

JavaScript

Nome
Leonardo

Nome
Francisco Theodoro



O que é JavaScript?

- É uma linguagem de programação de alto nível, interpretada e baseada em objetos
- Possui tipagem dinâmica
- É amplamente utilizada no desenvolvimento web para adicionar interatividade e dinamismo às páginas web.



Introdução



Tópicos da Introdução

- Impressão de valores na tela
- Declaração de variáveis
- Operadores Aritméticos
- Operadores de Comparação
- Operadores Lógicos

Impressão de Valores

Para imprimir algo em nosso terminal utilizamos o comando `console.log(parâmetro)`.

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

SAÍDA CONSOLE DE DEPURAÇÃO PROBLEMAS TER

[Running] node "c:\Users\leona\OneDrive\
Hello World!

Declaração de Variáveis

- var:
 - É a forma mais antiga de declarar variáveis e constantes. Atualmente, é considerada obsoleta e má prática.
- let:
 - Forma mais utilizada, atualmente, para declarar variáveis em JavaScript!
- const:
 - Método para se criar constantes no JavaScript. Se o objeto não precisar que seu valor seja alterado durante o código, busque sempre usar const.

```
var prova;  
let teste;  
const disciplina;
```

Operadores Aritméticos

Os operadores aritméticos são símbolos que nos auxiliam a efetuar contas matemáticas dentro da linguagem de programação.

- Soma (+);
- Subtração(-);
- Multiplicação(*);
- Divisão(/);
- Potência(** ou Math.pow());
- Raiz(Math.sqrt());
- Incremento (++);
- Decremento(--);
- Atribuição + Operação (op= valor)

```
let soma = 2 + 2; // Soma é 4
let subtracao = 5 - 2; // Subtração é 3
let multiplicacao = 2 * 2; // Multiplicação é 4
let divisao = 10 / 5; // Divisão é 2
let mod = 3 % 2; // Resto da divisão inteira é 1
let pot = 3 ** 2; //Potência é 9
//outra forma de potencia:
pot = Math.pow(3,2); // Potência também é 9
let raiz = 4 ** (1/2) //raiz é 2
//outra forma de raiz:
rad = Math.sqrt(4,2) // raiz também é 2
soma++ //incremento
soma-- //decremento
//atribuição com operação:
soma += 5; //soma = soma + 5
soma -= 5; //soma = soma - 5
soma *= 5; //soma = soma * 5
soma /= 5; //soma = soma / 5
subtracao **= 2; //soma = soma ** 2
```

Operadores de Comparação

Os operadores de comparação são símbolos que nos auxiliam a efetuar comparações dentro da linguagem de programação e nos retornam valores booleanos (True ou False).

- Igual(==);
- Diferente(!=);
- Maior(>);
- Menor(<);
- Maior ou Igual(>=);
- Menor ou Igual(<=);

```
let igual = (2==2); // true
let diferente = (2!=2); // false
let maior = 2 > 3; // false
let menor = 2 < 3; // true
let maiorIgual = 2 >= 1 //true
let menorIgual = 2 <= 1 //false
```


Operadores Lógicos

Os operadores lógicos são símbolos que nos auxiliam a efetuar operações lógicas dentro da linguagem de programação e nos retornam valores booleanos (True ou False)

- && – AND; Retorna “true” se todas as variáveis comparadas forem “true”;
- || – OR; retorna “true” se, pelo menos, uma variável de todas comparadas for “true”.

```
47 let ou = (menor || maior);  
48 let e = (maior && diferente);  
49 console.log(ou);  
50 console.log(e);
```

SAÍDA CONSOLE DE DEPURAÇÃO PROBLEMAS

[Running] node "c:\Users\leona\OneDr
true
false



Treinamento Técnico 24.2

Tipos de Variáveis



Tópicos dos Tipos de Variáveis

- Escopo de Variável;
- Tipos de Variáveis;
- Case Sensitive;
- Comentários;
- parseInt e parseFloat;

Escopo de Variáveis

- Escopo Global:
 - Relacionado a todo o código, a variável pode ser acessada de qualquer local do código.
- Escopo Local:
 - Variável é declarada entre chaves “{}” e só pode ser acessada dentro das chaves.

```
let matricula = 122115986;//global  
  
{  
    let nota1 = 9;//Local  
    let nota2 = 8;//Local  
}
```


Tipos de Variáveis

Os tipos mais conhecidos hoje em JavaScript são:

- object;
- string;
- Number(inteiro e float);
- boolean;
- function;
- Array (objeto);
- NAN (not a number);
- undefined.

```
let objeto = {propriedade: "valor", propriedade2: "valor2"}
let array = [0, 2, "Gabriel", "Bruno"]
let string = "string";
let numberInt = 23;
let numberFloat = 1.5
let boolean = true;

function imprimirValor (valor) {
  console.log(valor);
  return;
}

let imprimirValor2 = (valor) => {console.log(valor)}
imprimirValor(typeof array)
imprimirValor(numberFloat)
imprimirValor2(numberFloat)
```

Case sensitive e Tipagem Dinâmica

Case Sensitive - em JavaScript, temos a diferenciação entre letras maiúsculas e minúsculas.

Tipagem dinâmica - nossas variáveis recebem seu tipo “automaticamente” quando atribuímos o valor ao nome da variável declarada.

```
let num = 1  
let Num = 2  
console.log(num == Num)
```

Comentários

Para fazermos comentários em JavaScript temos 2 formas:

- Comentário por linha que utiliza o símbolo “//”
- Comentários em blocos que utiliza o símbolo “/**(conteúdo)*/”.

```
/*  
    Aqui estão os comandos utilizados na aula de JavaScript  
    Será disponibilizado junto da aula no github  
*/  
  
// Imprimindo valores  
//console.log("Hello World!")  
💡  
// Declaração de Variáveis
```

parseInt() e parseFloat()

São funções que convertem strings em número:

- parseInt() - analisa a string passada como parâmetro e caso encontre algum número antes de qualquer caractere, o transforma em inteiro e retorna seu valor.
- parseFloat - efetua a mesma coisa que o parseInt() porém transforma o número em float e o retorna.

```
89 let id = "123456ab"
90 let idFix = parseInt(id)
91 console.log(idFix)
92
93 let id2 = "7.89bc"
94 let id2Fix = parseFloat(id2)
95 console.log(id2Fix)
96
```

SAÍDA	CONSOLE DE DEPURACÃO	PROBLEMA
[Running]	node "c:\Users\leona\One	
123456		
7.89		



Treinamento Técnico 24.2

Arrays e Índices



Tópicos em Arrays

- Criação e Manipulação de Arrays;
- Índices;

Criação e Manipulação de Arrays

Como já falamos da tipagem dinâmica do JavaScript, para criar um array(lista) precisamos apenas atribuir dados entre colchetes([])

O array é separado em índices, que são como “páginas” para localizar os conteúdos que estão listados dentro dele.

Obs: O primeiro índice não é 1, mas o zero (0).

```
let toDoList = ["Estudar JS I", "Fazer exercícios do TT", "Chorar um pouco",  
               "Fazer mais exercícios do TT" ]  
  
toDoList[0] = "Estudar JS II"
```

Criação e Manipulação de Arrays

Alguns métodos para manipular arrays:

- `concat()` une dois arrays e retorna um novo array'
- `push()` adiciona um ou mais elementos no fim de um array e retorna o comprimento resultante do array'
- `pop()` remove o último elemento de um array e retorna esse elemento'
- `shift()` remove o primeiro elemento de um array e retorna esse elemento.

```
todoList = todoList.concat(todoList2)

todoList2.push("Dormir um pouco")
todoList2.pop();
todoList.unshift("Acordar")
todoList.shift()
```


Criação e Manipulação de Arrays

- Unshift(): Adiciona itens no começo do Array.
- map: cria um novo array com os resultados da chamada de uma função para cada elemento do array
- filter: cria um novo array com todos os elementos que passaram no teste implementado pela função fornecida
- reduce: aplica uma função contra um acumulador e cada elemento do array (da esquerda para a direita) para reduzi-lo a um único valor.

Criação e Manipulação de Arrays

- find: retorna o primeiro elemento no array que satisfaz a função de teste fornecida. Caso contrário, undefined é retornado
- includes: determina se um array contém um determinado elemento, retornando true ou false.

```
let numeroEncontrado = numeros.find(function(numeros){  
  return numeros == 3  
})  
  
let incluiDois = numeros.includes(2);
```

Índices

- Os índices são como páginas que expressam a localização de um dado dentro do array ou até de uma string

Condicionais



Tópicos em condicionais

- If - else
- Operador Ternário

If Else

Comandos que auxiliam quando é necessário tomar uma decisão que depende de alguma condição.

```
teste = 5

if(teste >= 7){
    console.log("Aprovado")
} else{
    console.log("Nota Insuficiente, faça a PF")
}
```

Operador Ternário

Exerce o mesmo fluxo condicional que o “if e else”, porém com um formato diferente.

```
//ternario  
teste >= 6 ? console.log("Aprovado") :  
console.log("Nota insuficiente, Faça a PF")
```

Iterações



Tópicos em iterações

- For
- Do-While
- While

For()

Laço que obedece a sintaxe: `for(variável, condição, incremento ou decremento) { bloco de código }`

```
for(let i = 0; i < 3; i++){  
  console.log("Estudar")  
  console.log("Dormir")  
}
```

Do While

Comando de iteração que é executado uma vez antes da condição se verificada e obedece a sintaxe: `do{bloco de código com incremento ou decremento}while(condição).`

```
let i = 0
do{
  console.log("Jogar")
  i+=1
}while(i<5)
```

While

Laço de iteração que obedece a seguinte estrutura: `while(condição){bloco de código com incremento ou decremento}`

```
let j = 0
while(j<10){
    console.log(j * 2)
    j++
}
```

Dúvidas???

