JavaScript

O elemento <script>

- Atributos opcionais do <script>:
  - async (somente para arquivos de scripts externos): Indica que o script deve ser carregado, porém ele não deve impedir que outros recursos da página também sejam carregados.
  - charset: a codificação utilizada no código indicado pelo atributo src.
  - defer (somente para arquivos de scripts externos): Indica que a execução do script pode ser seguramente adiada até depois que o documento for completamente analisado e exibido.
  - language: indicava a linguagem a utilizar, porém hoje em dia está Obsoleta.
  - src: indica o arquivo externo que contém o código.
  - type: substitui o atributo language. Pouco utilizado. Geralmente com valor “text/javascript”.

Estruturas

- SE
  - if (condição) código else código
- Repetição
  - do {

        código

    } while (expressão);
  - while (expressão) código
  - for (inicialização; expressão; incremento) código

Estruturas

- Repetição
  - for (property in expression) statement
  - Exemplo:

```
var teste = new Array(1,2,3,4,5);
for (var i in teste) alert( i );
```

- with () { }

```
x = Math.cos(3 * Math.PI) + Math.sin(Math.LN10)
y = Math.tan(14 * Math.E)
Por:
with (Math) {
   x = cos(3 * PI) + sin (LN10)
   y = tan(14 * E)
}
```

Estruturas

- Switch

```
switch ( expressão )  {
    case value: código
      break;
    case value: código
      break;
    case value: código
      break;
    case value: código
      break;
    default: código
}
```

Estruturas

- Funções

```
function Nome(arg0, arg1,…,argN) {
        código
        return
}
```

Objetos dinâmicos

- Criação de objetos dinâmicos com atributos

```
var pessoa = new Object();
pessoa.nome = "Jeiks";
pessoa.idade = 27;
alert( “O ”+
        pessoa.nome+
        “ possui ” +
        pessoa.idade );
```

Objetos dinâmicos

- Outras formas:

```
var pessoa = {
        nome : "Jeiks",
        idade : 27,
        5: true,
        2: false
    };
pessoa[5]
pessoa[2]
pessoa.nome
pessoa.idade
```

Arrays

- Criação de Arrays:
  - Sem tamanho ou dados definidos:

    var lista = new Array()

    ou

    var lista = [ ]
  - Com tamanho definido:

    var lista = new Array(5)

    ou

    var lista = [,,,,,]
  - Com dados definidos:

    var lista = new Array(“Jeiks”, 27, “Jacson”)

    ou

    var lista = [“Jeiks”, 27, “Jacson”]

Operações com Arrays

var lista = [“um”, “dois”, “três”]

- Tamanho:

  lista.length

- Concatenação de itens:

  lista.join(",") // um,dois,três

  lista.join("||") // um||dois||três

- Inserir itens com push:

  lista.push(“quatro”)

- Remover itens com pop:

  lista.push() // lista = [“um”, “dois”, “três”] – o retorno foi: “quatro”

.

Operações com Array

- Remover itens com shift:

  lista.shift() // lista = [“dois”, “três”] – o retorno foi: “um”

- Adicionar itens com unshift:

  lista.unshift(“um”) // lista = [“um”,“dois”, “três”]

- Reorganizando os valores da lista:

  lista.reverse() // lista = [“três”,“dois”,“um”]

  lista.sort() // lista = [“dois”, “três”, “um”]

- Concatenando listas:

  lista = [1,2,3]

  lista2 = [4,5,6]

  lista = lista.concat(lista2) // lista = [1,2,3,4,5,6]

- Localização:

  lista.indexOf(1) ; lista.lastIndexOf(1);

Operações com Array

- Operações de Iteratividade:

```
var numeros = [1,2,3,4,5,4,3,2,1];
var resultado = numeros.every( function(item, index, array){
      return (item > 2);
     });
alert(resultado);  // false, pois nem todos os números são maiores que 2

var resultado = numeros.some(function(item, index, array){
      return (item > 2);
});

alert(resultado);  // true, pois existe algum item maior que 2
```

Operações com Array

- Operações de Iteratividade:

```
var numeros = [1,2,3,4,5,4,3,2,1];
var resultado = numeros.filter(
        function(item, index, array){
          return (item > 2);
        });
alert(resultado);   // [3,4,5,4,3]
// ------------------------------------------------------------------//
var resultado = numeros.map(function(item, index, array){
      return item * 2;
});
alert(resultado);   //[2,4,6,8,10,8,6,4,2]
```

Operações com Array

- Operações de Iteratividade:
  - forEach

```
var numeros = [1,2,3,4];
numeros.forEach(function(item, index, array){
    //fazer algo com cada número
    alert( 'Item da posição ' + index +
            ': ' + item +'. Do vetor:' + array);
});
```

+ operações sobre funções

- Funções como valor:

```
function executaFuncao(funcao, parametro) {
    funcao(parametro);
}
function soma10( num ) {
    alert( num + 10 );
}
executaFuncao( soma10, 20 );
```

+ operações sobre funções

- Retorno de função:

```
function criaSoma10() {
    return function soma10( num ) {
      alert( num + 10 );
    }
}
temp = criaSoma10();
temp( 30 );
```

Operações com Strings

- Declaração:

```
var stringObject = new String("hello world");
var stringValue = "hello world";
```

- Métodos com caracteres:

```
var stringValue = "hello world";
alert(stringValue.charAt(1));      // "e"
    var stringValue = "hello world";
    alert(stringValue.charCodeAt(1));   // mostra "101"
var stringValue = "hello world";
alert(stringValue[1]);     //"e"
    var stringValue = "hello ";        var result = stringValue.concat("world","!");
    alert(result);            //"hello world!"
    alert(stringValue);       //"hello"
```

Operações com Strings

- Métodos com caracteres:

```
var stringValue = "hello world";
alert(stringValue.slice(3));            //"lo world"
alert(stringValue.substring(3));        //"lo world"
alert(stringValue.substr(3));           //"lo world"
alert(stringValue.slice(3, 7));         //"lo w"
alert(stringValue.substring(3,7));      //"lo w"
alert(stringValue.substr(3, 7));        //"lo worl"
    var stringValue = "    hello world    ";
    var trimmedStringValue = stringValue.trim();
    alert(stringValue);                 // "    hello world    "
    alert(trimmedStringValue);          // "hello world"
```

Operações com Strings

- Métodos com caracteres:

```
var stringValue = "hello world";
alert(stringValue.toLocaleUpperCase());    //"HELLO WORLD"
alert(stringValue.toUpperCase());          //"HELLO WORLD"
alert(stringValue.toLocaleLowerCase());    //"hello world"
alert(stringValue.toLowerCase());          //"hello world"

alert(stringValue.replace('l','K'));       //"jeKKo worKd"
alert(stringValue.split(' '));             // ['hello', 'world']
```

Objeto embutido Math

- O objeto Math possui 8 constantes:
  - **E**: constante do número de Euler. (2,718281828459045);
  - **LN2**: constante com o resultado do logaritmo natural na base 2. (0,6931471805599453);
  - **LN10**: constante com o resultado do logaritmo natural na base 10. (2,302585092994046);
  - **LOG2E**: constante com o resultado do logaritmo na base 2 do número de Euler. (1,4426950408889634);
  - **LOG10E**: constante com o resultado do logaritmo na base 10 do número de Euler. (0,4342944819032518);
  - **PI**: constante do pi (Π). (3,141592653589793);
  - **SQRT1_2**: constante com o resultado da raíz quadrada de meio. (√½ ≈ 0,7071067811865476);
  - **SQRT2**: constante com o resultado da raíz quadrada de 2 (√2 ≈ 1,4142135623730951);
- Para utilizar, basta acessar o elemento. Exemplos:
  - Math.E;
  - Math.LN2;

Objeto embutido Math

- Raízes e Potências
  - Método sqrt(): raiz quadrada.
    - Ex.: var1 = sqrt(4) é o mesmo que $\sqrt{4}$.
  - Método pow(): o valor da potência.
    - Ex.: var1 = pow(10, 3) é o mesmo que 10^3.
- Arredondamentos
  - Método round(): arredonda um número para o inteiro mais próximo.
  - Exemplos: o número 3.3 arredondado será 3. O número 3.8 arredondado será 4.
  - Método floor(): arredonda um número para o inteiro mais baixo;
  - Método ceil(): arredonda um número para o inteiro mais alto.
  - Método abs(): remove apenas a parte fracionada

Objeto embutido Math

- Trigonometria
  - sin(): retorna o valor de seno;
  - cos(): retorna o valor de cosseno;
  - tan(): retorna o valor da tangente;
  - asin(): retorna o valor do arco seno;
  - acos(): retorna o valor do arco cosseno;
  - atan(): retorna o valor do arco tangente;
- Maior e Menor
  - Método min(valor1, valor2): retorna o menor valor entre os parâmetros passados.
  - Método max(valor1, valor2): retorna o maior valor entre os parâmetros passados.

Objeto embutido Math

- Número Randômico
  - Método random(): retorna um número entre 0 e 1.
  - Exemplo:

```
var valor = Math.random() * 10;
valor = Math.round( valor );
```

Exercício

- Desenvolver funções e código em JavaScript para:
  1. Criar dinamicamente uma tabela com a quantidade de linhas especificada pelo usuário. Cada linha da tabela deverá conter um campo *input* do tipo *text*. A identificação de cada campo *input* deverá ser armazenada em um *array* global.
  2. Criar um *array* com todas as strings digitadas e apresentá-lo para o usuário.
  3. Criar uma string única com todas as strings digitadas e apresentá-la ao usuário.
  4. Fornecer ao usuário uma forma de pesquisar se determinada *string* foi digitada.
  5. Trocar um caractere por outro (fornecido pelo usuário) entre todas as strings digitadas.