

12 de Junho de 2020

Construindo Player de Música com HTML, CSS e JS

Ironhack Web Development Webinar

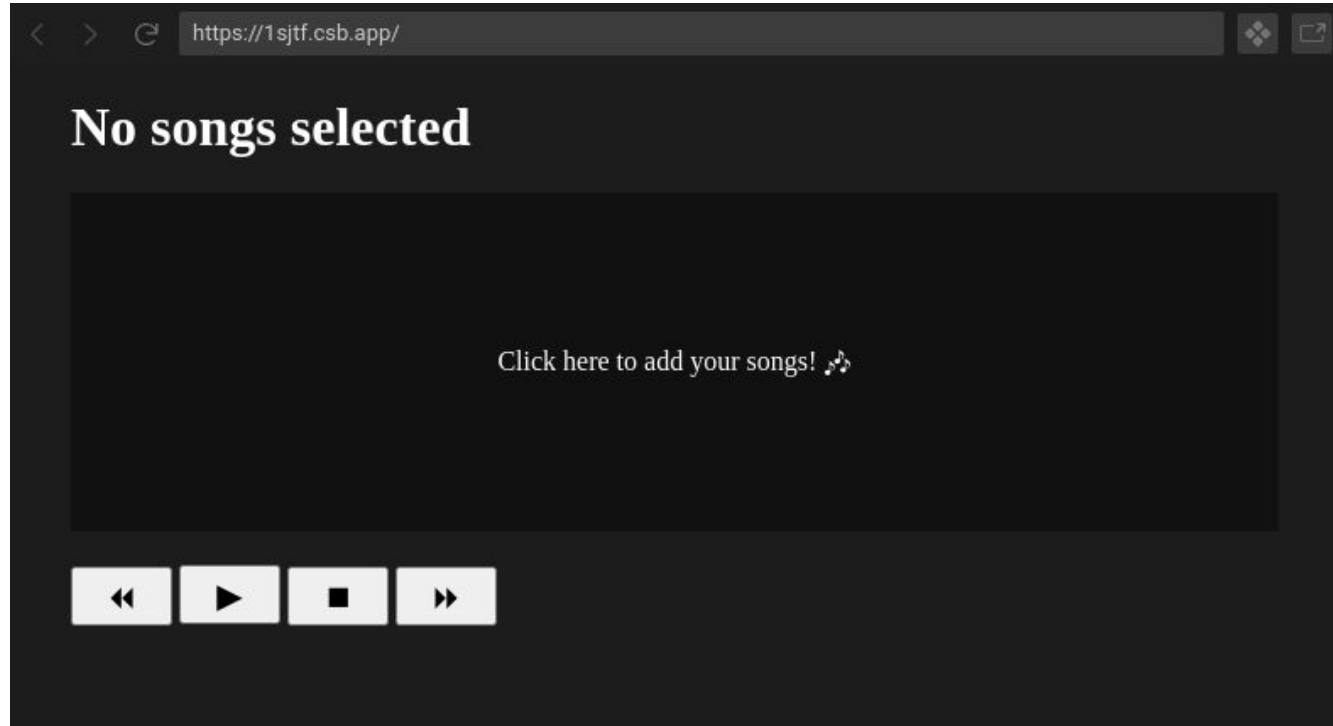


IRON
HACK

Ironhack Web Development Webinar

O que vamos construir?

Página Web de um Player de Música



Ironhack Web Development Webinar

O que vamos precisar?

O que vamos precisar?

Abra o link abaixo para ter acesso aos materiais que vamos precisar:

<https://tinyurl.com/ih-br-js-webinar>

Ironhack Web Development Webinar

Estrutura de Arquivos

[Codesandbox Starter Code](#)

Estrutura de Arquivos

```
src  |-- style.css
     |-- index.js
```

```
--- index.js
```

index.html

Ironhack Web Development Webinar

Introdução a JS (Javascript)

HTML x CSS x Javascript

Markup

.html

Conteúdo e Estrutura

Stylesheet

.css

Estilo e Layout

Programming

.js

Inteligência e Dinamismo

Introdução a JS

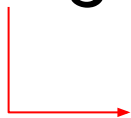
- Criada em 1995 por Brendan Eich para ser usada dentro do navegador da Netscape
- Estrutura inspirada na linguagem Java
- Teve alguns nomes
 - Mocha
 - Livescript
 - Javascript

Ironhack Web Development Webinar

Programando em Javascript

Comando básico do Javascript

```
console.log("Hello World")
```

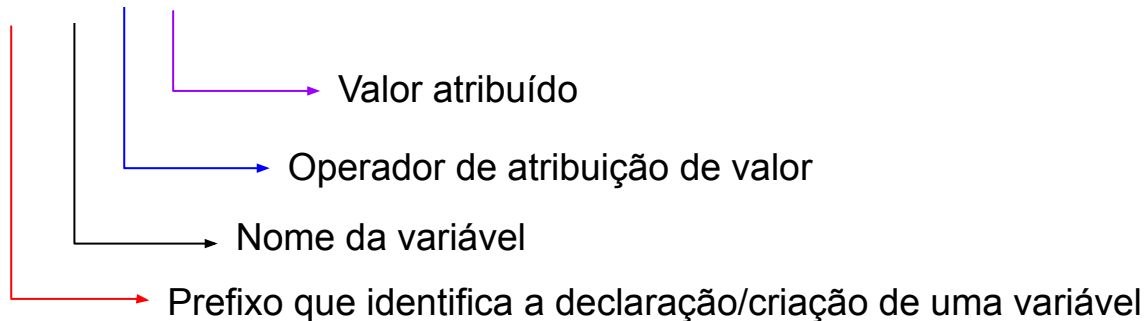


Mostra no console do navegador o(s) dado(s) passado(s) dentro dos "()

Declarando variáveis

- Guardam dados que serão manipulados/utilizados para executar uma determinada lógica
- Forma de declarar uma variável:

let **x** **=** **2**



Declarando variáveis

Prefixos utilizados:

let: utilizado para declaração de variáveis que poderão ter seus valores alterados dentro da nossa lógica

const: utilizado para declarar variáveis com valores constantes

Declarando variáveis

Exemplos:

```
let x = 2;  
console.log(x); // 2
```

```
x = "agora o valor desta variável é um texto";  
console.log(x); // "agora o valor desta variável é um texto"
```

```
const y = 2;  
console.log(y); // 2  
  
y = "example";  
console.log(y); // ???
```

Ironhack Web Development Webinar

Manipulando Elementos do HTML

Manipulando Elementos do HTML

Em todos os navegadores, existem variáveis globais que podemos acessar pelo Javascript e que nos permitem manipular nosso HTML

```
const bodyElement = document.querySelector("body");
```

Variável global do documento HTML

Método de captura de elementos

Nome do elemento HTML

Manipulando Elementos do HTML

Manipulação básica:

Capturar Elementos HTML - `const element= document.querySelector("body");`



Criar Eventos - eventos gerados a partir de alguma interação do usuário

- **onclick** --- gera um evento caso o usuário clique em um determinado elemento
- **onchange** --- gera um evento caso o usuário altere o conteúdo de um elemento



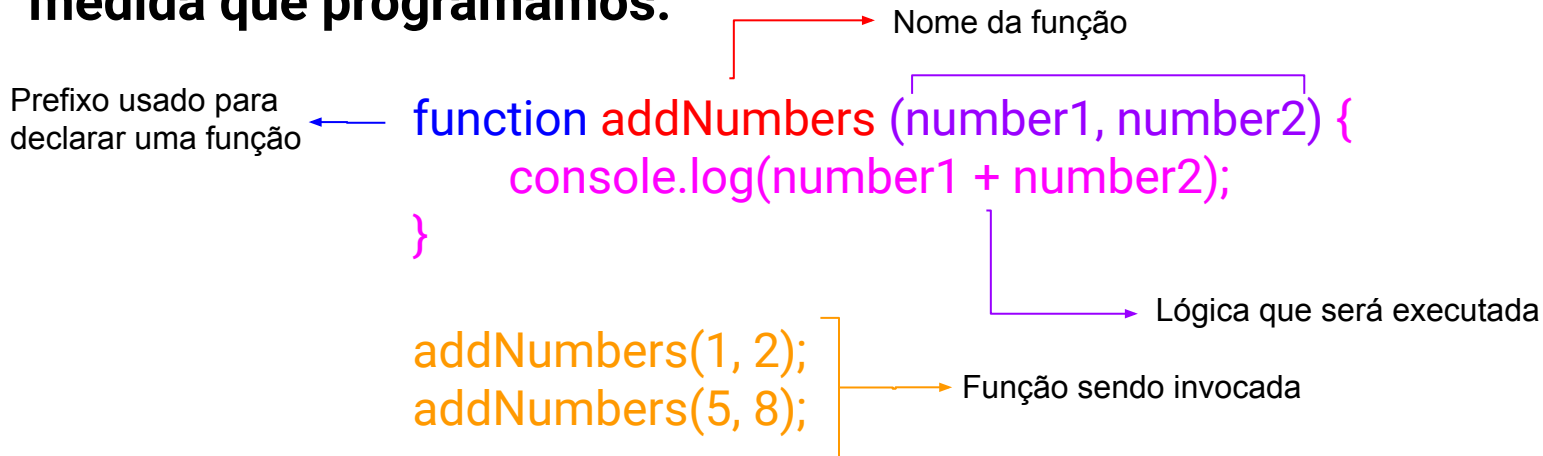
Manipulação de Elementos - alterar o HTML original dinamicamente

Ironhack Web Development Webinar

Funções

Funções

Funções são blocos de códigos / lógicas que poderão ser reutilizadas medida que programamos.



Ironhack Web Development Webinar

Estruturas de dados

Estruturas de dados

Number:

Estrutura usada para representar **números** e realizar operações matemáticas.

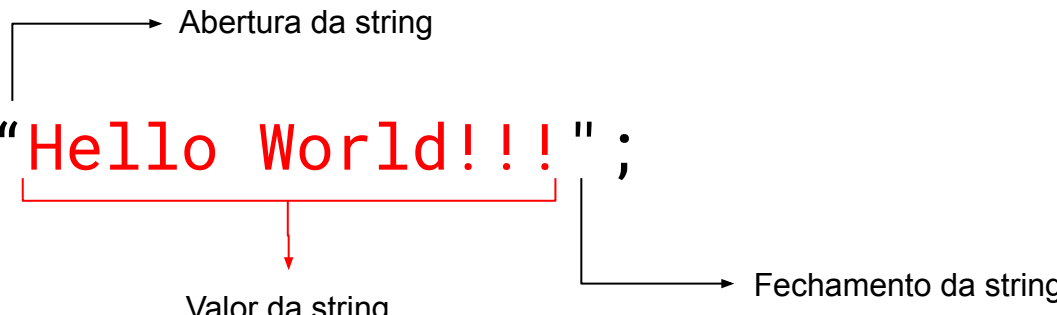
```
let number1 = 6;  
let number2 = 3;  
  
let sum = number1 + number2;  
let subtraction = number1 - number2;  
let multiplication = number1 * number2;  
let division = number1 / number2;  
let exponentiation = number1 ** number2;  
let module = number1 % number2;
```

Estruturas de dados

String:

Conjunto de caracteres utilizados para representar **texto**.

`const string = "Hello World!!!";`



The diagram illustrates the components of the string declaration `const string = "Hello World!!!";`. It features three annotations with arrows pointing to specific parts of the code: an arrow labeled "Abertura da string" (String opening) points to the opening double quote `"`; an arrow labeled "Valor da string" (String value) points to the text `Hello World!!!` inside the quotes; and an arrow labeled "Fechamento da string" (String closing) points to the closing double quote `"`.

Podemos utilizar `" "` ou `' '` para representar uma **string**.

Estruturas de dados

Array:

Estrutura utilizada para representar uma lista de dados.

`const array= ["Hello World", 3, 10, "Test!"] ;`

The diagram illustrates the components of an array declaration. A black arrow points from the text 'Abertura do array' to the opening square bracket '['. A red bracket spans the elements of the array, with a red arrow pointing down from its center to the text 'Conteúdo do array'. A black arrow points from the closing square bracket ']' to the text 'Fechamento do array'.

Abertura do array

Conteúdo do array

Fechamento do array

Estruturas de dados

Array:

Operações simples com arrays:

```
let fruitsList = ["maçã", "mamão", "manga", "melão"];  
  
console.log(fruitsList); // retorna o array completo  
  
console.log(fruitsList[0]); // retorna somente o primeiro elemento  
  
console.log(fruitsList[3]); // retorna somente o ultimo elemento  
  
console.log(fruitsList[5]); // undefined (?????)  
  
console.log(fruitsList.length); // retorna o tamanho do array
```

Estruturas de dados

Boolean

Dado utilizado em estruturas condicionais. Pode assumir os valores

True ou **False**

```
let boolean1 = true;
```

```
let boolean2 = false;
```

Estruturas de dados

Boolean:

Operadores lógicos: operadores utilizados para testar uma expressão e retornar **true** ou **false**

=== Operador EQUAL

>= Operador GREATER OR EQUAL THAN

!== Operador NOT EQUAL

<= Operador LOWER OR EQUAL THAN

> Operador GREATER THAN

&& Operador AND

< Operador LOWER THAN

|| Operador OR

Ironhack Web Development Webinar

Estruturas Condicionais

Estruturas Condicionais

Cláusula **If**:

Permite incluir condições que executam uma lógica caso a mesma for verdadeira (**true**) ou falsa (**false**)

```
let number1 = 6;
let number2 = 3;
if (number1 === number2) {
  console.log("os números são iguais");
} else {
  console.log("os números não são iguais");
}
```

Diagrama explicativo do código acima:

- Início da condição**: Indica o ponto de partida da cláusula `if`.
- Condição a ser testada**: Indica a expressão lógica dentro dos parênteses: `number1 === number2`.
- Lógica executada caso a condição **for** atendida**: Indica o bloco de código dentro das chaves que será executado se a condição for verdadeira.
- Lógica executada caso a condição **não for** atendida**: Indica o bloco de código dentro das chaves que será executado se a condição for falsa.

Ironhack Web Development Webinar

Perguntas e Respostas

Iniciando / Complementando Estudos

FreeCodeCamp: plataforma gratuita de estudo para HTML, CSS, JS e outros

<https://www.freecodecamp.org/learn>

