

INTRODUÇÃO A LÓGICA DE PROGRAMAÇÃO

A Lógica de Programação é necessária para pessoas que desejam trabalhar com desenvolvimento de sistemas e programas, ela permite definir a sequência lógica para o desenvolvimento. Então o que é lógica?

Lógica de programação é a técnica de encadear pensamentos para atingir determinado objetivo.

SEQUÊNCIA LÓGICA

Estes pensamentos podem ser descritos como uma sequência de instruções, que devem ser seguidas para se cumprir uma determinada tarefa.

Sequência Lógica são passos executados até atingir um objetivo ou solução de um problema.

Em nossa nova jornada, vamos aprender tudo que precisamos saber para que possamos destravar nossa mentalidade lógica e desenvolver assim projetos tanto básico como projetos com alto nível de complexidade utilizando a linguagem JavaScript.

INICIANDO NO JAVASCRIPT

O JavaScript, como o próprio nome sugere, é uma linguagem de scripting. Uma linguagem de scripting é comumente definida como uma linguagem de programação que permite ao programador controlar uma ou mais aplicações de terceiros. No caso do JavaScript, podemos controlar alguns comportamentos dos navegadores através de trechos de código que são enviados na página HTML.

Outra característica comum nas linguagens de scripting é que normalmente elas são linguagens interpretadas, ou seja, não dependem de compilação para serem executadas. Essa característica é presente no JavaScript: o código é interpretado e executado conforme é lido pelo navegador, linha a linha, assim como o HTML.

O JavaScript também possui grande tolerância a erros, uma vez que conversões automáticas são realizadas durante operações. Como será visto no decorrer das aulas, nem sempre essas conversões resultam em algo esperado, o que pode ser fonte de muitos bugs, caso não conheçamos bem esse mecanismo.

CONSOLE DO NAVEGADOR

Existem várias formas de executar códigos JavaScript em uma página. Uma delas é executar códigos no que chamamos de Console. A maioria dos navegadores desktop já vem com essa ferramenta instalada. No Chrome, é possível chegar ao Console apertando F12 e em seguida acessar a aba "Console" ou por meio do atalho de teclado Ctrl + Shift + C (assim como vemos na aula); no Firefox, pelo atalho Ctrl + Shift + K.

DEVELOPER TOOLS

O console faz parte de uma série de ferramentas embutidas nos navegadores especificamente para nós que estamos desenvolvendo um site. Essa série de ferramentas é o que chamamos de Developer Tools.

SINTAXE BÁSICA

OPERADORES

Podemos somar, subtrair, multiplicar e dividir como em qualquer linguagem:

Teste algumas contas digitando diretamente no console:

```
> 12 + 13 = 25
> 14 * 3 = 42
> 10 - 4 = 6
> 25 / 5 = 5
> 23 % 2 = 1
```

Para armazenarmos um valor para uso posterior, podemos criar uma variável:

```
> var resultado = 102 / 17;
> undefined
```

No exemplo acima, guardamos o resultado de 102 / 17 na variável resultado. O resultado de criar uma variável é sempre undefined. Para obter o valor que guardamos nela ou mudar o seu valor, podemos fazer o seguinte:

```
> resultado
> 6
> resultado = resultado + 10
> 16
> resultado
```

```
> 16
```

Também podemos alterar o valor de uma variável usando as operações básicas com uma sintaxe bem compacta:

```
> var idade = 10; // undefined  
> idade += 10; // idade vale 20  
> idade -= 5; // idade vale 15  
> idade /= 3; // idade vale 5  
> idade *= 10; // idade vale 50
```

TIPO DE DADOS

Não são apenas números que podemos salvar numa variável. O JavaScript tem vários tipos de dados.

NUMBER

Com esse tipo de dados é possível executar todas as operações que vimos anteriormente:

```
var pi = 3.14159;  
var raio = 20;  
var perimetro = 2 * pi * raio
```

STRING

Uma string em JavaScript é utilizada para armazenar trechos de texto:

```
var empresa = "TORNE-SE UM PROGRAMADOR";
```

Para exibirmos o valor da variável empresa fora do console, podemos executar o seguinte comando:

```
alert(empresa);
```

O comando *alert* serve para criação de popups com algum conteúdo de texto que colocarmos dentro dos parênteses. O que acontece com o seguinte código?

```
var numero = 30;  
alert(numero)
```

O número 30 é exibido sem problemas dentro do popup. O que acontece é que qualquer variável pode ser usada no alert . O JavaScript não irá diferenciar o

tipo de dados que está armazenado numa variável, e se necessário, tentará converter o dado para o tipo desejado.