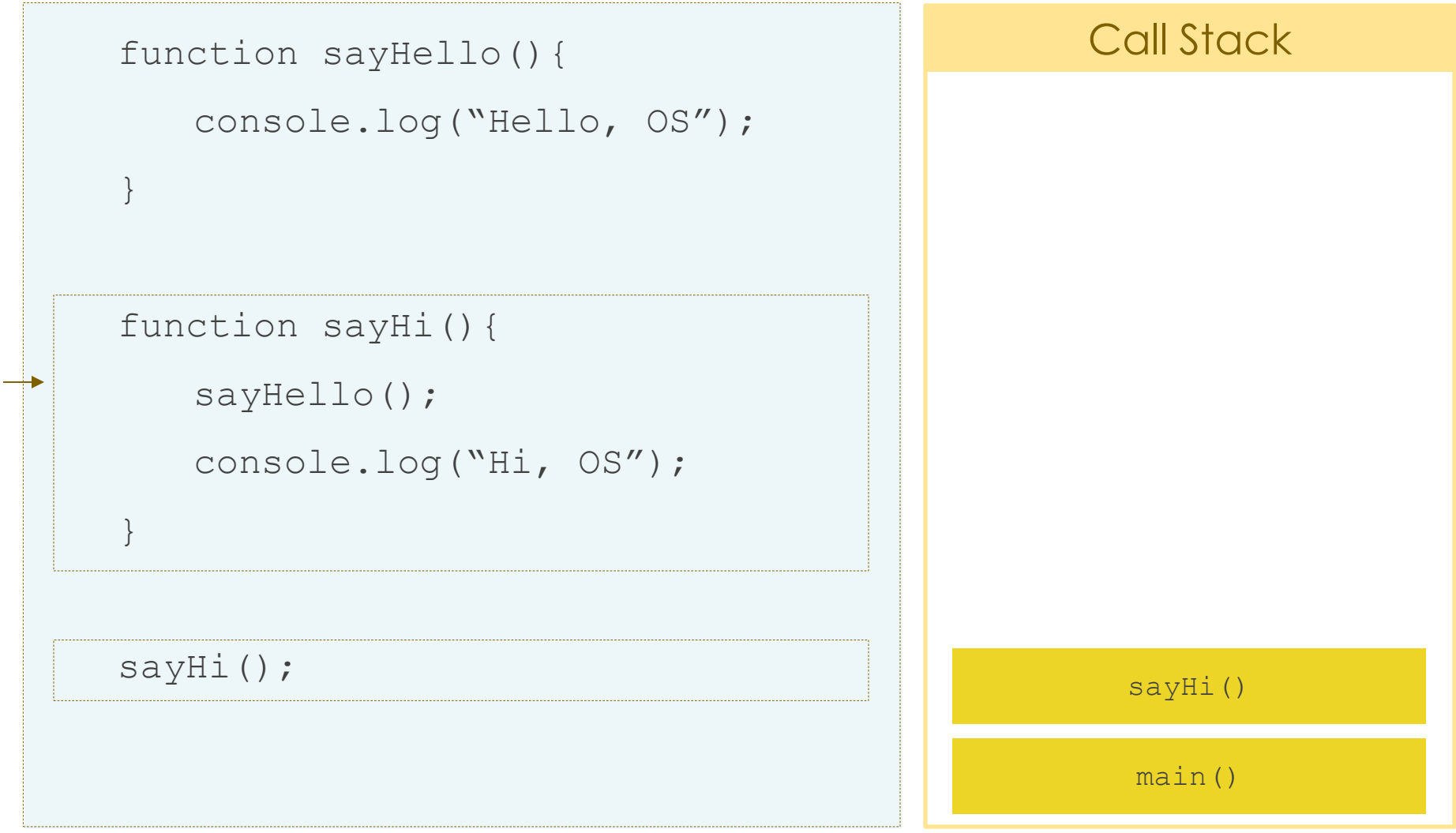
```
function sayHello(){
    console.log("Hello, OS");
}

function sayHi(){
    sayHello();
    console.log("Hi, OS");
}

sayHi();
```

Call Stack

main()

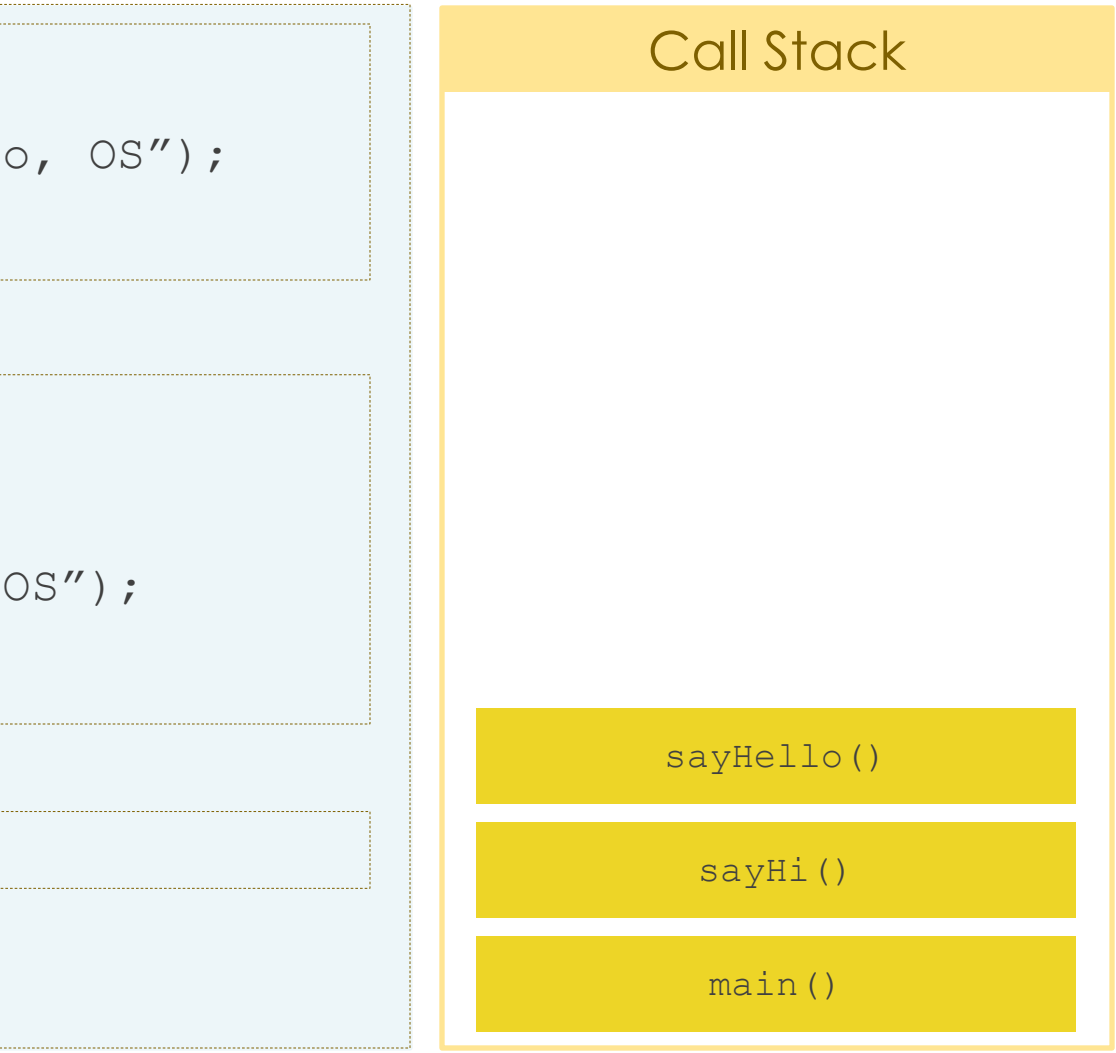
Call Stack

```
function sayHello(){
    console.log("Hello, OS");
}

function sayHi(){
    sayHello();
    console.log("Hi, OS");
}

sayHi();
```



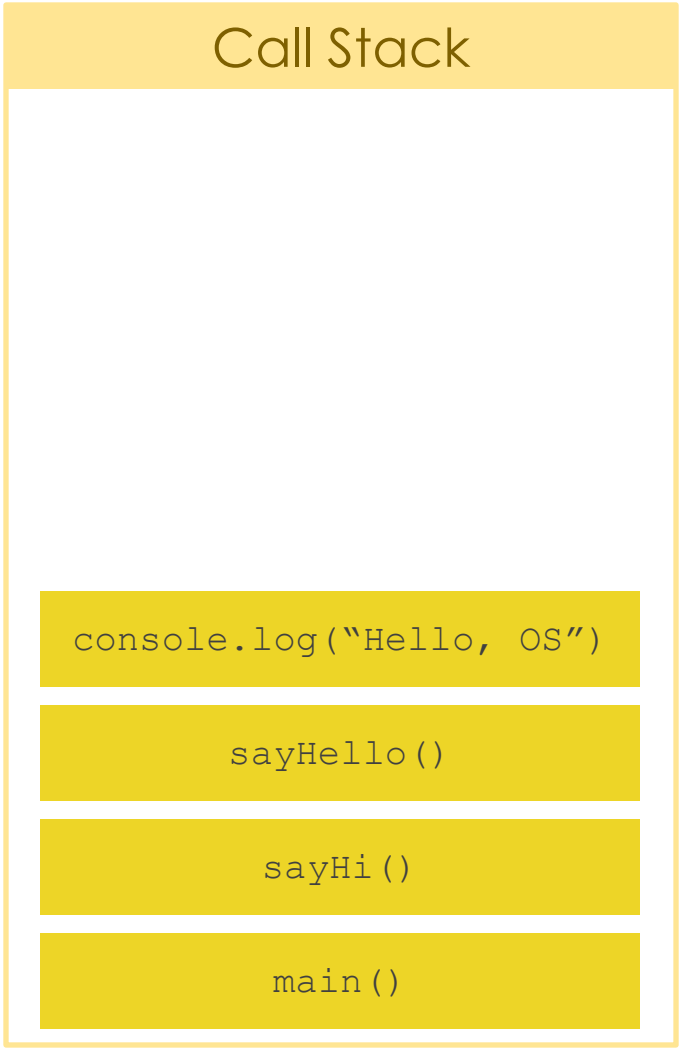
Call Stack

```
function sayHello(){
    console.log("Hello, OS");
}

function sayHi(){
    sayHello();
    console.log("Hi, OS");
}

sayHi();
```



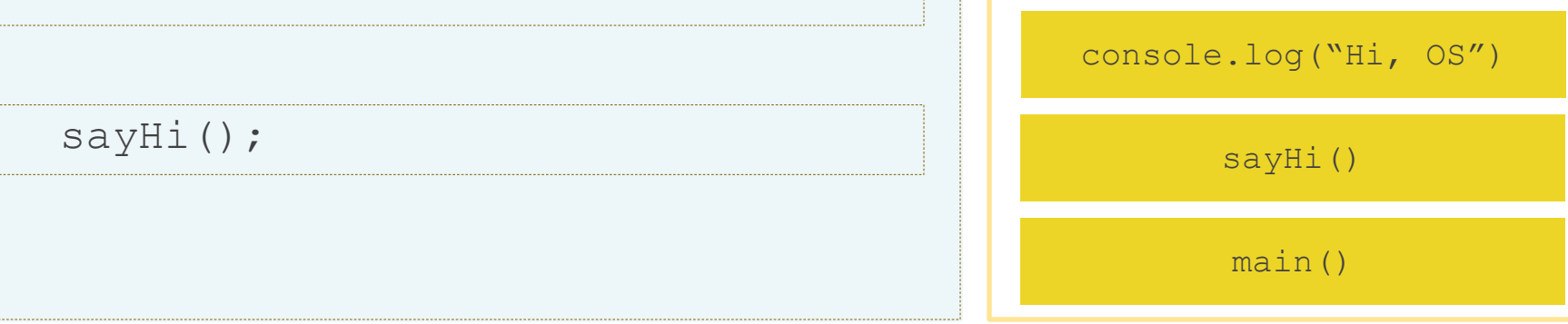
Call Stack

```
function sayHello(){
    console.log("Hello, OS");
}

function sayHi(){
    sayHello();
    console.log("Hi, OS");
}

sayHi();
```



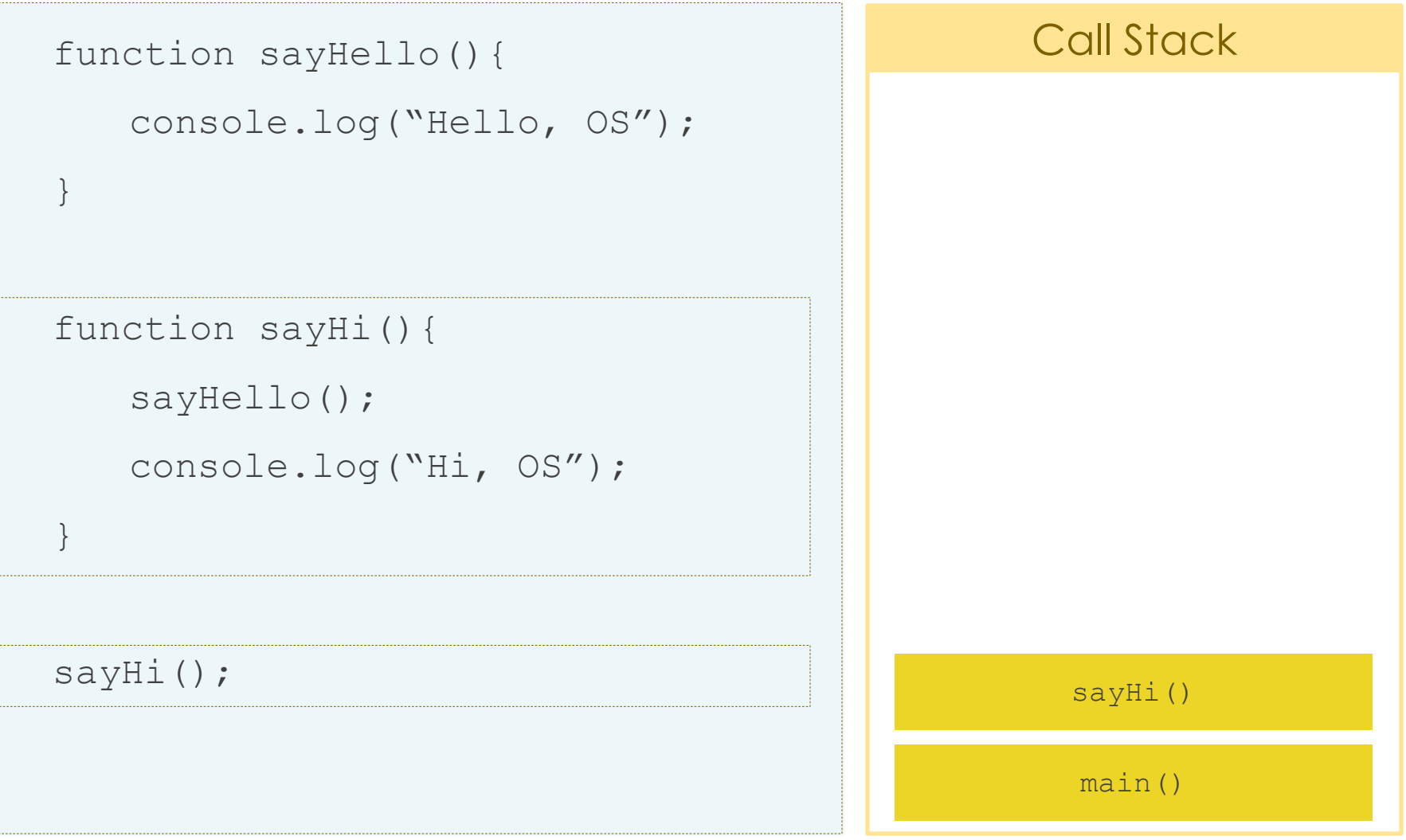
Call Stack

```
function sayHello(){
    console.log("Hello, OS");
}

function sayHi(){
    sayHello();
    console.log("Hi, OS");
}

sayHi();
```



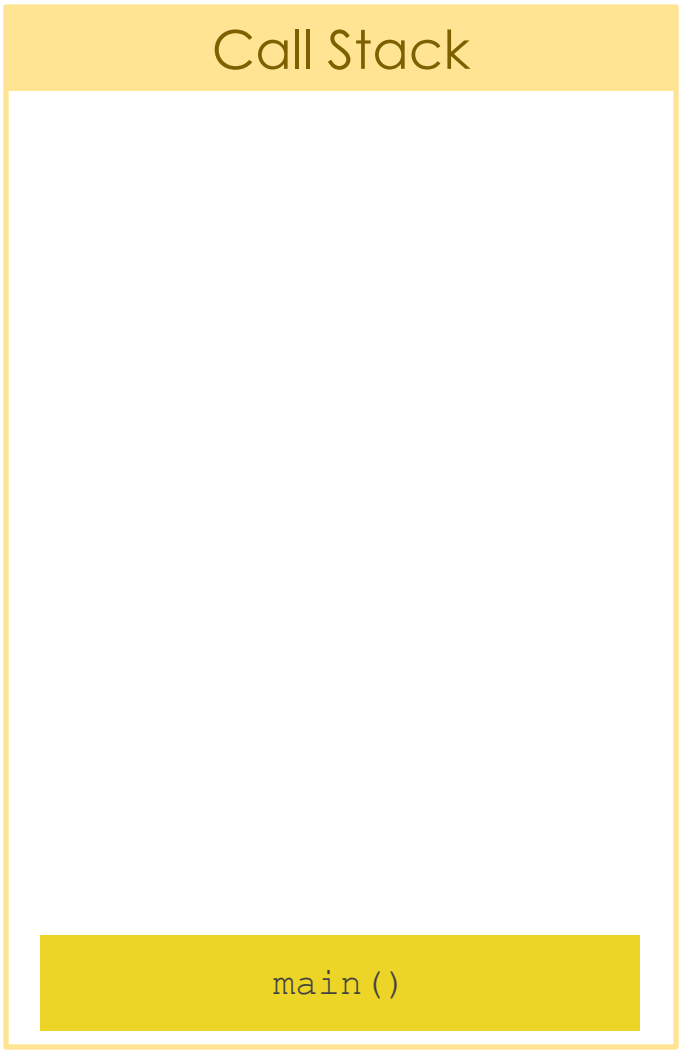
Call Stack

```
function sayHello(){
    console.log("Hello, OS");
}

function sayHi(){
    sayHello();
    console.log("Hi, OS");
}

sayHi();
```



Call Stack

```
function sayHello(){
    console.log("Hello, OS");
}

function sayHi(){
    sayHello();
    console.log("Hi, OS");
}

sayHi();
```



Blocking and Non-blocking Algorithms

JS

Event Loop

```
function sayHello(){
    console.log("Hello, OS");
}
setTimeout(sayHello,5000);
console.log("hi, OS");
```

Call Stack

Callback Queue

Event Loop

Event Loop

```
function sayHello(){
    console.log("Hello, OS");
}
setTimeout(sayHello,5000);
console.log("hi, OS");
```

Call Stack

main()

Callback Queue

Event Loop

Event Loop

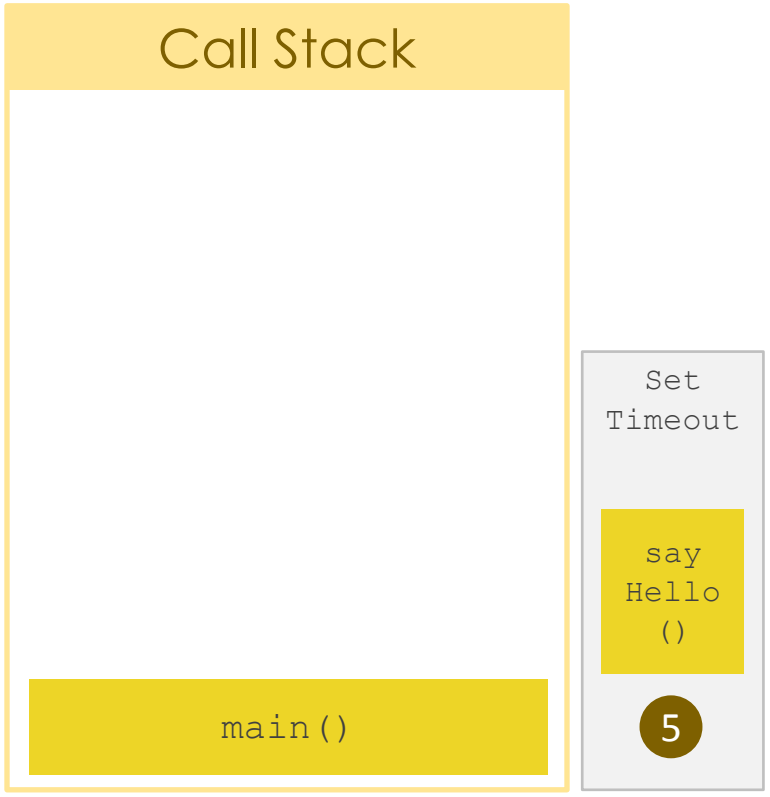
```
function sayHello(){
    console.log("Hello, OS");
}
setTimeout(sayHello,5000);
console.log("hi, OS");
```

Callback Queue

Event Loop

Event Loop

```
function sayHello(){
    console.log("Hello, OS");
}
setTimeout(sayHello,5000);
console.log("hi, OS");
```

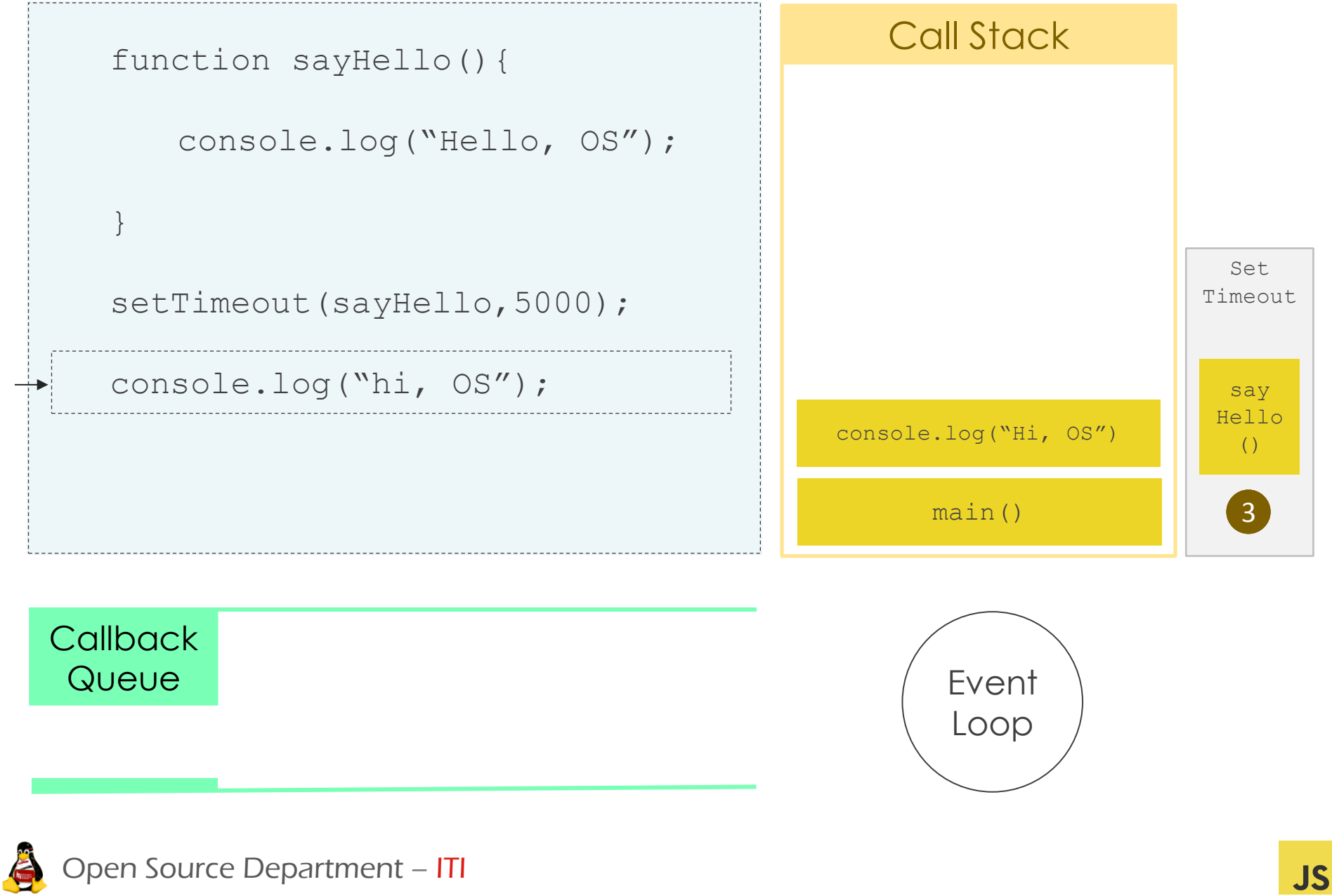


Callback Queue

Event Loop

Event Loop

```
function sayHello(){
    console.log("Hello, OS");
}
setTimeout(sayHello,5000);
console.log("hi, OS");
```



Event Loop

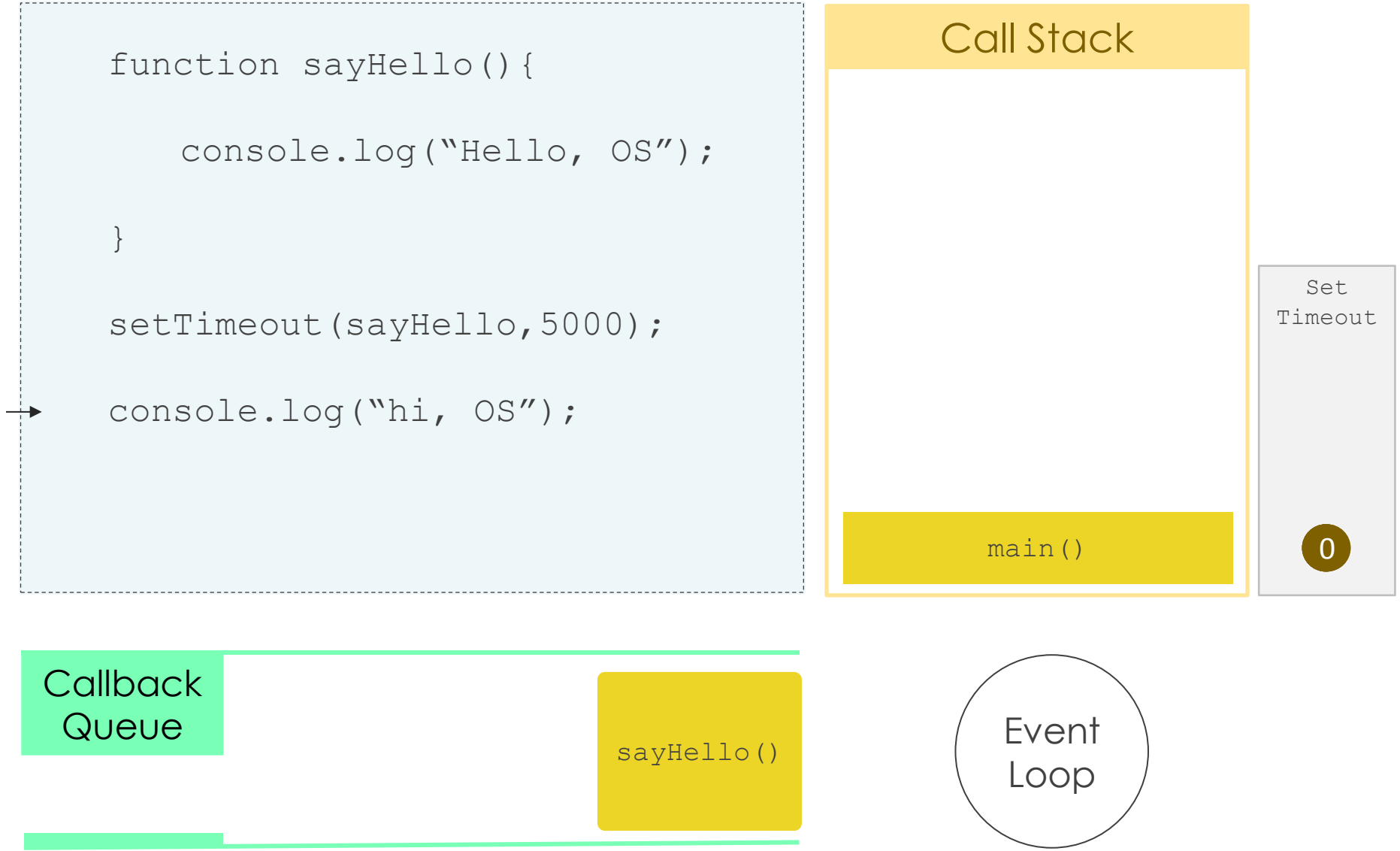
```
function sayHello(){
    console.log("Hello, OS");
}
setTimeout(sayHello,5000);
console.log("hi, OS");
```


Callback Queue

Event Loop

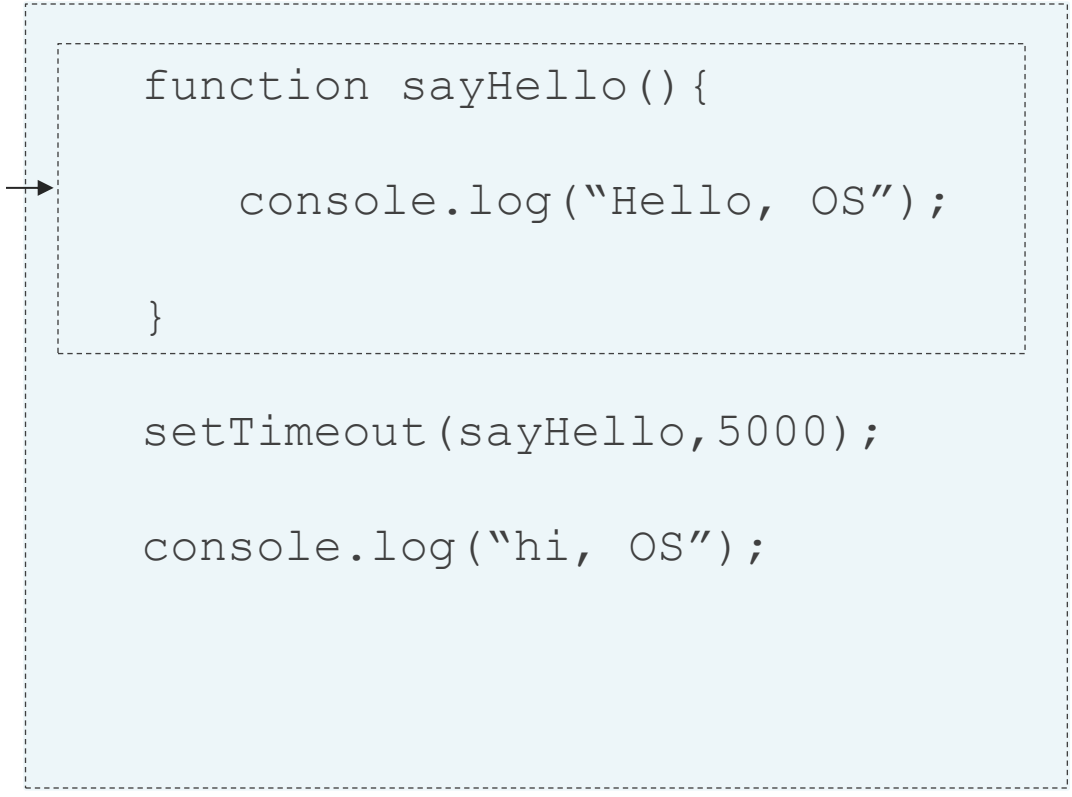
Event Loop

```
function sayHello(){
    console.log("Hello, OS");
}
setTimeout(sayHello,5000);
console.log("hi, OS");
```



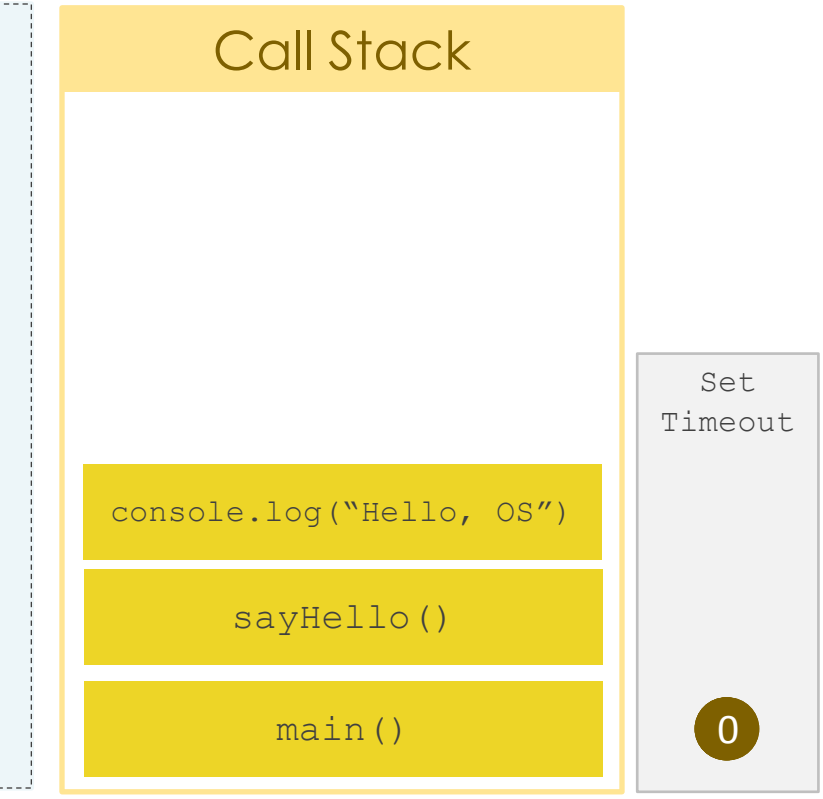
Event Loop

```
function sayHello(){
    console.log("Hello, OS");
}

setTimeout(sayHello,5000);

console.log("hi, OS");
```

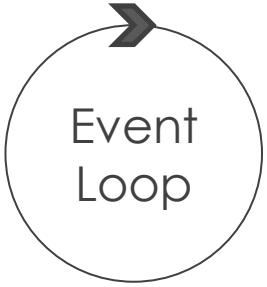


Event Loop

```
function sayHello(){
    console.log("Hello, OS");
}

setTimeout(sayHello,5000);

console.log("hi, OS");
```

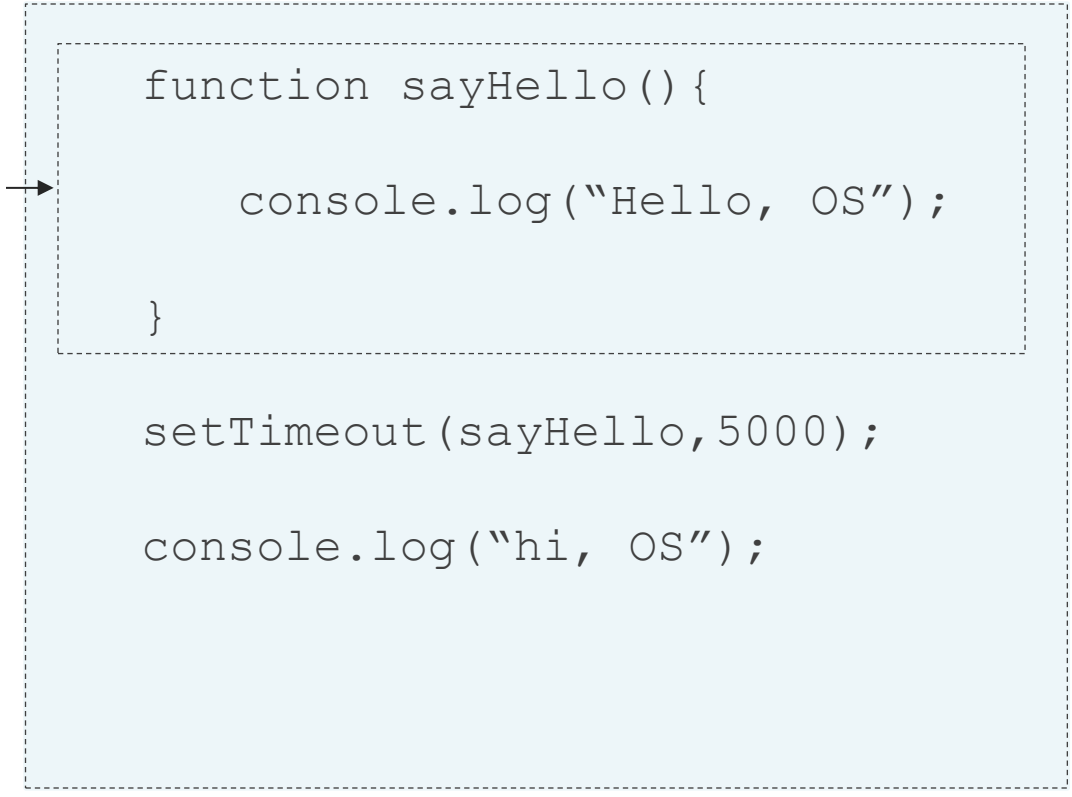


Callback Queue



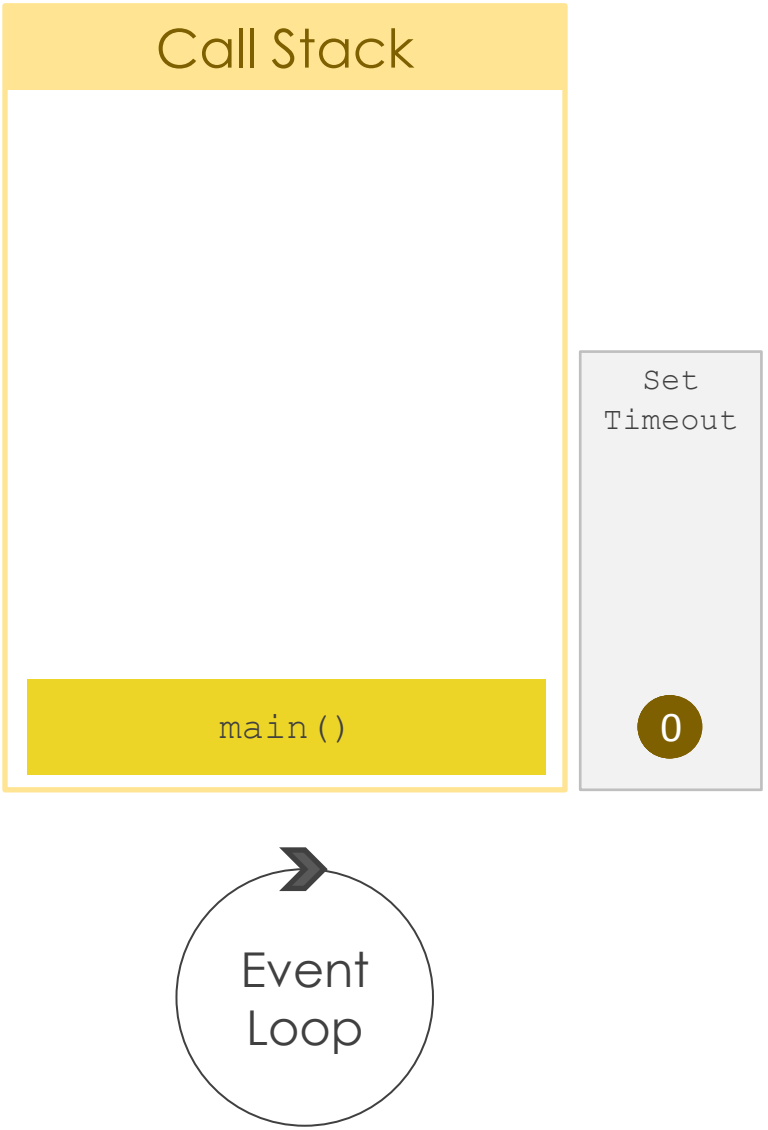
Event Loop

```
function sayHello(){
    console.log("Hello, OS");
}

setTimeout(sayHello,5000);

console.log("hi, OS");
```



Event Loop

```
function sayHello(){
    console.log("Hello, OS");
}
setTimeout(sayHello,5000);
console.log("hi, OS");
```



Event Loop

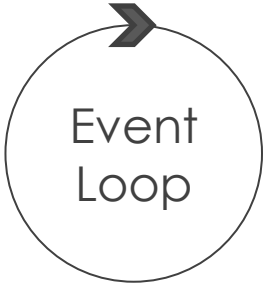
```
function sayHello(){
    console.log("Hello, OS");
}
setTimeout(sayHello,5000);
console.log("hi, OS");
```

Call Stack

Set Timeout

0

Callback Queue



ECMAScript 6

What's New?

JS

Constants

```
const   PI = 3.141593
```

let keyword

```
if(True){
    var x = 5
}
console.log("using var:", x)
```

```
if(True){
    let x = 5
}
```

Output:

```
> Using var: 5
```

let keyword

```
if(True){
    var x = 5
}
console.log("using var:", x)
```

```
if(True){
    let x = 5
}
console.log("using let:", x)
```

Output:

```
> Using var: 5

> ReferenceError: x is not defined
```

Classes

```
class Shape {
    constructor (id, x, y) {
        this.id = id
        this.move(x, y)
    }

    move (x, y) {
        this.x = x
        this.y = y
    }
}
```

```
class Rectangle extends Shape {
    constructor (id, x, y, width, height) {
        super(id, x, y)
        this.width  = width
        this.height = height
    }
}
```

Default Parameters

```
function doSum (x, y = 7, z = 42) {
    return x + y + z;
}
```

Calling It

```
doSum(2)            # output: 7
doSum(2,4)          # output: 9
doSum(2,4,10)       # output: 16
```

Export and Import

Index.js

```
export function sum (x, y) {
        return x + y
}

export var dept = "OS"
```

home.js

```
import * as i from "index";

console.log(i.dept);


//OR


import {sum} from "index";

sum(1,2);
```

What's Next

JavaScript

Web

ANGULARJS by Google

React

node JS

METEOR

Mobile

ionic

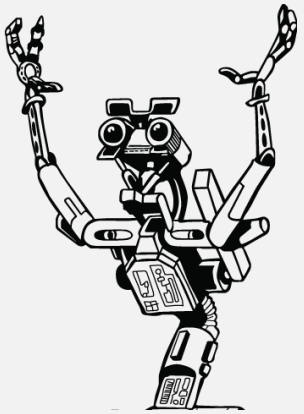
titanium

PhoneGap

Desktop

ELECTRON

Embedded Systems



Johnny-Five

JS

Workshop

Final Project

Team Building

Let's do it

Choose Your Game

Pick one or Invent new one

JS

Brainstorming

Sketch Your Idea

JS

Game Mock-ups

You can Use Balsamiq

Initial Commit

Write your first line of your Game

JS

THANK YOU

ahmedmowd@gmail.com