



Programação WEB

Linguagem JavaScript

Douglas Nassif Roma Junior

douglas.junior@grupointegrado.br

Conteúdo

- JavaScript
 - Comentários
 - Declarações
 - Fundamentos
 - Condicionais
 - Laços
 - Popup Boxes



Scripts como comentário

```
<html>
  <head></head>
  <body>
    <script type="text/javascript">
      <!-- document.write("Hello World!");
      //-->
    </script>
  </body>
</html>
```

Scripts como comentário

- A razão para se colocar um script como se fosse um comentário de código é porque navegadores que não possuem suporte a JavaScript poderiam exibir as instruções como se fosse conteúdo da página.

`<!--` antes da primeira instrução

`-->` após a última instrução

Onde Colocar o Script?

- JavaScripts na seção “body” serão executados ENQUANTO a página é carregada.
- JavaScripts na seção “head” serão executados quando INVOCADOS.

Exemplos

- Head section
Scripts contendo funções vão na seção “head” do documento. Assim podemos ter certeza de que o script já foi carregado antes da função ser invocada.
- Body section
Executa um script localizado na seção “body”
- External script
Mesmo comportamento da seção head

Exemplo – Head Script

```
<html>
  <head>
    <script type="text/javascript">
      function exibirMensagem() {
        alert("Essa função foi chamada com o evento 'onload'");
      }
    </script>
  </head>
  <body onload="exibirMensagem();">
  </body>
</html>
```

Exemplo – Body Script

```
<html>
  <head>
</head>
  <body>
    <script type="text/javascript">
      document.write("Esta Mensagem é escrita pelo JavaScript");
    </script>
  </body>
</html>
```


Exemplo Script Externo

- Qual o conteúdo, o nome e a localização do arquivo que contém o script externo?

```
<html>
  <head>
    <script type="text/javascript" src="biblioteca.js"></script>
  </head>
  <body>
    <p>O script está em um arquivo externo "biblioteca.js".</p>
  </body>
</html>
```

Onde Colocar Scripts?

- Se colocar na parte `HEAD` – temos certeza de que eles já foram carregados e podem responder a algum evento
- Se colocar na parte `BODY` – serão executados durante a carga da página – scripts para formatação/criação do documento
- Externos – para utilizar várias vezes/em várias páginas.
 - Não se deve colocar a tag `<script> </script>` no documento externo `.js`

Fundamentos de Programação

- Estrutura Condicional
 - IF
 - IF .. ELSE
 - IF .. ELSE IF .. ELSE
 - Switch

Estrutura Condicional

```
if (condição) {  
    // código para ser executado se condição verdadeira  
}
```

```
<script type="text/javascript">  
    //Escreve "Good morning" se a hora é menor que 10  
    var data = new Date();  
    var horas = data.getHours();  
    if (horas < 10){  
        document.write("<b>Good morning</b>");  
    }  
</script>
```

Estrutura Condicional

```
if (condição) {  
    // código executado se a condição for verdadeira  
} else {  
    // código executado se a condição não for verdadeira  
}
```

Estrutura Condicional

```
<script type="text/javascript">  
  //Se a hora é menor que 10 exibe a mensagem "Good morning".  
  //senão, exibe a mensagem "Good day".  
  var data = new Date();  
  var horas = data.getHours();  
  if (horas < 10) {  
    document.write("Good morning!");  
  } else {  
    document.write("Good day!");  
  }  
</script>
```

Estrutura Condicional

```
switch(n) {  
    case 1:  
        // código 1 aqui  
        break;  
    case 2:  
        // código 1 aqui  
        break;  
    default:  
        //código para ser executado se nenhum  
        //dos n casos anteriores for verdadeiro  
}
```

Popup boxes

- Em JavaScript há 3 tipos de Mensagens que podem ser enviadas ao usuário, sob a forma de “Popup boxes”
 - Alert Boxes
 - Prompt Boxes
 - Confirm Boxes

Popup boxes

- **Alert Box**

- Uma caixa de mensagem usada para exibir uma mensagem a qual devemos ter certeza de que o usuário leu.
- Quando uma caixa do tipo “alert box” é exibida, o usuário precisa clicar em OK para prosseguir.

- **Sintaxe:**

```
<script type="text/javascript">  
    alert("Texto para ser exibido");  
</script>
```

Popup boxes

```
<html>
<head>
  <script type="text/javascript">
    function disparaAlert() {
      alert("Olá está é uma Alert Box!");
    }
  </script>
</head>
<body>
  <input type="button" onclick="disparaAlert()" value="Exibir alert box" />
</body>
</html>
```

Popup boxes

- **Confirm Box**

- Uma caixa do tipo “Confirm box” é usada se você deseja que o usuário verifique ou aceite algo.
- Quando uma caixa Confirm Box é exibida, o usuário escolhe entre OK ou Cancel para prosseguir.
- Se o usuário clicar em OK, essa instrução retorna true, se o usuário clicar em Cancel, a instrução retorna false

- **Sintaxe:**

```
<script type="text/javascript">  
    confirm("Texto para ser exibido");  
</script>
```

```
<html>
  <head>
    <script type="text/javascript">
      function disparaConfirm() {
        var r = confirm("Escolha um botao");
        if (r == true){
          document.write("Voce escolheu OK!");
        } else {
          document.write("Voce escolheu CANCEL!");
        }
      }
    </script>
  </head>
  <body>
    <input type="button" onclick="disparaConfirm()"
                                              value="Exibir confirm box" />
  </body>
</html>
```

Popup boxes

- **Prompt Box**

- Uma caixa do tipo `prompt box` é usada se é preciso que o usuário informe algum valor de entrada antes de entrar em uma página.
- Quando uma caixa `prompt box` é exibida, o usuário precisa clicar em OK ou CANCEL após informar um valor de entrada.
- Se o usuário clicar em OK, o comando retorna o valor informado pelo usuário, caso ele clique em CANCEL, o comando retorna `NULL`.

- **Sintaxe:**

```
<script type="text/javascript">  
    prompt("Descrição aqui", "valor default")  
</script>
```

Popup boxes

```
<html>
<head>
  <script type="text/javascript">
    function disparaPrompt() {
      var name=prompt("Informe seu nome", "Harry Potter");
      if (name!=null && name!="") {
        document.write("Olá " + name + "! Seja bem vindo.");
      }
    }
  </script>
</head>
<body>
  <input type="button" onclick="disparaPrompt()"
                                             value="Exibir prompt box" />
</body>
</html>
```

Fundamentos de Programação

- Exercícios – Atividades

- Escrever scripts para:

- Ao entrar na página, ler o nome do visitante e o ano de nascimento.
 - Caso o usuário seja maior de idade (18 anos), Escrever no documento: “Esse conteúdo é permitido para você”
 - Caso contrário, escrever “Visite www.disney.com”

Fundamentos de Programação

- Inserir um botão na página com a funcionalidade de exibir a hora do sistema em uma mensagem do tipo `alert`.
- Exibir uma mensagem do tipo `prompt box` que permita que o usuário informe uma cor (padrão HTML) e exiba, no documento a frase “A Cor escolhida foi:” na cor escolhida pelo usuário.

Funções em JavaScript

- Para evitar que um script seja executado quando a página é carregada, podemos utilizar funções.
- A função só será executada quando um evento ou uma chamada a essa função ocorrer.
- Funções podem ser chamadas de vários locais dentro da mesma página ou mesmo por diferentes páginas, desde que elas estejam em um arquivo `.js`

Funções em JavaScript

- Podem ser inseridas tanto na seção `HEAD` quanto na `BODY` do documento.
- Para ter certeza de que a função terá sido carregada na memória quando for invocada, o mais prudente é inseri-la na seção `HEAD`

Funções em JavaScript

- Sintaxe

```
function nomeDaFuncao(var1,var2,...,varX) {  
    // conteúdo da função  
}
```

- `var1`, `var2`, etc, são os parâmetros da função.
- Caso não haja parâmetros, deve-se usar `()` sem nada dentro.
- Caso a função deva retornar algum valor, utilizamos uma instrução `return`.

Funções em JavaScript

- Variáveis declaradas dentro do corpo da função possuem escopo local (só podem ser acessadas dentro da função)
- O tempo de vida dessas variáveis também só existe enquanto a função está ativa. Quando a função encerra sua execução, as variáveis são destruídas

Funções em JavaScript

```
<html>
  <head>
    <script type="text/javascript">
      function minhaFuncao(txt) {
        alert(txt) ;
      }
    </script>
  </head>
  <body>
    <form>
      <input type="button" onclick="minhaFuncao('Hello')" value="Call function">
    </form>
  </body>
</html>
```

Laços de repetição

- For

```
for (início;condição;incremento) {  
    // código  
}
```

Laços de repetição

```
<html>
  <head></head>
  <body>
    <script type="text/javascript">
      for (i = 1; i <= 6; i++){
        document.write("<h" + i + ">This is header " + i);
        document.write("</h" + i + ">");
      }
    </script>
  </body>
</html>
```

Laços de repetição

- While..do

```
while (condição) {  
    // código  
}
```

- Do..While

```
do {  
    // código  
} while (condição);
```


Laços de repetição

- **break**
 - Interrompe o laço
- **continue**
 - “salta” o laço

Laços de repetição

```
<html>
<head></head>
<body>
  <script type="text/javascript">
    var i=0;
    for (i=0;i<=10;i++) {
      if (i==3) {
        break;
      }
      document.write("O número é" + i);
      document.write("<br />");
    }
  </script>
</body>
</html>
```

Laços de repetição

```
<html>
<head></head>
<body>
  <script type="text/javascript">
    var i=0;
    for (i=0;i<=10;i++){
      if (i==3) {
        continue;
      }
      document.write("Numero " + i);
      document.write("<br />");
    }
  </script>
</body>
</html>
```

Laços de repetição

- For.. In
- Para percorrer valores em um array

```
for (indice in array) {  
    // Código  
}
```

Laços de repetição

```
<html>
<head></head>
<body>
  <script type="text/javascript">
    var i;
    var mycars = new Array();
    mycars[0] = "Fusca";
    mycars[1] = "Fiat 147";
    mycars[2] = "Lada";
    for (i in mycars) {
      document.write(mycars[i] + "<br />");
    }
  </script>
</body>
</html>
```

Laços de repetição

- Escrever um script JavaScript para gerar a tabela de cores de HTML
- Para gerar todas as cores puras vamos utilizar três arrays Javascript com os possíveis valores para cada uma das cores RGB. Portanto, teremos três arrays como os que podem ser vistos a seguir:

```
var r = new Array("00", "33", "66", "99", "CC", "FF");  
var g = new Array("00", "33", "66", "99", "CC", "FF");  
var b = new Array("00", "33", "66", "99", "CC", "FF");
```

Laços de repetição

- Para escrever a tabela na página web, percorremos estes três arrays. Para isso, vamos utilizar loops aninhados.

```
Contamos desde 00 até FF com o valor R{  
    Contamos desde 00 até FF com o valor G{  
        Contamos desde 00 até FF com o valor B{  
            Imprimimos o valor atual de RGB  
        }  
    }  
}
```

Resultado

```
for (i=0;i<r.length;i++) {  
    for (j=0;j<g.length;j++) {  
        for (k=0;k<b.length;k++) {  
            //criamos a cor  
            var novacor= "#" + r[i] + g[j] + b[k];  
            //imprimimos a cor  
            document.write(novacor + "<br>");  
        }  
    }  
}
```


Laços de repetição

- Para inserir em uma grid e fazer com que as cores sejam efetivamente visualizadas, podemos usar uma tabela.

```
<table>  
  ...  
</table>
```

```
<table>
  <script type="text/javascript">
    var r = new Array("00","33","66","99","CC","FF");
    var g = new Array("00","33","66","99","CC","FF");
    var b = new Array("00","33","66","99","CC","FF");
    var i, j, k;
    for (i=0;i<r.length;i++) {
      for (j=0;j<g.length;j++) {
        document.write("<tr>");
        for (k=0;k<b.length;k++) {
          var novacor= "#" + r[i] + g[j] + b[k];
          document.write("<td bgcolor=\"\" + novacor + \"\">");
          document.write(novacor);
          document.write("</td>");
        }
        document.write("</tr>");
      }
    }
  </script>
</table>
```

Dúvidas

