

JavaScript
Hero to Herooo!

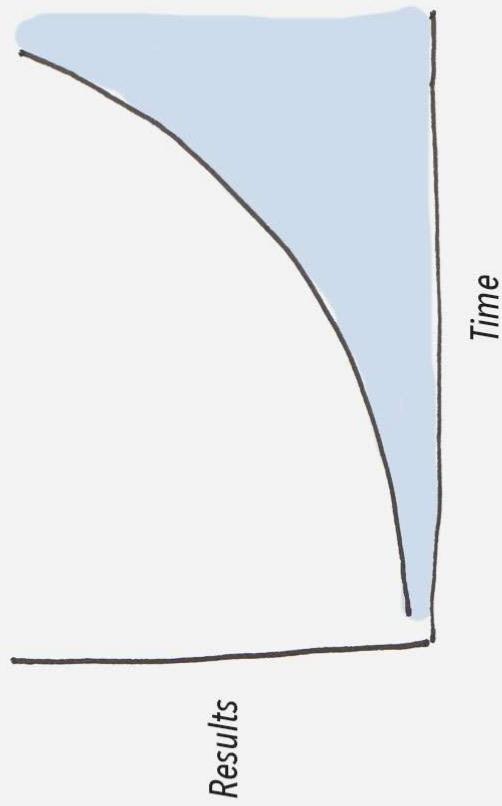
Sobre o curso

- Quero que aprendam a pensar como programadores.
- Quero que trabalhem em Exercícios/Projetos
- Aumento de dificuldade exponencial

Expectativa

EXPONENTIAL GROWTH

Improvements come slowly in the beginning, but your gains increase rapidly over time.



JamesClear.com

Regras

- Não há perguntas estúpidas.
- Crescimento > Conseguirmos fazer tudo

Parte
a

Objetivos

- Reveer conceitos básicos
- Reveer a **DOM** & e como modificar
- Reveer **data types & operadores**
- Declaração de variáveis e como usar
- Reunir valores em **objetos e arrays**
- Criar um projeto com estes conceitos!

Vamnossö
a

O que? Porque? Como?
JavaScrip!

O que é JS?

Copiar

```
let js = {  
    name: "JavaScript",  
    abbreviation: "JS",  
    isAwesome: true,  
    officialSpec: "ECMAScript",  
    birthYear: 1995,  
    creator: "Brendan Eich"  
};  
  
if (js.isAwesome) {  
    shout("Amo JS, mas por vezes ele não me ama!");  
}
```

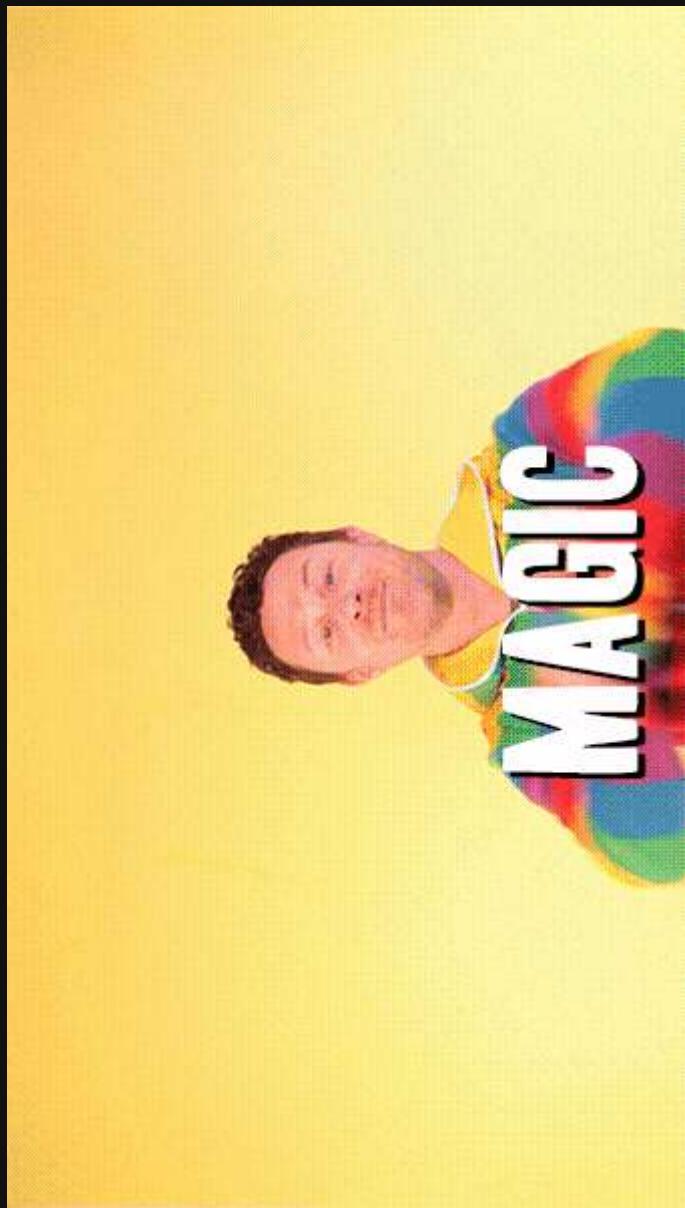
Por que precisamos de JS?

HTML & CSS & JS
||
BFF's!



YOU ARE MY BEST FRIEND

Como escrevemos JS?



Como Onde escrevemos JS?

- **Na consola do browser**
- Num ficheiro local usando um editor, ex: VS Code
- Num playground online ex: [CodePen](#),
[CodeSandbox](#)

DOM

Document Object Model

Arquitetura de um documento

```
1 <!DOCTYPE html>
2 <html lang="en-US">
3 <head>...</head>
4 <body>
5   <header>...</header>
6   <div id="board">...</div>
7 </body>
8 </html>
```

Copiar

Árvore de componentes

Ler a DOM com JS

O cerne da questão document!

Copiar

document

document

todo o documento HTML

Copiar

document.title

Copiar

document.title

Copiar

O título da página (documento)

document.body

Copiar

O elemento do corpo

document.body

Copiar

document.body.children

Copiar

```
document.body.children
```

Copiar

Todos os elementos dentro do body (corpo)

document.getElementById("board")

Copiar

`document.getElementById ("board")`

`document.querySelector ("#board")`

O (primeiro) elemento com o id="board"

Copiar

Copiar

`document.getElementById("board")`

`document.querySelector("#board")`

O (primeiro) elemento com o `id="board"`

Copiar

Copiar

document.getElementsByName("h1")

Copiar

document.getElementsByName("h1")

document.querySelectorAll("h1")

todos os elementos h1

document.getElementsByName("h1")

document.querySelectorAll("h1")

todos os elementos h1

Copiar

```
document.getElementsByName("player")
```

document.getElementsByName ("player")

Copiar

document.querySelectorAll (" .player")

Copiar

Todos os elementos com a classe class="player"

document.getElementsByName ("player")

Copiar

document.querySelectorAll (" .player")

Copiar

Todos os elementos com a classe class="player"

```
document.getElementsByName("player").length
```

Copiar

```
document.querySelectorAll(".player").length
```

Copiar

```
document.getElementsByName ("player").length
```

Copiar

```
document.querySelectorAll ("player").length
```

Copiar

O número de elementos com a classe
class="player"

Copiar

```
document.getElementById("player-1").textContent
```

Copiar

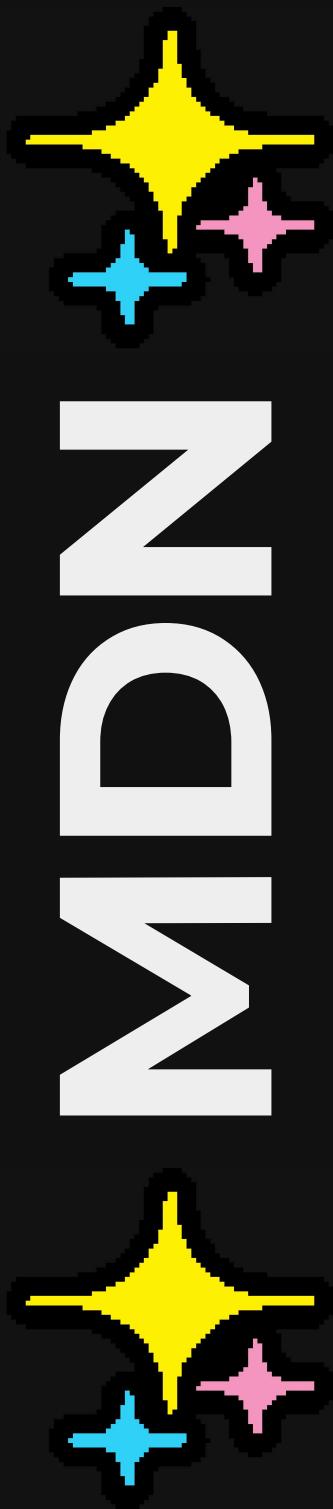
```
document.getElementById("player-1").textContent
```

O texto dentro do elemento com id="player-1"

*Mas Nelson como esperas que
eu me lembre de isto
tudo!?!?!*



As três letras mais importantes para qualquer programador web:



developer.mozilla.org

Editing the DOM with JS

```
document.title = "Bananas"
```

Copiar

```
document.title = "Bananas"
```

Copiar

muda o título da página
atenção as aspas duplas !

```
document.title = "Bananas"
```

Copiar

muda o título da página
atenção as aspas duplas !

Copiar

```
document.getElementById("player-1").textContent = "Batas"
```

Substituir o texto do elemento #player-1

Copiar

```
document.getElementById ("player-1") .textContent = "Batas"
```

Copiar

```
document.getElementById("player-1").append(" & AlunoS")
```

Copiar

```
document.getElementById("player-1").append(" & Alunos")
```

adiciona ao final do texto atual do elemento

Valores & Tipos de dados

Mas o que São?

"Tic Tac Toe"

9

"#board"

Valores

Valores

blocos de informações com os quais queremos
trabalhar

Valores

blocos de informações com os quais queremos
trabalhar
estas valores (dados) podem ser de diferentes
tipos

Dados de texto

ex. "Tic Tac Toe" ou "#board"

Dados de texto

ex. "Tic Tac Toe" ou "#board"

também conhecidos por...

strings



MUFFINCHAN

Tudo isto são strings:

- "Olá!"
- 'eu gosto de aspas simples'
- `Também ❤️ backticks`
- '867-5309'
- "42"

"42" vs. 42

são a mesma coisa?

`typeof`

retorna o tipo de um valor

Qual é o resultado?

```
typeof "42"
```

```
typeof 42
```

Copiar

Copiar

Números (numbers)



Tudo isto são números:

- 7
- 123078
- -6.75
- 1.21e10
- Infinity

JS tem dois tipos de dados:

- Tipos primitivos (ex: strings, números)
- Objetos (ex: document & etc)

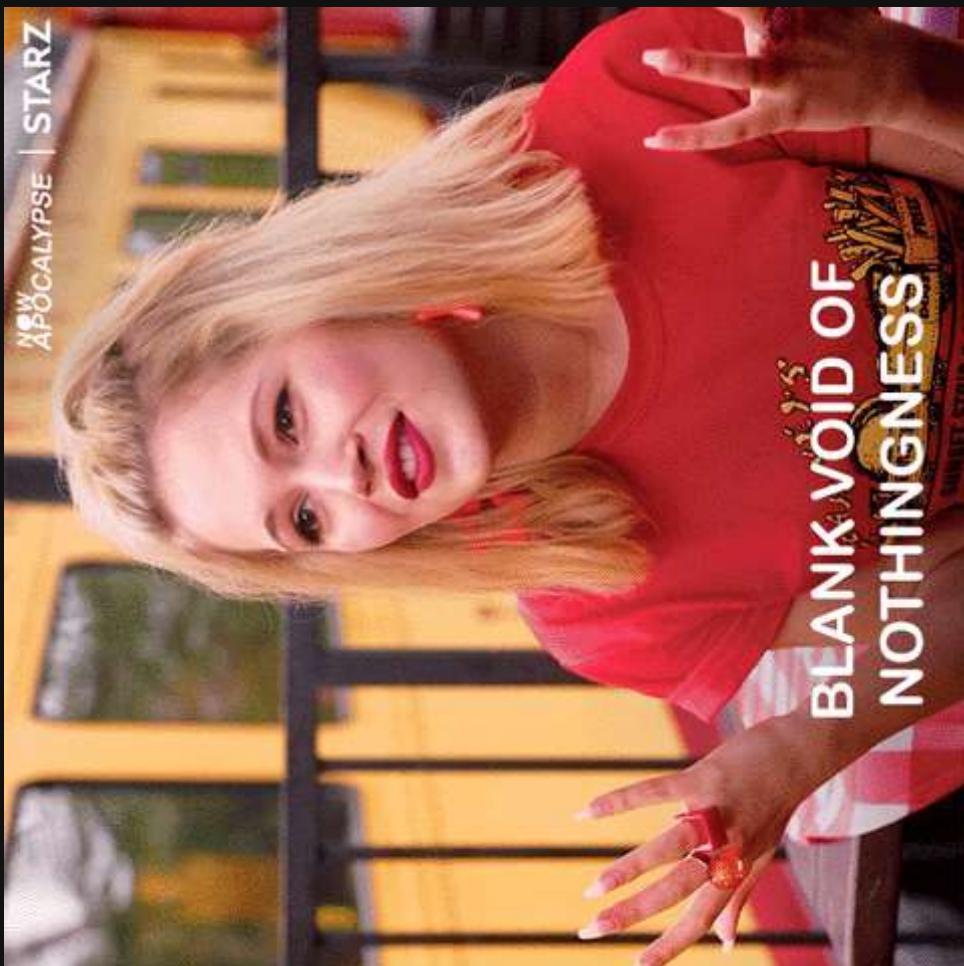
Tipos de dados primitivos

- string
- number
- boolean
- undefined
- null
- E outros ainda não precisamos

booleans



undefined & null



undefined
null
nada acidental
nada deliberado

Exercício

Qual é tipo de dado cada um destes valores?

- `false`
- `"true"`
- `document.title`
- `"some string".length`
- `null`

Trabalhair.com Strings

Do que são feitas as strings?



Do que são feitas as strings?



characters

Quantos caracteres existem nesta string?

"voçesnaosabiamassaeugostomuitodefruta".length

"bananas".length

Copiar

Copiar

Os caracteres estão numa ordem específica
Cada um recebe um número, que começa em 0



Os caracteres estão numa ordem específica
Cada um recebe um número, que começa em 0



também conhecido como "index" (índice)

Qual é o primeiro caractere na string?

"JAVASCRIPT" [0]

Copiar

Qual é o terceiro caractere na string?

"JAVASCRIPT" [2]

Copiar

Qual é o index de um carácter específico?

Qual é O index de um carácter específico?

"JAVASCRIPT".indexOf ("A")

Copiar

Qual é O index de um caracter específico?

"JAVASCRIPT" . indexOf ("A")

Copiar

"JAVASCRIPT" . indexOf ("T")

Copiar

Qual é O index de um carácter específico?

"JAVASCRIPT" . indexOf ("A")

"JAVASCRIPT" . indexOf ("T")

"JAVASCRIPT" . indexOf ("Q")

Copiar

Copiar

Copiar

Esta string contém outra string?

Esta string contém outra string?

"JAVASCRIPT".includes ("RIP")

Copiar

Esta string contém outra string?

"JAVASCRIPT".includes("RIP")

Copiar

"JAVASCRIPT".includes("LOL")

Copiar

Esta string começa com Outra string?

Esta string começa com outra string?

"JAVASCRIPT".startsWith("JA")

Copiar

Esta string começa com outra string?

"JAVASCRIPT".startsWith("JA")

"JAVASCRIPT".startsWith("HA")

Copiar

Copiar

Em que Index a substring começa?

Em que Index a substring começa?

```
"JAVASCRIPT".indexOf("VA")
```

Copiar

Em que Index a substring começa?

"JAVASCRIPT".indexOf("VA")

"JAVASCRIPT".indexOf("LOL")

Copiar

Copiar

Juntando strings

```
"JAVASCRIPT" + "!"
```

Copiar

"JAVASCRIPT".toLowerCase()

Copiar

Há muito mais que podemos fazer com strings ...



JavaScript > Standard built-in objects > String

Lembrar ✨ MDN ✨ é nosso amigo!

Exercícios

Operadores (Operators)



Parece que ja vimos alguns Operadores:

- "JAVASCRIPT" + "!"
- typeof "Cenas"

+ é um operador com vários tipos de uso:

- + é um operador com vários tipos de uso:
 - O que faz quando o utilizamos em strings?

- + é um operador com vários tipos de uso:
 - O que faz quando o utilizamos em strings?
 - Em que podemos usar para além de strings?

- + é um operador com vários tipos de uso:
 - O que faz quando o utilizamos em strings?
 - Em que podemos usar para além de strings?
 - Com quantos valores opera por vez?

Operadores aritméticos

- + adição
- - subtração
- * multiplicação
- / divisão

Ordem de operações

4 + 1 * 2 * 4 + 2

como na matemática, utilizamos () para agrupar

((4 + 1) * 2 * 4) + 2

Operadores de comparação

- > Maior que
- < Menos que
- \geq maior ou igual a
- \leq menor ou igual a

5 > 4

Que tipo de valor retorna?

Operadores de igualdade

estrito	relaxado	significado
<code>==</code>	<code>=</code>	igual
<code>!=</code>	<code>!</code>	diferente

Estes diferem? Como?

estrito	vs.	significado
1 === 1	vs.	1 == 1
"1" === "1"	vs.	"1" == "1"
1 === "1"	vs.	1 == "1"

Estes diferem? Como?

estrít o **vs.** **significado**

`1 === 1` *vs.* `1 == 1`

`"1" === "1"` *vs.* `"1" == "1"`

`1 === "1"` *vs.* `1 == "1"`

Deves sempre que possível usar a versão **estrita**

JS tem muitos operadores!

JS tem muitos operadores!

Alguém palpita de mais algum?

JS tem muitos operadores!

Algun palpite de mais algum?

Onde podemos encontrar o resto?

JavaScript > Expressions & Operators

MDN 

Expressões



São valores?

- `5 / 3 * 10`
- "Edit" + "Frontenders"
- `5 > 4 != 3 > 4`

uma expressão *avalia* (ou *resolve*) para um valor

JavaScript > JS Guide > Expressions & Operators



Variaveis



I will remember you

MakeAGIF.com

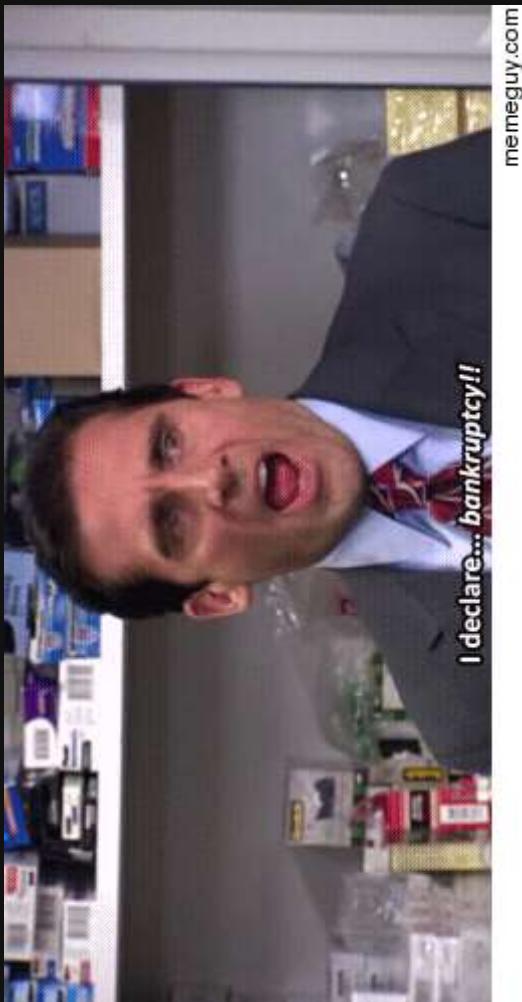
As variáveis permitem "recordarmos" valores

```
let remember = "2 de Agosto";
```

Copiar

O que podemos perceber deste código?

Declarando uma variável



let bankruptcy;

Copiar

Qual acham que é o valor bankruptcy?

Atribuindo uma variável

```
let minhaVariavel;  
minhaVariavel = "weeeeeeeeeee";
```

Copiar

Declarando & atribuindo ao mesmo tempo

```
let minhaVariavelLinda = "Olha mãe já faço isto numa linha
```

Copiar
SO

Declarando & atribuindo forevaaa

`const` declara e atribui uma "constante"

ou seja uma variável que não pode ser alterada

```
const minhaConstanteBonita = "Never gonna give you up";
```

Copiar

Utilizando uma variável

cenas - 10

outrasCenas.toUpperCase()

Como nomear variáveis

- variavelValida
- valida_mais_pouco_comum
- Human_TecnicamenteEstabem2
- ~~Oneoepi~~
- STIIIIIMMM



Mas o que realmente são variáveis??

Variáveis "contêm" valores?

variáveis "contêm" mas "apontam para" valores



O que acontece quando este código é executado?

```
let bajajas = ((4 + 1) * 2 * 4) + 2;
```

Copiar

O que acontece quando este código é executado?

```
let umaCena = "Algo importante";
let outraCena = umaCena;
umaCena = "Desculpem qualquer coisa...";
```

Copiar

Declaracões vs Expressões



I can't tell if that's
a statement or question

Uma expressão "questiona" JS por um valor

minhaLindaVariavel

6 + 4

document.getElementById ("bajajas")

Copiar

Uma declaração "diz" ao JS para fazer alguma coisa
(ex: declarar/atribuir uma variável)

```
let vinte = 6 + 14;  
minhaLindaVariavelDeclarada = "MUDAAAAA";  
let board = document.getElementById("bajajas");
```

Copiar

Veremos mais tipos de declarações no workshop

```
function adiciona (x, y) {
    return x + y;
}

let oMaior;
if (5 > 4) {
    oMaior = 5;
} else {
    oMaior = 4;
}

for (let caracter of "esta string") {
    console.log(caracter);
}
```

Copiar

Arrays

(aka: Listas)



Arrays são uma estrutura de dados em JavaScript, usada para armazenar e manipular "coleções" de valores.

Copiar Cena"

```
let muitasCenas = [ "cena" , "que cena" , "fazer cena" , "jonh
```



Assim como as strings, os arrays têm um `length` (comprimento)

```
muitasCenas.length
```

Copiar

E cada valor tem um índice (`index`)

```
muitasCenas[1]  
muitasCenas.indexOf("jonh cena")
```

Copiar

Como strings, podemos verificar se um array contém um determinado valor

```
multasCenas.includes("que cena")
multasCenas.includes("Triple H")
```

Copiar

Ao contrário das strings, podemos modificar arrays

Copiar

```
muitasCenas[1] = "faz-me uma cena";  
let jonhCena = muitasCenas.pop();  
muitasCenas.push(jonhCena.toUpperCase());
```

Que faz cada uma das linhas?

Arrays podem ser vazios, ou conter um único elemento

```
let foreverAloneArray = [];
let lonelyArray = ["lonely"];
```

Copiar

Arrays podem conter qualquer tipo de item/valores, ou misturar e combinar!

Copiar

```
let mistoDeCenas = ["cenas", 6, "Outra cena", false, document];
```

Com Arrays podemos fazer muitos cenas úteis!

O que é que faz cada uma delas

```
[ "c", "a", "d", "b" ].sort()
```

Copiar

```
["Bazinga", "incrível", "Incrível!"].join(" & ")
```

Copiar

```
[1, 2, 3].concat([4, 5, 6])
```

Copiar

JavaScript > Standard built-in objects > Array

Nosso amigo MDN !

Exercícios

Mutação de dados



Isto faz a mesma coisa?

```
let abcArray = ["a", "b", "c"];  
abcArray[1] = "d";  
abcArray;
```

Copiar

```
let abcString = "abc";  
abcString[1] = "d";  
abcString;
```

Copiar

mutável vs. **imutável**

Dados mutáveis podem ser editados (ex. Arrays)
Os dados "imutáveis" sempre permanecem iguais
(ex. strings & outros primitivos)

Isto faz a mesma coisa?

```
let numeros1 = [1, 2, 3];
let resultado1 = numeros1.push(4);
numeros1;
```

Copiar

```
let numeros2 = [1, 2, 3];
let resultado2 = numeros2.concat([4]);
numeros2;
```

Copiar

Algumas ações "transformam" um array (ex:

```
cenasAntigas.push(cenas))
```

...ou seja alterar o array "*in-place*"

Outras ações *não transformam* o array original,
mas criam uma nova cópia (ex:
`cenasAntigas.concat(outrasCenas))`

⚠ Usem o MDN para validar que ação é efetuada
por cada método! ⚡

As próprias variáveis também podem ser (im)mutáveis

O que acontece em cada caso?

```
let letCenas = "cena original";  
letVariable = "cena nova";
```

Copiar

```
const constCenas = "cena original";  
constVariable = "cena nova";
```

Copiar

variável imutável com valor mutável

O que acontece quando utilizamos `const` em um valor mutável como um array?

```
const numeros = [2, 6];
const sum = numeros[0] + numeros[1];

numeros[0] = 7

const novaSoma = numeros[0] + numeros[1];
```

Copiar

Se tiveres a escolha...

**Usar dados e variáveis imutáveis & é a melhor
escolha**

Mas porque?

