IOS – Instituto de Oportunidade Social

JS 08 - Objetos, Funções e Eventos

Objetivos da aula

- Compreender o uso de funções e sua sintaxe;
- Conhecer o conceito de arrow function;
- Entender abstração de objetos do mundo real;
- Conhecer os diferentes tipos de eventos em JavaScript. Tipos de Dados;

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.

A sintaxe para criar funções em JS é:

```
function myFunc(valor1, valor2) {
        return valor1 * valor2;
}
```

Funções

No exemplo mostrado anteriormente, a função com o nome **myFunc** espera receber **dois parâmetros** (valor1 e valor2) e **retorna** o resultado da multiplicação dos dois valores (return valor1 * valor2;). Quando o JavaScript encontra a **instrução return**, a função para de ser executada. Se a função foi chamada a partir de uma instrução, o JavaScript "retornará" para executar o código após a instrução de chamada. As **funções geralmente calculam um valor de retorno**. O valor de retorno é "retornado" de volta ao objeto que a chamou.

***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.*

IOS – Instituto de Oportunidade Social

Arrow Function

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!';
};
```

Basicamente, você deve **atribuir a função a uma variável** declarada com a palavra **const** e utilizar o operador **=>** para indicar o bloco da função. A palavra const pode ser ocultada se você não estiver usando o **Strict Mode**, mas vamos seguir as **boas práticas de programação**.

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;
};
```

IOS – Instituto de Oportunidade Social

Objetos

Objetos

Na vida real um carro é um **objeto** e esse objeto possui **propriedades** como, por exemplo: **nome**, **modelo** e **peso**, e **métodos** (ações a serem executadas) tais como: **ligar**, **dirigir**, **frear** e **parar**. As propriedades de um carro podem ser as mesmas, mas os **valores são diferentes** de carro para carro. Os métodos de um carro podem também ser os mesmos, mas a execução pode ser ligeiramente diferente de carro para carro, por exemplo, um carro pode ligar utilizando a chave e outro pode ser com um botão.

Objetos

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**.

A seguir os exemplos dos objetos **carro** e **pessoa** são bem interessantes, pois eles são formados por um **conjunto de valores** ou o que chamamos de **Array de objetos**. Falaremos mais sobre array de objetos mais adiante. Vamos abordar cada assunto com calma e aprofundar gradativamente.

Objetos

```
let marca = 'Fiat';

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

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

Objetos

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.

IOS – Instituto de Oportunidade Social

Vamos Praticar

Vamos Praticar

Apostila de JS

04.Javascript

Páginas 105 a 115

OBS: Acompanhar o passo a passo com o instrutor

IOS – Instituto de Oportunidade Social

Exercícios

Exercícios

Criar uma página HTML que emita uma **mensagem ao acessar** através de **evento**, na página deve constar uma equação do segundo grau a sua escolha Ex: $3x^2 + 4x - 5 = 0$ *(sendo a = 3, b = 3 e c = -5)* e um **botão** para efetuar o **cálculo** de **x1** e **x2**, o botão deve ter um **evento** para **calcular a equação** e **mostrar o resultado em tela** para o usuário. A primeira função deve ser no método tradicional e a segunda com **Arrow Function**. Funções abaixo:

```
const delta = b * b - 4 * a * c;
const x1 = (-b + Math.sqrt(delta)) / (2 * a);
const x2 = (-b - Math.sqrt(delta)) / (2 * a);
```