1. 红宝书中的第14章表单脚本
2. 获取表单

var form=document.getElementById("form1");

var firstForm=document.forms[0];

var myForm=document.forms["myForm"];

1. 提交表单

<!--通用提交按钮-->

<input type="submit" value="Submit Form"/>

<!--自定义提交按钮-->

<button type="submit">Submit Form</button>

<!--图像按钮-->

<input type="image" src="pic.png"/>

阻止submit事件的默认行为就会取消表单提交

EventUtil.addHandler(oForm,"submit",function(event){

event=EventUtil.getEvent(event);

EventUtil.preventDefault(event);

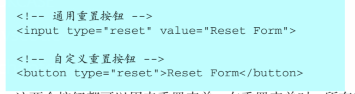
})

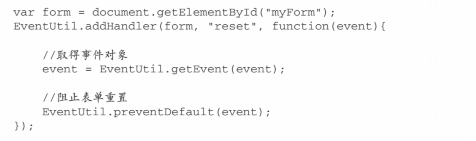
oForm.submit();

在以调用submit()的形式提交表单时，不会触发submit事件

避免重复提交表单方法：在第一次提交表单后禁用按钮，利用onSubmit事件处理程序取消后续的表单操作

1. 重置表单







1. 表单元素

//取得表单中的第一个字段

var field1=oForm.elements[0];

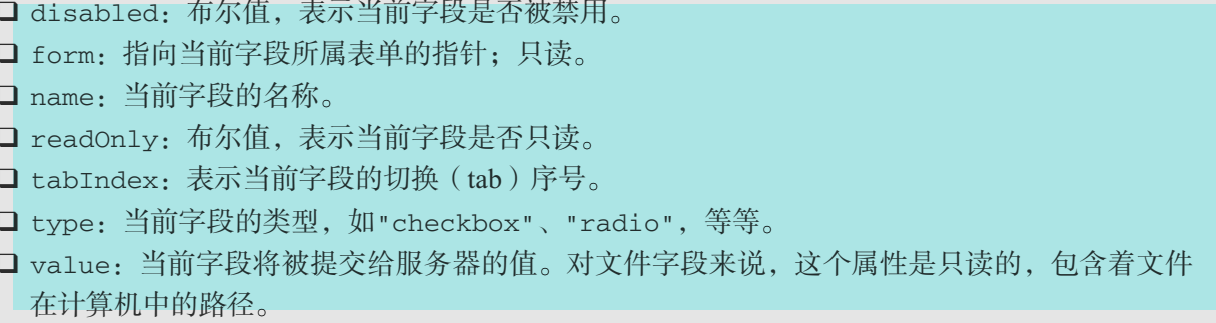
//取得名为color的字段

var aColor=oForm.elements["color"];

//取得表单中包含字段的数量

var iFieldCount=oForm.elements.length;

1. 表单字段属性



var oForm=document.getElementById("myForm");

var field=form.elements[0];

field.value="another value";

console.log(field.form===oForm);

field.focus();

field.disabled=true;

field.type="checkbox";//不推荐

避免多次提交-submit事件中禁用提交按钮

EventUtil.addHandler(oForm,"submit",function(event){

event=EventUtil.getEvent(event);

var target=EventUtil.getTarget(event);

var btn=target.elements["submitBtn"];

btn.disabled=true;

})

focus()获取焦点 不能为不可见的元素使用这个方法

blur()移除焦点

<input type="text" autofocus />

自动获取焦点

1. change事件

对于input和textarea元素，在它们失去焦点且value值改变时触发，对于select元素，在其选项改变时触发

1. 文本框脚本

单行文本：

<!--显示25个字符 输入不能超过50个-->

<input type="text" size="25" maxlength="50" value="initial value">

多行：

<textarea id="" cols="30" rows="10">initial value</textarea>

1. 选择文本

select() 选中所有文本

select事件-选择文本框的文本时就会触发

EventUtil.addHandler(oText,"select",function(event){

console.log(oText.value);

})

在调用select()方法时也会触发select事件

获得选取的文本

function getSelectedText(textBox){

if(typeof textBox.selectionStart=="number"){

return textBox.value.substring(textBox.selectionStart,textBox.selectionEnd);

}

else if(document.selection){

return document.selection.createRange().text;

}

}

选取文本

function selectText(textBox,startIndex,endIndex){

if(textBox.setSelectionRange){

textBox.setSelectionRange(startIndex,endIndex);

}

else if(textBox.createTextRange){

var range=textBox.createTextRange();

range.collapse(true);

range.moveStart("character",startIndex);

range.moveEnd("character",endIndex-startIndex);

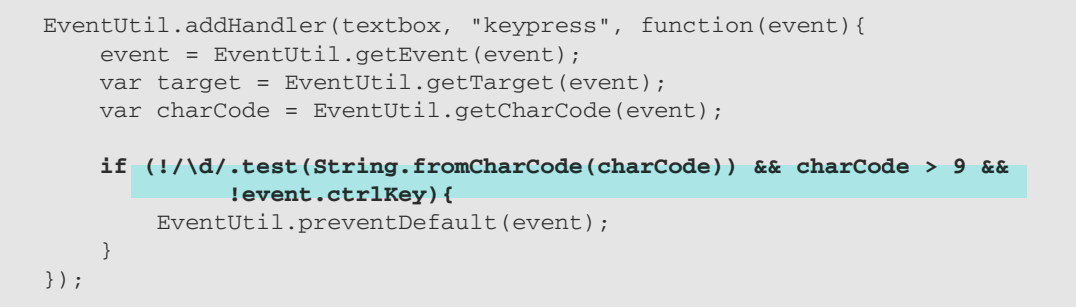
range.select();

}

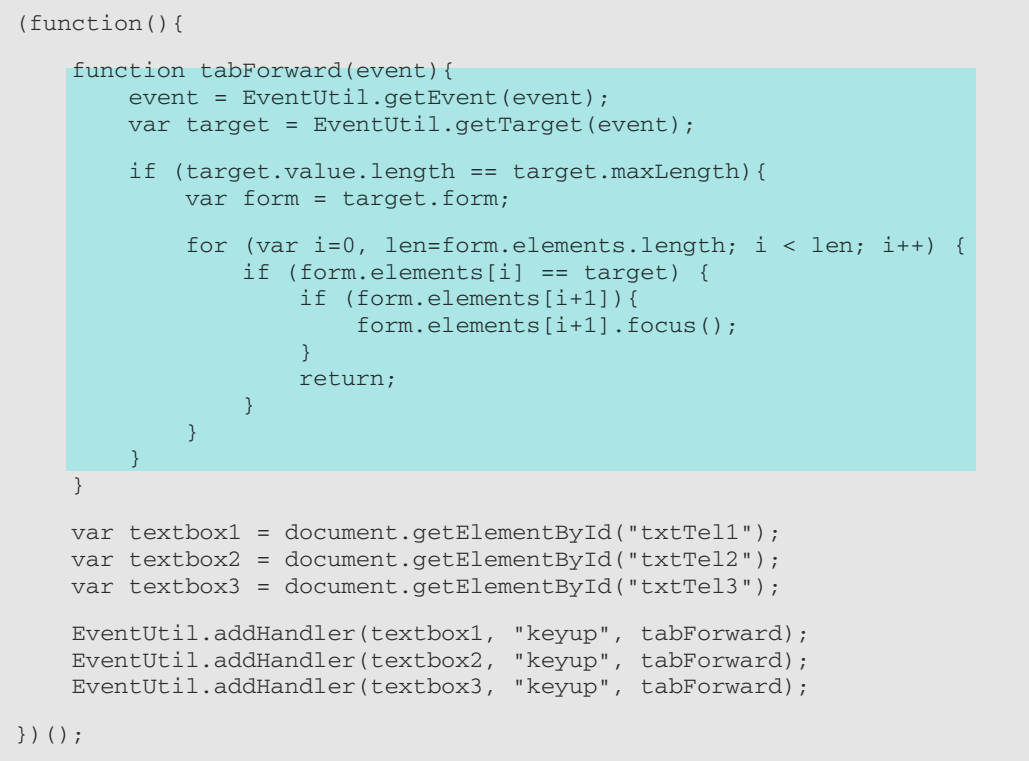
textBox.focus();

}

过滤输入



自动切换焦点



1. 约束api

必填字段

<input type="text" required>

输入类型：email url

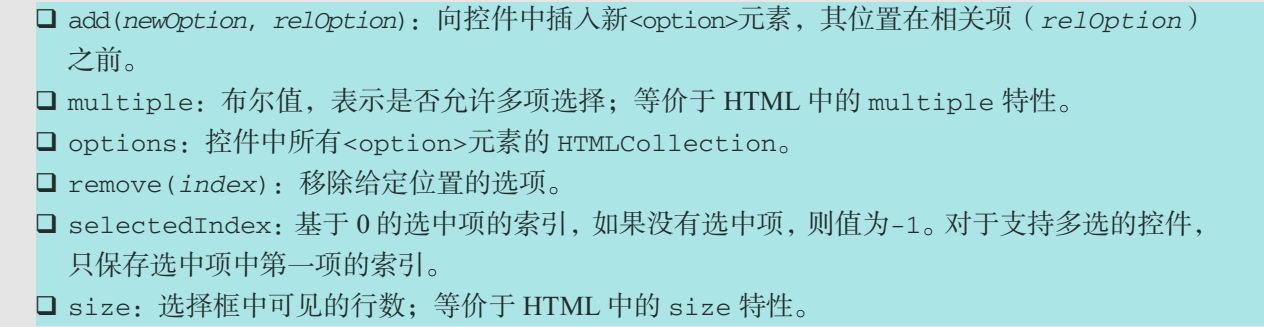
输入模式-pattern

检测有效性-checkValidity() validity

禁用验证-novalidate

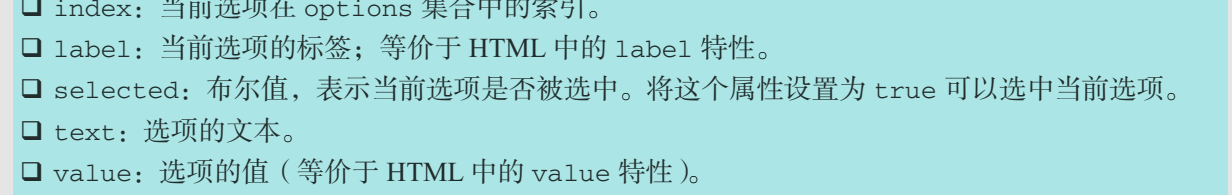
1. 选择框脚本

HTMLSelectElement的属性和方法



选择框的type属性，不是”select-one”，就是”select-multiple”

HTMLOptionElement的属性和方法



<select name="location" id="select1">

<option value="option1">choose1</option>

<option value="">choose2</option>

<option>choose3</option>

</select>

<script>

var oSelect=document.getElementById("select1");

console.log(oSelect[0].value);//option1

console.log(oSelect[1].value);//""

console.log(oSelect[2].value);//choose3

</script>

var text=oSelect.options[0].text;

var value=oSelect.options[0].value;

selectedIndex被选中元素的index

selected布尔值 元素是否被选中

getSelectedOptions(oSelect) 获取被选中的数组

添加选项

var oSelect=document.getElementById("select1");

var newOption=document.createElement("option");

newOption.appendChild(document.createTextNode("new choose"));

newOption.setAttribute("value","new select");

oSelect.appendChild(newOption);

var oSelect=document.getElementById("select1");

var newOption=new Option("new choose","new select");

oSelect.appendChild(newOption);//ie8及以前有问题

添加到其他位置用insertBefore

移除选项

oSelect.removeChild(oSelect.options[0]);

oSelect.remove(0);

oSelect.options[0]=null;

表单序列化

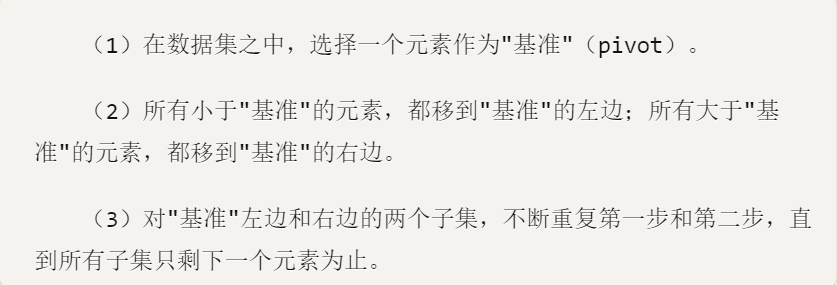
富文本编辑

1. vscode
2. 整段移动

选中按TAB右移，按SHIFT+TAB左移

1. 博客

1.快速排序



function quickSort(arr){

if(arr.length<=1){

return arr;

}

var pivotIndex=Math.floor(arr.length/2);//基准数的index

var pivot=arr.splice(pivotIndex,1)[0];

var left=[];

var right=[];

for(var i=0;i<arr.length;i++){

if(arr[i]<pivot){

left.push(arr[i]);

}

else{

right.push(arr[i]);

}

}

return quickSort(left).concat(pivot,quickSort(right));

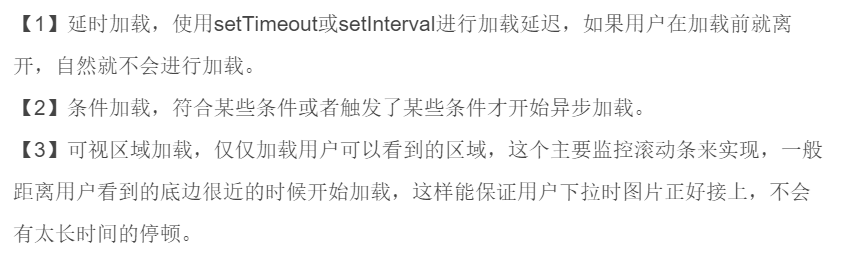
}

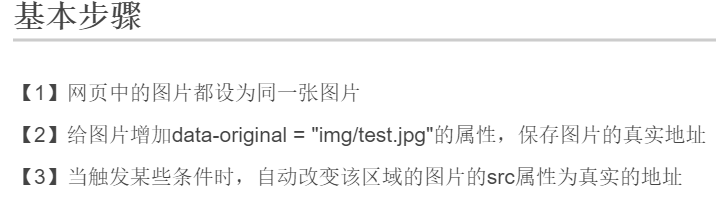
1. 字符串替换

str.replace(a,b);

1. 图片懒加载

图片延迟加载也称为懒加载，延迟加载图片或符合某些条件时才加载某些图片，通常用于图片比较多的网页。可以减少请求数或者延迟请求数，优化性能。





点击按钮显示图片

<button>加载图片</button>

<img src="#" alt="测试" data-original = "img/test.png">

<script>

var oBtn = document.getElementsByTagName('button')[0];

var oImg = document.images[0];

oBtn.onclick = function(){

oImg.src = "img/loading.gif";

if(oImg.dataset){

aftLoadImg(oImg,oImg.dataset.original);

}else{

aftLoadImg(oImg,oImg.getAttribute("data-original"));

}

}

function aftLoadImg(obj,url){

var oImg = new Image();

oImg.onload = function(){

obj.src = oImg.src;

}

oImg.src = url;

}

可视区显示图片

var oBtn = document.getElementsByTagName('button')[0];

var aImages = document.images;

loadImg(aImages);

window.onscroll = function(){

loadImg(aImages);

};

function loadImg(arr){

for( var i = 0,len = arr.length; i < len; i++){

if(arr[i].getBoundingClientRect().top < document.documentElement.clientHeight && !arr[i].isLoad){

arr[i].isLoad = true;

arr[i].style.cssText = "transition: ''; opacity: 0;"

if(arr[i].dataset){

aftLoadImg(arr[i],arr[i].dataset.original);

}else{

aftLoadImg(arr[i],arr[i].getAttribute("data-original"));

}

(function(i){

setTimeout(function(){

arr[i].style.cssText = "transition: 1s; opacity: 1;"

},16)

})(i);

}

}

}

function aftLoadImg(obj,url){

var oImg = new Image();

oImg.onload = function(){

obj.src = oImg.src;

}

oImg.src = url;

}

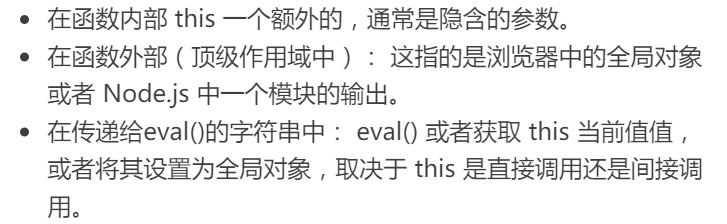
1. css选择器

CSS的一个核心特性是能向文档中的一组元素类型应用某些规则

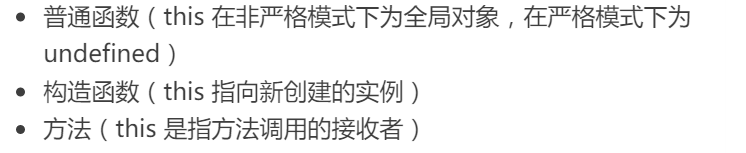
<http://www.cnblogs.com/xiaohuochai/p/4979514.html>

1. this

分类



* 1. 在函数中

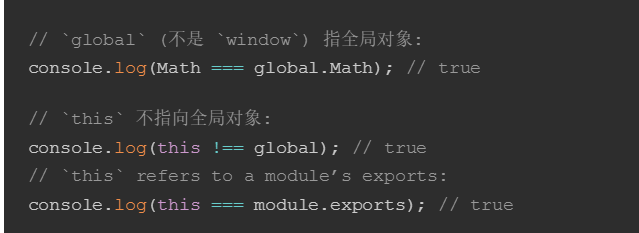


在普通函数中，可以通过call或apply进行函数调用，明确指定this的值

* 1. 在顶级作用域中

在浏览器环境下，this===window

在node环境下



* 1. 在eval中

间接调用eval()，this指向全局对象

直接调用，this与eval中的环境保持一致