

ARROW FUNCTIONS + FUNÇÕES DE ARRAYS



DEVinHouse

Parcerias para desenvolver a sua carreira

SENAI

<LAB365>

- Revisão e Classes
- Atividade
- Arrow Functions
- Funções de Arrays
 - `forEach()`, `map()`, `filter()`, `find()`, `reduce()`, `every()`, `some()`

- **Classe:** "gabarito" para construirmos objetos (instâncias)
- **Atributo:** variável pertencente à instância ou à classe (se estático)
- **Método:** função pertencente à instância ou à classe (se estático)
- **Herança:** atributos e métodos que uma subclasse herda da mãe
- **Polimorfismo:** atributos e métodos herdados podem ser alterados
- **Encapsulamento:** atributos e métodos de execução interna

REVISÃO | Classes

```
class Item {  
  static #nextId = 0  
  #id  
  descricao  
  
  constructor(desc = '') {  
    this.#id = Item.#nextId++  
    this.descricao = desc  
  }  
  
  get id() {  
    return this.#id  
  }  
  
  montaItem() {  
    const li = document.createElement('li')  
    li.innerHTML = this.descricao  
    return li  
  }  
}
```

Exemplo da estrutura básica de uma classe

```
class Tarefa extends Item {  
  #status  
  
  constructor(desc, stat = false) {  
    super(desc)  
    this.#status = stat  
  }  
  
  set status(novoStatus) {  
    this.#status = !!novoStatus  
  }  
  
  montaItem() {  
    const li = document.createElement('li')  
    const checkbox = document.createElement('input')  
    checkbox.type = 'checkbox'  
    checkbox.checked = this.#status  
    const texto = document.createTextNode(this.descricao)  
    li.appendChild(checkbox)  
    li.appendChild(texto)  
    return li  
  }  
}
```

Classe que herda características da superior e implementa polimorfismo

ATIVIDADE



Acesse **kahoot.it**
e digite o **PIN** que aparecerá
aqui na transmissão
(**acessar pelo smartphone**)



ARROW FUNCTIONS

() => { }

ARROW FUNCTIONS

- Arrow Functions foram introduzidas a partir do ES6 (2015)
- Nos permitem escrever de forma mais curta a sintaxe de um função
 - Arrow function: `() => {}`
 - Função comum: **function** `() {}`

ARROW FUNCTIONS

```
function funcaoComumA() {  
  console.log('Sou uma função A!')  
}  
  
const funcaoComumB = function () {  
  console.log('Sou uma função B!')  
}  
  
(function () {  
  console.log('Sou uma função anônima!')  
})();  
  
function quadrado(n) {  
  return n * n  
}  
  
function multiplica(a, b) {  
  return a * b  
}
```

Exemplo de definições de funções comuns

```
const funcaoComumA = () => {  
  console.log('Sou uma função A!')  
}  
  
const funcaoComumB =  
  () => console.log('Sou uma função B!')  
  
(() => console.log('Sou uma função anônima!'))()  
  
const quadrado = n => {  
  return n * n  
}  
  
// quando retorna o resultado de uma instrução  
// não precisa das chaves e do 'return'  
const multiplica = (a, b) => a * b
```

Exemplo de definições de arrow functions

ARROW FUNCTIONS

```
const vetor = [10, 20, 30]

function callback(item) {console.log(item)}
vetor.forEach(callback)

vetor.forEach(function (item) {
  console.log('Item: ' + item)
})

vetor.forEach(function (item) {console.log(item)})

vetor.forEach(function (item, indice) {
  console.log(`Item: ${item} - Idx: ${indice}`)
})

vetor.forEach(function (item, indice, array) {
  console.log(`${item} - ${indice}`, array)
})
```

Passando funções comuns por parâmetro

```
const vetor = [10, 20, 30]

const callback = item => console.log(item)
vetor.forEach(callback)

vetor.forEach((item) => {
  console.log('Item: ' + item)
})

vetor.forEach(item => console.log(item))

vetor.forEach((item, indice) =>
  console.log(`Item: ${item} - Idx: ${indice}`)
)

vetor.forEach((item, indice, array) => {
  console.log(`${item} - ${indice}`, array)
})
```

Passando arrow functions por parâmetro

ARROW FUNCTIONS

```
class Item {
  descricao
  constructor(desc) {
    this.descricao = desc
  }
  montaItem() {
    const li = document.createElement('li')
    const checkbox = document.createElement('input')
    checkbox.type = 'checkbox'

    checkbox.onclick = function () {
      // este this é o checkbox no evento de click
      console.log(this)
    }

    const texto =
      document.createTextNode(this.descricao)
    li.appendChild(checkbox)
    li.appendChild(texto)
    return li
  }
}
```

Em funções comum *'this'* é o contexto onde a função é chamada

```
class Item {
  descricao
  constructor(desc) {
    this.descricao = desc
  }
  montaItem() {
    const li = document.createElement('li')
    const checkbox = document.createElement('input')
    checkbox.type = 'checkbox'

    checkbox.onclick = () => {
      // este this é a instância da classe Item
      console.log(this)
    }

    const texto =
      document.createTextNode(this.descricao)
    li.appendChild(checkbox)
    li.appendChild(texto)
    return li
  }
}
```

Em arrow functions *'this'* é o contexto onde a função é criada

ARROW FUNCTIONS

```
// função criada fora de contexto
function mostraThis() {
  console.log(this.nome)
}

const fruta = {
  nome: 'Melão',
  imprimir: mostraThis
}

// chamada no contexto do objeto
fruta.imprimir()

// CONSOLE: 'Melão'
// Funções comuns redefinem this ao contexto
```

Em funções comuns *'this'* é o contexto onde a função é chamada

```
// função criada fora de contexto
const mostraThis = () => {
  console.log(this.nome)
}

const fruta = {
  nome: 'Melão',
  imprimir: mostraThis
}

// chamada no contexto do objeto
fruta.imprimir()

// CONSOLE: undefined (pois this é window)
// Arrow functions não redefinem this
```

Em arrow functions *'this'* é o contexto onde a função é criada

FUNÇÕES DE ARRAYS

```
[1, 2, 3].forEach(num => console.log(num))
```

FUNÇÕES DE ARRAYS

- Métodos disponíveis em instâncias de Array
- Acessadas como em objetos, utilizando o ponto (.)
- Facilita iterações e manipulação de arrays
- Desacoplamento do código!
 - **`nomeArray.nomeFuncao(funcaoParametro)`**
 - **`lista.forEach(item => console.log(item))`**

FUNÇÕES DE ARRAYS | Instâncias

- [.concat\(\)](#)
- [.copyWithin\(\)](#)
- [.entries\(\)](#)
- [.every\(\)](#)
- [.fill\(\)](#)
- [.filter\(\)](#)
- [.find\(\)](#)
- [.findIndex\(\)](#)
- [.flat\(\)](#)
- [.flatMap\(\)](#)
- [.forEach\(\)](#)
- [.includes\(\)](#)
- [.indexOf\(\)](#)
- [.join\(\)](#)
- [.keys\(\)](#)
- [.lastIndexOf\(\)](#)
- [.map\(\)](#)
- [.pop\(\)](#)
- [.push\(\)](#)
- [.reduce\(\)](#)
- [.reduceRight\(\)](#)
- [.reverse\(\)](#)
- [.shift\(\)](#)
- [.slice\(\)](#)
- [.some\(\)](#)
- [.sort\(\)](#)
- [.splice\(\)](#)
- [.toLocaleString\(\)](#)
- [.toString\(\)](#)
- [.unshift\(\)](#)
- [.values\(\)](#)

FUNÇÕES DE ARRAYS | Static

- Array.[from\(\)](#)
- Array.[isArray\(\)](#)
- Array.[of\(\)](#)

Array | https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array

FUNÇÕES DE ARRAYS | Instâncias

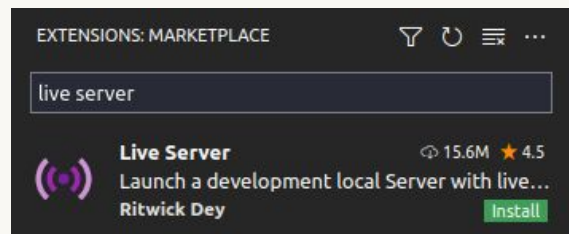
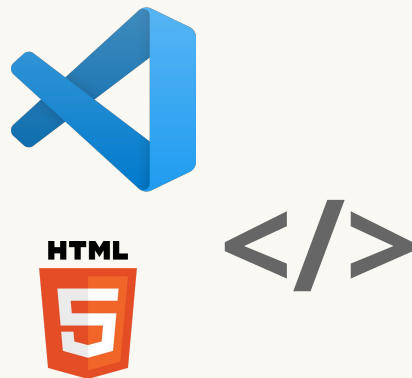
- [.forEach\(\)](#) executa uma dada função em cada elemento de um array
- [.map\(\)](#) executa a função para cada item e cria um novo array com todos returns
- [.filter\(\)](#) cria um novo array com os itens que passarem no teste da função fornecida
- [.find\(\)](#) retorna o valor do primeiro item que satisfizer a função de teste provida
- [.every\(\)](#) testa se todos os itens passam pelo teste implementado pela função fornecida
- [.some\(\)](#) testa se ao menos um dos itens passa no teste implementado na função
- [.includes\(\)](#) determina se um array contém um determinado elemento
- [.reduce\(\)](#) executa uma função reducer (fornecida por você) para cada elemento do array, resultando num único valor de retorno

Cada item possui um link clicável para a documentação da Mozilla

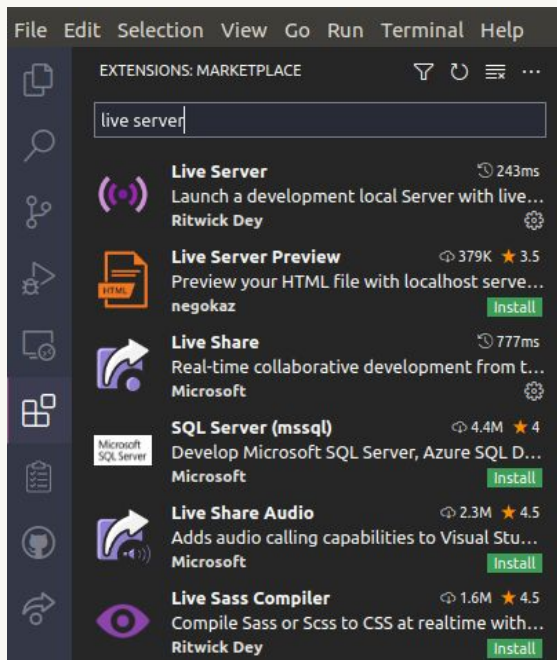
PARA A MÃO NA MASSA

- Instalar **VS Code**
(ou outro editor que se sentir mais confortável)
<https://code.visualstudio.com>
- Sugestão: Instalar extensão **Live Server** no **VS Code**
- Criar um arquivo **index.html** no seu editor

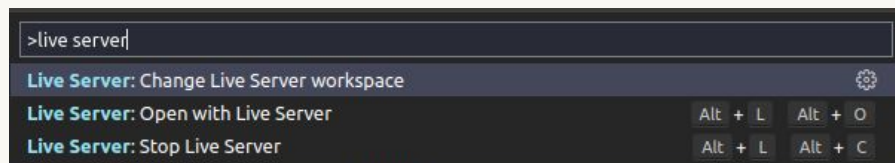
Code Sandbox | <https://codesandbox.io>
PlayCode | <https://playcode.io/new>
CodePen | <https://codepen.io/pen>
JSFiddle | <https://jsfiddle.net>



PARA A MÃO NA MASSA

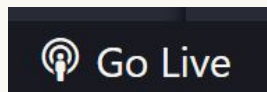


- **Ctrl + Shift + P**
Live Server: Open with Live Server



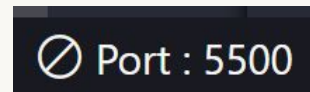
Start

- **Alt + L**
- **Alt + O**



Stop

- **Alt + L**
- **Alt + C**



MATERIAL COMPLEMENTAR



Arrow Functions vs. Functions em JavaScript | <https://youtu.be/S5Mn0qQzI-0>

JavaScript ES6 Arrow Functions Tutorial | <https://youtu.be/h33Srr5j9nY>

Arrow Functions | <https://youtu.be/AgOwGKB8D2M>

8 Must Know JavaScript Array Methods | <https://youtu.be/R8rmfD9Y5-c>

JavaScript: Funções de Array | https://www.youtube.com/playlist?list=PLYgzkrmJnLwo-pTURSTti_ljhjWB9BF7-

MATERIAL COMPLEMENTAR



Classes JavaScript | <https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Classes>

Arrow functions | developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Functions/Arrow_functions

JavaScript Arrow Function | https://www.w3schools.com/js/js_arrow_function.asp

Array | https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array



DEVinHouse

Parcerias para desenvolver a sua carreira

OBRIGADO!



<LAB365>