

Introdução

Javascript

O que é JavaScript?

- O Javascript é uma linguagem de programação de script.
- Javascript é incorporado a páginas Web, permitindo incrementar a apresentação e interatividade das mesmas.
- Esses scripts vão ser **interpretados** pelo browser, sem uso do servidor.
- Javascript foi desenvolvido inicialmente pela Netscape e a essa altura chamava-se LiveScript.
- Foi adotado no fim do ano de 1995, pela firma Sun (que também desenvolveu o Java).
- Tomou assim o seu nome atual: **Javascript**.

Javascript

Javascript X Java

- É importante saber que o Javascript é completamente diferente do Java.
- Mesmo que os dois sejam utilizados para criar programas em páginas Web.
- Mesmo que os dois usam o mesmo termo Java, temos aí duas ferramentas bem diferentes.

Javascript

Javascript X Java

■ Javascript:

- Código interpretado pelo browser no momento da execução.
- Códigos de programação simples mas para aplicações limitadas.
- Confidencialidade do código é nula (Código é visível).

■ Java:

- O código é compilado antes da sua execução.
- Linguagem de programação muito mais complexa mas mais poderosa.
- Segurança (Código compilado).

A Linguagem

Javascript

Algumas Considerações

- Os scripts podem estar localizadas em qualquer parte do código HTML, a única restrição é que devem começar e terminar com a declaração de Scripts.
- Por convenção costuma-se colocar todas as funções no início do documento (entre as TAGs `<HEAD>` e `</HEAD>`, isso para garantir que o código JavaScript seja carregado antes que o usuário interaja com a Home Page), ou seja, antes do `<BODY>`.

Javascript

Sintaxe

```
<html>  
<head>  
<script language=" Javascript" >  
//declarações  
</script>  
</head>  
<body>  
<!-- código html -->  
</body>  
</html>
```

Javascript

Exemplo

```
1 <html>
2 <head>
3 <title>exemplo javaScript</title>
4 <script language= "javascript" >
5     document.write ("Aqui já é javascript!");
6 </script>
7 </head>
8 <body>
9     <BR />Esta linha está escrita em html
10 </body>
11 </html>
```


Javascript

Falta Suporte?

- Browser não suporta JavaScript?
- Utilize os comentários HTML `<!-- e -->`.
- Exemplo:
`<script>`
`<!--`
`document.write (" Aqui já é JavaScript");`
`-->`
`</script>`

Javascript

Comentários

```
1 <html><head>
2 <title>JavaScript</title>
3 <script>
4     document.write ("Aqui já é JavaScript");
5     //comentário de uma linha
6
7     /*
8         comentário de multiplas linhas
9         comentário de multiplas linhas
10    */
11 </script>
12 </head>
13 <body>
14     <BR />Aqui é HTML
15 </body>
16 </html>
```

Conceitos Preliminares

Javascript

Algumas Conceitos OO - Classes

- Uma classe é um tipo definido pelo usuário que contém o molde, a especificação para os objetos. A classe envolve, associa, funções e dados, controlando o acesso a estes, defini-la implica em especificar os seus atributos (dados) e seus métodos (funções).

Javascript

Algumas Conceitos OO - Classes

- Uma classe categoriza objetos semelhantes.
- Classe é uma descrição de um ou mais objetos com os mesmos atributos e serviços.
- Objetos pertencentes à classe são ditos **INSTÂNCIAS** da classe.
- Instanciação: ato de criar (instanciar) objetos de uma classe.

Javascript

Algumas Conceitos OO - Objetos

- Objeto é qualquer indivíduo, lugar, coisa, evento, tela, relatório ou conceito...
- Ou seja, qualquer elemento que seja **aplicável ao projeto do sistema**.
- Os objetos possuem dados e funcionalidade que definem seus comportamentos.
- Possuem: atributos (estado) e serviços (comportamento).

Javascript

Algumas Conceitos OO - Objetos - Exemplo

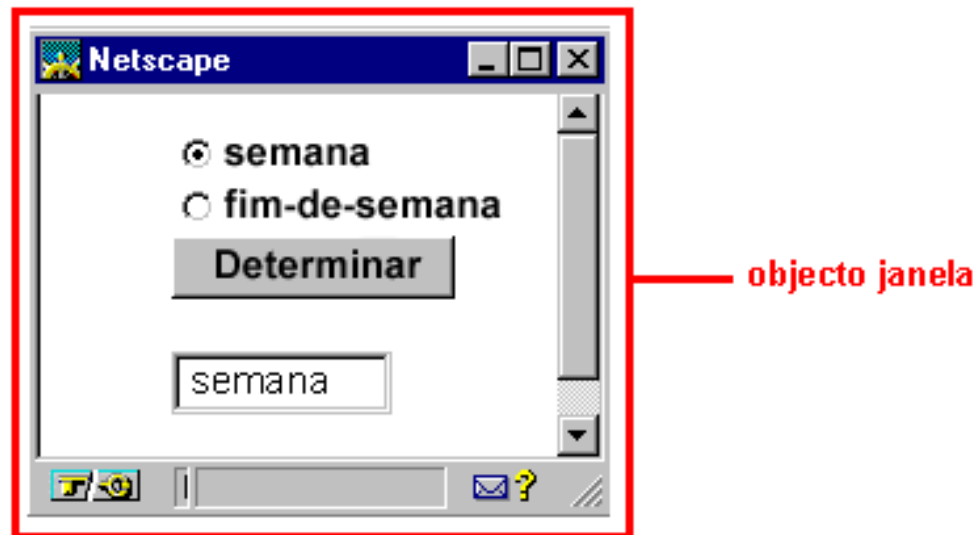
- Carregou-se a seguinte página:



Javascript

Algumas Conceitos OO - Objetos - Exemplo

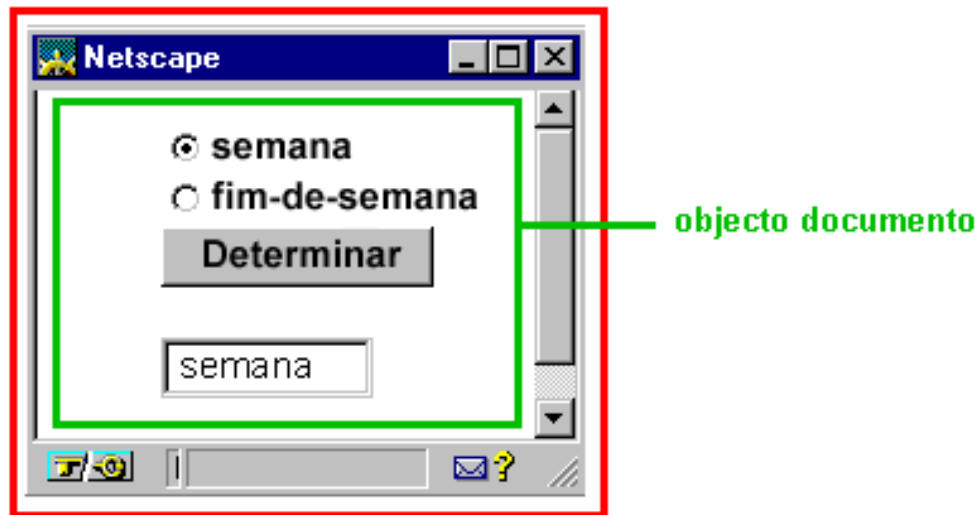
- Esta página aparece em uma janela.
- Esse é o objeto **JANELA**.



Javascript

Algumas Conceitos OO - Objetos - Exemplo

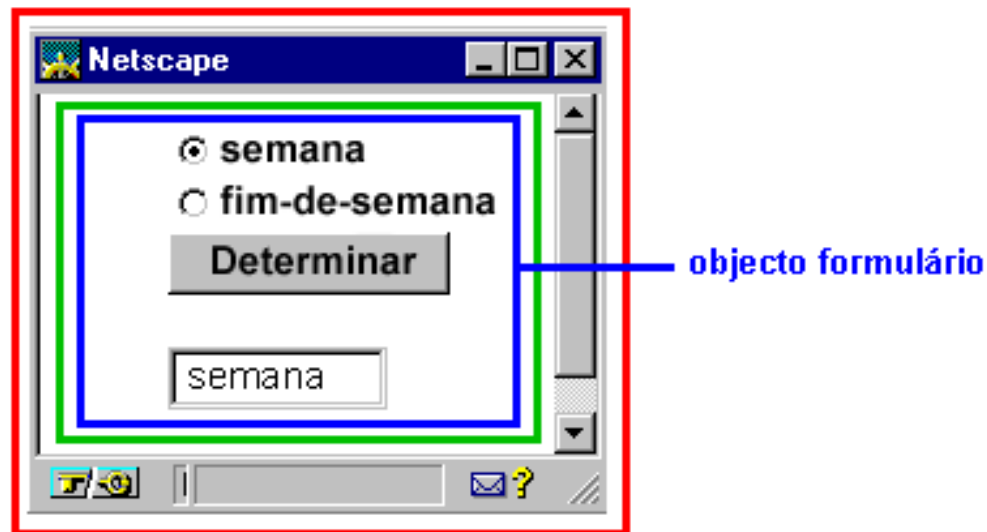
- Nesta janela, há um documento HTML, é o objeto **DOCUMENTO**.
- Isto quer dizer: o objeto **JANELA** contém o objeto **DOCUMENTO**.
- Isso nos traz a noção de hierarquia, que será sempre utilizada em Javascript.



Javascript

Algumas Conceitos OO - Objetos - Exemplo

- Neste documento, temos um formulário.
- É o objeto **FORMULÁRIO**.
- Continuando: o objeto **JANELA** contém um objeto **DOCUMENTO** que contém um objeto **FORMULÁRIO**.



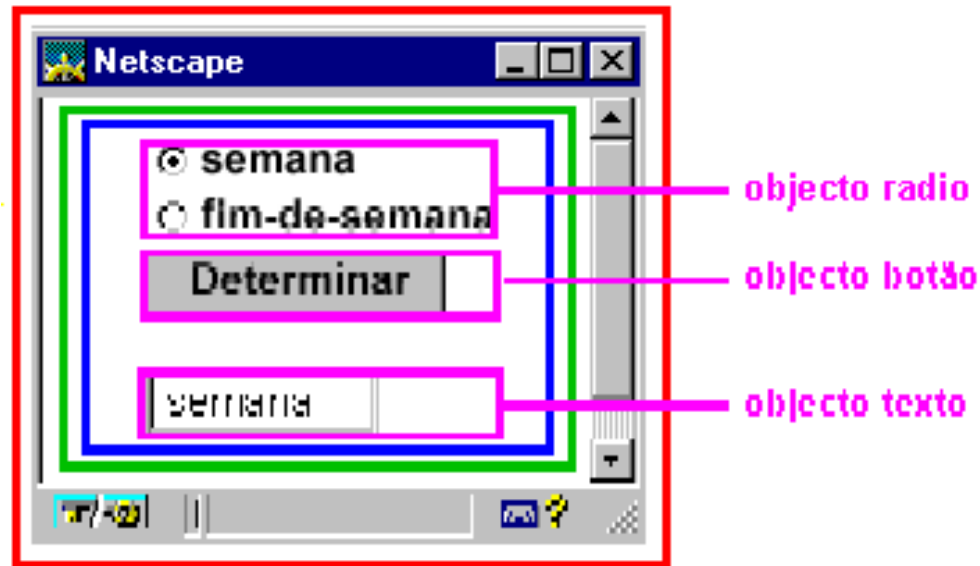
Javascript

Algumas Conceitos OO - Objetos - Exemplo

- Neste formulário encontramos 3 objetos.
- Botões rádio, botão clássico e uma área de texto.
- São: objeto **RADIO**, objeto, **BOTÃO** e objeto **TEXTTO**.
- Em outras palavras: o objeto **JANELA** contém o objeto **DOCUMENTO** que contém o objeto **FORMULÁRIO** que por sua vez contém os objetos **RADIO**, **BOTÃO** e **TEXTTO**.

Javascript

Algumas Conceitos OO - Objetos - Exemplo



Javascript

Algumas Conceitos OO - Objetos - Exemplo

A hierarquia do exemplo é então:

janela	>	documento	>	formulário	>	radio
					>	botão
					>	texto

Javascript

Algumas Conceitos OO - Objetos

- Para acessar um objeto, é necessário dar o caminho completo do objeto começando pelo objeto mais externo até o objeto de referência.
- Seja por exemplo para o botão clássico "Determinar":
`window.document.form.button`.
- Javascript não é propriamente uma linguagem orientada a objetos tal o como C++ ou Java.
- Podemos dizer, entretanto, que Javascript é uma linguagem baseada em objetos.

Métodos

Javascript

Métodos

- As ações de um objeto, que são chamadas de métodos podem ser acessadas e manipuladas através da seguinte sintaxe:
`objeto.metodo();`
- É muito simples identificar um método.
- As ações são verbos no infinitivo.
- Outra forma de identificar um método, são os parênteses que o acompanham.

Javascript

Métodos - write()

- Agora já estamos aptos a entender o comando **document.write()**.
- Trata-se do método **write** do objeto **document**.
- Ou seja, representa a capacidade que o objeto document tem de escrever na tela.

Javascript

Métodos - write() - Exemplo

```
1 <html>
2 <head>
3 <title>Aplicacao JavaScript</title>
4 <script language="Javascript">
5 document.write( "<center><font color='#FF0000'>Alô Mundo!!! </font></center><BR />" );
6 document.write( "<center><font color=\"#FF0000\">Usando aspas duplas </font></center><BR />"
7 // O caracter de "\"" indica ao JavaScript que as aspas fazem parte do texto
8 </script>
9 </head>
10 <body>
11 Aqui é HTML
12 </body>
13 </html>
```

Javascript

Métodos - alert()

- Janela de alerta.
- As janelas de alerta são usadas para dar ao usuário informações importantes relativas à página que ele está visitando, ou sobre ações por ele praticadas.
- Sintaxe: **alert (“mensagem ”)**.

Javascript

Métodos - alert() - Exemplo

```
1 <html>
2 <head>
3 <title>usando janelas de alerta</title>
4 <script language = "javascript">
5 alert ("Seja bem vindo. esta página foi escrita utilizando javascript.");
6 </script>
7 </head>
8 <body>
9 </form>
10 </body>
11 </html>
```

Variáveis

Javascript

Variáveis

- Sintaxe: `var <nome da variavel>.`
- Podemos declarar mais de uma variável em uma mesma instrução **var**, separando-as por vírgula.
- Exemplo:
`var titulo;`
`var titulo, nome, autor;`

Javascript

Variáveis

- Procure usar variáveis de nomes curtos e que lembrem seu significado.
- Adote um critério para criação de variáveis.
- Os nomes das variáveis não podem conter espaços em branco ou caracteres especiais.

Javascript

Exemplos

```
1 <HTML>
2 <HEAD>
3 <TITLE>Variáveis</TITLE>
4 <script language = "javascript">
5 var a, b;
6 a = 5;
7 b= "meu valor é alfanumérico, pois estou entre aspas";
8 document.write ("o valor da variável a é: " + a);
9 document.write ("<p>" + "o valor da variável b é: " + b + "</p>");
0 document.write ("<p> o valor da variável b impresso em  negrito e na  cor vermelha é: " +
1 "<b><font color='red'>" + b + "</b></p>");
2 </script>
3 </SCRIPT>
4 </HEAD>
5 <BODY>
6 </BODY>
7 </HTML>
```


Javascript

Exemplos

```
1 <html>
2 <head>
3 <script language = "javascript">
4 var a, b;
5 a = 5;
6 b= "eu sou o string 6";
7 alert (a);
8 alert("o valor da variável b é: \n " + b);
9 </script>
10 </head>
11 <body>
12 Página usando variáveis para serem exibidas em janelas de alerta.
13 </body>
14 </html>
```

Javascript

Exemplos

- Bônus: método **prompt()**.

```
1 <html>
2 <head>
3 <title>O método prompt</title>
4 <script language="Javascript">
5 var num = prompt( 'Informe o número','');
6 document.write( 'O numero digitado foi ' + num);
7 </script>
8 </head>
9 <body>
10 </body>
11 </html>
```

Tipos de Datos

Javascript

Tipos de Dados

■ Numéricos:

- São números que podem ser tanto inteiros quanto de ponto flutuante.
- Exemplos: 55, 10, 10.45, 2e11.

■ Lógicos:

- São dados que podem receber dois valores: true (verdadeiro) ou false (falso).

■ Strings:

- São cadeias de caracteres.
- Valores strings são valores delimitados por apóstrofo(') ou por aspas(").
- Exemplos: "laranja", "casa123", "1537".

■ Nulo:

- O null é um valor especial e representa um objeto nulo.

Operadores

Javascript

Operador de Concatenação

- O operador de concatenação é representado pelo sinal de adição +.
- O operador + justapõe uma string a uma outra variável.
- Exemplo:

```
var a = "Programação para";  
var b = "Web II";  
document.write(a + b);
```
- Esse código vai imprimir: Programação para Web II.

Javascript

Operadores Aritméticos

- São operadores que manipulam valores numéricos.
- Operador de adição: $+$.
- Operador de subtração: $-$.
- Operador de multiplicação: $*$.
- Operador de divisão: $/$.
- Operador de resto: $\%$.

Javascript

Operadores Aritméticos - Exemplo

```
1 <html>
2 <head>
3 <title>Operadores</title>
4 <script language="Javascript">
5 var a, b, c, d;
6 a = 8;
7 b = 2;
8 c = (a/b);
9 d = (a%b);
10 document.write("O resultado da divisão (a/b) é " + c + "<BR />");
11 document.write("O resto da divisão de (a%b) é " + d);
12 </script>
13 </head>
14 <body>
15 </body>
16 </html>
```


Javascript

Operadores Lógicos

- São operadores que comparam dois valores e retornam duas possíveis saídas:
 - **true** se a comparação for verdadeira.
 - **false** se a comparação for falsa.
 - O valor **true** é equivalente a 1.
 - O valor **false** é equivalente a 0.

Javascript

Operadores Lógicos

- Operador de igualdade: `==`.
- Operador de diferença: `!=`.
- Operador maior: `>`.
- Operador menor: `<`.
- Operador maior ou igual: `>=`.
- Operador menor ou igual: `<=`.
- Operador de conjunção (operador E): `&&`.
- Operador de disjunção (operador OU): `||`.

Javascript

Operadores Lógicos - Exemplo

Exemplo na aula.

Javascript

Operadores de Incremento e Decremento

- Soma (ou diminui) uma unidade de variáveis do tipo inteiras.
- Incremento: `++`.
- Decremento: `--`.
- Exemplo:
 - `var a,b,c;`
 - `a = 5;`
 - `b = a++;` // b recebe o valor 6
 - `c = a--;` // c recebe o valor 4

Conversão para Números

Javascript

Conversão para Números

- O Javascript dispõe de duas funções para conversão de valores alfanuméricos em valores numéricos:
- **parseInt()** - converte um valor alfanumérico em um valor numérico inteiro.
- **parseFloat()** - converte um valor alfanumérico em um valor numérico de ponto flutuante.

Javascript

Conversão para Números - Exemplo

```
1 <html><head>
2 <script language="javascript">
3   var a, inteiro, flutuante;
4   a ="4.25kg";
5   inteiro = parseInt(a);
6   flutuante = parseFloat(a);
7   document.write("O valor da variável a é: " + a);
8   document.write("<p> O valor numérico inteiro da variável a é:  " + inteiro);
9   document.write("<p> O valor numérico decimal da variável a é:  " + flutuante);
10 </script>
11 </head>
12 <body>
13 </body>
14 </html>
```

Javascript

Conversão para Números - Exercício

- Usar o método **prompt()** para solicitar que o usuário entre com dois números.
- Calcular a soma desses dois números;
- Exibir a soma em uma janela de alerta.

Funções para Manipulação de Strings

Javascript

Manipulação de Strings - Conversão de Caixa

- A conversão de caixas de uma string permite converter todo o seu conteúdo em letras maiúsculas ou minúsculas.
- Isto é feito por intermédio de dois métodos específicos de strings.
- **toUpperCase()**: Converte todos os caracteres contidos em uma string em letras maiúsculas.
- **toLowerCase()**: Converte todos os caracteres contidos em uma string em letras minúsculas.

Javascript

Manipulação de Strings - Conversão de Caixa

Sintaxe:

- `string.toUpperCase()`
- `string.toLowerCase()`

```
1 <script language = "javascript">
2 var a, b;
3 a = "Olá pessoal ! ";
4 b = "Sejam bem vindos ao JavaScript";
5 document.write ("<p><b> Usando o método toUpperCase</p></b>");
6 document.write (a.toUpperCase( ));
7 document.write (b.toUpperCase( ));
8 document.write ("<p><b> Usando o método toLowerCase </p></b>");
9 document.write (a.toLowerCase( ));
10 document.write (b.toLowerCase( ));
11 </script>
```

Javascript

Manipulação de Strings - Substrings

- Podemos definir uma substring como sendo parte de uma string.
- No Javascript, uma substring é manipulada pelo método `substr()`.
- Sintaxe: `substr(PosiçãoInicial, NúmeroDeCaracteres)`.
- Exemplo:
 - `Pais = "BRASIL"`.
 - `Pais.substr(0,2)` - retorna BR.
 - `Pais.substr(2,2)` - retorna AS.
 - `Pais.substr(3,1)` - retorna S.

Estruturas de Controle

IF

Javascript

Estruturas de Controle - IF

- É a estrutura SE vista em algoritmos.
- É uma estrutura de controle utilizada para tomar decisões.
- É uma condicional que realiza operações em função de uma expressão.
- Funciona da seguinte maneira: primeiro se avalia uma expressão, se o resultado for positivo realizam-se dentro do bloco ({ }).

Javascript

Estruturas de Controle - IF

- A sintaxe da estrutura IF é a seguinte:
if(condição){
ações a realizar em caso positivo ...
}

Javascript

Estruturas de Controle - IF

```
1 <head>
2   <script language="Javascript">
3     var valor;
4     valor = 5;
5
6     if(valor == 5){
7       document.write("A variavel valor eh igual a " + valor);
8     }
9   </script>
10 </head>
```

Javascript

Estruturas de Controle - IF e ELSE

- A sintaxe da estrutura IF combinada com ELSE é a seguinte:

```
if(condição){  
ações a realizar em caso positivo ...  
}  
else{  
ações a realizar em caso negativo...  
}
```

Javascript

Estruturas de Controle - IF e ELSE

```
1 <head>
2   <script language="Javascript">
3     var a, b;
4     a = 5;
5     b = 10;
6
7     if(a == b){
8       document.write("A variavel a tem o mesmo valor da variavel b");
9     }
10    else{
11      document.write("As variaveis a e b tem valores diferentes");
12    }
13  </script>
14 </head>
```

Javascript

IF e ELSE - Exemplo: Método confirm()

- `window.confirm()`: aguarda confirmação do usuário.

```
1 <head>
2   <script language="Javascript">
3     var confirmacao;
4     confirma = window.confirm("Essa página não pode ser acessada por menores de 18 anos! Você tem mais de 18 anos?");
5
6     if(confirma == true){
7       document.write("Você se declarou como maior de 18 anos e deseja visualizar a página por sua conta.");
8     }
9     else{
10      document.write("Você se declarou menor de 18 anos e não pode visualizar a página");
11    }
12  </script>
13 </head>
```

Javascript

Estruturas de Controle - IF ELSE IF

```
if(condição1){  
ações para condição 1 satisfeita ...  
}  
else if(condição2){  
ações para anteriores não satisfeitas mas condição 2 satisfeita ...  
}  
else if(condição3){  
ações para anteriores não satisfeitas mas condição 3 satisfeita ...  
}  
else{  
nenhuma das condições anteriores foram satisfeitas.  
}
```

Javascript

Estruturas de Controle - IF ELSE IF

```
1 <script language="Javascript">
2   var valor;
3   valor = 5;
4
5   if(valor == 3){
6       document.write("A variavel valor eh igual a 3");
7   }
8   else if(valor == 4){
9       document.write("A variavel valor eh igual a 4");
10  }
11  else if(valor == 5){
12      document.write("A variavel valor eh igual a 5");
13  }
14  else{
15      document.write("A variavel valor nao eh igual a 3 nem a 4 e nem a 5");
16  }
17 </script>
```

Javascript

Estruturas de Controle - IF: Exercícios

1. Use o método `prompt()` para solicitar ao usuário sua idade. Verificar se o usuário é maior ou menor de idade e imprimir isso na tela.
2. Use o método `prompt()` para solicitar ao usuário para entrar com um número. Verificar se o número fornecido pelo usuário é ímpar ou par e imprimir isso na tela.
3. Usando o método `prompt()`, solicite ao usuário para entrar com 2 números. Verifique se o 1º número é maior, menor ou igual ao segundo e imprima isso na tela.

FOR

Javascript

Estruturas de Repetição - FOR

- A estrutura de repetição **for** nos permite executar um bloco de instruções diversas vezes.
- O número de vezes que o bloco será executado é controlado por uma variável.
- A sintaxe da estrutura **for** é a seguinte:
for(iniciação ; condição ; incremento){
conjunto de instruções ...
}

Javascript

Estruturas de Repetição - FOR

- A estrutura de repetição **for** nos permite executar um bloco de instruções diversas vezes.
- O número de vezes que o bloco será executado é controlado por uma variável.
- A sintaxe da estrutura **for** é a seguinte:

```
for(x = 0 ; x <= 10 ; x++){  
document.write( "x igual a:" + x + "<br>" );  
}
```

Javascript

Estruturas de Repetição - FOR

As três expressões que ficam entre parênteses têm a finalidade de:

- Iniciação: Determinar o valor inicial da variável de controle do laço.
- Condição: Expressão que deve ser **verdadeira** para que o bloco seja executado. Assim que a expressão for **falsa** o bloco não é mais executado.
- Incremento: Passo de alteração da variável de controle do laço.

Javascript

Estruturas de Repetição - FOR

```
1 <script language="Javascript">
2 var i;
3
4 for(i = 0 ; i <= 10 ; i++){
5     document.write("o valor da variavel i eh: " + i + "<br>");
6 }
7 </script>
```

WHILE

Javascript

Estruturas de Repetição - WHILE

- O **while** é a estrutura de repetição mais simples.
- Com o **while** testa-se uma condição e executa-se um bloco de comandos enquanto a condição for **verdadeira**.
- A sintaxe do **while** é a seguinte.

```
while(condição){  
conjunto de instruções ...  
}
```

Javascript

Estruturas de Repetição - WHILE

```
1 <script language="Javascript">
2 var i;
3 i = 0;
4 while(i <= 10){
5     document.write("o valor da variavel i eh: " + i + "<br>");
6     i++;
7 }
8 </script>
```

DO..WHILE

Javascript

Estruturas de Repetição - DO..WHILE

- A estrutura de repetição **do..while** funciona de maneira bastante semelhante ao **while**.
- A diferença básica é que a expressão é testada ao final do bloco de comandos.
- Ou seja, primeiro executa-se o bloco de comandos e depois verifica-se se a condição é **verdadeira**.

Javascript

Estruturas de Repetição - DO..WHILE

- Se a condição for **verdadeira**, o bloco de comandos é executado novamente.
- Se a condição for **falsa**, o bloco de comandos não é executado novamente.
- A sintaxe do **do..while** é a seguinte:
do{ conjunto de instruções ... }
while(condição);

Javascript

Estruturas de Repetição - DO..WHILE

```
1 <script language="Javascript">
2 var i;
3 i = 0;
4 do{
5     document.write("o valor da variavel i eh: " + i + "<br>");
6     i++;
7 }
8 while(i <= 10);
9 </script>
```

Javascript

Estruturas de Repetição - Exercício

- Imprimir todos os **números pares** entre 200 e 300:
 - Utilizando a estrutura de repetição **for**.
 - Utilizando a estrutura de repetição **while**.

SWITCH

Javascript

Estruturas de Controle - SWITCH

- A estrutura de controle **switch** é utilizada para escolher um bloco de comandos entre vários de acordo com o valor de uma variável.
- O **switch** é um substituto do **IF-ELSE-IF** visto anteriormente.
- A sintaxe do **switch** é a seguinte:

```
switch(variável){  
  case constante1:  
    conjunto de instruções;  
    break;  
  case constante2:  
    conjunto de instruções;  
    break;  
  default:  
    conjunto de instruções;
```

Javascript

Estruturas de Controle: SWITCH

```
1 <script language="Javascript">
2 var turno;
3 turno = parseInt(window.prompt("Digite o turno do curso: 1-manha / 2-tarde / 3-noite:", ""));
4 switch(turno){
5   case 1:
6     document.write("Turno da manha.");
7     break;
8   case 2:
9     document.write("Turno da tarde.");
10    break;
11  case 3:
12    document.write("Turno da noite.");
13    break;
14  default:
15    document.write("Turno nao existente.");
16
17 }
18 </script>
```

Javascript

Estruturas de Repetição - Exercício 1

Fazer um algoritmo que:

- Leia dois números e um sinal de operação (+, -, * ou /).
- Exiba numa **janela de alerta** o **resultado** da operação dos números.