




JS

RESPONSIVE WEB DEVELOPMENT

Prof. Alexandre Carlos de Jesus
Prof. Israel Marques Cajai Júnior
Prof. Luis Carlos de Souza Silva





O QUE É ?

LINGUAGEM NATIVA DA WEB.

MELHORA A USABILIDADE E A INTERATIVIDADE DO
USUÁRIO COM A PÁGINA.

COM O LANÇAMENTO DO NODE.JS, COMEÇOU A
SERVIR NO LADO DO SERVIDOR.

JS





ONDE USAR ?

**PODEMOS PROGRAMAR ARDUINO COM A BIBLIOTECA
JOHNNY-FIVE.**

**PODEMOS CRIAR APLICATIVOS DESKTOP COM O
FRAMEWORK ELECTRON.**

JS

**PODEMOS USAR O BANCO DE DADOS MONGODB, QUE
UTILIZA A SINTAXE JAVASCRIPT EM SUAS QUERIES.**

Operadores aritméticos

+

SOMA

-

SUBTRAÇÃO

/

DIVISÃO

*

MULTIPLICAÇÃO

%

MÓDULO

**

POTÊNCIA

Operadores relacionais

>

MAIOR QUE

<

MENOR QUE

>=

MAIOR OU IGUAL A

<=

MENOR OU IGUAL A

==

IGUALDADE
evite usar

!=

DESIGUALDADE
evite usar

===

IGUALDADE
ESTRITA

!==

DESIGUALDADE
ESTRITA

Operadores lógicos

&&

E (AND)

Todos verdadeiros =
TRUE

||

OU (OR)

Apenas um verdadeiro =
TRUE

!

NÃO (NOT)

Inverte resposta

RETORNAM APENAS TRUE / FALSE

IF

Permite a criação de testes lógicos que determinam se uma condição é verdadeira ou falsa.

```
if (teste lógico){  
    condição verdadeira}  
else{  
    condição falsa}
```

SWITCH

Compara uma expressão com os valores de cada caso. No fim de cada caso deve possuir um comando break. Caso nenhum valor seja igual à expressão, executa o bloco de comandos default (não obrigatório).

```
switch(expressão){  
    case n:  
        bloco de comando;  
        break;  
    case n:  
        bloco de comando;  
        break;  
    default:  
        bloco de comando;  
}
```


FOR

É uma estrutura de repetição com um número definido de vezes que algum determinado bloco de comandos será executado. Você deve definir: o início, o fim e o incremento.

```
for (let i = 0 ; i <= 9 ; i++){  
    bloco que será executado...  
}
```

Neste exemplo a variável *i* é iniciada valendo 0, o laço deve repetir enquanto *i* for menor ou igual a 9, o laço é incrementando automaticamente em 1.

WHILE

Usado basicamente quando não sabemos o número de repetições, mas pode ser usado também quando sabemos quantas vezes o laço deve ser executado.

```
while (condicionamento) {  
    bloco que será executado...  
    pode ter vários comandos...  
}
```



DO...WHILE

Devido a sua sintaxe, executa o laço pelo menos uma vez.

```
let i = 0; // variável inicial
do {
    bloco que será executado...
}
while (i <= n); // até quando será executado.
```

OBJETO STRING

Propriedade length - retorna o tamanho da string.

Método toLowerCase() - retorna a string em letras minúsculas.

Método toUpperCase() - retorna a string em letras maiúsculas.

Método charAt() - retorna o caracter contido em uma determinada posição.

Método indexOf('character') - verifica em que posição está determinado caracter, caso não encontre retorna -1. Inicia a busca pelo primeiro caracter.



OBJETO STRING

Método indexOf('caracter', posição) - verifica em que posição está o caracter, começando na posição determinada.

Método lastIndexOf('caracter') - verifica em que posição está determinado caracter, caso não encontre retorna -1. Inicia a busca pelo último caracter.

Método substring(início,fim) - retorna um pedaço da string, você pode determinar o começo e fim deste retorno.



OBJETO STRING

Método slice(posição inicial, posição final) - retorna uma cópia de parte da string. Você deve passar como parâmetros o índice que deve iniciar a cópia e o índice final. Cuidado pois não pega o conteúdo do índice final, ele é usado para indicar onde deseja finalizar a cópia. Caso não seja declarado, pegará o conteúdo até o final da string.

Método split('caracter', tamanho) - separa uma string em um array de strings. Você pode usar qualquer caractere existente na string para fazer a separação. Caso nada seja informado o array terá como único índice a string inteira. Podemos também definir a quantidade de elementos que existirão nesse novo array. Os dois parâmetros são opcionais.



OBJETO STRING

Método `replace('conteúdo', 'novo conteúdo')` - Substitui um determinado conteúdo na string. Caso o conteúdo seja um único caracter, ele irá aplicar a alteração no primeiro caracter da string encontrado, ignorando os demais. Utilizando Expressão Regular ou alguma função podemos fazer a substituição de todos os caracteres.



OBJETO MATH

Math.ceil() - retorna o próximo valor inteiro.

Math.floor() - retorna o valor inteiro anterior.

Math.round() - caso o número tenha casas decimais entre 0 e 49, ele retorna o valor inteiro anterior. Caso o número tenha casas decimais entre 50 e 99, ele retorna o próximo valor inteiro.

Math.pow() - potência de um número. (lembrar do operador **)



OBJETO MATH

Math.max() - retorna o maior valor de um intervalo.

Math.min() - retorna o menor valor de um intervalo.

Math.random() - sortear um número entre 0 e 1.

Math.sqrt() - raiz quadrada de um número.

Math.cbrt() - raiz cúbica.

toFixed() - define o número de casas decimais.

parseInt() - retorna a parte inteira de um número.

parseFloat() - retorna o número e suas respectivas casas decimais.



ARRAY

Declarando um array

```
var nome =["valor 1", "valor 2", "valor 3", ...];
```

Tamanho do array

```
Nome do array.length;
```

Adicionar um item no fim do array

```
Nome do array.push("conteúdo");
```



ARRAY

Remover um item no fim do array

Nome do array.pop();

Adicionar um item no início do array

Nome do array.unshift("conteúdo");

Remover um item no início do array

Nome do array.shift();



ARRAY

Remover um item no fim do array

Nome do array.pop();

Adicionar um item no início do array

Nome do array.unshift("conteúdo");

Remover um item no início do array

Nome do array.shift();

Remover um item mas manter o índice

delete nome do array(índice);



ARRAY

Percorrer uma parte do array

Nome do array.slice(início,fim);

Copiar uma parte do array

Variável = nome do array.slice(início,fim);

Copiar todo o array por referência

Novo array = array a ser copiado - O que faz em um altera o outro

Copiar todo o array usando spread operator(...)

Novo array = [... array a ser copiado]- O que faz em um não altera o outro



ARRAY

Tornando um array em uma string

Nome da variável = Nome do array.join(' ');

Concatenando arrays

Nome da variável = nome do array1.concat(nome do array2);

Concatenando arrays usando spread operator(...)

Nome da variável = [...nome do array1, ...nome do array2, ...nome do array3];



Referências

<https://developer.mozilla.org/pt-BR/docs/Aprender/JavaScript>

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/String

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Math

https://developer.mozilla.org/pt-BR/docs/Web/JavaScript/Reference/Global_Objects/Array

https://developer.mozilla.org/pt-BR/docs/DOM/Referencia_do_DOM

A large, stylized yellow logo consisting of the letters 'J' and 'S' in a bold, sans-serif font, representing JavaScript.