# HTML+CSS

## 一、样式表(Css)：行内样式表，内部样式表，外部样式表。

## 二、样式表优先级：行内样式表>内部，外部样式表（内部和外部样式表优先级一样大，谁在后面，谁优先显示）

## 三、四大选择器

元素（标签）选择器、通配符（\*）选择器、类（.）选择器、id选择器（#）

权重：

标签1

Class 10

Id 100

行内 1000

!important最高

## 四、Css两大特性

继承性（优先级最低）、层叠性（子元素覆盖父元素）

## 五、Css优先级

优先级由高到低：!important>行内样式>ID选择器>类选择器>标签选择器>继承样式

color:red!important;

## 六、并集选择器

p,a,input{}

## 七、交集选择器（两个选择器，一个为标签选择器，一个为类选择器）

p.computer{}

## 八、后代选择器（只执行后代）

h1 em {

color: red;

}只执行所有的em（文字倾斜）

## 九、子元素选择器（只执行子元素）

p>em {

color: red;

}

## 十、相邻兄弟选择器（只执行兄弟，后面那个）

h1+p {

color: red;

}

执行h1后面第一个p标签

## 十一、伪类选择器

a:link {

font-size: 20px;

}

a:visited {

color: lime;

}

a:hover {

color: red;

}

a:active {

font-size: 30px;

}

input:focus {

background-color: yellow;

}

li:first-child {

font-size: 30px;

} /\*ul里面第一个li\*

爱恨原则顺序不能变，不然执行不了

## 十一、伪元素选择器

p:first-child:first-letter {

color: #ff0000;

font-size: xx-large;

}

p:first-child:first-line {

color: #0000ff;

font-variant: small-caps;

}

div:first-line{

color: #0000ff;

font-variant: small-caps;

}

h1:before {

content: url(book.png)"这里可以加文字";

}

h2:after {

content: "搜索";

}

## 十二、背景颜色

background-color: red;

background-color: #ffff00 ;

background-color: rgba(128,128,128,0.5);

## 十三、背景属性

background-image: url(https://ss0.bdstatic.com/5aV1bjqh\_Q23odCf/static/superman/img/logo\_top\_86d58ae1.png);

background-repeat: repeat-x;

repeat-y；

no-repeat;

background-position:90% 20%;

100px -200px;

center center;/\*水平后垂直\*/

background: #ffff00 url(https://ss0.bdstatic.com/5aV1bjqh\_Q23odCf/static/superman/img/logo\_top\_86d58ae1.png) no-repeat 50% 50%;

## 十四、字体属性

text-align: left;

right;

center;

text-decoration: overline;

line-through;

underline;

font-family: FangSong,'黑体';

英文不用加单引号，中文要

font-size: 20px;

网页默认字体大小为16px

font-size: xx-large;

把字体的尺寸设置为不同的尺寸，从 xx-small 到 xx-large。

xx-small最小

x-small较小

small小

medium中等

large大

x-large较大

xx-large最大

默认值：medium。但各浏览器对这个尺寸理解可能不太一样，所以现在用的不多。

font-style: oblique; 强制倾斜oblique也为倾斜

italic;

font-variant: small-caps; 字-体变形（小体大写）

font-weight: 600; 600以上则加粗

normal；

bold；

font:italic bold 36px/72px '微软雅黑';（36px/72px字大小和行高）

p {font: 20px/24px “微软雅黑”;} /\* 正确 \*/必须写

p {font: italic bold 20px/24px ;} /\* 错误 \*/

## 十五、字体单位

CSS中常见的单位

• px–最常见的，页面按照精准样式展示;属于绝对单位；

• em–基准点为父节点字体的大小，如果自身定义了font-size按自身来计算；也就是说这个em不是一个固定值，属于相对单位；

• rem–相对单位，r可以看作为 ”root“,结合em，理解为相对根节点HTML的字体大小来计算；

• vw：viewpoint width，视窗宽度，1vw等于视窗宽度的1%。

• vh：viewpoint height，视窗高度，1vh等于视窗高度的1%。

• %:百分比

div{

width: 100%;

height: 100%;

background: #ff0;

}平铺屏幕

## 十六、css文本属性（p标签）

color: red;

direction: rtl;

direction 属性规定文本的方向 / 书写方向。

ltr 默认。文本方向从左到右。

rtl 文本方向从右到左。

letter-spacing: -5px;

line-height: 1.5;

200%；

5px;

text-indent: 2em;

text-shadow:2px 2px 2px #fff；

X轴，Y轴，模糊程度(不可是负值)，阴影颜色

4个参数中，X和Y是必须的，一个文字可多个阴影

text-transform: capitalize;首字母大写

text-transform: uppercase;全部字母大写

text-transform: lowercase;全部字母小写

white-space: pre;预排版，怎么打怎么显示

white-space: nowrap; 不换行

word-spacing: 10px; 英文间距

letter-spacing: 10px;中文间距

## 十七、CSS尺寸属性

max-width: 50px;最大宽度

min-width：50px;最小宽度

min-height: 300px;最小高度

十七、列表属性

list-style-type: decimal;（123）

list-style-type: square;（正方形）

list-style-type: circle;（空心圆）

list-style-position: inside；

list-style-position: outside；（默认）

list-style-image: url(arr.gif);

## 十八、表格属性

border-collapse: collapse; 合并边框

border-spacing: 10px 20px;边框间距（水平 垂直）

caption-side: bottom; 标题方向

empty-cells: hide; 空单元格隐藏

table-layout: auto; 默认。表示列宽度由单元格内容设定。

table-layout: fixed;表示列宽由表格宽度和列宽度设定。

## 十九、内容属性

h1:after {

content: "基础";

}文字

h2:before {

content: url("book.png");

} 图片

a:after {

content: attr(href);

}链接

## 二十、内外边距属性

（1）ex和 em 不同的是，ex 是指所用字体中小写 x 的高度，通常为字体高度（em）的一半

（2）margin可以为负值，而border和padding不支持负值。

（3）outline与border的区别：outline是不占空间的，即不会增加额外的width或者height，不会象border那样影响元素的尺寸或者位置。

## 二十一、浮动

（1）position: absolute;clip: rect(0px 80px 60px 0px);

clip 属性剪裁绝对定位元素。

rect (top, right, bottom, left)

这个属性用于定义一个剪裁矩形。对于一个绝对定位元素，在这个矩形内的内容才可见。出了这个剪裁区域的内容会根据 overflow 的值来处理。剪裁区域可能比元素的内容区大，也可能比内容区小。

（2）overflow: scroll;盒子滚动条

overflow: hidden;溢出隐藏

（3）<p id="hideinfo" style="display: none;">这条信息默认是隐藏的</p>

<button onclick="document.getElementById('hideinfo').style.display= '';">显示</button>

<button onclick="document.getElementById('hideinfo').style.display= 'none';">隐藏</button>

（4）float: left;

清除浮动clear: left；/both

## 二十二、定位（子绝父相）

1、position: absolute;不占空间

（1）父元素没position: absolute;则基于浏览器定位

（2）父元素有定位，则基于父元素定位

2、position: relative;

相对于原来自己顶位，原来的位置依旧占空间

3、position: fixed;

相对于浏览器窗口定位，不占空间

4、z-index层叠等级属性

当对多个元素同时设置定位时，定位元素之间有可能会发生重叠。在CSS中，要想调整重叠定位元素的堆叠顺序，可以对定位元素应用z-index层叠等级属性，其取值可为正整数、负整数和0。

比如： z-index: 2;

注意：

z-index的默认属性值是0，取值越大，定位元素在层叠元素中越居上。

如果取值相同，则根据书写顺序，后来居上。

后面数字一定不能加单位。

只有相对定位，绝对定位，固定定位有此属性，其余普通文档流，浮动，静态定位都无此属性，亦不可指定此属性。

## 二十三、下载协议：

a连接href中ftp为下载协议

## 二十四、緢点

a连接定义name

a连接定义href中#+name

# CSS3

## 二十四、兄弟选择器

h2~p {

color: red;

}h2到后面的第一个p所有标签

兄弟选择器

语法 说明

元素 1 ~元素 2

第1 个元素之后，所有的元素2 都会被选择，且这些元素和第一个元

素拥有同一个父元素（两个元素之间不一定要相邻）

## 二十五、属性选择器

（1）<td>[title^="l"]</td>

title属性值为“l开头”的td元素会被选择

（2）<td>[title$="o"]</td>

title属性值为“o结尾”的td元素会被选择

（3）<td>[title\*="x"]</td>

title属性值“包含x”的td元素会被选择

## 二十六、background-clip裁剪

border-box表示背景被裁剪到边框盒；

padding-box表示背景被裁剪到内边距框；

content-box表示背景被裁剪到内容框。

## 二十七、background-origin

border-box表示从border区域开始显示背景图像；

padding-box表示从padding区域开始显示背景图像；

content-box表示从content区域显示背景图像。

## 二十八、background-size

例如：background-size:95% 300px;

设置背景图像的高度和宽度。第一个值设置宽度，第二个值设置高度。如果只设置一个值，则第二个值会被设置为 "auto"。

background-size:cover;表示把背景图像扩展至足够大，以使背景图像完全覆盖背景区域。背景图像的某些部分也许无法显示在背景定位区域中。

background-size:contain;表示把图像图像扩展至最大尺寸，以使其宽度和高度完全适应内容区域。

## 二十九、多背景图片

多背景(CSS3)

以逗号分隔可以设置多背景，可用于自适应布局 做法就是 用逗号隔开就好了。

- 一个元素可以设置多重背景图像。

- 每组属性间使用逗号分隔。

- 如果设置的多重背景图之间存在着交集（即存在着重叠关系），前面的背景图会覆盖在后面的背景图之上。

- 为了避免背景色将图像盖住，背景色通常都定义在最后一组上，

例如：

background:url(test1.jpg) no-repeat scroll 10px 20px/50px 60px ,

url(test1.jpg) no-repeat scroll 10px 20px/70px 90px ,

url(test1.jpg) no-repeat scroll 10px 20px/110px 130px #aaa;

## 三十、渐变背景

body{

background: yellow linear-gradient(0deg,yellow,blue) no-repeat;

/\*background: black radial-gradient( circle 200px at center center, orange,black) no-repeat;\*/

height: 500px;

}/\*0deg角度（从下到上，顺时针）由黄到蓝，底色为黄色\*/

1）线性渐变

语法：linear-gradient(方向,颜色1,颜色2……)

渐变的方向（可选）：可以设置角度值（单位deg），也可以是以下值：

to bottom 从上至下；to bottom right 从左上至右下；to right 从左至右；to up right 从左下至右上；to up 从下至下；to up left 从右下至左上；to left 从右至左；to bottom left 从右上至左下

body{

background: black radial-gradient( circle 100px at center center, orange,black) no-repeat;

height: 500px;

}

2）径向渐变

语法：radial-gradient (方向,颜色1,颜色2……)

扩散的方向：可以设置角度值（单位deg），也可以是以下值：

at center center； at left top； at center top； at right top； at right center； at right bottom； at center bottom； at left bottom

例如：

body{

background: black radial-gradient( circle 200px at center center, orange,black) no-repeat;

height: 500px;

}

表示放射源的形状为圆，圆心坐标在中心center center,半径为200px(可不写)

ellipse (默认): 指定椭圆形的径向渐变。

circle ：指定圆形的径向渐变

## 三十一、新增文本属性（（1）3个属性完成）

（1）text-overflow有两个属性值，clip和ellipsis。clip 表示不显示省略标记（…），而是简单的裁切；ellipsis代表当对象内文本溢出时显示省略标记（…）

需要注意的是，要想这两个属性起真正的作用，需要配合overflow:hidden;white-space:nowrap;使用。

overflow:hidden;代表着超出文本的部分不显示

（2）word-break 属性规定自动换行的处理方法。

normal 使用浏览器默认的换行规则。

break-all 允许在单词内换行。

keep-all 只能在半角空格或连字符处换行，显示整个单词。

white-space:nowrap;代表强制文本在一行显示。

## 三十二、圆角边框border-radius

例如：border-radius:15px 0px 15px 0px;

如果设置1个值，表示4个圆角都使用这个值。

如果设置两个值，表示左上角和右下角使用第一个值，右上角和左下角使用第二个值。

如果设置三个值，表示左上角使用第一个值，右上角和左下角使用第二个值，右下角使用第三个值。

如果设置四个值，则依次对应左上角、右上角、右下角、左下角（顺时针顺序）。

如果任意一个值为0，则这个角是矩形。不允许为负值。

那么以斜杠/分开后面的参数是怎么回事呢？

第一个参数表示圆角的水平半径，第二个参数表示圆角的垂直半径，所以可以画一个左右不对称的圆角。

例如：

.div1{border-radius: 2em/1em}

圆角半径设置为高度的一半则为圆。

## 三十三、盒子阴影(CSS3)

语法格式：

box-shadow:水平阴影 垂直阴影 模糊距离（虚实） 阴影尺寸（影子大小） 阴影颜色 内/外阴影；

1. 前两个属性是必须写的。其余的可以省略。

2. 外阴影 (outset) 是默认值，不用写。如果想要内阴影设置为inset

## 三十四、resize调整窗体宽度和高度

<div style="resize: both;">该窗体可以调整宽度和高度</div>

<div style="resize: horizontal;">该窗体可以调整宽度</div>

<div style="resize: vertical;">该窗体可以调整高度</div>

## 三十五、outline-offset外轮廓扩展

outline-offset: 10px;向外扩展

outline-offset 属性对轮廓进行偏移，并在边框边缘进行绘制。

## 三十六、变形

（1）2D变形(CSS3) transform

skew(angleX ,angleY)

表示元素沿着 x 轴方向倾斜angleX角度，沿着y轴方向倾斜angleY角度，其中 angleY可以省略，表示 Y 轴方向不倾斜。

skewX(angleX)：表示元素沿着 x 轴方向倾斜angleX 角度。

skewY(angleY)：表示元素沿着 y 轴方向倾斜 angleY 角度。

transform-origin

表示元素旋转的中心点，

默认值为 50% 50%。

第一个值表 示元素旋转中心点的水平位置，它还可以赋值 left、right、center、长度、百分比；

第二个值表示元素旋转中心点的垂直位置，它还可以赋值 top、bottom、 center、长度、百分比。

（2）translate(x,y)

表示元素水平方向移动 x，垂直方向移动 y，其中 y 可以省略，表示垂直方向没有位移

translateX(x)：表示元素水平方向移动 x。

translateY(y)：表示元素垂直方向移动y。

rotate(angle)

表示元素顺时针旋转 angle 角度，angle 的单位通常为 deg

scale(x,y)

表示元素水平方向缩放比为 x，垂直方向上缩放比为y，其中y可以省略，表示y和 x相同，以保持缩放比。

scaleX(x)：表示元素水平方向缩放比为 x。

scaleY(y)：表示元素垂直方向上缩放比为 y。

skew(angleX ,angleY)

表示元素沿着 x 轴方向倾斜angleX角度，沿着y轴方向倾斜angleY角度，其中 angleY可以省略，表示 Y 轴方向不倾斜。

skewX(angleX)：表示元素沿着 x 轴方向倾斜angleX 角度。

skewY(angleY)：表示元素沿着 y 轴方向倾斜 angleY 角度。

## 三十七、perspective:200;

设置成透视效果，透视效果为近大远小。该属性值用来设置 3D 元素距离视图的距离，单位为像素，已经内置了，我们只需要写具体数值即可，默认值为 0。当为元素定义 perspective 属性时，其子元素会获得透视效果，而不是元素本身。

## 三十八、过渡

transition:property transition-duration transition-timing-function transition-delay

transition: 要过渡的属性 花费时间 运动曲线 延迟时间;

transition-property设置哪条 CSS 使用过渡none/all/CSS 属性名称列表

transition-duration 设置过渡的过渡时间，单位是 秒 s （ms 毫秒）例如： 0.5s 这个s单位必须写

transition-timing-function设置过渡的时间曲线linear匀速/ease逐渐慢下来/ease-in加速/ease-out减速/ease-in-out先加速后减速/ ，默认是ease

transition-delay 设置过渡的延迟时间（何时开始），默认是 0s，立即开始

如果有多组属性变化，用逗号隔开。

例如：transition: width 0.6s ease 0s, height 0.3s ease-in 1s; //执行0.6s，0s后开始

如果想要所有的属性都变化过渡， 写一个all 就可以，例如：

transition: all 0.6s;

## 三十九、动画animation

div {

width:100px;

height:100px;

background:red;

position:relative;

animation:turnaround 5s infinite;

/\*-moz-animation:turnaround 5s infinite;

-webkit-animation:turnaround 5s infinite;

-o-animation:turnaround 5s infinite;\*/

}

@-moz-keyframes turnaround {

0% {top:0px; left:0px; background:red;}

25% {top:0px; left:100px; background:blue;}

50% {top:100px; left:100px; background:yellow;}

75% {top:100px; left:0px; background:green;}

100% {top:0px; left:0px; background:red;}

}

@keyframes name{

from{

}

to{

}

}

animation:name duration delay count state

animation-name 使用@keyframes 定义的动画

animation-delay 设置动画的延迟时间

animation-iteration-count 设置动画播放次数数字/infinite

animation-play-state设置动画播放状态paused/running

animation-timing-function 设置动画的时间曲线linear/ease/ease-in/easeout/ease-in-out

animation-direction 设置动画反向播放 normal/alternate;

animation-fill-mode: forwards动画保持最终状态

## 四十、多列

(1)column-width: auto;

column-width: 100px;

列宽度column-width属性

Internet Explorer 10以上 和 Opera 支持 column-width 属性。

Firefox 支持替代的 -moz-column-width 属性。

Safari 和 Chrome 支持替代的 -webkit-column-width 属性。

注意：Internet Explorer 9 以及更早版本的浏览器不支持 column-width 属性。

列数column-count属性

列间距column-gap属性：column-gap属性不能单独设置，只有通过column-count属性为元素进行分列后才能设置分列间距，不能为负值。

(2) 列边框column-rule属性

column-rule:5px dashed red;

column-span 属性规定元素应横跨多少列。

1 元素应横跨一列。

all 元素应横跨所有列。

## 四十一、box-sizing（大小用百分比）

Width表内容的宽度

Internet Explorer、Opera 以及 Chrome 支持 box-sizing 属性。

Firefox 支持替代的 -moz-box-sizing 属性。

CSS3的box-sizing属性

语法：box-sizing : content-box || border-box || inherit;

当为content-box时，将采取标准模式进行解析计算，默认值

当为border-box时，将采取怪异模式解析计算

当为inherit时，将从父元素来继承box-sizing属性的值

标准盒模型

在标准模式下的盒模型:盒子总宽度/高度=width/height+padding+border+margin

IE下的怪异盒模型

在怪异模式下的盒模型:盒子的总宽度和高度width/height是包含内边距padding和边框border宽度在内的

盒子总宽度/高度=width/height + margin = (内容区宽度/高度 + padding + border )+ margin;

也即是说 width = 内容区宽度 + padding + border

box-sizing的运用场景：假如项目布局使用的是自适应的布局方式，div给出的宽度是百分比的形式，即框占窗口宽度的50%，但边界和内边距是用像素来表示的怎么办？为了避免这种问题，可以使用属性box-sizing来调整框模型。使用border-box，来将框模型更改成这个新的模型。

## 四十二、弹性盒子（浮动+清除浮动）

（1）display:flex;

flex-direction:row;横向排布

flex-wrap: nowrap;

justify-content:space-between;平均分布

（2）CSS3 弹性盒，是一种当页面需要适应不同的屏幕大小以及设备类型时确保元素拥有恰当的行为的布局方式。

弹性盒模型的语法基础概念

Flex是Flexible Box的缩写，意为"弹性布局"，用来为盒状模型提供最大的灵活性。

任何一个容器都可以指定弹性布局

设置 flex 布局以后，子元素的 float,clear,vertical-align 等样式全部失效

display:flex;或 display:inline-flex;

弹性父元素属性

display:flex;

flex-direction: row|row-reverse|column| column-reverse指定了弹性子元素在父容器中的位置。

justify-content:flex-start|flex-end|center|space-between| space-around 水平对齐方式

align-items:flex-start | flex-end | center | baseline | stretch 垂直对齐方式

flex-wrap: nowrap|wrap|wrap-reverse|initial|inherit;

align-content:当flex-wrap起作用发生换行时，该属性控制每行的对齐方式

弹性子元素属性

order:num 数值小的排在前面。可以为负值

flex 属性用于指定弹性子元素如何分配空间，按比列分配

有关CSS3弹性盒布局方式的更多详情请查看

<https://www.jianshu.com/p/5856c4ae91f2>

## 四十三、css属性hack（IE的兼容问题，少的兼容写在后面，继承关系）

background-color: pink;

background-color: red\9; /\* ie 6.7.8.9.10 \*/

background-color: purple\0; /\* ie 8.9.10 \*/

\*background-color: blue; /\*ie6.7专属 只有这2个浏览器识别 \*/

\_background-color: lime; /\* ie6专属 只有这个浏览器识别 \*/

# 浏览器兼容

## 总结：

-moz-

-webkit-

-o-

# JQ

## 一、页面加载后完成

用$代替$(document).ready(function(){

//执行代码

})。

$(function(){

//执行代码

})

## 二、选择器

### （1）子代（层级）选择器

子代选择器是在给定父元素下匹配所有的子元素，子元素只能是自己的孩子元素，不能是孙子、曾孙里面的元素。

//匹配所有后代元素

$('div span')

//匹配直接子元素

$('div>span')

//匹配所有紧接在 prev 元素后的 next 元素(相邻选择器，指后面的一个弟弟，直接的，不能隔开)

//$("prev + next")

//匹配 prev 元素的所有同辈 siblings 元素(相邻兄弟选择器，指后面的弟弟)

$("prev ~ siblings")

### （2）子元素 （初级教材无该内容）

• :first-child 匹配第一个子元素

• :last-child 匹配最后一个子元素

• :first-of-type 选择所有相同的元素名称的第一个兄弟元素

• :last-of-type 选择所有相同的元素名称的最后一个兄弟元素

• :nth-child 匹配其父元素下的第 N 个子或奇偶元素,下标1开始

• :nth-last-child() 选择所有他们父元素的第 n 个子元素。计数从最后一个元素开始到第一个

• :nth-last-of-type() 选择的所有他们的父级元素的第 n 个子元素，计数从最后一个元素到第一个

• :nth-of-type() 选择同属于一个父元素之下，并且标签名相同的子元素中的第 n个

• :only-child 如果某个元素是父元素中唯一的子元素，那将会被匹配

• :only-of-type 选择所有没有兄弟元素，且具有相同的元素名称的元素。

### （3）位置选择器

• :first匹配第一个元素，$('selector:first')

• :last获取最后一个元素，$('selector:last')

• :even匹配所有索引值为偶数的元素，从0开始算起，$('selector:even')

• :odd匹配所有索引值为奇数的元素 ，从0开始算起，$('selector:odd')

• :eq匹配一个给定索引值的元素，从0开始索引，$('selector:eq(n)')

• :gt匹配所有大于给定索引值的元素，，从0开始索引，从指定索引的下一个开始被选取,不包括本身。$('selector:gt(n)')

• :lt匹配所有小于给定索引值的元素，从0开始索引，从指定索引的下一个开始被选取,不包括本身。$('selector:lt(n)')

### （4）属性选择器

基于元素的属性（属性名或属性值或属性名和属性值）选取元素。以<button type="btn btn-large"></button>为例

• [attribute] 匹配包含给定属性的元素,不限定属性值，$('[type]')

• [attribute='value']匹配给定的属性和属性值的元素 $('[type="btn"]')

• [attribute^='value'] 匹配给定的属性值以特定值开头的元素$('[type^="btn"]')

• [attribute$='value'] 配给定的属性值特定值结尾的元素$('[type$="large"]')

• [attribute\*='value'] 匹配给定的属性值包含给定值的元素$('[type\*="btn"]')

## 三、JQ事件（选绑做）

### (1)事件的绑定

分三步：选中需要使用脚本进行事件响应的元素节点（选），声明需要在选中节点上响应的事件（绑），指定当事件发生时需要运行的代码（做）。

简单说：选择元素、指定事件和调用代码。

使用bind()绑定

使用简单绑定

事件:bind（“click”，function（）{}）

做：click、dblclick（双击）、hover（function(){}）(不用bind也行)

### (2) 键盘事件：当用户操作键盘时发生

• keydown()事件：当键盘或按钮被按下时，发生 keydown 事件，按住不放会反复触发。

• keyup()事件：当键盘键被松开时发生 keyup 事件。

• keypress()事件：当键盘或按钮被按下时，发生 keypress 事件，键入字符。

### (3)鼠标事件

• click()事件：用户在元素单击鼠标左键，并在相同元素上松开左键时触发。

• dblclick()事件：在用户完成迅速连续的两次点击之后触发，双击的速度取决于

操作系统的设置。一般双击事件在页面中不经常使用。

• mousedown()事件：在用户敲击鼠标键时触发，跟 keydown 事件不一样，该事

件仅在按下鼠标时触发。

• mouseup()事件：在用户松开鼠标时触发，如果在与按下鼠标的元素相同元素

上松开，那么 click 事件也会触发。

• mousemove:用户移动鼠标触发。

• mouseover:用户鼠标移动到一个元素上触发。

• mouseout:用户鼠标从一个元素上移开时触发。

mouseenter:当鼠标穿过元素时触发。

mouseleave:当鼠标离开元素时触发。

• hover()事件： 一个模仿悬停事件（鼠标 over 及 out）的方法。

### (4) UI事件：

当与浏览器UI本身（而不是网页）交互时发生的事件，有

•load : Web页面加载完成

•unload: Web页面正在卸载

•error:浏览器遇到javascript错误或者不存在的资源

•resize:浏览器窗口大小变化

•scroll:用户使用滚动条移动页面。

$(function(){

$(window).scroll(function(){

console.log("滚动！");

});

})

### (5) 焦点事件：

元素得到或失去焦点时触发。

focus:元素得到焦点

blur:元素失去焦点

$("#login").submit(function(){

alert("欢迎使用");

});

### (6)表单事件：

当用户与表单元素进行交互时发生。

• change:复选框、单选框或单选按钮值发生变化时触发。

• submit事件：当提交表单时，会发生 submit 事件。

• reset事件：当单击了表单重置按钮时触发。

• select事件：用户在表单域用选择文本时触发。

$("#user").focus(function(){

$("#user-tip").html("手机号码、邮箱、注册账号");

})

$("#user").blur(function(){

$("#user-tip").html("");

})

### (7) 事件对象：

事件对象在事件发生时事件的状态信息。事件对象会作为参数传递给任何事件处理程序或事件监听的函数。如果需要传递一个参数给处理函数，事件对象会作为匿名封装函数的第一个参数传递进去（自动发生），常用e来表示 。

• e.target: 最初触发事件的 DOM 元素

• e.type:发生事件的类型

• e.cancelable:是否可以撤销事件在这个元素上的默认行为

• e.preventDefault() 阻止默认事件行为的触发

• e.stopPropagation() 防止事件冒泡到 DOM 树上，也就是不触发的任何前辈元素上的事件处理函数

• e.pageX 鼠标相对于文档的左边缘的位置

• e.pageY 鼠标相对于文档的上边缘的位置

• e.which 针对键盘和鼠标事件，这个属性能确定你到底按的是哪个键或按钮

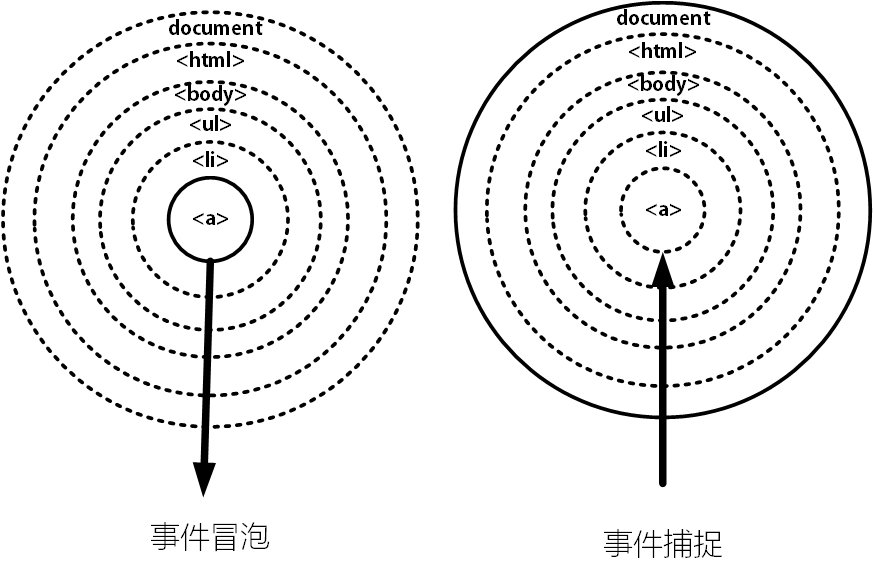
### (8) 事件冒泡

多个嵌套元素绑定了相同的事件（如click)时，会触发各元素的事件。只有当代码在一个元素和其祖先元素或后代元素上都有事件处理程序时，事件流才会变得非常重要。

HTML元素都位于另一些元素中。如果移动鼠标到一个链接上，或者点击一个链接，同样会把鼠标移到它的父元素上，或者点击它的父元素。

事件捕获：从最不精确的对象(document 对象)开始触发，然后到最精确(也可以在窗口级别捕获事件，不过必须由开发人员特别指定)

事件冒泡：事件按照从最特定的事件目标到最不特定的事件目标(document对象)的顺序触发。



## 四、DOM

### （1）内部插入（增加孩子）

• append() 向每个匹配的元素内部最后面追加内容

$("ul").append($("<li>添加的列表</li>"));

• appendTo() 把所有匹配的元素追加到另一个指定的元素集合中

$("<li>添加的列表</li>").appenTo($("ul"));

• prepend() 向每个匹配的元素内部最前面前置内容

• prependTo() 把所有匹配的元素前置到另一个、指定的元素集合中

### （2）外部插入（增加兄弟）

• after() 在每个匹配的元素之后插入内容

• before() 在每个匹配的元素之前插入内容

• insertAfter() 把所有匹配的元素插入到另一个、指定的元素集合的后面

• insertBefore() 把所有匹配的元素插入到另一个、指定的元素集合的前面

### （3）获取和设置元素节点内容

• text() - 设置或返回所选元素的文本内容

• html() - 设置或返回所选元素的内容（包括 HTML 标记）

• val() - 设置或返回表单字段的值

### （4）获取和设置元素的属性

获取属性：$('selector').attr('属性名称');

设置属性：$('selector').attr('属性名称','属性值');

$("img").click(function(){

alert($(this).attr("src"));

$(this).attr("src","img/sunflower.jpg")

})

### （5）获取和设置元素样式

$('selector').css('样式属性'[,'样式属性值'])：设置或者返回被选元素的一个或多个样式属性

$('selector').addClass('类名')：添加类样式

$('selector').removeClass('类名')：删除类样式

$('selector').toggleClass('类名')：切换类样式

### （6）替换

• replaceWith() 将所有匹配的元素替换成指定的 HTML 或 DOM 元素

$("div span").replaceWith("<p>新内容</p>");

• replaceAll() 用匹配的元素替换掉所有 selector 匹配到的元素

### （7）删除

• empty() 删除匹配的元素集合中所有的子节点，并不删除元素本身，只清空内容。

• remove() 从 DOM 中删除所有匹配的元素及子节点

### （8）克隆

• clone() 克隆匹配的 DOM 元素并且选中这些克隆的副本(克隆他以及他的孩子)

$("ul").click(function(){

$("ul").clone().appendTo($("body"));

})

## 五、js与jq的转换

<p>js转jq对象</p>

<script type="text/javascript">

var p=document.getElementsByTagName(("p")[0]);

$(p).css("color","#f00");

</script>

<p>jq转js对象</p>

<script type="text/javascript">

var p=document.getElementsByTagName(("p")[0]); $p=$("p:eq(1)");

$p[0].style.color="#00f";

</script>

## 六、效果

显示

show(speed,easing,fn)时间，速度，函数

隐藏

hide(speed,easing,fn)时间，速度，函数

切换

toggle(speed,easing,fn)时间，速度，函数

easing:linear（匀速）、swing（加速）

淡入

fadeIn(“slow”,”swing”,function(){})

淡出

fadeOut(“slow”,”swing”,function(){})

淡入淡出切换

fadeToggle(“slow”,”swing”,function(){})

淡入到

fadeTo(“slow”,0.5，”swing”,function(){})

总：0.5为透明度

淡入淡出切换只改透明度

显示隐藏切换改大小、高度、透明度

上拉

slideUp(“fast”,”linear”,function(){})

下拉

slideDown(“fast”,”linear”,function(){})

上拉下拉切换

slideToggle(“fast”,”linear”,function(){})

动画执行

$("ul").click(function(){

$(this).animate({height:"40px",marginTop:"+=100px"},"1000",function(){})

})

动画停止

$("#stop").click(function(){

$(“ul”).stop();

})

## 七、Ajax

<button>加载数据HTML</button>

<button>加载数据JSON</button>

<table>

<tr>

<th>书名</th>

<th>作者</th>

<th>价格</th>

</tr>

</table>

<script type="text/javascript">

$("button:eq(0)").click(function(){

$("body").load("data/content.html");

})

$("button:eq(1)").click(function(){

$.get("data/books.json",function(data){

// var html = '';

// for(var)

// console.log(data);

$("table").append(data.toString());

},"json");

})

</script>

# H5

1. 表单

input：

value默认显示控件上的值

placeholder显示提示的内容

progress进度条，属性max等于100，value自定义

# JS

## 一、数组

array.pop();删除并返回数组的最后一个元素

array.unshit();向数组最前面添加一个元素

array.shit();向数组最前面删除并返回第一个元素

# 总结：

一、HTML不区分大小写、JS区分大小写

二、==先类型先转换为数值再比较，===直接比较

三、p {font: 20px/24px “微软雅黑”;} /\* 正确 \*/必须写

p {font: italic bold 20px/24px ;} /\* 错误 \*/

四、font-family: FangSong,'黑体';

英文不用加单引号，中文要

五、网页默认字体大小为16px

六、font-variant: small-caps; 字-体变形（小体大写）

direction 属性规定文本的方向 / 书写方向。

ltr 默认。文本方向从左到右。

rtl 文本方向从右到左。

text-transform: capitalize;首字母大写

text-transform: uppercase;全部字母大写

text-transform: lowercase;全部字母小写

white-space: pre;预排版，怎么打怎么显示

white-space: nowrap; 不换行

word-spacing: 10px; 英文间距

letter-spacing: 10px;中文间距

（2）margin可以为负值，而border和padding不支持负值。

parseInt("0.12"); //0  
parseInt(".12"); //NaN

遇到非数字，停止转换

H5新增表单元素

* datalist
* keygen
* output