Introducción al Frontend

Clase 03

En clases pasadas...

HTML es **semántico**

Real
No fake

HTML ES SOBRE:

LA INFORMACION

NO COMO SE VE

Real
No fake

CSS

CSS es **declarativo**

Real
No fake

the savior rem
(root em)

Real
No fake

Javascript

The Web That Never Was - Dylan Beattie

NDC { Sydney }

17-21 September 2018

Inspiring Software Developers since 2008

ndcsydney.com

```
~h1{This is a heading}

This is a paragraph.

This is another paragraph,
with some ~b{bold text} in
it!
```

NDC { Sydney }
17-21 September 2018

Inspiring Software Developers
since 2008

ndcsydney.com

```
~h1{This is a heading}

This is a paragraph.

This is another paragraph,
with some ~b{bold text} in
it!
```

NDC { Sydney }
17-21 September 2018

Inspiring Software Developers
since 2008

ndcsydney.com

```
~html(lang:en){
 ~metadata{
   ~title{It's HTML, Jim, but not as we know it...}
 }
 ~document{
   ~h1{Hello, World!}
   ~p{
     Using HTML, we can make paragraphs, and even
     ~link(surf:grid!cern.ch/hello.html){link}
     to other documents into our paragraphs!
   }
   ~img(surf:grid!cern.ch/images/test.jpg){Test Image}
 }
}
```

NDC { Sydney }
17-21 September 2018

Inspiring Software Developers
since 2008

ndcsydney.com

The Web That Never Was - Dylan Beattie

```
~html(lang:en){
 ~metadata{
   ~title{Hello World in HyperLISP}
 }
 ~code(type=text/hyperlisp){

   (bind (document "load")
     (lambda (element event)
        (alert "Hello World!"))))

 }
 ~document{
   ~h1{Hyperlisp is the future, you know!}
}
```

NDC { Sydney }
17-21 September 2018

Inspiring Software Developers

The Web That Never Was - Dylan Beattie

22,444 views • Nov 13, 2018

NDC Conferences
106K subscribers

The story of the web is a story about freedom. It's a story about information, about breaking down barriers, about creating new ways for people to communicate, to collaborate, and to share their ideas. It’s also a story that has as much do with marketing, money and meetings as it does with

SHOW MORE

I think the main point is...
So many of these systems we use and rely on everyday are there not because somebody worked really really hard to find the best solution, or to make the best decision, or with the best possible team...

They are there because people found something that was good enough and thought about what we can do with it, **what is next?**

Real
No fake

“Simple, dynamic, and accessible to non-developers.”

Easy to grasp syntactically; dynamic, to reduce verbosity and speed up development; and powerful.

Real
No fake

Brendan Eich

10 días

Real
No fake

Real
No fake

const **all the way,**
if mutate then use let

variables and values

number

booleans

strings

`Down on the sea`

"Lie on the ocean"
'Float on the ocean'

String interpolation

null **and** undefined

Non-zero value

0

null

undefined

Just JavaScript

Hey there!

I'm Dan Abramov.

Functions

in JS

Default parameters

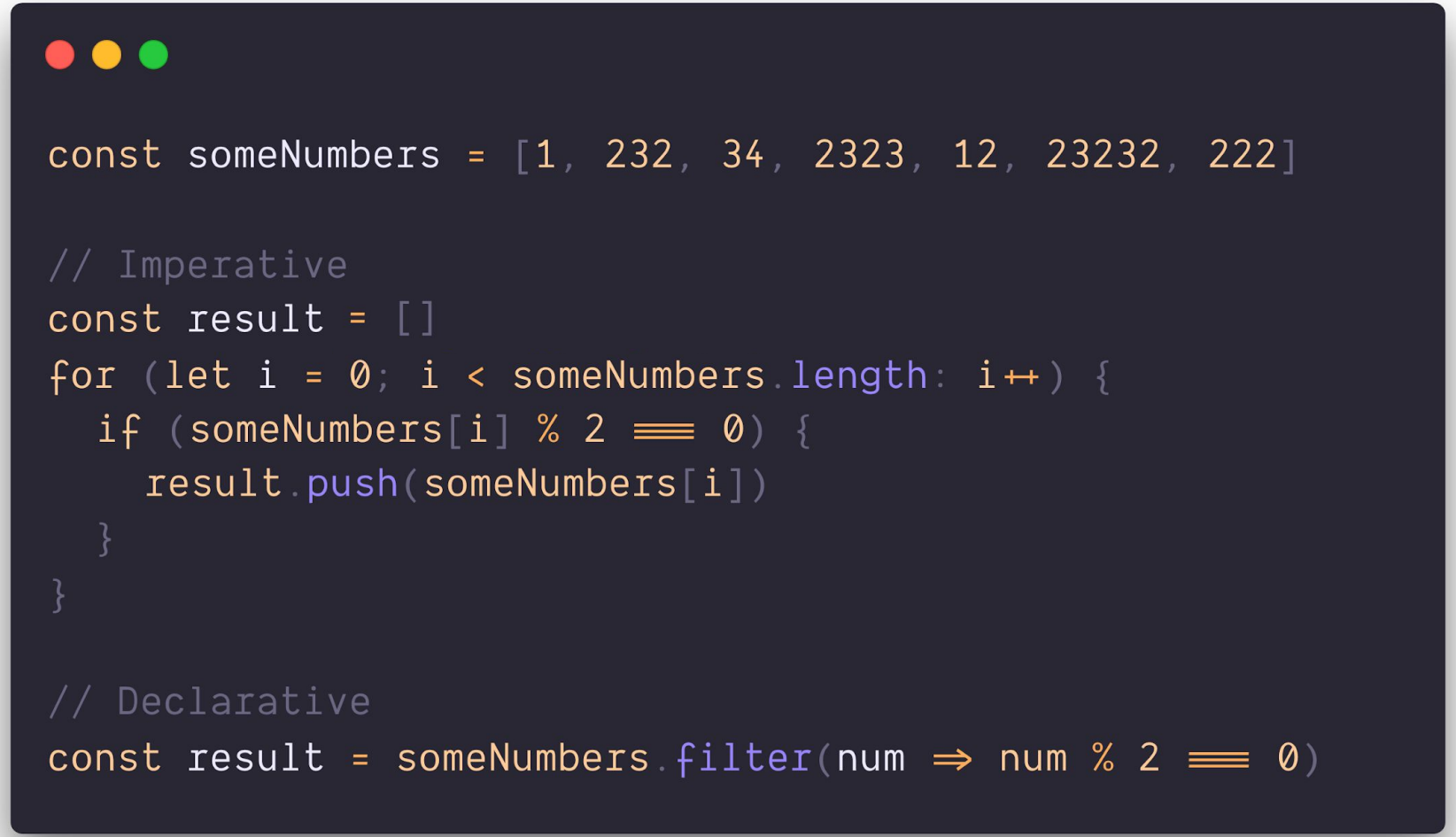
This feature is pretty neat. It allows you to set a default value if none is provided.

```
1  function sayMsg(msg='This is a default message.') {
2    console.log(msg);
3  }
4
5  sayMsg(); // This is a default message.
6  sayMsg('This is a different message!'); // This is a different message!
```

gistfile1.js hosted with ❤️ by GitHub view raw

Arrays in JS

DECLARATIVE PROGRAMMING


const someNumbers = [1, 232, 34, 2323, 12, 23232, 222]
// Imperative
const result = []
for (let i = 0; i < someNumbers.length; i++) {
if (someNumbers[i] % 2 === 0) {
result.push(someNumbers[i])
}
}
// Declarative
const result = someNumbers.filter(num => num % 2 === 0)

Useful Array Methods →

https://cutt.ly/oxeH2Pd

Destructuring

Chains

https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Array

Objects in JS

```
const object = {
    name: "Saitma",
    race: "Human",
    superpower: "Super strength"
    color: "yellow"
}
```

Destructuring

```
const add = ({ x, y }) => {
  return x + y
}
```

```
const data = {
  name: 'Harry',
  location: {
    city: 'Dehradun',
    state: 'Uttrakhand'
  },
  car: {
    name: 'Ferrari'
  }
}


const bikeName =  data.bike?.name ?? 'No bike';
console.log(bikeName);    // No bike
```

Modules

in JS

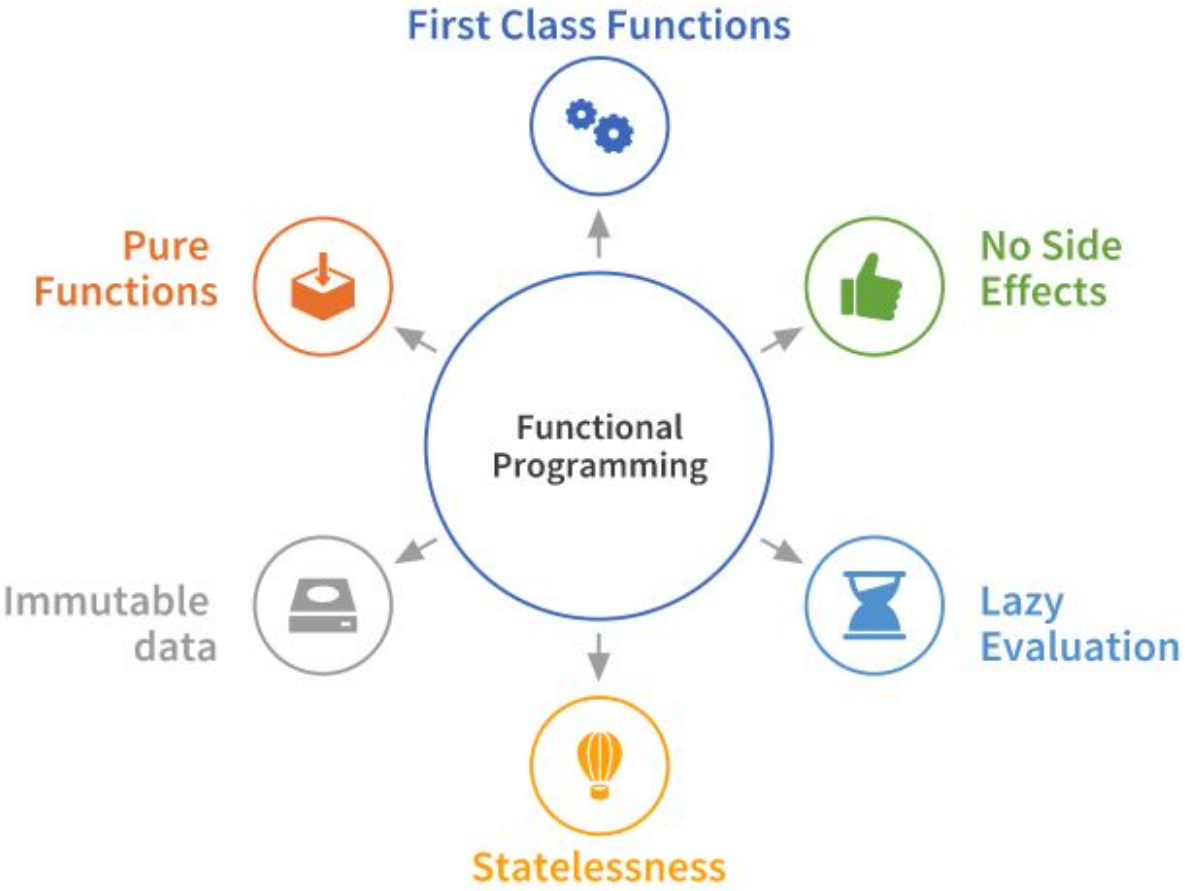
import export

Default export

Named export

DOM

Closures



Simplificación

What is functional programming?

> Functional programming is a programming paradigm—a style of building the structure and elements of computer programs—that treats computation as the evaluation of mathematical functions and avoids changing-state and mutable data—Wikipedia

Simplification

So how do we know if a function is `pure` or not? Here is a very strict definition of purity:

- It returns the same result if given the same arguments (it is also referred as `deterministic`)
- It does not cause any observable side effects

Simpli
ficación

It returns the same result if given the same arguments

Imagine we want to implement a function that calculates the area of a circle. An impure function would receive `radius` as the parameter, and then calculate `radius * radius * PI`:

```
const PI = 3.14;

function calculateArea(radius) {
  return radius * radius * PI;
}

calculateArea(10); // returns 314.0
```

Simplification

It does not cause any observable side effects

Examples of observable side effects include modifying a global object or a parameter passed by reference.

Now we want to implement a function to receive an integer value and return the value increased by 1.

```
let counter = 1;

function increaseCounter(value) {
  counter = value + 1;
}

increaseCounter(counter);
console.log(counter); // 2
```

Simplificación

Immutability

> Unchanging over time or unable to be changed.

after it's created. **If you want to change an immutable object, you can't. Instead, you create a new object with the new value.**

Simpli
ficación

pure functions + immutable data = referential transparency

Simplification

Functions as first-class entities

Simpli
ficación

Higher-order functions

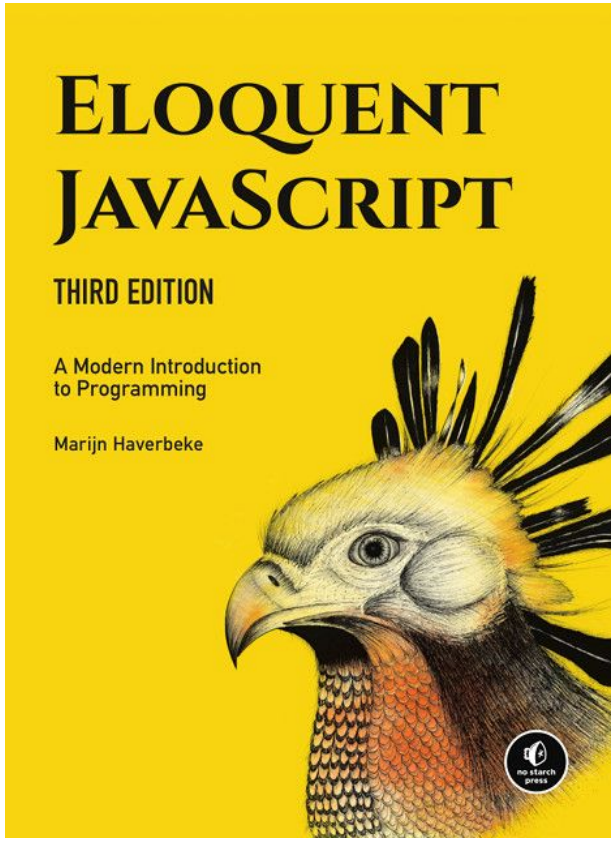
When we talk about higher-order functions, we mean a function that either:

- takes one or more functions as arguments, or
- returns a function as its result

The `doubleOperator` function we implemented above is a higher-order function because it takes an operator function as an argument and uses it.

You've probably already heard about `filter`, `map`, and `reduce`. Let's take a look at these.

Simpli
ficación



eloquentjs-es.thedojo.mx

¿Qué es React?

Es una librería de JavaScript de código abierto, creada y mantenida por Facebook que nos permite crear interfaces de usuario y aplicaciones en la web (y móvil) usando programación reactiva y funcional.