

=== Observe o formulário a seguir:

## Fale conosco



Entre em Contato:

Dados do Usuario:

Nome:

Email:

Telefone 1:

Telefone 2:   -

Figura 1 - Formulário com entrada de dados

### 1 - Caixa de texto Nome:

Nome: <input type="text" name="nome" size="30" maxlength="30" placeholder="Primeiro Nome" required>

-- a propriedade **maxlength** limita a quantidade de caracteres digitados para no máximo 30

-- a propriedade **placeholder** apresenta na caixa uma mensagem que normalmente indica o que e como deve ser digitado o dado nesta caixa.

-- a propriedade **required** obriga a digitação nesta caixa

### 2- Caixa de texto Email:

Email:<input type="email" name="email" size="30" maxlength="30" placeholder="Endereço de Email" required>

-- a propriedade **type="email"** valida se a digitação é referente a um endereço de email, solicitando a digitação do @

### 3- Caixa de texto Telefone 1:

Telefone 1: <INPUT type="text" maxlength="14" size="12" name="telefone" placeholder="(99)99999-99999" onkeypress="return Mascara(window.event.keyCode)">

-- o evento **onkeypress** aciona um método que será executado, neste caso, sempre que for teclado algo. A cada tecla digitada nesta caixa será acionado o evento "Mascara", processando o parâmetro enviado (window.event.keyCode), ou seja o código ASC II da última tecla digitada, e retornado para a caixa, para uma nova digitação.

### 4 – Caixas de texto Telefone 2:

Telefone 2:

```
(<input type="text" name = "cod-reg" size="1" maxlength = "2" placeholder ="DDD" required  
onkeypress="return blokletras(window.event.keyCode)">)
```

```
<input type="text" name="prefixo" size="2" maxlength="5" placeholder="prefixo" required onkeypress="return  
blokletras(window.event.keyCode)">
```

```
- <input type="text" name="sufixo" size="2" maxlength="4" placeholder="sufixo" required onkeypress="return  
blokletras(window.event.keyCode)"> </p>  
</fieldset>
```

-- definição de "tamanho da caixa de texto", "quantidade limite de caracteres digitado", "definição de como e o que deve ser digitado" e acionamento do método com passagem de parâmetro no evento "onkeypress".

## 5 – Métodos Java Script:

A seção de códigos Java Script deve ser aberta e fechada e dentro da seção "Head"

```
2 <head>  
3   <title>Validação na Entrada de dados </title>  
4  
5   <script language=javascript>  
6  
7       function blokletras(keypress)  
8       {  
9           ... comandos  
10      }  
11  
12      function Mascara (keypress)  
13      {  
14          ... comandos  
15      }  
16  
17  </SCRIPT>  
18 </head>  
19  
20 <body>  
21
```

Figura 2 - Seção para inserção de métodos Java Script

## 6 – Método Java Script blokletras:

Este método consiste em checar na tabela ASC II se o código da última tecla digitada (keypress) corresponde a algum código pertencente a "categoria numérica" (códigos de 48 a 57 – de 0 a 9). Chamada ao método:

**onkeypress="return blokletras(window.event.keyCode)"**

É acionado através de um evento, neste caso o "onkeypress", e possui envio de parâmetro e retorno.

O parâmetro enviado é o código ASC II da última tecla digitada (**window.event.keyCode**).

O retorno é booleano, onde caso o código ASC da última tecla digitada pertença a categoria de valores numéricos, será "True" **permitindo uma nova digitação**, caso contrário será "False" **bloqueando a digitação** da tecla.

```
9
10 <script language=javascript>
11
12 function bloklettras(keypress)
13 {
14     // campo senha - bloqueia letras
15
16     if(keypress>=48 && keypress<=57)
17     {
18         return true;
19     }
20     else
21     {
22         return false;
23     }
24 }
25 </SCRIPT>
26
27 </head>
```

Figura 3 - Método para bloquear entrada de letras na digitação

## 7 – Método Java Script Mascara

```
21 function Mascara (keypress)
22 {
23     // telefone 1 - bloqueia letras e insere na máscara automaticamente
24     if(keypress>=48 && keypress<=57)
25     {
26         separador1 = '(';
27         separador2 = ')';
28         separador3 = '-';
29         conjunto1 = 0;
30         conjunto2 = 3;
31         conjunto3 = 9;
32         if (eval(document.cadastro.telefone.value.length) == conjunto1)
33         {
34             document.cadastro.telefone.value = document.cadastro.telefone.value + separador1;
35         }
36         if (eval(document.cadastro.telefone.value.length) == conjunto2)
37         {
38             document.cadastro.telefone.value = document.cadastro.telefone.value + separador2;
39         }
40         if (eval(document.cadastro.telefone.value.length) == conjunto3)
41         {
42             document.cadastro.telefone.value = document.cadastro.telefone.value + separador3;
43         }
44         return true;
45     }
46     else
47     {
48         return false;
49     }
50 }
51 }
```

Figura 4 - Método para mascarar o telefone

**Linha 24:** verifica se o parâmetro enviado está dentro da sequência numérica da tabela ASC, caso esteja continua a partir da linha 26, caso não esteja, retorna “false”(linhas 46 a 49) impedindo que está tecla seja digitada dentro da caixa de texto.

**Linhas 26 a 28:** atribui valores a variáveis, sendo: abertura e fechamento dos parênteses do DDD e o traço que separa o prefixo do sufixo.

**Linhas 29 a 31:** atribui valores a variáveis que são os “pontos de parada” para concatenação de um dos símbolos criados nas linhas 26 a 28.

**Linha 32:** verifica se o tamanho (length) da “digitação na caixa de texto” até então, atingiu o “primeiro ponto de parada, se sim, vai para linha 34.

**`eval(document.cadastro.telefone.value.length) == conjunto1`**

**eval:** Converte para valor numérico

**document:** referência JavaScript a um objeto na página

**cadastro:** propriedade “name” do formulário criado

**telefone:** propriedade “name” da caixa de texto

**value.length:** quantidade de caracteres dentro desta caixa de texto (até o momento)

**Linha 34:** concatena na caixa de texto o “separador do momento”.

E assim vai para os “if”s das linhas 36 e 40. Ao concluir as concatenações, assume o retorno “True” (linha 44) permitindo que está tecla seja digitada dentro da caixa de texto.

---