

IOS – Instituto de
Oportunidade Social

Aula JS 03 - Objetos, Funções e Eventos



- > Funções
- > Arrow Function
- > Objetos
- > Eventos

IOS – Instituto de
Oportunidade Social

Funções



> Funções

Funções são usadas em programação para executar ações que são rotineiramente executadas em um programa. Uma função é um bloco de código implementado para executar uma tarefa em particular. Para executar uma função devemos sempre invocá-la (chamá-la) dentro do código

> Funções



Importante! O nome de funções segue as mesmas regras de variáveis deve começar por letras ou underline (_) e pode conter números e não pode conter caracteres especiais. O nome também não pode ser uma palavra-chave ou palavra reservada. Programadores geralmente usam somente nomes de funções como camelCase ou CamelCase, isto é, a primeira letra de cada palavra deve ser escrita em maiúscula. Isso padroniza o código e facilita identificar o que é variável e o que é função.

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Arrow Function



O conceito de Arrow Function foi introduzido em 2015 no ES6. Acostume-se bem com essa forma mais prática de declarar funções, pois muitos exemplos que você encontrará na internet usará esse tipo de declaração.

A sintaxe de uma arrow function é:

```
const hello = () => {  
  return 'Olá Arrow Function!';  
};
```


No exemplo mostrado anteriormente, a função não tem nenhum parâmetro de entrada (dentro dos parênteses da função está vazio). Toda função pode ter zero ou mais parâmetros de entrada. Mas também podemos fazer uma versão da função `addNums` como arrow function, vejamos o exemplo:

```
const addNums2 = (num1 = 1, num2 = 1) => {  
  return num1 + num2;  
};
```

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Objetos



Objetos

JS



Propriedades	Métodos
car.name = Chevrolet	car.start()
car.model = Celta	car.drive()
car.weight = 890kg	car.brake()
car.color = grey	car.stop()

No JS, um objeto é uma coleção de dados e/ou funcionalidades relacionadas (que geralmente consistem em diversas variáveis e funções — que são chamadas de propriedades e métodos quando estão dentro de objetos). Por enquanto, pense que toda declaração de variável ou função feita no JS é um objeto.

As variáveis mostradas abaixo são objetos:

```
let marca = 'Fiat';
```

```
const carro = {  
  type: 'Fiat',  
  model: '500',  
  color: 'white',  
};
```

```
const pessoa = {  
  firstName: 'John',  
  lastName: 'Doe',  
  age: 50,  
  eyeColor: 'blue',  
};
```

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Eventos



Eventos são ações executadas quando algo acontece na página web, ou seja, é a reação algum estímulo ou interação em elemento HTML. Esses eventos do HTML normalmente chamam funções do JavaScript. Eventos HTML são coisas que o navegador pode fazer ou algo que o usuário pode fazer

Os principais eventos do HTML são:

Event	Description
onchange	Um elemento HTML é alterado.
onclick	O usuário clica em um elemento HTML.
onmouseover	O usuário move o mouse sobre o elemento HTML.
onmouseout	O usuário tira o mouse de cima do elemento HTML.
onkeydown	O usuário pressiona um Tecla do teclado.
onload	O navegador termina de carregar a página.

> Vamos Praticar

IOS – Instituto de
Oportunidade Social

Exercício



Crie um arquivo *index.html*

Crie no arquivo *index.html* uma *tag button* com o evento *onclick* chamando a função ***alertCookie()*** o nome do *button* será ***gerando cookie***

Crie um arquivo *main.js*

Crie no arquivo *main.js* uma ***função simples*** chamando o ***alertCookie()*** contendo um alert com a seguinte mensagem ‘*Você quer um cookie*’ e imprime a mesma mensagem no *console.log*

Verifique o resultado no seu Browser

Crie no arquivo *index.html* uma segunda *tag button* com o evento *onclick* chamando a função ***alertSucesso()*** o nome do button será ***Mensagem de sucesso***

Crie no arquivo *main.js* uma ***Arrow Function*** chamando o ***alertSucesso()*** contendo um alert com a seguinte mensagem '*Mensagem gerada com sucesso*' e imprime a mesma mensagem no console.log

Verifique o resultado no seu Browser

No arquivo main.js crie um objeto chamado notebook e adicione cinco propriedades com seus respectivos valores

exiba o objeto no console.log

Obs: para imprimir um objeto no console é diferente de exibir uma função: para exibir um objeto é `console.log(notebook)` sem os parênteses junto ao nome do objeto

Crie uma função que faça a conversão de anos para dias e exiba o resultado no console.log

— — — — — — — — — —

Envie seus exercício para o GitHub e disponibilize o link no moodle na respectiva atividade