**用法**

所用到的js文件你都能在dist文件夹里找到

Dom使用方法

<script src="jquery.js" type="text/javascript"></script>

<script src="inputmask.js" type="text/javascript"></script>

<script src="inputmask.???.Extensions.js" type="text/javascript"></script>

通过jquery

<script src="jquery.js" type="text/javascript"></script>

<script src="inputmask.js" type="text/javascript"></script>

<script src="jquery.inputmask.js" type="text/javascript"></script>

<script src="inputmask.???.Extensions.js" type="text/javascript"></script>

或者通过全集成的js

<script src="jquery.js" type="text/javascript"></script>

<script src="jquery.inputmask.bundle.js" type="text/javascript"></script>

**默认掩码定义**

9：数字

a：字母

\*：数字字母

在扩展中还有更多的定义

**掩码类型**

**静态掩码**

这是最基础的掩码方式，输入的长度跟掩码的长度一样

$(document).ready(function(){

$(selector).inputmask("aa-9999"); //静态掩码

$(selector).inputmask({mask: "aa-9999"}); //静态掩码

});

**可选掩码**

定义一部分可选的掩码，使用[]包裹

例子：

$('#test').inputmask('(99) 9999[9]-9999');

这个掩码允许输入 (99) 99999-9999 或者 (99) 9999-9999

输入 => 12123451234 mask => (12) 12345-1234 (触发 complete)  
输入=> 121234-1234 mask => (12) 1234-1234 (触发complete)  
输入=> 1212341234 mask => (12) 12341-234\_ (触发incomplete)

**跳过可选部分字符**

可以配置按什么字符时可跳过可选部分

skipOptionalPartCharacter:””

输入 => 121234 1234 mask => (12) 1234-1234 (触发complete)

当clearMaskOnLostFocus设置为true时（默认），在可选部分在结尾部位时且你没填充可选部分，失去焦点时会自动清除掉可选的掩码部位

**可选掩码显示greedy**

默认为true，当设置为false时，input上将只显示最简单的掩码，可选部分不会显示出来

$(selector).inputmask({ mask: "9[-9999]", greedy: false });

将只显示 \_ ，本来是 \_-\_\_\_\_

**动态掩码**

动态部分使用{}

{n} 前面的掩码得输入n次

{n,m} 前面的掩码得输入n到m次

{+}代表前面的掩码至少得输入1次，{\*}代表前面的掩码至少0次

$(selector).inputmask("aa-9{1,4}"); =>ab-12 或者 ab-123 或者 ab-1234 或者 ab-12345

**交换掩码**

使用符号 |

a|9 => b 或者 1

(aaa)|(999)

**预处理掩码**

Mask可以定义为一个function

**定义自定义掩码**

**validator**(chrs,maskset,pos,strict,opts)

返回值 为true或false或者一个命令对象

chrs你输入的字符

maskset 目前只发现里面的buffer数组有用处，是输入值组成的数组，console.log一下，你就清楚了

pos 当前输入的位置 从0开始

strict 目前还不知是什么

opts 插件的选项

**命令对象选项**

pos

c

caret

remove

insert

refreshFromBuffer

**cardinality 占位数**

这个指定掩码占多少个字符

**prevalidator**(chrs,maskset,pos,strict,opts)

在cardinality占位数达到前进行输入的限制

**definitio**nSymbol 暂时没发现有什么用

我发现官方文档的例子有错，用官网的例子，IP地址的掩码无法使用，在validator方法里面的所用到的buffer，应该得用buffer.buffer才正常

**placeholder**

默认为\_

**set defaults 设置默认值**

Inputmask.extendDefaults({placeholder:”1”});//设置默认参数

Inputmask.extendDefinitions({});//设置自定义掩码

Inputmask.extendAliases({}); //设置别名

继承别名

Inputmask.extendAliases({

"myMask":{

alias:"numeric",

placeholder:" ",

…其它属性

}

});

$(selector).inputmask("myMask");

所有的事件回调函数都在选项中设置

**方法**

**mask**

创建一个输入限制的input

$(selector).inputmask({mask:"99-9999"});

或者

Inputmask({mask:"99-9999"}).mask(document.querySelectorAll(selector));

或者

var im=new Inputmask("99-9999");

im.mask(document.querySelectorAll(selector));

或者

Inputmask("99-999").mask(selector);

**unmaskedvalue 取得输入的原始值**

var value=$(selector).inputmask("unmaskedvalue");

**Value unmasking 将值与掩码匹配，取得原始值**

Inputmask.unmask("23/03/1973",{alias:"dd/mm/yyyy"});//23031973

Inputmask.unmask("12-345","99-999");//12345

**Remove 移除inputmask**

$(selector).inputmask("remove");

或者

var input = document.getElementById(selector);

if(input.inputmask)

input.inputmask.remove();

或者

Inputmask.remove(document.getElementById(selector));

**getemptymask 取得空的掩码值数组**

$(selector).inputmask("getemptymask");

**hasMaskedValue 是否使用掩码组件**

$(selector).inputmask("hasMaskedValue"); //true或者false

**isComplete 验证是否通过符合掩码**

$(selector).inputmask("isComplete"); //true或者false

**getmetadata 取得掩码值**

$(selector).inputmask("999-aaa");

$(selector).inputmask("getmetadata");//999-aaa

**setvalue 设置值**

$(selector).inputmask("setvalue","123123");

或者

$(selector).val("123123213");

**option 获取或者设置选项的值**

$(selector).inputmask("option","placeholder");//取得placeholder的值

$(selector).inputmask("option",{

placeholder:" "

});

//设置选项placeholder的值

**format 按掩码格式返回值**

Inputmask.format("10092015",{alias:"dd/mm/yyyy"});//10/09/2015

Inputmask.format("20150910","y-m-d");//2015-09-10

**isValid 判断一个值与掩码是否匹配**

Inputmask.isValid("23/03/2001",{alias:"dd/mm/yyyy"}); //true或者false

**Options 选项**

**placeholder 占位符**

$(selector).inputmask("yyyy-mm-dd",{placeholder:"\*"});//单个

$(selector).inputmask("y-m-d",{placeholder:"yyyy-mm-dd"});//多个

**optionalmarker 定义可选掩码符号**

optionalmarker:{start:"[",end:"]"}

**quantifiermarker 定义重复掩码的符号**

quantifiermarker: { start: "{", end: "}" }

**groupmarker 定义组符号**

groupmarker: { start: "(", end: ")" },

**alternatormarker 定义交换符号**

alternatormarker: "|",

**escapeChar 转义符号**

escapeChar:"\\"

**mask**

**oncomplete 当字符匹配后触发事件**

**onincomplete 当字符还未匹配，且失去焦点时**

**oncleared 当清空时**

**repeat 指定 mask重复次数**

**greedy 前面介绍过**

**autoUnmask 提交时是否自动去掉掩码**

默认是false 如果设置为true，比如 12-345 提交时或val()时得到的值是12345

**removeMaskOnSubmit 在表单提交之前移除inputmask**

**clearMaskOnLostFocus 失去焦点时隐藏掩码格式**

默认为true

**insertMode 输入模式**

默认为true 为插入 false为重写 功能没区别，只是看起来不一样而已

**clearIncomplete 在失去焦点时且未匹配时清除值**

默认为false

**aliases 别名**

$(selector).inputmask("date");//这儿的date就是定义的一个别名掩码

或者

$(selector).inputmask({alias:"date"});

使用并扩展或重写别名属性

$(selector).inputmask("date",{clearIncomplete:true});

或者

$(selector).inputmask({alias:"date",clearIncomplete:true});

**onKeyDown 输入事件**

function(event,buffer,caretPos,ops){}

**onBeforeMask 在进掩码格式之前**

function(initialValue,opts){}

initialValue 初始值

return 返回一个处理过的值做为input的值

**onBeforePaste 在粘贴之前**

function(pasteValue,opts){}

pasteValue 粘贴的值

return 返回一个处理过的值做为粘贴值

**onBeforeWrite 输入之前触发**

**onUnMask 在unmaskedvalue操作前触发**

**showMaskOnFocus 获得焦点时显示掩码**

默认为true

**showMaskOnHover 经过时显示掩码**

默认为true

**onKeyValidation 当按下键后，验证事件**

function(key,result){}key为输入的字符对应在ascii编码，result为key值是否符合掩码，不符合为false，符合为一个对象{pos:位置}

**skipOptionalPartCharacter 这个前面有说过了**

**showTooltip 鼠标放在input上面会提示掩码**

默认为false，如果tooltip没设置则为默认的参数中的掩码

**tooltip 提示语**

默认为参数中的掩码

**numericInput 按照数字输入方向**

默认为false，设置为true后，输入方向由右边开始

**rightAlign 内容右对齐**

默认为false，在某些数字别名中默认为true

**undoOnEscape 按esc会撤销刚才输入的**

默认为true，不过我设置为false也跟true一样

**radixPoint 指定小数点用什么符号表示**

**groupSeparator 指定数字类型中分隔符号**

默认为, 在currency 中有效果

**radixFocus**

好像是可以直接点击小数点后面

**nojumps 不自动跳过固定的掩码**

默认为false

**nojumpsThreshold 不知干什么用的**

**keepStatic 说不清，看例子**

$(selector).inputmask({ mask: ["+55-99-9999-9999", "+55-99-99999-9999", ], keepStatic: true });

输入 1212345123 => 结果 +55-12-1234-5123 再输入4 => 变成 +55-12-12345-1234

**positionCaretOnTab 不知有什么用**

**tabThrough 能过tab键跳过固定的掩码**

**definitions**

**ignorables**

**isComplete 输入每个正确的值时都会触发**

function(buffer,option){}

**canClearPosition 删除时触发**

function(maskset, position, lastValidPosition, opts,option){}

返回true或false

**postValidation 验证每个位置时触发**

function(buffer, opts){}

返回true或false

**staticDefinitionSymbol 不懂**

**Supported markup options 支持的标签选项**

**RTL attribute 反方向输入**

<input id="test" dir="rtl" />

**readonly attribute 只读**

<input id="test" readonly="readonly" />

**disabled attribute 禁用**

<input id="test" disabled="disabled" />

**maxlength attribute 最大输入长度**

<input id="test" maxlength="4" />

**data-inputmask attribute 使用data-inputmask属性**

你也可以使用data-inputmask属性，它将自动转换为json对象做为inputmask的选项

<input data-inputmask="'alias': 'date'" />

<input data-inputmask="'mask': '9', 'repeat': 10, 'greedy' : false" />

$(document).ready(function(){

$(":input").inputmask();

});

**data-inputmask- attribute 使用data-inputmask-属性名**

所有的选项都能被定义在data-inputmask-属性名中

<input data-inputmask-mask="9" data-inputmask-repeat="10"

data-inputmask-greedy="false" />

$(document).ready(function(){

$(":input").inputmask();

});