JavaScript Básico

Mariana da Costa Lisboa

Conceitos Básicos

Imprimindo Valor no Terminal:

Para imprimir um valor no terminal (talvez a tarefa mais básica e fundamental de qualquer linguagem de programação), é preciso usar o comando:

```
console.log(“hello world”)
```

Variáveis:

No JavaScript existem três tipos de variáveis, cada uma para a sua determinada função.

```
var nome = “Mariana”
let idade = 17
const altura = 1.48
```

Mas qual a lógica por trás de cada uma?

A variável do tipo **var** é o modo antigo de se declarar variáveis em JavaScript, porém é bastante usada, pois pode ser utilizada em várias ocasiões.

A variável do tipo **let** é usada para variáveis que podem sofrer alterações no futuro.

A variável do tipo **const** é utilizada para variáveis constantes, ou seja, que não sofrerão alterações futuras.

Tipos de Dados:

No JavaScript existem alguns tipos de dados, como:

- String: usadas em casos de textos;
- Number: usadas em casos de números, independente de serem inteiros ou fracionados;
- Boolean: usadas em casos de valores booleanos (verdadeiro ou falso);
- Null: usadas em caso de variáveis vazias permanentemente;
- Undefined: usadas em caso de variáveis vazias temporariamente.

Operadores:

Operadores Aritméticos:

Os operadores aritméticos são usados em alguns cálculos, segue os exemplos:

- + soma
- - subtração

- * multiplicação
- / divisão
- % resto de divisão
- ++ incremento
- – decremento

Operadores de Atribuição:

Geralmente são usados em atribuições de variáveis:

- = atribuição simples
- =+ atribuição e soma
- =- atribuição e subtração
- =* atribuição e multiplicação
- =/ atribuição e divisão

Operador es de Comparação:

Usadas exatamente para comparar valores, exemplos:

- == igualdade do dado
- != desigualdade do dado
- === igualdade do dado e de seu tipo
- !== desigualdade do dado e de seu tipo
- > maior que
- < menor que
- >= maior igual que
- <= menor igual que

Operadores Lógicos:

Na estrutura de repetição são usados alguns operadores lógicos para ajudar a especificar melhor qual será o resultado obtido, veja alguns exemplos:

- &&: é usado para o “e”;
- ||: é usado como o “ou”;
- !: é usado como o “não”

Operadores de Concatenação:

O “+” é um operador de concatenação entre um texto e uma variável ou de uma variável e outra.

Listas

Vamos a uma rápida passada pelas listas (variáveis compostas, ou seja, que recebem mais que um dado).

```
var frutas = [“banana”, “maçã”, “uva”, “jabuticaba”]
console.log(frutas) //será exibido toda a lista
console.log(frutas[2]) //será exibido a fruta que ocupa a segunda posição,ou seja, “uva”
```

Dá para fazer mais coisas com listas, como adicionar novos elementos, remover outros, mas isso é assunto para uma aula mais detalhada (mesmo assim deixa atividades preparadas).

```
frutas.push(“pera”) //será adicionado uma nova fruta na lista
frutas.splice(1) //remove o elemento que ta na posição 1
```

Estruturas de Controle

Estrutura de Condição:

A estrutura de condição é o famoso “se”, “se não” e “então”, vamos para o código que dá para entender melhor.

```
var idade = 17
if(idade < 17){
        console.log(“menor de idade)
}
else if(idade == 18){
        console.log(“tem 18 anos”)
}
else{
        console.log(“maior de idade”)
}
```

Estrutura de Repetição:

A estrutura de repetição, como o próprio nome diz, ele repete, automatiza tarefas repetitivas, facilitando assim a vida do desenvolvedor. Por exemplo, para fazer um contado de 0 até 5, como eu faria sem uma estrutura de repetição:

```
console.log(0)
console.log(1)
console.log(2)
console.log(3)
console.log(4)
console.log(5)
```

Agora vamos tentar com uma estrutura de repetição:

```
var i;
```

```
for (i = 0; i < 6; i++){
        console.log(i)
}
```

Tem o modo de repetição **while** também, que é usada quando não sabemos a quantidade de repetições necessária, porém não será passada hoje.

```
var i;
while (i <=5){
        console.log(i)
        i++
}
```

Funções

As funções são usadas exatamente para reduzir a quantidade de linhas de código e facilitar/organizar o chamado de instruções, vamos da uma olhada:

```
function contagem(){
        for (var i = 0; i < 5; i++){
                console.log(i)
        }
}

contagem()
```

Nesse exemplo o uso de uma função parece inútil,porém numa situação onde existe uma parte de código que será repetida inúmeras vezes, as funções poupa muito estresse.