



# CURSO DE HTML5, CSS3, JAVASCRIPT

Ministrado por:

- Bruno Teles dos Santos
- Matheus Vieira
- Allan Macedo

by NU△GE  
IT's everywhere.



# Parte 3

JAVASCRIPT

# JAVASCRIPT

É uma linguagem de script que lhe possibilita adicionar um novo nível de interatividade e função às páginas Web.

Um script é uma seqüência de instruções (ou, até mesmo um programa) que são interpretados ou executados por um outro programa e não pelo processador do computador.

HTML5  
CSS3  
JAVASCRIPT



Diferentemente de HTML, se houver erros nos scripts o programa interromperá a execução e poderá até travar o computador. Normalmente você poderá visualizar a mensagem de erro gerado em seu navegador.



# JAVASCRIPT

JavaScript é uma linguagem de terceira geração, o que significa que ela é prima do C, Pascal e BASIC.

Existem semelhanças, mas existem também diferenças fundamentais:

- JavaScript é independente de formatação.
- JavaScript é uma linguagem interpretada.
- JavaScript é altamente portátil e independente de hardware.
- JavaScript se encaixa facilmente em outro software, como os navegadores.

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## O que posso fazer?

- Adicionar mensagens que rolam na tela ou alterar as mensagens da linha de status do navegador.
- Validar conteúdo de um formulário.
- Fazer cálculos;
- Trabalhar com datas, horas e fuso horário;
- Exibir mensagens para o usuário tanto como parte de uma página da Web como em caixas de alertas;



HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## O que posso fazer?

- Fazer animações de imagens ou imagens que mudam quando você move o cursor sobre elas;
- Detectar o navegador em utilização e exibir conteúdo diferente para navegadores diferentes;
- Detectar plug-ins instalados e notificar o usuário se um plug-in foi exigido;
- E muito mais, é só usar a criatividade!



HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Em uma página HTML

Para incluir JavaScript em uma página HTML utilizamos as tags `<script>` conteúdo `</script>`.

Podemos também indicar a linguagem de script a ser utilizada e a versão:

```
<script language="javascript">
```

```
...
```

```
</script>
```

```
// JavaScript é a linguagem padrão do navegador.
```



# JAVASCRIPT

## Regras

- O script é sensível a letras maiúsculas e minúsculas (sensitive case), portanto, digite em minúsculo o que estiver em minúsculo e em maiúsculo o que estiver em maiúsculo!
- Não utilize caracteres especiais (acentos, asteriscos, pontos...), exceto como conteúdo de variáveis!

HTML5  
CSS3  
JAVASCRIPT





# JAVASCRIPT

## Onde utilizar?

Podemos incluir as tags de Javascript em vários lugares do código HTML:

- Em um Arquivo Externo

```
<!DOCTYPE html>
<html>
<head>
<title> Javascript2 </title>
  // Chamando um Arquivo Externo contendo JavaScript.
  <script language="javascript" src="arquivo_externo.js"></script>
</head>...
```

HTML5  
CSS3  
JAVASCRIPT

# JAVASCRIPT

## Onde utilizar?

- No Cabeçalho do Documento

```
<!DOCTYPE html>
<html>
<head>
<title> Javascript1 </title>
  <script language="javascript">
    alert("Seja bem-vindo(a)!"); // JavaScript no Cabeçalho da página HTML.
  </script>
</head>...
```

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Onde utilizar?

- Dentro de Tags, utilizando Tratadores de Eventos

```
<!DOCTYPE html>
<html>
<head>
  <title> Javascript3 </title>
</head>
<body>
  <input type="button" value="Mensagem" onclick="alert('Ola!');" >
  <!-- JavaScript em uma "tag" HTML através de um Tratador de Eventos.-->
</body>
```

HTML5  
CSS3  
JAVASCRIPT

# JAVASCRIPT

## Onde utilizar?

- No Corpo do Documento

```
<!DOCTYPE html>
<html>
<head>
  <title> Javascript4 </title>
</head>
<body>
  <script language="javascript">
    // JavaScript no Corpo de uma página HTML
    document.write("Iniciando em JavaScript");
  </script>
</body>
```

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Comentários

Nós podemos escrever anotações dentro do código fonte sem que elas sejam executadas como comandos do Javascript, para isto, precisamos indicar que as linhas são apenas comentários.

```
// Comentando uma linha.
```

```
/* Comentando  
várias linhas... */
```



# JAVASCRIPT


## Escrevendo em uma página

Você deve ter percebido que utilizamos várias vezes nos exemplos o comando `document.write( )`.

Em JavaScript nós o chamamos de Método, porque ele executa uma função (ação), ou seja, com ele você pode escrever no documento, que é o corpo da página em HTML

### Exemplo:

```
<script language="javascript">  
  document.write("Escrevendo no documento");  
</script>
```

 O conteúdo dos parênteses deve vir entre aspas ou apóstrofes, a menos que seja uma variável. Todos os comandos em JavaScript terminam com ponto e vírgula (;).

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Caixas Pop-up

As caixas Pop-up também são métodos em JavaScript. Com elas podemos interagir com o usuário exibindo mensagens, recebendo dados através da caixa de diálogo e confirmações ou não de informações.



HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Alert

Exibe uma caixa de mensagem e aguarda um clique do usuário no botão <OK> para fechá-la.

### Exemplo:

```
<script language="javascript">  
  alert("Seja muito bem-vindo(a)!");  
</script>
```



HTML5  
CSS3  
JAVASCRIPT





# JAVASCRIPT

## Prompt

Abre uma caixa de diálogo para que o usuário possa entrar com dados.

### Exemplo:

```
<script language="javascript">  
  origem = prompt("Informe o seu país de origem","Brasil");  
</script>
```



Neste exemplo, o texto após a vírgula e dentro dos parênteses, será exibido no campo de resposta como resposta padrão, se preferir deixar a resposta em branco, não inclua conteúdo nem espaços entre as aspas.

A resposta poderá ser armazenada em uma variável para ser tratada posteriormente pelo programa.

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Confirm

Exibe uma caixa de confirmação contendo dois botões: <ok> (sim) e <cancel> (não).

### Exemplo:

```
<script language="javascript">  
  resp = confirm("Será que vai chover?");  
</script>
```



A resposta também poderá ser armazenada em uma variável para ser tratada posteriormente pelo programa.

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Caracteres de Escape

Os caracteres de escape são usados em Strings para:

- Evitar que certos caracteres causem erros dentro do programa;
- Adicionar formatação (como retorno de carro) a Strings.

### Exemplo:

```
<script language="javascript">  
  alert("Um retorno de carro está\rbem no meio desta linha!");  
  alert("\rIsto não saiu como deveria!\r"disse ela");  
  alert("Esta linha tem uma tabulação\taquí.");  
</script>
```

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Variáveis globais e locais

Criar variáveis em JavaScript é muito fácil, é dar um nome e atribuir um valor a ela e estará criada!

### Exemplo:

```
Nome = "Maria"; //Variavel global
```

```
Function nomeDaFuncao(){  
  var idade = 15; //Variavel local  
}
```



HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Nomes das variaveis

- Nomes de variáveis só podem conter letras maiúsculas e/ou minúsculas, números e sublinhado, portanto, não podem conter espaços, hífen ou outros caracteres especiais;
- O primeiro caractere do nome da variável deve ser uma letra ou o sublinhado;
- Javascript é “case sensitive”, portanto, letras maiúsculas são diferentes de minúsculas.
- Não há limite oficial no comprimento de nomes de variáveis, portanto, vale o bom senso.
- Utilize sempre nomes mnemônicos, ou seja, que traduzam o conteúdo da variável.

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Tipos de dados

Em JavaScript você não precisa definir uma variável especificando o tipo de dados que ela armazenará (exceto em casos raros), ela se adaptará ao tipo de dados do conteúdo a ela atribuído.



HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Tipos de dados básicos

- **Number** - Números Inteiros ou de Ponto Flutuante.  
Ex.: 0; 1; -3; 42; 3,1415926535897932384626433832795 ou 3.76e2.
- **String** - Trata-se de uma cadeia de caracteres (entre aspas)  
Ex.: "Meu nome é Guilherme".
- **Boolean** - Booleanos ou Lógicos; possuem dois valores: Verdadeiro ou Falso.  
Ex.: true; false.

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Tipos de dados numéricos especiais

- **Positive infinite** - Infinito positivo.
- **Negative infinite** - Infinito negativo.
- **0** - zero positivo e negativo.
- **NaN** - Not a Number (Não é um Número). Usualmente é gerado como resultado de uma operação matemática que não faz sentido, por exemplo, divisão de qualquer número por zero.

HTML5  
CSS3  
JAVASCRIPT



Não há possibilidade de digitar nenhum dos valores acima. Os valores relacionados com infinito resultam quando se ultrapassa o limite de  $10^{308}$  ou  $10^{-308}$ . Exemplo, multiplicando  $1.0e300$  por si mesmo. Eles não são muito úteis para nós





# JAVASCRIPT

## Tipos de dados especiais

- **Null** - Valor nulo, quando não há nada de útil disponível.
- **Undefined** - Indefinido. Na maioria das vezes é o mesmo que null. Sua presença indica que algo saiu errado em seu JavaScript.



HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Conversão de tipos

O JavaScript trata conversões entre tipos de dados para você sempre que ele pode

### Exemplo:

```
total= 40;  
document.write("O total é " + total);
```

A instrução imprimirá a mensagem: "O total é 40".

A função de `document.write()` trabalha com Strings, portanto, o interpretador de JavaScript automaticamente converte quaisquer não-Strings na expressão em Strings antes de desempenhar a função.

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Funções parseInt e parseFloat

- `parseInt(string)` //Converte uma string em um número inteiro.
- `parseFloat(string)` //Converte uma string em um número com ponto flutuante.

A duas funções efetuarão a leitura da string desde o seu início e retornarão uma versão numérica.

### Exemplo:

```
string_a = "45";  
Num_b = 10;  
num_a = parseInt(string_a);  
alert(num_a + num_b);
```

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Outros métodos de conversão

- **String** Converte o tipo de uma variável em String.
- **Number** Converte o tipo de uma variável em numérico.
- **Boolean** Converte o tipo de uma variável em booleano.

### Exemplo:

```
1. a = 2;  
b = String(a);  
//Converte a variável numérica "a" em String.  
alert(typeof(b));
```

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Operadores

Os operadores precisam agir sobre alguma coisa para que possam operar. Eles podem funcionar com um só dado, que os tornam operadores unários; com dois, binários ou com três, ternário.



HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Operadores Aritméticos

### Adição (+)

```
a = 2 , b = 3; //Lê-se: a recebe 1 e b recebe 3  
c = a + b;  
document.write("<br>" + c); // o valor de "c" será 5
```

### Subtração (-)

```
a = 3 , b = 1;  
c = a - b;  
document.write("<br>" + c); // o valor de "c" será 2
```



HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Operadores Aritméticos

### Divisão (/)

a = 8 , b = 2;

c = a / b;

document.write("<br>" + c); // o valor de "c" será 4

### Multipliação (\*)

a = 2 , b = 3;

c = a \* b;

document.write("<br>" + c); // o valor

### Módulo ou Resto da Divisão(%)

a = 5 , b = 2;

c = a % b;

document.write("<br>" + c); // o valor de "c" será 1

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Operadores Relacionais

Os operadores relacionais são usados em comparações:

Menor que (<)

```
a = 5 , b = 6;  
alert(a < b);
```

Menor ou igual a (<=)

```
a = 3 , b = 5;  
alert(a <= b);
```

Maior que (>)

```
a = 4 , b = 6;  
alert(b > a);
```





# JAVASCRIPT

## Operadores Relacionais

Maior ou igual a ( $\geq$ )

`a = 5 , b = 7;`

`alert(b  $\geq$  a);`

Igual a ( $=$ )

`a = 3 , b = 3;`

`alert(a = b);`

Não igual a ( $\neq$ )

`a = 3 , b = 4;`

`alert(a  $\neq$  b);`



HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Operadores Lógicos

Eles são utilizados em instruções condicionais. (&& e || são binários e ! unário)

### E (&&)

```
a = 6 , b = 3, c = 1;  
alert((a > b) && (b > c)); // (a é maior que b) e (b maior que c)
```

### Ou (||)

```
a = 6 , b = 3, c = 1;  
alert((a > b) || (b == c)); // (a é maior que b) ou (b é igual a c)
```

### Negação (!)

```
a = 6 , b = 3, c = 1;  
alert(a != b); // (a não é igual a b)
```

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Operadores Atribuição

### Atribuição Plena (=)

`a = 3;` // A Variável “a” recebe o valor 3.

`a = b = c = 5;` // As variáveis “a”, “b” e “c” recebem o valor 5.

### Compostos:

`x =+ 3;` // É o mesmo que `x = x + 3;`

`x =- 3;` // É o mesmo que `x = x - 3;`

`x =* 3;` // É o mesmo que `x = x * 3;`

`x =/ 3;` // É o mesmo que `x = x / 3`



HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Operadores Diversos

### Condicional - Ternário (? :)

```
1. x = 1, y = 2; (x > y)?alert("Sim, x é Maior que y"):  
alert("Não, x é Menor que y");
```

Se a expressão condicional ( $x > y$ ) for verdadeira, executará a sentença após a interrogação (?), caso contrário, executará a sentença após os dois pontos (:).

### Concatenador de string - Binário (+)

```
a = "São", b = "Paulo";  
c = a + b; // O conteúdo de "c" será "São Paulo".
```

### Operador de tipo - Unário (typeof)

```
mens = "Olá!";  
alert(typeof(mens)); // O tipo "string" será exibido na caixa de alerta.
```

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Funções

Funções são grupos de instruções em JavaScript que podem ser tratadas como uma unidade. Elas são executadas apenas quando o programa as solicita, uma ou mais vezes. Normalmente definimos as funções dentro do cabeçalho da HTML e podemos chamá-las em qualquer parte do script.

Uma função é definida a partir da palavra-chave function, seguida de seu nome e de parênteses. O conteúdo da função (sentenças) deve estar inserido entre chaves.



HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Funções

```
<!DOCTYPE html>
<html>
  <head>
    <title> Funções </title>
    <script language="javascript">
      function nome_da_função (){
        sentença1; // Linhas de instruções da função.
        sentença2; ...
      }
    </script>
  </head>
  <body>
    Corpo da página...
    <script>
      nome_da_função(); //Chamada da função para que ela possa ser executada.
    </script>
  </body>
</html>
```

HTML5  
CSS3  
JAVASCRIPT

# JAVASCRIPT

## Funções sem parametros

A função é executada sem receber parâmetros e/ou retornar valores.

### Exemplo:

```
<script language="javascript">  
  function mens(){ // Definição da função no cabeçalho da página.  
    alert("Ola!");  
  }  
</script>
```

```
<script language="javascript">  
  mens(); // Chamada da função a partir do corpo da página.  
</script>
```

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Funções com parâmetros

Uma função pode receber parâmetros. Os dados que serão enviados à função deverão estar inseridos entre os parênteses da chamada da mesma. A função receberá e armazenará os dados nas variáveis contidas entre os parênteses de sua definição, na respectiva ordem em que foram enviados.

### Exemplo:

```
<script language="javascript">  
    function mens(quem1 , quem2){ // Definição da função e das variáveis quem1 e quem2.  
        alert("Ola " + quem1 + ",olá " +quem2);  
    }  
</script>
```

```
<script language="javascript">  
    nome1 = "Pedro";  
    nome2 = "Paulo";  
    mens(nome1,nome2); //Chamada da função mens() passando os parâmetros nome1 e nome2.  
</script>
```

HTML5  
CSS3  
JAVASCRIPT





# JAVASCRIPT

## Funções com retorno de valores

As funções podem retornar valores para o ponto de sua chamada.

### Exemplo:

```
<script language="javascript">  
  function media(a,b,c){ // Função recebendo parâmetros em a, b e c.  
    calculo =(a+b+c)/3; // Cálculo dos valores armazenados nas variáveis.  
    return calculo; // Função retornando o resultado do cálculo.  
  }  
</script>  
<script language="javascript">  
  result = media(3,5,8); // Chamada da função media() passando os valores (3, 5, e 8).  
  alert(result);  
  // Exibindo o retorno do cálculo que foi atribuído a variável result.  
</script>
```

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Condição if/else

O if é a instrução condicional principal do JavaScript. Esta instrução tem o mesmo significado da palavra em inglês: “se”.

### Exemplo:

```
<script language="javascript">  
  a = 5;  
  if (a > 2){  
    alert("a é maior que 2"); //Exibe a mensagem "a é maior que 2".  
  }  
</script>
```

Esta instrução inclui uma condição entre os parênteses ( $a > 2$ ). Se a condição for verdadeira, a sentença que se segue: `alert("a é maior que 2")` será executada; caso contrário, não faz nada e o JavaScript pula para o próximo comando após o ponto e vírgula.

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Condição if/else

O else (senão) trata o caso contrário, ou seja, se condição for falsa:

### Exemplo:

```
<script language="javascript">  
  if (a > 2) {  
    alert("a é maior que 2");  
    a = 0;  
  }else{  
    alert("a é menor ou igual a 2"); //Será executada se "a" não for maior que 2.  
  }  
</script>
```

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Instrução switch

JavaScript possui a instrução switch que permite combinar vários testes da mesma variável ou expressão em um único bloco de instruções.

### Exemplo:

```
<script language="javascript">
  dia_semana = 5;
  switch(dia_semana) {
    case 0 :
      document.write("Domingo");
      break;
    case 1:
      document.write("Segunda-feira");
      break;
    case 2:
      document.write("Terça-feira");
      break;
```

```
    case 3:
      document.write("Quarta-feira");
      break;
    case4:
      document.write("Quinta-feira");
      break;
    case 5:
      document.write("Sexta-feira"); //Será impresso "Sexta-feira".
      break;
    case 6:
      document.write("Sábado");
      break;
    default:
      alert(" Dia da semana inválido!");
  }
</script>
```

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Instrução switch

- A variável que terá o seu valor testado deverá estar entre os parentes da instrução switch;
- As sentenças do switch deverão estar contidas entre as chaves ({ }).
- Cada instrução case possui um valor específico que será comparado com o valor da variável. Se o valor do case corresponder ao da variável, as instruções após os dois pontos (:) serão executadas; caso contrário, o próximo case será testado.
- A instrução break é colocada em cada case.
- Se um dos casos for satisfeito, então o switch poderá ser finalizado.
- O default é opcional. Se nenhuma das instruções case for satisfeita, as instruções do default serão executadas.

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Array

Uma Matriz ou Array permite armazenar vários dados separadamente dentro de uma única variável, formando um conjunto. Usualmente, todos esses dados possuem um esquema de conexão.

Os Arrays simplificam o seu código porque diminui o número de variáveis que você poderia criar, com nomes similares. (Ex.: dias da semana).



HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Array

Os Arrays podem conter strings, números, objetos ou outros tipos de dados.

### Exemplo:

```
notas = new Array(5);  
notas[0] = 8.5;  
notas[1] = 5.0;  
notas[2] = 10.0;  
notas[3] = 9.0;  
notas[4] = 4.5;
```



# JAVASCRIPT

## Array

Os Arrays podem conter strings, números, objetos ou outros tipos de dados.

### Exemplo:

```
notas = new Array(5);  
notas[0] = 8.5;  
notas[1] = 5.0;  
notas[2] = 10.0;  
notas[3] = 9.0;  
notas[4] = 4.5;
```

- A palavra-chave **new** define a variável notas como um novo objeto do tipo Array, notas agora possuirá todas as propriedades e métodos do objeto Array;

HTML5  
CSS3  
JAVASCRIPT





# JAVASCRIPT

## Array

- O número entre parênteses indica a quantidade de elementos do Array, no caso 5;
- O índice do primeiro elemento é 0 (zero);
- Para atribuímos valores aos elementos, utilizarmos o nome do Array seguido do índice correspondente ao elemento, inserido entre colchetes ([ ]):
- notas[0] = 8.5;
- notas[1] = 5.0;



# JAVASCRIPT

## Array

Para ler o conteúdo de um Arrayé só utilizar a mesma notação que se utilizou na atribuição de valores.

A seguinte instrução exibe os valores dos primeiros três elementos do Array notas:

```
document.write("Notas: " + notas[0]+ ", " + notas[1]+ " e " + notas[2]);
```



HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Array de String

Para ler o conteúdo de um Array é só utilizar a mesma notação que se utilizou na atribuição de valores.

### Exemplo:

```
dia_semana = new Array(7);  
dia_semana[0] = "Domingo";  
dia_semana[1] = "Segunda-feira";  
dia_semana[2] = "Terça-feira";  
dia_semana[3] = "Quarta-feira";  
dia_semana[4] = "Quinta-feira";  
dia_semana[5] = "Sexta-feira";  
dia_semana[6] = "Sábado";
```

```
dia_semana = new Array("Domingo","Segunda-feira","Terça-feira","Quarta-feira","Quinta-feira","Sexta-feira","Sábado");
```

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Array

Assim como o objeto String, Array também possui a propriedade length, só que ela informa o número de elementos pertencentes ao Array (o que é útil no caso de você criar um Array sem informar o tamanho e solicitar ao usuário que entre com a quantidade de dados).

### Exemplo:

```
pontos = new Array(20);  
document.write(pontos.length); //Será impresso 20.
```



HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Loop for

O JavaScript possui recursos que fazem o computador desempenhar tarefas repetitivas para você.

O loop for é o primeiro que utilizaremos para criar loops (laços, voltas ou repetições)

### Exemplo:

```
for (i= 1; i< 5; i++){  
    document.write("Esta é a linha ",i, "<br>");  
}
```

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Loop while

Diferente do loop for, o loop while não necessita obrigatoriamente de um contador para controlá-lo, em vez disso, ele executa enquanto (while) uma condição for verdadeira. Se a condição iniciar como falsa, as instruções não serão executadas.

### Exemplo:

```
valor = new Array(3,2,4,1,5);  
i = 0;  
total = 0;  
while (total < 10){  
    total += valor[i];  
    i++;  
}  
document.write("Total = " + total); //Resultado: Total = 10
```

HTML5  
CSS3  
JAVASCRIPT



# JAVASCRIPT

## Loop do..while

O terceiro loop é o do...while (faça...enquanto). A diferença em relação ao loop while é que no do...while, a condição é testada no final do loop, portanto, as instruções inseridas no do são executadas pelo menos uma vez.

### Exemplo:

```
valor = new Array(3,2,4,1,5);  
cont = 0;  
total = 20;  
do {  
    total += valor[cont];  
    cont++;  
}while (total < 10)
```

```
document.write("Total = " + total); //  
Resultado: Total = 23
```

HTML5  
CSS3  
JAVASCRIPT



OBRIGAD@



by NUAGE  
IT's everywhere.