



JAVASCRIPT

Fundamentos do Desenvolvimento Web

Leonardo Sommariva

JavaScript

- JavaScript é uma linguagem de programação interpretada (FLANAGAN e FERGUSON, 2002);
- Originalmente implementada como parte dos navegadores web para que scripts pudessem ser executados do lado do cliente;
- Dessa forma permite interação com o usuário sem a necessidade deste script passar pelo servidor;
- Realizando comunicação assíncrona e alterando o conteúdo do documento exibido.

JavaScript – Breve Histórico

- Foi desenvolvido em 1995 por Brendan Eich, da *Netscape*;
- A Microsoft desenvolveu um dialeto compatível chamada JScript e foi incluída no Internet Explorer 3.0;
- Javascript e JScript são tão similares que os dois termos são comumente usados de forma intercambiável.

JavaScript

- **Compatibilidade:** Suportado pelos principais navegadores;



JavaScript – Principais características

- Oferece tipagem dinâmica: tipo de variáveis não são definidos;
- É interpretada;
- Oferece suporte a expressões regulares;
- Suporta elementos de sintaxe de programação estruturada, como o C (por exemplo, *if*, *while*, *switch*)

JavaScript – Principais características

- Baseada em objetos;
- Objetos JavaScript são arrays associativos;
- Os nomes da propriedade de um objeto são strings:
`obj.x = 10` e `obj["x"] = 10`
- são equivalentes, o ponto neste exemplo é apenas sintático.

JavaScript – Principais características

- Avaliação em tempo de execução: JavaScript inclui a função ***eval*** que consegue executar em tempo de execução comandos da linguagem que estejam escritos em uma string;
- Exemplo: `eval("x * y");`
- JavaScript não apresenta suporte a programação multi-thread.

JavaScript

- Baseadas em protótipo: JavaScript usa protótipos em vez de classes para o mecanismo herança. É possível simular muitas características de orientação a objetos baseada em classes com protótipos.

JavaScript – Uso em páginas web

- Uso primário de JavaScript é escrever funções que são embarcadas ou incluídas em páginas HTML e que interagem com o Modelo de Objeto de Documentos (DOM) da página.
- **Por exemplo:**
 - Abrir uma nova janela com controle programático sobre seu tamanho, posição e atributos;
 - Validar valores de um formulário para garantir que são aceitáveis antes de serem enviados ao servidor;
 - Mudar imagens à medida que o mouse se movimenta sobre elas.

JavaScript – Uso em páginas web

```
<html lang="pt-BR">
  <head>
    <meta charset="UTF-8" />
    <title>Sommariva</title>
    <script>
      window.onload = function () {
        document.getElementById("hello").addEventListener("click", function () {
          alert("Bem-vindo Alunos!");
        }, false);
      };
    </script>
  </head>
  <body>
    <noscript>Seu navegador não suporta JavaScript ou ele está desabilitado.</noscript>
    <button id="hello">Dizer "Olá"</button>
  </body>
</html>
```

JavaScript – Exemplo de Scripts

Script Simples

```
<script>
    // Mostra um alerta de Confirmar e Cancelar.
    if (confirm('Escolha "Ok" ou "Cancelar" para ver a mensagem correspondente.')) {
        alert('Você apertou o botão "OK"'); // mostra um alerta para resposta "OK"
    } else {
        alert('Você apertou o botão "Cancelar"'); // mostra um alerta para resposta "Cancelar"
    }
};
</script>
```

Comentários

```
<script>
    // Mostra um alerta de Confirmar e Cancelar.
    // Este comentário ocupa uma única linha

    /* Já este comentário
    é mais longo e utiliza
    várias linhas */
</script>
```

JavaScript – Exemplos de Scripts

Função

```
<script>
    function nomeDaFuncao( /*parâmetros*/ ) {
        /* código que será executado */

        return/*Valor retornado*/;
    }
</script>
```

Hierarquia do Objetos

```
<script>
    //Construtor
    function Exemplo() {
        this.propriedade = 'Isso é uma propriedade.',
        this.metodo = function() {
            return 'Isso é um método';
        }
    }

    var objeto = new Exemplo(); //Instância do construtor "Exemplo"

    //Alerta os respectivos textos na tela
    alert(objeto.propriedade),
    alert(objeto.metodo());
</script>
```

JavaScript – Exemplos de Scripts

- Algoritmo números perfeitos

```
<script>
function perfectNumbers(max) {
    var i, j, k,
        perfects = [];

    for (i = 0; i++ < max; ) {
        for (j = k = 0; ++j < i; ) {
            if (i % j === 0) {
                k += j;
            }

            if ((j + 1) === i && k === i) {
                perfects.push(k);
            }
        }
    }

    return perfects.join(', ');
}

alert('Números perfeitos entre 0 e 5000: ' + perfectNumbers(5000));
</script>
```

JavaScript - Variáveis

- As variáveis devem ser criadas seguindo algumas regras:
 - Devem começar por uma letra, \$ ou _
 - Após a primeira letra, podem conter números;
 - Por convenção, começam com letra minúscula;
 - São **case-sensitive**.

JavaScript - Variáveis

```
var name = "Leonardo";
```

```
var top10 = [1,2,3,4,5,6,7,8,9, 10];
```

```
var $scope = {title: "JavaScript", version: "1.8.5"};
```

```
var _person = "Leonardo";
```

```
var numberCode = 789654;
```

```
var a = 10;
```

```
var b = 9;
```

JavaScript – Criando um number

```
var nota = 10;
```

```
nota.toExponential(2); // 100
```

```
nota.toFixed(2); //10.00
```

```
nota.toPrecision(1); //1e+1
```

```
nota.toString(); // “10”
```

```
nota.valueOf(); // 10
```


JavaScript – Fique atento as exceções

- $3 / 0 = \text{Infinity}$
- `"String" * 10 = NaN`

E assim por diante....

JavaScript – Math API

- Similiar a classe Math da linguagem Java
- abs – valor absoluto do número;
- floor – valor inteiro do número;
- log- logaritmo natural do número (base E);
- min – Retorna o menor número;
- max – Retorna o maior número;
- pow – potência do número;
- random – Gera um número randômico;
- round – Arredonda o número para o inteiro mais próximo;
- sin – Seno do número;
- sqrt – Raíz quadrada do número;
- tan – tangente do número

JavaScript - Strings

- Uma String é composta por uma sequência de zero ou mais caracteres com as seguintes características:
- São imutáveis;
- Podem ser declaradas com aspas simples ou duplas

JavaScript – Strings API

- `charAt` – Retorna o char da posição;
- `charCodeAt` – Retorna o código do char da posição;
- `concat` – Concatena duas strings;
- `indexOf` – Retorna o índice da última ocorrência do char;
- `lastIndexOf` – Retorna o índice da ultima ocorrência do char;
- `match` – Retorna um array resultante da busca com `RegExp`;
- `replace` – Substitui parte da String por outra;
- `search` – retorna a posição da String ou `RegExp`;
- `slice` – Extrai parte da String, retornando uma nova;
- `split` – Divide a String com base na expressão regular informada;
- `substring` – Extrai a parte da String desejada;
- `toLowerCase` – Gera uma nova String em letra minúscula;
- `toUpperCase` – Gera uma nova String em letra maiúscula;
- `trim` – remove os espaços em branco do início e fim da String;
- `valueOf` – Retorna o valor primitivo da String

JavaScript - Boolean

```
var isento = true;
```

```
var estudante = false;
```

```
isento.toString();// "true"
```

```
estudante.valueOf();// false
```

JavaScript - Function

- Uma função é um objeto que contém um bloco de código executável. Ela pode ser atribuída a uma variável, passada como parâmetro ou mesmo retornada de uma outra função.

JavaScript - Function

```
<script>  
    function soma(a,b) {  
        return a+b;  
    }  
</script>
```

somma; // [Function: soma]
soma(2,2); // 4
soma.name; // 'soma'
soma.length; // 2

Referências

- FLANAGAN, David; FERGUSON, Paula. JavaScript: The Definitive Guide. 4th ed. [S.I.]: O'Reilly & Associates, 2002;
- BRANAS, Rodrigo. Desvendando a linguagem JavaScript. 2014.