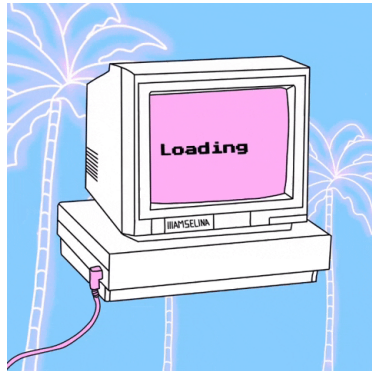
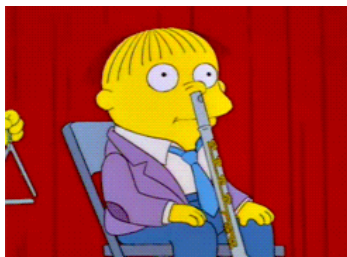


Asynchronous JavaScript



Synchronous? Asynchronous? Wdym?



Consider the code below...

```
/* payload from stars wars API
{
  results: [
    {
      "name": "Luke Skywalker",
      "mass": "77",
      "skin_color": "fair",
      "eye_color": "blue",
      ...
    },
    ...
  ]
}
*/

const data = fetch('https://swapi.dev/api/people')
  .then(res => res.json())
  .then(res => res.results)

console.log(data[0].name)
```

Now:

```
//the line below is evaluated immediately  
  
console.log(data[0].name)
```

Later:

```
//the line below is evaluated at a later time  
  
const data = fetch('https://swapi.dev/api/people')  
    .then(res => res.json())  
    .then(res => res.results)
```

Consider the code below...

```
function now () {  
  return 21  
}  
  
function later () {  
  answer = answer * 2  
  console.log("Meaning of life:", answer)  
}  
  
var answer = now()  
  
setTimeout(later, 1000)
```

Now:

```
function now() {  
  return 21  
}  
  
function later() { .. }  
  
var answer = now()  
  
setTimeout( later, 1000 )
```

Later:

```
answer = answer * 2;  
console.log( "Meaning of life:", answer );
```

Callbacks

```
$.get('https://swapi.dev/api/people', function(data) {  
  console.log(data) /* yay got some data! */  
})
```

Callbacks

```
$.get('https://swapi.dev/api/people', function(data) {  
  console.log(data) /* yay got some data! */  
})
```

this can easily turn into...callback hell

```
$.get('https://swapi.dev/api/people', function(data) {  
  $.get(data.anotherUrl, function(data) {  
    $.get(data.anotherUrl, function(data) {  
      $.get(data.anotherUrl, function(data) {  
        $.get(data.anotherUrl, function(data) {  
          ...  
        })  
      })  
    })  
  })  
})  
})  
})  
})
```


Promises

And we are finally here to talk about Promises in JavaScript!



Promises

And we are finally here to talk about Promises in JavaScript!



Before we jump into looking at code...let's have a little story time!

Promise constructor

```
new Promise((resolve, reject) => {  
  resolve('success!')  
  ...  
  reject('error!')  
}))
```

live coding guided project

examples and some explanations borrowed from You Don't Know JS by Kyle Simpson

If you'd like to dive deeper into JS, I highly recommend reading his books!

His books can be found free here:

<https://github.com/getify/You-Dont-Know-JS/blob/1st-ed/README.md>