



# Introdução ao Javascript

Parte 1

Prof. Victor Hugo Lopes



# agenda

Introdução e histórico

Características técnicas

Propriedades, métodos e eventos

Inserção de Scripts em páginas e Sintaxe básica

# introdução

**JavaScript** é uma linguagem de programação interpretada do tipo client-side: todo algoritmo fica disponível no lado cliente da aplicação.

**Java vs JavaScript**: Java é uma linguagem de programação orientada a objetos que foi desenvolvida pela Sun, inicialmente, para projetar aplicações para Internet, bastante poderosa, e com curva de aprendizagem longa. Já o JavaScript é uma linguagem de script, sendo desenvolvido pela empresa Netscape como suporte à interações dinâmicas no navegador.

JavaScript não é nem parente da linguagem Java!!!

Java pode criar programas completamente independentes, ao passo que a função do JavaScript é tão somente operar nas tags HTML.



# introdução

**JavaScript** provê aos programadores web capacidade de construção de web páginas que podem desempenhar algumas tarefas:

- ▮ Ler elementos de documentos e escrever novos elementos e textos em documentos;
- ▮ Manipular ou movimentar textos;
- ▮ Realizar cálculos matemáticos;
- ▮ Reagir a eventos, como um click de um botão;
- ▮ Interpretar a data e hora corrente do sistema;
- ▮ Capturar características dos usuários distintos, como navegador e versão, resolução de tela, etc;
- ▮ Realizar filtragem de dados digitados em formulários, construção de máscaras de validação de dados.

# introdução

Visto que JavaScript foi desenvolvido pela NetScape, seus navegadores a incluem desde a versão 2, mas o NetScape foi descontinuado. A Microsoft criou uma variante chamada Jscript (cópia descarada do JS), incluída no IE na versão 3.

Depois a linguagem foi padronizada pelo ECMA, ISO e W3C as duas empresas buscaram elementos parecidos nos navegadores.

Esta padronização garantiu a atual compatibilidade com os diversos navegadores atuais do mercado.

# Características técnicas

Imperativa e estruturada: Suporta características da linguagem C, onde os comandos são baseados em linguagem imperativa (“Faça isso!”), e podem ser construídos de maneira estruturada:

- Fluxo de interpretação top-down e sequencial;
- Estruturas de seleção simples (IF), composta (IF-ELSE), aninhadas (IF-ELSE-IF ...) e múltipla (SWITCH);
- Estruturas de repetição (FOR, WHILE).

# Características técnicas

Permite a modularização, isto é, a construção de blocos de comandos que podem ser separados em partes independentes para construção da aplicação.

Permite a orientação a objetos: é quase inteiramente baseada em objetos, onde os objetos podem ser trabalhados como arrays associativos.

As propriedades e seus valores podem ser adicionadas, alteradas ou removidas em tempo de execução.

# Características técnicas

**Segurança:** JavaScript representa uma potencialidade crítica para programadores maliciosos rodarem em clientes web. Navegadores são projetados para conterem este risco, no qual tentam impedir que scripts do navegador realizem tarefas que não são próprias do navegador, impedindo a criação de arquivos locais, ou acesso à pastas e arquivos.

Além disso, os navegadores limitam pela regra da mesma origem, onde scripts de um web site não possam ter acesso a informações como nomes de usuário, senhas e cookies enviados de outro site.

Porém, estas regras podem possuir bugs identificados por hackers!





# DOM

JavaScript opera nos elementos existentes em uma web página mediante uma interface que permite completa condição para criar, modificar e remover elementos na página. Esta interface é o DOM (Document Object Model).

- ▮ Cada objeto possui certas propriedades que o descrevem. Um parágrafo, por exemplo, possui propriedades que indicam o tamanho da fonte, a cor do texto, o alinhamento dos caracteres, entre outros.
- ▮ Estes objetos também possuem métodos, que descreve uma ação do objeto. Um botão em uma página pode ter um método que realiza a troca da sua cor de fundo, por exemplo.
- ▮ Por fim, estes objetos também estão expostos a eventos, que são as ações realizadas pelos usuários ou demais objetos da aplicação. O parágrafo possui um evento de passar o mouse sobre, um botão possui o evento de ser clicado.

# DOM

A grande maioria dos navegadores implementam o DOM.

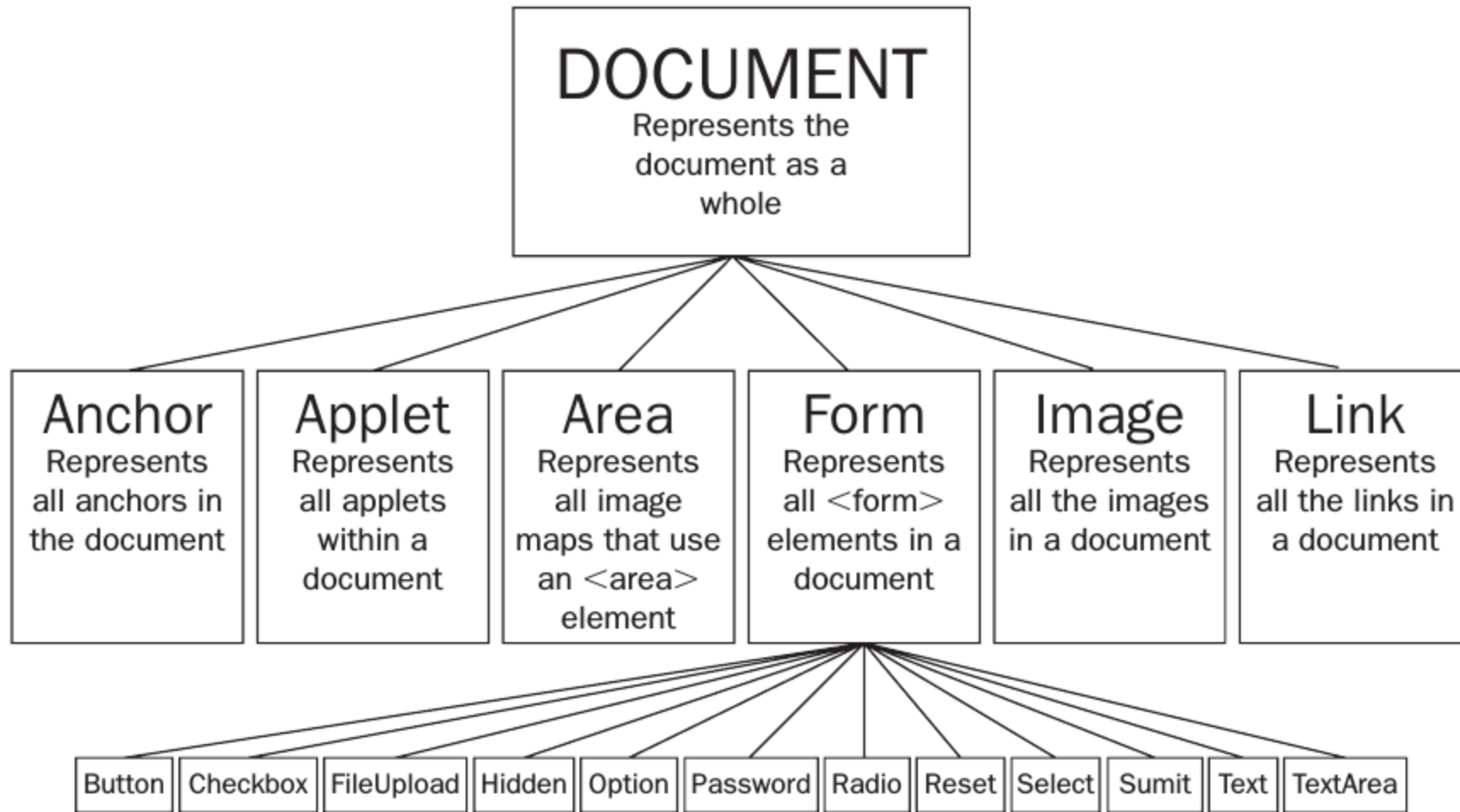
Em resumo, o DOM descreve como você pode:

- Capturar uma propriedade específica da página ou de um objeto específico;
- Executar métodos que promovam ações em objetos ou no documento todo;
- Reagir a eventos de usuários e objetos.

O principal objeto do DOM é o object.

Pense no DOM como uma interface entre o browser e a linguagem de programação, assim como um controle remoto de TV, que facilita a ação de trocar o canal, ou alterar o volume do áudio.

# DOM



# Sintaxe

JS pode ser utilizado na forma embebida no documento HTML ou importado de um arquivo .js, isto é, pode ser utilizado na forma interna ou externa.

Em qualquer uma das duas formas, só será válido se a interpretação de JS estiver habilitada no navegador do cliente.

Para se utilizar a forma interna, utiliza-se o marcador `<script>`, sendo recomendável a utilização da propriedade `type` para indicar que se trata de um código JavaScript:

```
<script type="text/javascript"></script>
```



# Sintaxe

Ex.:

```
<html>
  <body>
    <p>
      <script type="text/javascript">
        document.write("Escrito pelo Javascript");
      </script>
    </p>
  </body>
</html>
```

# Sintaxe

Ex.:

```
<html>
  <body>
    <p>
      <script type="text/javascript">
        document.write("Escrito pelo Javascript");
      </script>
    </p>
  </body>
</html>
```

JS é case-sensitive,  
i.e., document é  
diferente de Document

Abertura do script

Comando javascript: usando o DOM  
Para acessar o objeto document.  
O ponto (.) permite acessar uma  
propriedade ou método do objeto  
Document. O método Write  
cria uma nova linha no  
documento html, exatamente onde  
o script foi embutido. Com  
o texto informado.

# Sintaxe

Onde colocar os códigos JS?

Scripts JS podem ser declarados dentro da seção `<head>`, ou no meio do corpo, como no exemplo. Mais qual a diferença entre as duas abordagens?

Ao colocar dentro do corpo, o script é rodado automaticamente durante o carregamento do html, isto é, assim que a tag for renderizada pelo navegador.

Já na abordagem de colocar dentro da seção `<head>`, o script será forçado a aguardar todo o carregamento da página, para depois ser executado. Esta abordagem é válida para métodos que precisam ser chamados à partir de interações dos usuários, ou deve aguardar o completo carregamento dos recursos.

# Sintaxe

## JS externo

O exemplo visto é de um JS interno. A outra forma de trabalho é manter o script JS em um arquivo externo, de extensão .js, e importá-lo no arquivo que for utilizá-lo.

Esta abordagem favorece a reusabilidade de código, principalmente se utilizado com o paradigma da Orientação a Objetos.

Nesta abordagem, o carregamento do arquivo externo é parecido com a importação de um css externo. Deve ser declarada na seção <head>.

```
<script type="JavaScript" src="scripts/validation.js"></script>
```



# Sintaxe

JS externo

Quando utilizar script externo:

- Se seu script for utilizado por mais de uma página, e não há necessidade de reescrever várias vezes o mesmo script;
- Se for cômodo alterar um único script e gerar impactos em outros arquivos;
- Construir páginas mais limpas (mesma ideia do css externo!).

# Sintaxe

## Comentários em JS

Basicamente há duas formas de construção de comentários em JS: comentários em única linha e linhas múltiplas.

```
<script type="text/javascript">  
    document.write("My first JavaScript") // comment goes  
here  
</script>
```

```
/* This whole section is commented  
out so it is not treated as a part of  
the script. */
```

# Sintaxe

Normas básicas da sintaxe

Cada linha deve ser terminada com “;”

JavaScript difere minúsculas de maiúsculas

{ } são utilizadas para delimitação de blocos

# Sintaxe

## Comentários em JS

Basicamente há duas formas de construção de comentários em JS: comentários em única linha e linhas múltiplas.

```
<script type="text/javascript">  
    document.write("My first JavaScript") // comment goes  
here  
</script>
```

```
/* This whole section is commented  
out so it is not treated as a part of  
the script. */
```



# Sintaxe

## Variáveis e Operadores

Assim com em outras linguagens, as variáveis são valores que são guardados temporariamente na memória para serem utilizadas pelo restante do programa. Elas precisam de um nome (identificador) para ser referenciada.

O identificador deve ser criado seguindo as regras já conhecidas.

JS não é fortemente tipada, e variáveis podem ser declaradas em qualquer ponto do script (antes de ser utilizada, claro!!), e não possuem definição estática de tipo.

Em sua declaração, pode ser precedida da palavra `var`.

# Sintaxe

## Variáveis e Operadores

### Operadores matemáticos:

+ : adição

- : subtração

\* : multiplicação

/ : divisão

%: resto da divisão

++ : incremento

-- : decremento

### Operadores relacionais:

> : maior que

< : menor que

>= : maior ou igual a

<= : menor ou igual a

== : igual a

!= : diferente de

+= : soma e alocação

-= : subtração e alocação

\*= : produto e alocação

/= : divisão e alocação

### Operadores lógicos

&&: and

|| : or

! : not

# Sintaxe

Variáveis e Operadores

Ex.:

```
<script type="text/javascript">  
    var a1 = 10;  
    a2 = 10;  
  
    document.write(a1+a2+" - ");  
    a2+=100;  
    document.write(a2);  
</script>
```

# JavaScript

## Exercitando:

1) É necessário que as variáveis tenham a definição de um tipo antes de serem utilizadas em um programa JavaScript. Certo ou Errado? - (CESPE 2012, TCE-ES, auditor de TI)

2) HTML, DHTML, JavaScript e PHP são linguagens utilizadas no desenvolvimento de sites da World Wide Web. A seu respeito é correto afirmar que: (FCC 2012, TJ-RJ, análise de sistemas)

- a) o código de uma aplicação JavaScript deve ser interpretado pelo servidor HTTP ao passo que o código de uma aplicação PHP deve ser interpretado pelo cliente HTTP.
- b) o código de uma aplicação JavaScript deve ser interpretado pelo cliente HTTP ao passo que o código de uma aplicação PHP deve ser interpretado pelo servidor HTTP.
- c) tanto o código de uma aplicação JavaScript como o código de uma aplicação PHP devem ser executados pelo cliente HTTP.
- d) tanto o código de uma aplicação JavaScript como o código de uma aplicação PHP devem ser executados pelo servidor HTTP.
- e) o código de uma página HTML deve ser interpretado pelo cliente HTTP ao passo que o código de uma página DHTML deve ser interpretado pelo servidor HTTP.

3) No que diz respeito aos operadores em JavaScript, sejam as variáveis do mesmo tipo  $a=3$  e  $b=5$ . As operações  $a != b$  e  $a != b \ \&\& \ b == 5$  retornarão, respectivamente, os seguintes valores: (CONSULPLAN 2012, TSE, programador)

- a) verdadeiro e falso.
- b) falso e verdadeiro.