



# Linguagem para Web I

# JavaScript

## Parte 3 - Funções, Condicionais e Loops

Versão de 27.10.2012



# JavaScript - Parte 3

1. Função (o que é, e funções do JavaScript);
2. Como criar e chamar uma função;
3. Condicionais;
4. Loops;
5. Leitura e Escrita de dados para testes.



# 1. Função

É um bloco de comandos que é definido uma vez mas que pode ser executado ou chamado várias vezes. Elas podem receber informações para realizar uma operação (parâmetros) e retornar valores.

O JavaScript possui suas próprias funções. Mas o programador pode criar suas próprias funções personalizadas.



# 1. Funções do JavaScript

Algumas funções do JavaScript

Função	Descrição
<b>parseInt(valor)</b>	Converte uma string para número, retornando a parte inteira de um valor ou NaN caso não seja possível a conversão.
<b>parseFloat(valor)</b>	Converte uma string para número, retornando a parte inteira e decimal de um valor ou NaN caso não seja possível a conversão.
<b>Number(valor)</b>	Converte uma string em um número ou retorna NaN caso não seja possível a conversão. A decisão se será um número inteiro ou decimal, vai depender do parâmetro passado a função.
<b>toString(valor)</b>	Converte qualquer tipo de valor em um string.
<b>isNaN(valor)</b>	Verifica que o valor passado é um NaN. Caso seja retorna true, caso contrário, false.
<b>isFinite(valor)</b>	Retorna true, caso o valor passado como parâmetro seja finito e false caso seja intinite.
<b>str.replace(termo, novoTermo)</b>	Procura ocorrências em uma string para substituição.
<b>escape(valor)</b>	Retorna a string convertida para o formato url-encoding de 7 bits, muito usado para criação de cookies.
<b>unescape(valor)</b>	Retorna a string convertida para o formato de 7 bits, de acordo com a tabela de caracteres Latin-1.



# 1. Funções do JavaScript

```
var texto1 = "10";
var texto2 = "458.78";
var msg = "10alunos";
var msg2 = "10.523 bilhões";
var nome = "João Balão";

var a = parseInt(texto1); // atribui o número 10
var b = parseFloat(texto2); // atribui o número 458.78
var c = parseInt(texto2); // atribui o número 458
var d = a + c; // atribui o número 468
var e = parseInt(nome); // atribui NaN
var f = escape(nome) ; // atribui Jo%E3o%20Bal%E3o
var g = unescape(f); // atribui João Balão
var h = parseInt(msg); // atribui o número 10
var i = parseFloat(msg2); // atribui o número 10.523
var sobren = "Silva";
var nome = nome.replace("Balão", sobren); // atribui João Silva a nome
```



# JavaScript - Parte 3

1. Função (o que é, e funções do JavaScript);
2. Como criar e chamar uma função;
3. Condicionais;
4. Loops;
5. Leitura e Escrita de dados para testes.



## 2. Como criar funções

Deve-se criar quando há rotinas que se repetem várias vezes no mesmo HTML ou em vários documentos de um site. Assim evita-se repetição de código.

```
<script>  
function falaComMundo() {  
    alert("Olá Mundo!");  
}  
</script>
```

**cria** a função

```
<h1>Criando uma function</h1>  
<p><a onClick="falaComMundo();" href="#">Oi</a></p>  
<p><button onMouseOver="falaComMundo();">Tudo bem?</button></p>
```


**chama** a função



## 2. Como criar funções

Funções podem receber valores, conhecidos como parâmetros. Passe parâmetros usando os parênteses e eles agirão como variáveis locais daquela função.

```
<script>
// espera receber um valor em dolar
function converteParaReais(dolar) {
// multiplica o valor recebido por 2 e guarda numa variavel
    var convertido = dolar*2;
    alert("R$" + convertido);
}
</script>
```



**cria** a função

```
<h1>Criando uma function</h1>
<!-- passa o valor 35 como parâmetro ao clicar -->
<p><button onClick="converteParaReais(35);">Converte $35</button></p>
```



**chama** a função



## 2. Como criar funções - Exercício

Com base no exemplo anterior, crie uma função que é chamada no clique de um *button*. Ela recebe um valor em Celsius, converte para Fahrenheit e exibe em um alerta. Para converter um valor em Fahrenheit usa-se a fórmula:

$$F = (9/5) * C + 32$$

Crie 3 botões, eles servem para testar os valores em celsius: **25**, **100** e **32**

Formato	Entrega
Individual	Em anexo por email para <a href="mailto:professor@rodrigomuniz.com">professor@rodrigomuniz.com</a>
Arquivo <b>graus.html</b>	Não esqueça nome, período e turno
Até 23h59 do dia 29/10	Assunto do email: <b>[Web LPW] Exercício JS graus</b>



## 2. Como criar funções - Exercício

Uma função que é chamada no clique de um *button*. Ela recebe um valor em Celsius, converte para Fahrenheit e exibe em um alerta. Para converter um valor em Fahrenheit usa-se a fórmula:

$$F = (9/5) * C + 32$$

```
<script>
// espera receber um valor em Celsius
function converteParaF(c) {
    var convertido = (9/5) * c + 32;
    alert(convertido + " Fahrenheit");
}
</script>
```

```
<h1>Convertendo</h1>
<button onClick="converteParaF(25);">Converte 25</button>
<button onClick="converteParaF(100);">Converte 100</button>
<button onClick="converteParaF(32);">Converte 32</button>
```



## 2. Como criar funções

Funções também podem retornar valores para serem guardados em uma variável para uso futuro.

```
<script>
// espera receber um valor em reais
function converteParaDolar(reais) {
// multiplica o valor recebido por 2 e guarda numa variavel
    var convertido = reais*2;
    return convertido;
}
var valorFinal = converteParaDolar(450);
alert("R$" + valorFinal); //alerta R$900
</script>
```



# JavaScript - Parte 3

1. Função (o que é, e funções do JavaScript);
2. Como criar e chamar uma função;
3. Condicionais;
4. Loops;
5. Leitura e Escrita de dados para testes.



## 3. Condicionais ➡ if

Condicionais executa ou pula comandos dependendo do valor especificado em uma expressão usando operadores. São conhecidos como pontos de decisão. Abaixo temos um exemplo do condicional *if*:

```
var meuNumero = 10;
if (meuNumero == 10) {           //true (alerta)
    alert("Igual a 10!");
}
if (meuNumero < 7) {             //false (não alerta)
    alert("Menor que 7!");
}
if (meuNumero > 9) {            //true (alerta)
    alert("Maior que 9!");
}
```



## 3. Condicionais ➡ else

Com o *else* é possível executar um código caso a expressão seja falsa.

```
var meuNumero = 4;  
if (meuNumero == 10) {  
    alert("Igual a 10!");  
} else { //false (alerta)  
    alert("Menor que 7!");  
}
```



## 3. Condicionais ➡ else if

Com o *else if* é possível executar um código caso a expressão seja falsa mas especificando uma nova condicional.

```
var meuNumero = 4;
if (meuNumero == 10) {
    alert("Igual a 10!");
}
else if (meuNumero > 2) { //true (alerta)
    alert("Maior que 2!");
}
else {
    alert("Nem é 10, nem é maior que 2!");
}
```



### 3. Condicionais ➡ switch

Com o *switch case* é possível escolher entre vários blocos de código quando se depende do valor de apenas uma expressão.

```
var dia = new Date().getDay();
switch (dia) {
  case 0:
    alert("Hoje é Domingo");
    break;
  case 1:
    alert("Hoje é Segunda");
    break;
  case 2:
    alert("Hoje é Terça");
    break;
  case 3:
    alert("Hoje é Quarta");
    break;
  case 4:
    alert("Hoje é Quinta");
    break;
  case 5:
    alert("Hoje é Sexta");
    break;
  case 6:
    alert("Hoje é Sábado");
    break;
}
```





# JavaScript - Parte 3

1. Função (o que é, e funções do JavaScript);
2. Como criar e chamar uma função;
3. Condicionais;
4. Loops;
5. Leitura e Escrita de dados para testes.



## 4. Loops ➡ while

Loops são usados para executar uma parte do código várias vezes (para escrever vários elementos em uma lista ou preencher os valores de um *array*, por exemplo).

O *while* quer dizer “enquanto”...

```
var contador = 0;
document.write( "<ul>" );
//enquanto contador for menor que 10
while (contador < 10) {
    document.write( "<li>" + contador + "</li>" );
    contador++;
}
document.write( "</ul>" );
```

→  
Resultado no browser

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9



## 4. Loops ➡ for

O uso do *for* é mais comum que o *while*. Ele é usado para loops que atendem a um padrão e usam um contador.

```
document.write( "<ul>" );
```

```
//declara contador ; testa contado ; incrementa
```

```
for (contador = 0 ; contador < 10 ; contador++) {
```

```
    document.write( "<li>" + contador + "</li>" );
```

```
}
```

```
document.write( "</ul>" );
```

Resultado no browser

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9



# JavaScript - Parte 3

1. Função (o que é, e funções do JavaScript);
2. Como criar e chamar uma função;
3. Condicionais;
4. Loops;
5. Leitura e Escrita de dados para testes.



## 5. Leitura e escrita de dados para testes

Vamos usar **document.write()**, **window.prompt()**, **console.log()** e **alert()** para fazer nossos testes iniciais.



```
<html>
<head>
  <title>Entrada de Dados</title>
</head>
<body>
  <h1>Entrada de Dados</h1>
  <script>
    var nome = window.prompt("Por favor, digite seu nome","");
    document.write("Olha que nome bonito: " + nome);
  </script>
</body>
</html>
```

## 5. Leitura e escrita de dados para testes - Exercício

Use o seu arquivo **graus.html** como base para criar o **graus2.html**. Ao ser executado este arquivo vai pedir um valor em celsius ao usuário e em seguida apresentará o valor convertido para Fahrenheit em um alerta.

Formato	Entrega
Individual	Em anexo por email para <a href="mailto:professor@rodrigomuniz.com">professor@rodrigomuniz.com</a>
Arquivo <b>graus.html</b>	Não esqueça nome, período e turno
Até 23h59 do dia 29/10	Assunto do email: <b>[Web LPW] Exercício JS graus 2</b>



# Referências

**FLANAGAN**, David. JavaScript O Guia Definitivo.

Editora: Bookman, 2004.

Professor **Rodrigo Muniz**

[professor@rodrigomuniz.com](mailto:professor@rodrigomuniz.com)



:-)

# Obrigado!

[professor@rodrigomuniz.com](mailto:professor@rodrigomuniz.com)