Criando máscaras de input com jQuery Mask Plugin

31/08/2017

Em muitos sistemas/páginas precisamos criar máscara em campos de entrada de informações para garantir que estas estejam no formato esperado. Para facilitar esta atividade podemos utilizar o plugin [jQuery Mask Plugin](https://igorescobar.github.io/jQuery-Mask-Plugin/" \t "_blank) que facilita bastante.

Primeiramente, para utilizamos o plugin devemos baixa-lo, e para isto podemos baixar o zip pelo link fornecido pela página do plugin, ou clonar o repositório.

|  |  |
| --- | --- |
| 1 | git clone https://github.com/igorescobar/jQuery-Mask-Plugin |

Depois precisaremos inserir o arquivo do plugin na nossa página, este se encontra no diretório *dist* do arquivo baixado, e adicionar também o plugin base do *jQuery*.

|  |  |
| --- | --- |
| 1  2 | <script type="text/javascript" src="jquery-3.2.1.min.js"></script>  <script type="text/javascript" src="jquery.mask.min.js"></script> |

Neste ponto já podemos utilizar o plugin para criar uma máscara em algum campo *input*, por exemplo colocar uma mascara de data nos campos com *class=”data”*.

|  |  |
| --- | --- |
| 1  2  3 | $(document).ready(function(){      $('.data').mask('00/00/0000');  }); |

Como podemos ver basta utilizar um seletor para o campo de queremos inserir a máscara e passar por parâmetro o formato:

‘0’: Um digito obrigatório  
‘9’: Um digito opcional  
‘#’: Um digito com recurção  
‘A’: Uma letra de a até z (maiúsculas ou minusculas) ou um digito  
‘S’: Uma letra de a até z (maiúsculas ou minusculas) sem digito

Com os símbolos acima podemos criar diversos tipos de máscaras, sendo que qualquer simbolo que não for um dos acima será adicionado automaticamente na máscara. Podemos ver exemplos de máscaras abaixo:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | $('.data').mask('00/00/0000');  $('.tempo').mask('00:00:00');  $('.data\_tempo').mask('00/00/0000 00:00:00');  $('.cep').mask('00000-000');  $('.tel').mask('00000-0000');  $('.ddd\_tel').mask('(00) 0000-0000');  $('.cpf').mask('000.000.000-00');  $('.cnpj').mask('00.000.000/0000-00');  $('.dinheiro').mask('000.000.000.000.000,00' , { reverse : true});  $('.dinheiro2').mask("#.##0,00" , { reverse:true}); |

Como você deve ter notado, nas últimas duas máscaras, é passado um objeto como segundo parâmetro com a propriedade *reverse: true*, isto é feito para que a máscara seja preenchida da direita para a esquerda e não da esquerda para a direita com é o padrão, isso é feito nestas máscaras para primeiro ser digitado os centavos e depois os valores inteiros.

Um recurso interessante deste plugin é que podemos customizar os símbolos criando nossas próprias expressões regulares para eles:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | $('.cor\_hex').mask('#xxxxxx' , {                                      translation: {                                          'x': {                                              pattern: /[a-fA-F0-9]/                                          },                                          '#' : ''                                      }                                  }); |

Na máscara acima é passado um segundo objeto de parâmetro com o atributo *translation* que terá como atributos os novos símbolos para nossa máscara. Por exemplo, adicionamos um simbolo ‘x’ que ira aceitar as letras de ‘a’ até ‘f’ (tanto maiúsculas ou minusculas) e dígitos de 0 até 9. Para nossa máscara aceitar o caractere ‘#’ como um literal ao invés do simbolo padrão para o digito recursivo, o sobrescrevemos.

Também é possível passar um *placeholder* como um parâmetro para a máscara.

|  |  |
| --- | --- |
| 1 | $('.placeholder').mask("00/00/0000", { placeholder: "\_\_/\_\_/\_\_\_\_" }) ; |

Caso seja necessário recuperar o valor do campo sem a máscara, pode-se utilizar a função *cleanVal*.

|  |  |
| --- | --- |
| 1 | var unmask\_value = $('.data').cleanVal(); |