p标签: 会忽略换行和多余空格 换行用br

</pre>会保留换行和空格 适合显示计算机代码

<hr />水平线

中文乱码: 加入<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">

<big>大号字 <em>着重字 <sub>下标子 <sup>上标字

邮件:<a href="mailto:hnbhnb0001@126.com?cc=someoneelse@microsoft.com&bcc=andsomeoneelse2@microsoft.com&subject=Summer%20Party&body=You%20are%20invited%20to%20a%20big%20summer%20party!">其中20%用以代替空格，cc抄送，bcc密送，subject主题，body内容``

<address>地址样式</address>

鼠标移到缩略词上显示详细文本: <abbr title="etcetera">etc.</abbr> <acronym title="World Wide Web">WWW</acronym> 在IE5中仅acronym有效

<bdo dir="rtl">如果浏览器支持bro(bi-directional override)从右向左输出</bdo>

长引用:<blockquote>自动换行和外边距</blockquote> 短引用:<q>自动加引号</q>

显示计算机代码格式的标签<kbd><samp><code>

<del>删除字效果</del> <ins>插入字效果</ins>

<a target="\_blank">在新窗口中打开链接

<a href="#C4">跳到本页某处的链接</a> <h2><a name="C4">这是要跳的那一处</a></h2> 网址中加#C4也可以

<a href="/index.html" target="\_top">跳出框架</a>

垂直框架<frameset cols="25%,50%,25%"> <frame src="/example/html/frame\_a.html" noresize="noresize"><frame src="/example/html/frame\_b.html"><frame src="/example/html/frame\_c.html"> </frameset>其中cols改为rows则为水平框架，noresize="noresize”表示不可拖拉,frameset不能放在body中，能放在header中

混合框架<frameset rows="50%,50%"><frame src="/example/html/frame\_a.html"><frameset cols="25%,75%"><frame src="/example/html/frame\_b.html"><frame src="/example/html/frame\_c.html"></frameset></frameset>

导航框架:<frameset cols=“120,\*"><frame src="/example/html/html\_contents.html"><frame src="/example/html/frame\_a.html" name="showframe"></frameset>其中120是像素, 注:导航框架网页中链接的target=“show frame”

内联框架，老的浏览器可能不支持<iframe src="/i/eg\_landscape.jpg"></iframe>

table: 边框大小:border=“8”默认是”0” 表头<th> <caption>表标题</caption> <th rowspan=“2”>跨两行</th> <th colspan="2">跨两列</th> 单元格边距<table cellpadding=“10”> 单元格边框大小cellspacing 表格背景颜色或图像<table bgcolor="red" background="/i/eg\_bg\_07.gif">单元格颜色图像同理 单元格属性align table边框样式:属性frame=“box”、”above”、”below”、”hsides”、”vsides”

列表:<ul><li>咖啡</li></ul> 其中ul改ol则为有序列表，ul元素(圆点)符号:属性type=“disc”(实心圆点)、”circle”(空心圆点)、”square”(实心方点) ol元素符号:type=“A”(大写字母列表)、”a”(小写字母列表)、”I”(大写罗马字母列表)、”i”(小写罗马字母列表) 嵌套列表的元素符号默认与父级不同 定义列表<dl><dt></dt><dd></dd></dl>

form表单:

input属性type=“password”、”checkbox”、”radio”、”button”、number

下拉框<select name="cars"><option value="volvo">Volvo</option></select> option属性selected="selected”默认选项

<fieldset><legend>边框</legend></fieldset> 其中legend是说明以哪个为基准

提交表单<form action=“action.php” method=“get”><input type=“submit” value=“submit”></form>

body属性:background、bgcolor=“#d0d0d0” img属性:align(默认bottom) alt=“无法显示图像时的替换文本”

<a href=“…”><img src="/i/eg\_buttonnext.gif" /></a>图像链接

图片点击区域映射<img usemap=“#planetmap” src="./uploads/1242\_2208.png" width="145" height="126"/><map name="planetmap"><area shape=“circle” coords=“0,0,82,126” href=“…”/></map>

<a href=“…”><img src=“…” ismap /></a>鼠标移动到图片上可以看到状态栏的坐标变化

css样式: <head><style type="text/css">h1 {color: red} p {color: blue}</style></head>

<a href=“…” style="text-decoration:none">无下划线的链接，否则会默认带下划线</a>

head中引入外部样式<link rel="stylesheet" type="text/css" href="/html/csstest1.css" >

head中<title>网页标题</title> <base target="\_blank" />所有链接都将在新窗口中打开 重定向<meta http-equiv="Refresh" content="5;url=http://www.w3school.com.cn" />其中5表示5秒后跳转

<noscript>不支持JavaScript的浏览器将显示该文本</noscript>

css: <style type=“text/css”>body {background-color: yellow} h1 {background-color: #00ff00} h2 {background-color: transparent} p.no2 {background-color: rgb(250,0,250)} </style> 调用p.no2: <p class=“no2”>…</p>

div与span区别: 前者头尾自动换行，后者不会

背景图片:body {background-image:url(test.png);(注:若显示不出来再试试background-image: url("../images/bg3.jpg");) background-repeat: no-repeat;}其中no-repeat改为repeat则重复图片，默认是repeat; 只垂直重复图片repeat-y; 只水平重复图片repeat-x 背景图上放置图像background-attachment:fixed;background-position:center;或background-position: 30% 20%;或 50px 100px； 大小background-size: 50% 50%; fixed为适应firefox和Opera且背景图不随着滚动

所有属性写在一个声明中:body{ background: #ff0000 url(/i/eg\_bg\_03.gif) no-repeat fixed center; }

背景图完全显示不考虑变形background-size: 100% 100%;

字符间距{letter-spacing: 2px} 行间距{line-height: 200%(或10px或2)}

对齐:{text-align: center}

文本修饰线:{text-decoration: overline} line-through underline blink none

文本缩进:{text-indent: 1cm}

大小写:{text-transform: uppercase} lowercase capitalize(自动首字母大写)

文本不折行，替换为空格，p标签和textarea等通用{white-space: nowrap}

单词间距:{word-spacing: 30px(或-0.5em);}注: 这是单词间距，区别于letter-spacing

字体系列:p.serif{font-family:"Times New Roman",Georgia,Serif}按顺序如果没有一种字体，用下一种字体

字体大小:{font-size: 200%} 1em=16px

字体风格:{font-style:normal} italic斜体 oblique倾斜

小型大写字母{font-variant:small-caps;}

加粗{font-weight:900;} 400=normal 700=bold

一个声明内:{font:italic bold 12px/30px arial,sans-serif;} 其中12px/30px分别是大小和行距

链接: a:link {color:#FF0000;}/\* 未被访问的链接 \*/ a:visited {color:#00FF00;}/\* 已被访问的链接 \*/ a:hover {color:#FF00FF;}/\* 鼠标指针移动到链接上 \*/a:active {color:#0000FF;} /\* 正在被点击的链接 \*/ 注:这个顺序不能变

ul{list-style-type: disc} circle square none decimal decimal-leading-zero(0开头的数字标记) lower-roman upper-roman lower-alpha upper-alpha lower-greek(希腊字母) lower-latin upper-latin 其他:http://www.w3school.com.cn/cssref/pr\_list-style-type.asp

ol{list-style-type: decimal} lower-roman upper-roman lower-alpha upper-alpha

图像作列表符号ul {list-style-image: url('/i/eg\_arrow.gif')}

符号位置: ul {list-style-position: inside} outside

一个声明: ul {list-style: square inside url('/i/eg\_arrow.gif')}

table,th,td{border:1px solid blue;}

table重叠边框table {border-collapse:collapse;}

table {width:100%;} th{height:50px;}

td{text-align:right; vertical-align:bottom; padding:15px; background-color:green; color:white;}

table中的单元格自适应内容大小table {table-layout: automatic/(fixed)}

单为空白单元格时table {empty-cells:hide;}

p {border: medium double rgb(250,0,255)}其中border-style double(dotted,dashed, solid,groove,ridge,inset,outset) border-width border-color border-bottom border-bottom-style

p{margin-left:2cm}

外边距合并: 两个标签的外边距相遇时，合并外边距，取最大值

定位:p {position:relative;left:-20px}相对于正常位置左边移-20px，理解:内正外负，理同top、right、bottom

h1 {position:absolute;left:100px;top:150px}相对于页面

fixed: 相对于正常位置，上右下左不是移动距离

浮动:img {float:right} 也可用于span和div等

li为行内元素(不换行): li {display:inline}

div{ clear:left; } 清楚左侧相邻元素的浮动

通配符选择器: \* {color:red;}

类选择器: .class{…}

多类选择器: .important.warning {background:silver;} <p class="important warning">… 注: 在 IE7 之前的版本中，不同平台的 Internet Explorer 都不能正确地处理多类选择器。

id选择器p#intro {font-weight:bold;} <p id="intro">…</p>

em标签: 斜体显示重要文本。 效果同i标签

属性选择器\*[title] {color:red;} a[href][title]{color:red;}

自定义属性样式img[alt]{border: 5px solid red;}

根据具体属性值选择a[href="http://www.w3school.com.cn"] {color: red;}

p[class~="important"]{color: red;}可匹配<p class="important warning">和<p class="important">

子串匹配属性选择器：a[href\*="w3school.com.cn"] {color: red;}( 匹配属性值中包含指定值的每个元素。)

特定属性选择类型：\*[lang|="en"] {color: red;}(匹配en或以en开头的lang属性)

后代选择器: h1 em {color:red;}( 只对 h1 元素中的 em 元素应用样式,不管中间的嵌套)

子元素选择器: h1 > strong {color:red;}相比后代选择器有严格的层次

相邻兄弟选择器: h1 + p {margin-top:50px;}( 选择紧接在 h1 元素后出现的段落，h1 和 p 元素拥有共同的父元素) //没有h1 - p

伪类: a:link {color: #FF0000} visited hover active first-child

p > i:first-child { font-weight:bold; }

p:first-child i {color:blue;}, 示例: <div><p>1</p><p>2</p></div>

lang伪类q:lang(no){quotes: "~" "~"} <q lang="no">…</q>

:first-line 伪元素 :first-letter 伪元素

:before{content:url("./uploads/uncheck.png");} 伪元素 :在元素的内容前面插入新内容。

:after 伪元素: 在元素的内容之后插入新内容

居中.center{margin:auto;} IE5:用text-align:left;

p{max-height: 10px}

内联元素p {display: inline} 不显示的元素div {display: none}//布局也不保留

块级元素span{display: block}

使元素不可见: {visibility:visible} hidden//布局保留

<span style="cursor:auto">光标移到这里的形状 </span> cursor: crosshair(加号)、default(默认箭头)、pointer(手指)、move(十字箭头)、e-resize(竖线东箭头)、右(ne-resize)东北箭头、nw-resize (西北箭头)、n-resize(竖线北箭头)、se-resize(东南箭头)、(sw-resize)西南箭头、(s-resize)竖线南箭头、(w-resize  
text)竖线西箭头、(text)工字光标、(wait)刷新、(help)问号、not-allowed禁用

ul{overflow:hidden;} 防止 li 元素出现在列表之外。

html4 : <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/html4/loose.dtd"> <meta http-equiv="Content-Type" content="text/html;charset=utf-8">

html5: <!DOCTYPE html> <meta charset="utf-8"> <!--[if lt IE 9]>

<script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script>

<![endif]-->

语义元素: section article header footer nav(链接列表) aside(元素页面主内容之外的某些内容) <figure><figcaption>图标题</figcaption><img src=…/> </figure>

canvas绘图:<script>$(document).ready(function () {var c = document.getElementById("myCanvas"); var cxt=c.getContext("2d"); cxt.fillStyle="#FF0000"; cxt.fillRect(0,0,150,75);})</script> <canvas id="myCanvas"> </canvas> 其他: 折线:cxt.moveTo(10,10); cxt.lineTo(150,50); cxt.lineTo(10,50); cxt.stroke(); 填充圆: cxt.fillStyle="#FF0000"; cxt.beginPath(); cxt.arc(70,18,15,0,Math.PI\*2,true); cxt.closePath(); cxt.fill();其中arc(中心x,中心y,半径,起始弧度,结束弧度) 梯度填充: var grd=cxt.createLinearGradient(0,0,175,50); grd.addColorStop(0,"#FF0000"); grd.addColorStop(1,"#00FF00"); cxt.fillStyle=grd; cxt.fillRect(0,0,175,50);其中: createLinearGradient(开始点x,开始y,结束x,结束y) 绘图: var img=new Image();

img.src="/i/eg\_flower.png" cxt.drawImage(img,0,0);

svg矢量图: <?xml version="1.0" standalone="no"?> <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"> <svg width="100%" height="100%"> <circle cx="100" cy="50" r="40" stroke="black" stroke-width="2" fill="red"/></svg> <svg height="190"><polygon points="100,10 40,180 190,60 10,60 160,180" style="fill:red;stroke:blue;stroke-width:3;fill-rule:evenodd;" /></svg>

嵌入svg: <embed src="Slice 1.svg" width="300" height="100" type="image/svg+xml"/>(不是写在svg标签内) 或<iframe src="rect.svg" width="300" height="100"> </iframe>

svg中的形状元素rect, circle, ellipse, line, polygon, polyline, path

高斯滤镜: <defs> <filter id="Gaussian\_Blur"> <feGaussianBlur in="SourceGraphic" stdDeviation="3" /> </filter> </defs> 引: filter:url(#Gaussian\_Blur)"

线性渐变:<defs><linearGradient id="orange\_red" x1="0%" y1="0%" x2="100%" y2="0%"><stop offset="0%" style="stop-color:rgb(255,255,0);stop-opacity:1"/><stop offset="100%" style="stop-color:rgb(255,0,0);stop-opacity:1"/></linearGradient></defs> 形状中引: fill="url(#orange\_red)"

放射渐变: <radialGradient id="grey\_blue" cx="50%" cy="50%" r="50%" fx="50%" fy="50%">…

地理定位: navigator.geolocation.getCurrentPosition(showPosition); 其中function showPosition(position){x.innerHTML="纬度: " + position.coords.latitude + "<br>经度: " + position.coords.longitude;}

客户端存储(属于js): 永久存储: localStorage

localStorage.sitename, localStorage.setItem(key,value); localStorage.getItem(key); localStorage.removeItem(key); localStorage.clear();localStorage.key(index);

sessionStorage用法同localStorage,不过是对session的存储,当关闭浏览器会删除

var site = new Object; site.keyname = 'keyname'; var str = JSON.stringify(site);将对象转为字符串; var siteNew = JSON.parse(str);将字符串转对象

缓存:<html manifest="demo.appcache">//lymark??最好结合真实环境

接收服务器消息: <script>if(typeof(EventSource)!=="undefined"){ var source=new EventSource("demo\_sse.php"); source.onmessage=function(event){…event.data…}} </script> 其中demo\_sse.php: <?php header('Content-Type: text/event-stream');

header('Cache-Control: no-cache'); $time = date('r'); echo "data: The server time is: {$time}\n\n"; flush(); ?>//lymark尝试结合计时器测试及时消息

响应式web设计?

为了确保页面可自由缩放可以在 <head> 元素中添加以下 <meta> 标签:

<meta name="viewport" content="width=device-width, initial-scale=1">

框架:

bootstrap(响应式布局,封装好的UI控件)

(某些环境下的浏览器不支持) 样式引用连接：<link href="http://cdn.static.runoob.com/libs/bootstrap/3.3.7/css/bootstrap.min.css" rel="stylesheet">或<http://apps.bdimg.com/libs/bootstrap/3.3.4/css/bootstrap.min.css>

foundation:(支持全平台)

CDN: <!-- css 文件 -->

<link rel="stylesheet" href="http://static.runoob.com/assets/foundation-5.5.3/foundation.min.css">

<!-- jQuery 库 -->

<script src="http://static.runoob.com/assets/jquery/2.0.3/jquery.min.js"></script>

<!-- JavaScript 文件 -->

<script src="http://static.runoob.com/assets/foundation-5.5.3/js/foundation.min.js"></script>

<!-- modernizr 文件 -->

Modernizr让组件能在所有浏览器上正常运行<script src="http://static.runoob.com/assets/foundation5.5.3/js/vendor/modernizr.js"></script>

<small>浅色副标题 kbd键盘连接键 <hr />水平线是浅灰

按钮: <a>, <button>, 或 <input>可通用一部分class

图片: <img … class="img-rounded"> 选项: "img-rounded"(圆角),"img-circle"(圆形),"img-thumbnail"(圆角边框) , "img-responsive"支持响应式,存在最小宽度

辅助类: 插入符<span class="caret">

浮动: <div class="pull-left"> pull-right center-block

显示/隐藏: <div class="show"> show hidden//布局会变

响应式显示/隐藏: visible-xs相反hidden-xs(<768px), -sm(768~992px) -md(992~1200px) -lg(>=1200px)

字体图标: <span class="glyphicon glyphicon-user">

<http://www.runoob.com/bootstrap/bootstrap-glyphicons.html>

下拉菜单(需要bootstrap的js文件): <div class="dropdown">( 或上拉dropup)<button type="button" class="btn dropdown-toggle" id="dropdownMenu1 " data-toggle="dropdown"> <ul class="dropdown-menu" role="menu" aria-labelledby="dropdownMenu1"><li role="presentation"><a role="menuitem" tabinde x="-1" href="#">Java</a>附加: <li role="presentation" class="dropdown-header">下拉菜单的标题

按钮组: <div class="btn-group btn-group-lg"><button type="button" class="btn btn-default">button1</button></div>

输入框组<div class="input-group"><span class="input-group-addon">@</span> <input type="text" class="form-control" placeholder="twitterhandle"><input type="radio">

tab菜单: <ul class="nav nav-tabs"><li class="active"><a href="#">Home</a></li>

其中"nav nav-tabs":标签式导航菜单,其他选项: "nav nav-pills"胶囊式导航菜单，"nav nav-pills nav-stacked"垂直的胶囊式导航菜单

自动调节导航标签大小,让导航菜单和父元素等宽<ul class="nav nav-pills nav-justified"> 禁用链接<li class="disabled">

下拉标签: <li class="dropdown"><a class="dropdown-toggle" data-toggle=" dropdown" href="#"><ul class="dropdown-menu">

标签对应内容: <div class="tab-content"><div id="home" class="tab-pane fade in active"><div id="menu1" class="tab-pane fade"> //ul标签栏li, a中的href="#home" data-toggle="tab"

导航栏: <nav class="navbar navbar-default" role="navigation"><div class="navbar-header"> <a class="navbar-brand" href="#">菜鸟教程</a></div><div><ul class="nav navbar-nav">

下拉式id导航: <nav class="navbar navbar-default" role="navigation"><div class="navbar-header"><button type="button" class="navbar-toggle" data-toggle="collapse" data-target="#example-navbar-collapse"></div>… <div class="collapse navbar-collapse" id="example-navbar-collapse">

对齐<ul class="nav navbar-nav navbar-left">

固定到顶部<nav class="navbar navbar-default navbar-fixed-top"> 底部navbar-fixed-bottom 静态顶部(默认)navbar-static-top 倒置的导航栏(黑色背景白色文本)navbar-inverse

面包屑导航<ol class="breadcrumb">,改分隔符或其他样式ol.breadcrumb > li + li:before {color: #CCCCCC;content: "/ ";padding: 0 5px;}

分页<ul class="pagination"><li><a href="#">&laquo;</a></li><li class="active" > <a href="#">1</a></li> <li class="disabled" ><a href="#">2</a></li>

翻页<ul class="pager"> &larr;左箭头 &rarr;右箭头

标签<span class="label label-default">其中label-default其他选项label-primary, success, info, warning, danger

徽章与标签相似, 边角更加圆滑<span class="badge">

超大屏幕<div class="jumbotron">

页面标题(在网页标题四周添加适当的间距)<div class="page-header">

子元素含container，文字越界并出现水平滚动条，加上width:100%

缩略图<div class="col-xs-10 col-sm-10 col-md-10 col-lg-10"><a href="#" class="thumbnail"><img …></a></div>

alert: <div class="alert alert-success alert-dismissable"><a href="#" class="alert-link"></a> <button type="button" class="close" data-dismiss="alert" aria-hidden="true">&times;</button></div>其中success其他选项info, warning, danger

进度条(IE9及之前的版本和旧版的Firefox不支持该特性，Opera12不支持动画): <div class="progress progress-striped active "><div class="progress-bar progress-bar-success" style=" width: 40%;"></div></div>其中progress-striped带条纹的进度条，active表示动画,把多个progress-bar放在progress中可实现堆叠

多媒体对象<div class="media"><a><img class="media-object pull-left" src="./uploads/uncheck.png"></a><div class="media-body"><h4 class="media-heading">媒体标题</h4><ul class="media-list"><li class="media">li</li></ul></div></div>

列表组<ul class="list-group"><li class="list-group-item active">…</li>

带head: <div class="list-group"><a href="#" class="list-group-item active"><h4 class="list-group-item-heading">justHead</h4></a><a href="#" class="list-group-item"><h4 class="list-group-item-heading">head</h4><p class="list-group-item-text">text</p> </a></div>

面板<div class="panel panel-default panel-primary "><div class="panel-heading">< div class="panel-body"> 面板脚注<div class="panel panel-default"><div class="panel-body"> cell格式<table class="table">

well: 引起内容凹陷显示或插图效果的容器<div class="well well-lg">…</div>

模态框: data-toggle="modal", data-target="#myModal"或href="#myModal"控制特定的模态框，<div class="modal fade" id="myModal"><div class="modal-dialog"><div class="modal-content">模态框内容</div></div></div>当模态框被切换时，它会引起内容淡入淡出; aria-labelledby="myModalLabel"，该属性引用模态框的标题; aria-hidden="true" 用于保持模态窗口不可见，直到触发器被触发为止; <div class="modal-header">为模态窗口的头部定义样式; class="close"为模态窗口的关闭按钮设置样式; data-dismiss="modal"关闭模态窗口; class="modal-body"为模态窗口的主体设置样式; class="modal-footer"为模态窗口的底部设置样式;

其他: <script>$(function () { $('#myModal').modal({keyboard: false,//ESC退出无效backdrop: 'static',//点击模态框外部时不会关闭模态框show:false初始化时不显示模态框})});</script>

与model()一起的相关方法: $(function () { $('#myModal').modal('show')});其中show其他选项:hide,toggle

事件$('#myModal').on('hide.bs.modal',function() {alert('当调用 hide 实例方法时触发');})其中hide.bs.modal其他选项: show.bs.modal(show开始时调用), shown.bs.modal(show结束时调用), hidden.bs.modal

JS下拉菜单$('.dropdown-toggle').dropdown('toggle')

滚动监听: <div data-spy="scroll" data-target="#navbar-example" style="height: 200px;overflow:auto; position: relative;"> <h4 id="ios">iOS</h4> <p>ios内容</p> <h4 id="svn">SVN</h4><p>svn内容</p></div>，联动的目标: <nav id="navbar-example" class="navbar navbar-default"> <ul class="nav navbar-nav"><li><a href="#ios">iOS</a></li> <li><a href="#svn">SVN</a></li></ul>

.scrollspy('refresh')//???

值传递$('#myScrollspy').on('activate.bs.scrollspy', function (){$("#activeitem").html ($(".nav li.active > a").text());})其中nav中的li标签属性是active的

标签页: 切换<a href="#home" data-toggle="tab">//其中home是相应div的id

初始选中的标签$(function () {$('#myTab li:eq(0) a').tab('show');});

事件: $('a[data-toggle="tab"]').on('shown.bs.tab', function (e) {…$(e.target).text ()…$(e.relatedTarget).text()//前一个被选择的标签…});

提示工具:

<a href="#" data-toggle="tooltip" title="Example tooltip">悬停的链接</a>，在script中要加入$(function () { $("[data-toggle='tooltip']").tooltip(); });

其他属性: data-placement默认top，data-animation，data-html，data-selector，data-title，data-trigger，data-content，data-delay，data-container。 插件方法: .tooltip(options) .tooltip('toggle') show hide destroy

事件: show.bs.tooltip当调用 show 实例方法时立即触发该事件; shown.bs.tooltip; hide.bs.tooltip; hidden.bs.tooltip

弹出框: <a href="#" data-toggle="popover" data-content="弹出的内容">弹出链接</a>其他属性、方法和事件与提示工具差不多

警告框: <div class="alert alert-warning"><a href="#" class="close" data-dismiss="alert">&times;</a><strong>警告</strong>有问题。</div>

$(".close").click(function(){$("#myAlert").alert();//让所有警告框都带有关闭功能??}); .alert('close')//关闭所有警告框 事件: close.bs.alert closed.bs.alert

按钮:加载状态: data-loading-text="Loading..." $(".btn").click(function(){$(this). button('loading').delay(1000).queue(function() {$(this).button('reset'); $(this).dequeue(); });});//还原.button('reset') //加载完后的内容data-complete-text="Loading finished" .button('complete');

折叠: <div id="accordion"><div><a data-toggle="collapse" data-parent="#accordion" href="#collapseOne">显示的连接</a><div id=" collapseOne" class="panel-collapse collapse in">折叠的内容</div></div></div>

class="collapse in"一开始显示内容, class="collapse"一开始隐藏内容

JS中的方法: $(' .panel-collapse').collapse('toggle'), show, hide

事件show.bs.collapse shown hide hiden

轮播<div id="myCarousel" class="carousel slide"><ol class="carousel-indicators"> <li data-target="#myCarousel" data-slide-to="0" class="active"></li><li data-target= "#myCarousel" data-slide-to="1"></li></ol><div class="carousel-inner"><div class= "item active"><img src='./uploads/1242\_2208.png'></div><div class="item"><img src='./uploads/uncheck.png'></div></div><a class="carousel-control left" href="# myCarousel" data-slide="prev">&lsaquo;</a><a class="carousel-control right" href="#myCarousel" data-slide="next">&rsaquo;</a></div>

其他: <div class="carousel-caption">标题 1</div>

方法.carousel({interval: 2000})//pause, wrap .carousel('cycle')从左到右轮播 pause number循环轮播到某个特定的帧 prev next

事件slide.bs.carousel开始过渡时调用，slid.bs.carousel完成过渡时调用

附加导航插件<ul class="fi" data-spy="affix" data-offset-top="125">锁定元素位置；//其中data-offset-top是从哪个位置开始有偏移, div.fi{top:30px;}设置与顶部的距离 或$(document).ready(function(){$(".fi").affix({offset: {top: 125}});});

AngularJS(各种及时交互)

src="http://cdn.static.runoob.com/libs/angular.js/1.4.6/angular.min.js"

ng-app指令告诉 AngularJS该元素是 AngularJS 应用程序 的"所有者"

ng-model 指令把输入域的值绑定到应用程序变量 name

ng-bind 指令把应用程序变量 name 绑定到某个段落的 innerHTML,类似{{…}},为避免初始加载闪烁的问题，可以用ng-bind

ng-init 指令初始化 AngularJS 应用程序变量

控制器 <div ng-app="myApp" ng-controller="myCtrl"> <input type="text" ng-model="firstName"></div> var app = angular.module('myApp', []); app.controller('myCtrl', function($scope) {$scope.firstName= "John"; $scope.lastName= "Doe";});

ng-repeat重复指令：<li ng-repeat="x in names">{{ x }}</li>每个项会克隆一次 HTML 元素

自定义指令: <runoob-directive></runoob-directive> app.directive ("runoobDirective", function() { return { template:"<h1>指令!</h1>"    };}); (restrict: "E")其他自定义指令的方式<div runoob-directive></div>(restrict:"A") <div class="runoob-directive"></div>(restrict: "C") <!-- directive: runoob-directive -->(restrict : "M",replace : true,)//被注释掉也能调用

验证输入: <form name="myForm" ng-app=""><input type="email" name= "myAddress"><span ng-show="myForm.myAddress.$error.email">ng-show会在返回true的情况下显示</span></form>

ng-model 指令可以为应用数据提供状态值(invalid, dirty, touched, error): valid如果输入的值是合法的则为 true, dirty如果值改变则为true, touched如果通过触屏点击则为 true。必须是form中的元素,且按name引用,不是ng-model

ng-model 指令基于它们的状态为 HTML 元素提供了 CSS 类：input.ng-invalid {background-color: lightblue;}

其他ng-empty ng-not-empty ng-touched ng-untouched ng-valid ng-invalid ng-dirty ng-pending ng-pristine

作用域scope. $rootScope定义的值，可以在各个controller 中使用。??lymark

控制器中的结果在页面显示要用回调

{{ lastName | uppercase }}格式化为大写 lowercase小写 currency格式化数字为货币格式 <li ng-repeat="x in names | orderBy:'country'">根据表达式排列数组 <li ng-repeat="x in names | filter:test ">过滤,用于数组

服务: app.controller('customersCtrl', function($scope, $location) {$scope.myUrl = $location.absUrl();});用$location服务可以返回当前页面的URL地址

$http服务:$http.get("welcome.htm").then(function (response) {$scope.myWelcome= response.data;}); //then其他选项success

$timeout服务: $timeout(function () {$scope.myHeader = "How are you today?";}, 2000);

$interval(计时器)服务: $interval(function () {$scope.theTime = new Date(). toLocaleTimeString();}, 1000);

自定义服务: app.service('$hexafy', function() {this.myFunc = function (x) {return x.toString(16);}}); 调用: $scope.hex = $hexafy.myFunc(255); 或factory

自定义过滤app.filter('myFormat',['hexafy', function(hexafy) {return function(x) {return hexafy.myFunc(x);};}]); 调用: {{x | myFormat}}// hexaf见上文

选择框: <select ng-model="selectedName" ng-options="x.site for x in names"></select>或<select><option ng-repeat="x in names">{{x}}</option> </select>区别: 用ng-repeat选择的值是字符串,用ng-options选择的值是对象,可以这样{{selectedSite.site}} 对象做数据源ng-options="x for (x, y) in sites"

表格: <tr ng-repeat="x in names"><td>{{ x.Name }}</td><td>{{ x.Country }}</td> </tr> 样式: table tr:nth-child(odd){…} table tr:nth-child(even) {…} 索引:$index

行间判断<td ng-if="$odd" style="background-color:#f1f1f1">//ps:如果不满足if,则不会创建这个td

button: ng-disabled=true/false可否点击

ng-show="true/false"元素是否可见，与ng-hide相反

事件<button ng-click="count = count + 1">其中<p>{{ count }}</p>

复制$scope.user = angular.copy($scope.master)

API: angular.lowercase() uppercase isString isNumber

包含: <div ng-include="'myUsers\_List.html'">//其中双引号中的单引号必须要

跨域包含: <body ng-app="myApp"><div ng-include="'http://c.runoob.com/ runoobtest/angular\_include.php'"></div><script> var app = angular.module('myApp', [])app.config(function($sceDelegateProvider) {$sceDelegateProvider. resourceUrlWhitelist(['http://c.runoob.com/runoobtest/\*\*']);});</script></body>//其中angular\_include.php中:<?php header('Access-Control-Allow-Origin:\*'); echo '<b style="color:red">我是跨域的内容</b>';?>

动画: <script src="http://cdn.static.runoob.com/libs/angular.js/1.4.6/angular-animate.min.js"></script> <body ng-app="ngAnimate"><input type="checkbox" ng-model="myCheck"><div ng-hide="myCheck"> .ng-hide {height: 0;width: 0;background-color: transparent;top:-200px;left: 200px;} //transition: all linear 0.5s;

@keyframes myChange {from {height: 100px;} to {height: 0;}} div.ng-hide {animation: 0.5s myChange;}

依赖注入: mainApp.value("defaultInput", 5);//即向mainApp.controller传参数

路由: <div ng-view></div>其中div内的HTML根据路由变化, angular.module('routingDemoApp',['ngRoute']).config(['$routeProvider',function($routeProvider){ $routeProvider.when('点击的href="#/"链接', {template: '显示的内容'}) }]);//其中['ngRoute']和$routeProvider暂时看成固定写法吧

React框架(标签的封装):

安装: <script src="https://cdn.bootcss.com/react/15.4.2/react.min.js"></script>

<script src="https://cdn.bootcss.com/react/15.4.2/react-dom.min.js"></script>

<script src="https://cdn.bootcss.com/babel-standalone/6.22.1/babel.min.js"></script>

JSX: <script type="text/babel">ReactDOM.render(<div>…</div>, document. getElementById('example')); </script> 表达式如{1+1}

其中不能用if else，用三目

内联样式var myStyle = {fontSize: 100,color: '#FF0000'}; ReactDOM.render(<h1 style = {myStyle}>菜鸟教程</h1>,);{/\*注释...\*/}

插入(标签)数组: var arr = [<h1>一</h1>,<h2>二</h2>,];ReactDOM.render(

<div>{arr}</div>,…);

组件var HelloMessage = React.createClass({render: function() {return <h1>Hello {this.props.name}</h1>;}});ReactDOM.render(<HelloMessage name="Runoob" />,…);//这里this.props.name="Runoob"

状态: getInitialState: function() {return {liked: false};}, handleClick: function (event) {this.setState({liked: !this.state.liked});},//setState会自动调用 this.render render: function() {…return (<p onClick={this.handleClick}> <b>{text}</b> </p>);…}

为props中的属性设置默认值: getDefaultProps: function() {return {name: 'Runoob'};},//但会被<HelloMessage name="Runoob" />覆盖

state可以根据与用户交互来改变, props不可变

props验证: propTypes: {title: React.PropTypes.string.isRequired,},//要开启调试器，这里如果title取数字会报错,不取参数则不显示title,其他的内容还是会显示。 与string相应的其他验证: <http://www.runoob.com/react/react-props.html>

replaceState: 与setState()类似, 原state不在nextState中的状态都会被删除//??lymark

判断组件挂载状态:bool isMounted()

生命周期:

componentWillMount 在渲染前调用,在客户端也在服务端。

componentDidMount在第一次渲染后调用，只在客户端。可以在这个方法中调用setTimeout, setInterval或者发送AJAX

componentWillReceiveProps 在组件接收到一个新的prop时被调用。这个方法在初始化render时不会被调用。

shouldComponentUpdate 返回一个布尔值。在组件接收到新的props或者state时被调用。在初始化时或者使用forceUpdate时不被调用。

componentWillUpdate在组件接收到新的props或者state但还没有render时被调用。在初始化时不会被调用。

componentDidUpdate 在组件完成更新后立即调用。在初始化时不会被调用。

componentWillUnmount在组件从 DOM 中移除的时候立刻被调用。

React AJAX: this.serverRequest = $.get(this.props.source, function (result) { var lastGist = result[0];…} .bind(this));

onChange事件：<input onChange={this.handleChange} />其中handleChange: function (event) {this.setState({value: event.target.value});},

属性调用: this.refs.myInput.focus();//其中<input type="text" ref="myInput" />

Ext(DOM操作):

初始化Ext.onReady(function() {… });

按id获取某个元素var myDiv = Ext.get('myDiv'); myDiv.highlight();黄色高亮显示然后渐退 Ext.select('div').highlight();按标签获取 myDiv.center();垂直居中 myDiv.setOpacity(.25); myDiv.addClass('red');

Ext.get('myButton').on('click', function(){alert("You clicked the button");});

函数作参数var paragraphClicked = function(e) {Ext.get (e.target).highlight();}Ext.select('p').on('click', paragraphClicked);

面向对象:Ext.define('My.sample.Person', {name: 'Unknown', constructor: function(name) {if (name) {this.name = name;}}, eat: function(foodType) {alert (this.name + " is eating: " + foodType);} }); var bob = Ext.create('My.sample.Person', 'Bob'); bob.eat("Salad");

Ext.define('My.own.Window', {extend: 'Ext.Component',//继承 config: {title: 'Title Here',bottomBar: {height: 50,resizable: false}}//config初始化 applyTitle: function (title) {if (!Ext.isString(title) || title.length === 0) {alert('Error: Title must be a valid non-empty string');}else {return title;}}//调set方法时验证}); 调用: var myWindow = Ext.create('My.own.Window', {title: 'Hello World',bottomBar: {height: 60}}//改config里的内容); alert(myWindow.getTitle());//调get方法 myWindow.setTitle('Something New');//调set方法

类方法: statics: {factory: function(brand) {return this({brand: brand});}},

CSS3:

div{border:2px solid;border-radius:25px; box-shadow: 10px 100px 5px #888888; }

边框图片{border-image/-webkit-border-image/-o-border-image:url(border.png) 30 30 round/stretch;}两个参数则表示“上下，左右”，四个参数是上右下左,九切片

background-origin:border-box/content-box;位置以border还是content为准

background-clip: border-box/content-box;从指定位置裁剪

从上到下渐变: #grad1{background: (-webkit-/-o-/-moz-)linear-gradient(red, blue);}

从左到右渐变: #grad1 {height: 200px;background: -webkit-linear-gradient(left, red , blue); /\* Safari 5.1 - 6.0 \*/ background: -o-linear-gradient(right, red, blue); /\* Opera 11.1 - 12.0 \*/ background: -moz-linear-gradient(right, red, blue); /\* Firefox 3.6 - 15 \*/ background: linear-gradient(to right, red , blue); /\* 标准的语法（必须放在最后） \*/}

按角度渐变: #grad1{background: (-webkit-/-o-/-moz-)linear-gradient(180deg,red, blue);}，其中可以接两个以上的颜色参数表示多个渐变结点，颜色参数也可以用rgba(255,0,0,0),最后一个参数是透明度。

重复渐变: background: repeating-linear-gradient(red, yellow 10%, green 20%);

径向渐变: background: radial-gradient(red, green);

圆形径向渐变: background: radial-gradient(circle, red, yellow, green);默认ellipse

background: radial-gradient(60% 55%, closest-side,blue,green,yellow,black);其中第一个参数是中心位置，第二参数决定渐变最大/最小范围

重复径向渐变: background: repeating-radial-gradient(red, yellow 10%, green 15%);

文字阴影: h1{text-shadow: 5px 5px 5px #FF0000;}

#boxshadow::after {…;z-index: -1;box-shadow: 0 15px 20px rgba(0, 0, 0, 0.3);}伪元素添加阴影效果

显示溢出内容的方式:<div style="text-overflow:ellipsis/clip/'>>';">

允许长文本换行,即使分裂单词:p{word-wrap:break-word;}或p{word-break: break-all;}

字体: @font-face{font-family: myFirstFont;src: url ('Sansation\_Light.ttf'),url('Sansation\_Light.eot'); /\* IE9 \*/}

2D旋转: div{ (-ms-/-webkit-)transform:rotate(30deg);}

位移:transform: translate(50px,100px);

放大/缩小: transform: scale(2,3);

transform: skew(30deg,20deg);其中第一个参数是逆时针倾斜,高度不变;第二个参数是顺时针倾斜,宽度不变

用矩阵变换: transform:matrix(0.866,0.5,-0.5,0.866,0,0);

3D旋转: transform: rotateX(120deg); transform: rotateY(130deg);其他跟2D类似

过渡动画: div{width:100px; transition:width 10s; transform 2s;} div:hover{width:300px; transform: rotate(180deg);}

div{background:red;transition-property:width;transition-duration:1s;transition-timing-function:linear;transition-delay:2s;}同div { transition: width 1s linear 2s;}

keyframes动画:@(-webkit-)keyframes myfirst{from {background:red;}to {background:yellow;}} div{( -webkit-)animation:myfirst 5s;}

结点动画: @keyframes myfirst{0% {background:red;}25% {background:yellow;}50% {background:blue;}100% {background:green;}} 其他属性: animation-iteration-count: infinite;播放次数 animation-direction: alternate;方向 animation-play-state: running;播放状态,默认running

多列: column-count: 3;文字分列布局 column-gap: 40px;列间隙 column-rule-style: solid;分割线样式 column-rule-width: 1px; column-rule-color: lightblue;

column-span: all;某个元素横跨多少列,firefox不支持

column-width:100px;指定列的宽度

可由用户调整大小: div{resize:both;overflow:auto;}

box-sizing:border-box;确定的方式定义某个区域的具体内容

超出边缘绘制轮廓: outline:2px solid red;outline-offset:15px;

圆角: border-radius: 8px; 或border-radius: 50%;

阴影:box-shadow: 0 0 2px 1px rgba (0, 140, 186, 0.5);其中3参决定虚化范围，4参决定虚化宽度

响应式布局: max-width: 100%;

<div class="container"><img …><div class= "topleft"></div></div>其中.container {position: relative;} .topleft {position: absolute;}

滤镜: filter: grayscale(100%);其他效果见: <http://www.runoob.com/cssref/css3-pr-filter.html>

initial关键字: color:initial保持原始样式

响应布局@media only screen and (max-width: 700px){.responsive {width: 49.99999%; margin: 6px 0;}} @media only screen and (max-width: 500px){.responsive {width: 100%;}}

模态var modal = document.getElementById('myModal') //myModal是个div, var img = document.getElementById('myImg'); var modalImg = document.getElementById("img01"); img.onclick = function(){modal.style.display = "block";modalImg.src = this.src; }

阴影: .button1 {box-shadow: 0 8px 16px 0 rgba(0,0,0,0.2)}参数依次:水平位置、垂直位置、模糊距离、阴影尺寸、阴影颜色、inset将外部阴影(outset)改为内部阴影

button的span标签后面加内容.button span:after {content: '»'; transition: 4s;} hover状态下透明度变化.button:hover span:after {opacity: 1;right: 0;}

button被点击: .button:active {transform: translateY(4px);}//移动y坐标4px

悬停未激活链接的背景色 a:hover:not(.active) {background-color: #ddd;}

左上角圆角: border-top-left-radius: 5px;

div{box-sizing: border-box;}宽高包含padding和border在内

弹性盒子: .flex-container {display: -webkit-flex;display: flex;}从左到右排列 从右到左body {direction: rtl;} .flex-container {flex-direction: row;}其他row-reverse横向从右到左，column，column-reverse

.flex-container {justify-content: flex-start}左对齐,其他: flex-end右对齐, center中心对齐, space-between两端对齐, space-around不考虑内部元素margin的均匀布局

.flex-container {align-items: flex-start} 元素顶对齐,纵向文字自适应，其他: flex-end底对齐，center竖直居中，stretch默认,纵向拉伸，baseline基线对齐

flex-wrap子元素换行方式: flex-wrap: nowrap;不换行,子元素可能超出边界;其他wrap换行; wrap-reverse反转wrap排列

align-content：wrap基础上行的竖直对齐方式，align-content: flex-start | flex-end | center | space-between | space-around | stretch

.first { order:1;}设置元素顺序,当前位置若有其他元素，依次后移

.flex-item:first-child {margin-right: auto;}元素边缘弹性

在两轴都居中.flex-container { display: flex; } .flex-item { margin: auto; }

align-self：跟align-items没区别？

.item1 { flex: 2; } .item2 { flex: 1; } 这两个元素比例是2:1

多媒体查询@media screen and (min-width: 480px) {body {background-color: lightgreen;}} lightgreen颜色应用的最小视窗是480px

@media screen and (max-width: 699px) and (min-width: 520px) {…}

伪元素添加属性ul li a:after {content: " (" attr(data-email) ")";font-size: 12px;font-style: italic;color: #666666;}

适配手机要加上<meta name="viewport" content="width=device-width, initial-scale=1" />

weui：<http://res.wx.qq.com/open/libs/weui/1.1.2/weui.min.css>

actionsheet: <div class="weui-mask\_transparent actionsheet\_\_mask" id="mask"></div> <div class="weui-actionsheet"><div class="weui-actionsheet\_\_menu"> <div class="weui-actionsheet\_\_cell">示例菜单</div>…</div></div></div>

article： <article class="weui-article"><section></section></article>

bage: <div class="weui-cell"><div class="weui-cell\_\_hd" style="position: relative;"><span style="vertical-align: middle;font-size: 20px">单行列表</span><span class="weui-badge" style="position: absolute;top: -.4em;right: -.4em;">8</span></div></div>

innerHTML不会加载js，.html会

cell: title cells>cell>hd,bd,ft bd采用自适应布局

dialog: div>weui-mask, weui-dialog>hd,bd,ft>weui-dialog\_\_btn js中fadein btn的parents中的div fadeout

flex布局：weui-flex>weui-flex\_\_item

weui-footer: weui-footer weui-footer\_fixed-bottom(weui-footer\_\_text(Copyright © 2008-2016),weui-footer\_\_links(weui-footer\_\_link))

九宫格: .weui-grids>a. weui-grid>div. weui-grid\_\_icon>img, p. weui-grid\_\_label

加载相关: 菊花i.weui-loading线中字div. weui-loadmore weui-loadmore\_line>span

input相关：div.weui-cells\_\_title div.weui-cells weui-cells\_radio>label.weui-cell weui-check\_\_label>div.weui-cell\_\_bd, div weui-cell\_\_ft>input.weui-check, span. weui-icon-checked 输入<input class="weui-input" type="number" pattern="[0-9]\*" placeholder="请输入qq号"/>

对勾大图标<i class="weui-icon-success weui-icon\_msg"></i>

tabbar: div.weui-tab>div.weui-navbar>div.weui-navbar\_\_item , div.weui-tab\_\_panel 如果要显示哪个bar\_\_item，就添加weui-bar\_\_item\_on

返回之前页面的链接：href="javascript:history.back();"

panel: weui-panel weui-panel\_access

picker: 用到weui.js:

<script src="https://res.wx.qq.com/open/libs/weuijs/1.1.1/weui.min. js"></script>

weui.picker([{label: '飞机票',value: 0 }], {onChange: function (result) {console.log (result);},onConfirm: function (result) {console.log(result);}});

weui.datePicker({start: 1990,end: new Date().getFullYear(),onChange: function (result) {console.log(result);},onConfirm: function (result) {console.log(result);}});

Preview表单预览：div.weui-form-preview(div.weui-form-preview\_\_hd(label.weui-form-preview\_\_label,weui-form-preview\_\_value),div.weui-form-preview\_\_bd(p(label,value)),ft)

进度条：.weui-progress(.bar(.\_\_inner-bar js\_progress)) js: setTimeout(func, 20) .width(++)

weui-slider: .weui-slider(.weui-slider\_\_inner(.weui-slider\_\_track, .weui-slider\_\_handler)) 与官方demo不同的地方：

1 e.changedTouches应该是e.originalEvent.changedTouches;

2 startLeft是parseInt($sliderHandler.css('left')).

weui-tabbar: jquery的siblings('.selected')方法：查找每个元素的所有类名含 "selected" 的所有同胞元素。

weui-tab(weui-tab\_\_panel, weui-tabbar(a.weui-tabbar\_\_item(weui-tabbar\_\_icon , weui-tabbar\_\_label)))

toast：通用的手机前方菊花框：div( div.weui-mask\_transparent , div.weui-toast( i.weui-loading weui-icon\_toast , p.weui-toast\_\_content ) )

uploader: div.\_\_db(ul.\_\_files(li.\_\_file(div.\_\_file-content)) , div\_\_input-box(input.\_\_input type=”file” accept=”image/\*”)) js中：.on(‘change’,function(e){ e.target.files … url.createObjectURL(file)})

未解: layerslider ge\_posts\_slider header

jquery的$(function(){…})可以添加初始化方法，js的window.onload只能有一个初始化方法