

PROGRAMAÇÃO WEB Parte VII

Prof. Me. Felipe Diniz Dallilo



Alterando valores dos campos

Para alterar o valor de um campo no formulário é necessário apenas aplicar o “reverso da atribuição” como ilustrado abaixo:

```
<input type="text" id="id1"/>  
<script>  
  document.getElementById('id1').value = 'Teste';  
</script>
```

* Caso não houver nenhum evento para “disparar” uma função no JS, é importante que o script esteja após o HTML.

Exemplo

Abaixo um exemplo simples de um Código javascript inserindo um valor em um componente HTML:

```
<html>
  <head>
    .....
  </head>
  <body>
    <input type="text" id="id1"/>
    <script>
      document.getElementById('id1').value = 'Teste';
    </script>
  </body>
</html>
```

DIV

A tag <div> define uma divisão ou uma seção em um documento HTML. Com a mesma é possível:

- Adicionar componentes HTML
- Aplicar folhas de estilos (mudar cor da seção, redimensionar tamanho, mudar o tipo da fontes,).
- Adicionar código Javascript.

DIV

- Sintaxe HTML de uma div:

- `<div>`
 //Código HTML aqui
`</div>`

Para associar ao Javascript e CSS é necessário adicionar um **identificador** na div.

```
<div id="testeDiv">  
</div>
```


Adicionando html dinamicamente na página!

Através das DIV's é possível adicionar código dinamicamente na página html, segue exemplo abaixo:

```
function gravar(){  
    var div =  
    document.getElementById("divResultado");  
    div.innerHTML = "<h1> Olá mundo ! </h1>";  
}
```

* Lembrando que é necessário a tag <div> no HTML com o identificador "**divResultado**"

Adicionando html dinamicamente na página!

- O código também pode ser incremental:
- ```
function gravar(){
```
- ```
    var div =
```
- ```
 document.getElementById("divResultado");
```
- ```
    div.innerHTML = div.innerHTML + "<h1> Olá
```
- ```
 mundo! </h1>";
```
- ```
}
```
- * Lembrando que é necessário a tag <div> no HTML com o identificador "**divResultado**"

Adicionando html dinamicamente na página!

- O valor inserido em um formulário também pode ser utilizado para gerar o código dinamicamente como ilustra o exemplo abaixo:
- ```
function gravar(){
```
- ```
    var titulo = document.getElementById("txtTitulo").value;
```
- ```
 var div = document.getElementById("divResultado");
```
- ```
    div.innerHTML = div.innerHTML + "<h1>" + titulo
```
- ```
 + "</h1>";
```
- ```
}
```
- * Lembrando que é necessário a tag <div> no HTML com o identificador **"divResultado"**
- * * Lembrando também que é necessário um "input type text" no HTML com o identificador "txtTitulo"

Exemplo

```
<html>
<head>
  <title>teste</title>
  <meta charset="UTF-8" />
  <script type="text/javascript">
    function gravar(){
      var titulo = document.getElementById("txtTitulo").value;
      var subtítulo = document.getElementById("txtSubtítulo").value;
      var div = document.getElementById("divResultado");

      div.innerHTML = div.innerHTML + "<h1>" + titulo + "</h1>" + "\n" + subtítulo ;
    }
  </script>
</head>
<body>
  <div>
    <label>Titulo:</label>
    <input type="text" id="txtTitulo"/> <br>
    <label>Subtítulo:</label>
    <input type="text" id="txtSubtítulo"/>
    <button id="btnEnviar" onclick="gravar()" >Gravar</button>
  </div>
  <div id="divResultado">
  </div>
</body>
</html>
```

For

- A sintaxe para fazer um laço de repetição “for” no Javascript pode ser dividida da seguinte forma :

```
for (1; 2; 3) {  
  4;  
}
```

1 – Criação da variável

2 – Condição

3 – Incremento

4 – Bloco executado dentro do loop

For

Exemplos incremento padrão:

```
for (i = 0; i < 10 ; i++) {  
    alert(i);  
}
```

For

Exemplos incremento customizado:

- ```
for (i = -19; i < 10 ; i = i +5) {
 alert(i);
}
```

# For

Definição de variável fora do laço:

```
var i = 2;
var len = 10;
var text = "";
for (; i < len; i++) {
 alert(len);
}
```



# For

Definição de variável e incremento fora da declaração do for:

```
var i = 0;
var len = 10;
for (; i < len;) {
 alert(i);
 i++;
}
```

# While

- Outra, possibilidade de laço de repetição no Javascript é o While, que pode ser dividido em duas partes:
- while (1) {  
    2  
}
- 1 – Condição
- 2 – Bloco executado dentro do loop

# While

Exemplo:

```
while (i < 10) {
 text += "O número é: " + i;
 i++;
}
```

# Do while

- Caso seja necessária a execução de um bloco dentro do loop (mesmo quando a condição não é satisfeita), utilizar um Do While:

```
do {
```

```
 1
```

```
}while (2);
```

1 – Instrução

2 - Condição

# Do while

Exemplo:

- ```
do {  
    text += "O número é: " + i;  
    i++;  
}  
while (i < 10);
```


Laços não efetivos

Cuidado com a forma de incremento utilizada e a condição de parada do while, isto pode gerar um laço que nunca será utilizado ou um loop eterno.

```
i = 0;
while (i < 10) {
    text += "O número é: " + i;
    i--; //sempre será menor que 10.
}
```

Referências

- Lemay, Laura; Colburn, Rafe; Tyler, Denise. Aprenda a Criar Páginas Web com HTML e XHTML em 21 Dias. Editora Pearson. 2013. 1ª Edição. ISBN: 9788534614283
- Chak, Andrew. Como Criar Sites Persuasivos. Editora Pearson. 2012. 1ª Edição. ISBN: 9788534615112
- Fábio Flatschart; Clécio Bachini; Cesar Cusin. Open Web Platform. Editora Pearson. 2013. 1ª Edição. ISBN: 9788574526140