nth-of-type() 选择器

* p:nth-of-type(2){

background:#ff0000;

}

规定属于其父元素的第二个 p 元素的每个 p

* p:nth-of-type(odd){

background:#ff0000;

}

p:nth-of-type(even){

background:#0000ff;

}

奇数p和偶数p是两种不同的颜色

* 使用公式 (an + b)

第an+b(n=0,1,2,3…)个p被选中。从第一个p开始

p:nth-of-type(3n+0){

background:#ff0000;

}

指定下标是 3 的倍数的所有 p 元素的背景色

ul>li:nth-child() span

属性选择器

input[type='submit']

p中套p时很奇怪，没有总结出规律

<!DOCTYPE html>

<html>

<head>

<style>

p:nth-of-type(3)

{

background:#ff0000;

}

</style>

</head>

<body>

<h1>这是标题</h1>

<p>第一个段落。<p>子段落</p>444444</p>

<p>第五个段落。</p>

<p>第六个段落。</p>

</body>

</html>

这段代码谁的背景色也没有改变

奥，知道为什么了

1.内联元素却不能包含块元素，它只能包含其他的内联元素

2.一些块元素不可以包含另一些块元素

p中不要包含任何其他块级元素

CSS **list-style**属性

ul{

list-style:square inside url('/i/arrow.gif');

}

就是这三个属性一起写

ul 和li都能

1. list-style-type
2. list-style-position
3. list-style-image

1.list-style-type属性

|  |  |
| --- | --- |
| none | 无标记。 |
| disc | 默认。标记是实心圆。 |
| circle | 标记是空心圆。 |
| square | 标记是实心方块。 |
| decimal | 标记是数字。 |
| decimal-leading-zero | 0开头的数字标记。(01, 02, 03, 等。) |
| lower-roman | 小写罗马数字(i, ii, iii, iv, v, 等。) |
| upper-roman | 大写罗马数字(I, II, III, IV, V, 等。) |
| lower-alpha | 小写英文字母The marker is lower-alpha (a, b, c, d, e, 等。) |
| upper-alpha | 大写英文字母The marker is upper-alpha (A, B, C, D, E, 等。) |
| lower-greek | 小写希腊字母(alpha, beta, gamma, 等。) |
| lower-latin | 小写拉丁字母(a, b, c, d, e, 等。) |
| upper-latin | 大写拉丁字母(A, B, C, D, E, 等。) |
| hebrew | 传统的希伯来编号方式 |
| armenian | 传统的亚美尼亚编号方式 |
| georgian | 传统的乔治亚编号方式(an, ban, gan, 等。) |
| cjk-ideographic | 简单的表意数字 |

2）list-style-position

|  |  |
| --- | --- |
| inside | 列表项目标记放置在文本以内，且环绕文本根据标记对齐。 |
| outside | 默认值。保持标记位于文本的左侧。列表项目标记放置在文本以外，且环绕文本不根据标记对齐。 |

**display:none**

不显示

一个ul的li设置border的时候，中间的border要比两端的border粗，因为中间的是两个border摞在一起

解决方法:

ul>li{

border: 1px solid #000;

border-bottom: none;

}

ul>li:**last-child**{

border-bottom: 1px solid #000;

}

<div class="box">

<h1></h1> 列表加标题

<ul>

<li></li>

<li></li>

</ul>

</div>

* li标签里边有个span,而且span是inline-block的话，span的line-height会继承li的line-height, 如果此时li有padding，那么就会发生问题

padding

**li**

**span**

line-height

变长了！

解决方法:span的line-height属性设置为normal

* 子元素浮动之后父元素的高度无法被撑起，解决方法

父元素加属性overflow:hidden

* 实战学的标准开头

1. \*{
2. **padding**: 0px;
3. **margin**: 0px;
4. **text-decoration**: none;  //取消a标签下划线
5. **list-style**: none;
6. }
7. html, body{
8. **width**: 100%;
10. }

**表单**

<form>

First name:<br>

<input type="text" name="firstname" readonly="readonly"><br>

//单选框

<input type="text" name="sex" placeholder="man">Male<br>

<input type="text" name="name1" disabled="disabled" >//禁用的input元素

</form>

<form>

<input type="radio" name="sex" value="male" checked>Male

<br>

<input type="radio" name="sex" value="female">Female

</form>

<input type="text" name="n1" oninput="f1()">

<script type="text/javascript">

function f1(){

alert('fff')

}

</script>

submit不跳转的方法：

form标签加属性target="rfFrame"

form外

<iframe id="rfFrame" name="rfFrame" src="about:blank" style="display:none;"></iframe>

* input可以直接设置width和height

placeholder 属性提供提示信息。该提示会在输入字段为空时显示，并会在字段获得焦点时消失。

CSS **font**属性

至少要指定字体大小font-size和字体系列font-family

未设置的属性会使用其默认值

* font-style 规定字体样式

normal

italic 斜体

oblique 倾斜

如果一个字体没有倾斜字体样式，则italic无效，oblique有效，拉斜了

* font-variant 规定字体异体

font-variant设置小型大写字母的字体显示文本，这意味着所有的小写字母均会被转换为大写，但是所有使用小型大写字体的字母与其余文本相比，其字体尺寸更小

取值：normal small-caps

* font-weight 规定字体粗细

normal bold bolder lighter

100 200 300 400 500 600 700 800 900

定义由粗到细的字符。400 等同于 normal，而 700 等同于 bold。

* font-size/line-height 规定字体尺寸和行高
* font-family 规定字体系列

font:14px/1.3 'Segoe UI',Arial, sans-serif;

有逗号的值都是并列；网上的一般都只设置size和family

不设置html的css属性时，body就是根节点，整个页面的属性就是body的属性

设置html的css属性后，根节点变为html

min-height

可能取值：

length 定义元素的最小高度。默认值是 0。

% 定义基于包含它的块级对象的百分比最小高度

* box-shadow: 10px 10px 5px #888888 inset;

 

内阴影 外阴影

h-shadow 必需。水平阴影的位置。允许负值。

v-shadow 必需。垂直阴影的位置。允许负值。

blur 可选。模糊距离。上面的5px

spread 可选。阴影的尺寸。

color 可选。阴影的颜色

inset 可选。将外部阴影 (outset) 改为内部阴影

* text-shadow: h-shadow v-shadow blur color;

text-shadow: 5px, 5px, 3px, #000;

文字阴影

可以设置多个阴影，用逗号隔开

transition:0.25s all;

div{

width:100px;

height:100px;

transition:width 2s;

-moz-transition:width 2s; /\* Firefox 4 \*/

-webkit-transition:width 2s; /\* Safari and Chrome \*/

-o-transition:width 2s; /\* Opera \*/

}

div:hover

{

width:300px;

}

* transition-property 规定设置过渡效果的 CSS 属性的名称
* transition-duration 规定完成过渡效果需要多少秒或毫秒。必须设置
* transition-timing-function 规定速度效果的速度曲线
* linear 速度相同
* ease 慢速开始，然后变快，然后慢速结束
* ease-in 规定以慢速开始的过渡效果（等于 cubic-bezier(0.42,0,1,1)）。
* ease-out 规定以慢速结束的过渡效果（等于 cubic-bezier(0,0,0.58,1)）。
* ease-in-out 规定以慢速开始和结束的过渡效果（等于 cubic-bezier(0.42,0,0.58,1)）。
* cubic-bezier(n,n,n,n) 在 cubic-bezier 函数中定义自己的值。可能的值是 0 至 1 之间的数值。
* transition-delay 定义过渡效果何时开始

默认值transition: all 0 ease 0

这玩意就是跟hover配合的，或者跟transform配合

transition: transform 0.1s, opacity 0.3s;

transform: translate(10px, 3px)

* translate(10px, 3px) 平移水平向右10px，垂直向下3px
* scale(0.7, 1.5) 缩放比例
* rotate(30deg) 顺时针旋转角度
* rotateX(30deg) 沿着x轴3D旋转

rotateY(angle)

rotateZ(angle)

* skew(x-angle,y-angle) 定义沿着 X 和 Y 轴的 2D 倾斜转换

p:after

{

content:"台词：-";

}

在p标签后插入新内容

:before和:after伪元素在页面中生成的元素在缺省情况下是“内联(inline)”

content中的内容

* [String] 使用引号包括一段字符串，将会向元素内容中添加字符串。

a:after { content: "↗"; }

* attr() 调用当前元素的属性，将图片的 Alt 提示文字或者链接的 Href 地址显示出来。

a:after { content:"(" attr(href) ")"; }

* url() uri() – 用于引用媒体文件。

h1::before { content: url(logo.png); }

* counter() – 调用计数器，可以不使用列表元素实现序号功能。

h2:before {

counter-increment: chapter;

content: "Chapter " counter(chapter) ". "

}

a:hover::before, a:hover::after { position: absolute; }

a:hover::before { content: "\5B"; left: -20px; }

a:hover::after { content: "\5D"; right: -20px; }

鼠标悬浮时添加左右方括号

* before和after我看到大部分例子写成position: absolute然后调整left、right、top或者bottom，absolute是相对于最近的有定位的父级元素定位的，所以必须把原来的元素设置成position: relative。这样看来，其实a和a::before是父子关系

清除浮动

1.

.clearfix:after{content: "";display: block;clear: both;} <!--clearfix是预清浮动的父级元素-->

.clearfix{\*zoom:1;}

2.空div clear:both

鼠标指针变成

cursor:pointer; 一只手，指向连接类型

cursor:wait; 程序正忙

text 文本

default

a:link

a:visited

a:hover

a:active

未访问连接、已访问连接、悬浮、鼠标按下

CSS**渐变Gradients**

* 线性渐变Linear Gradients

#grad1 {

height: 200px;

background: -webkit-linear-gradient(red, blue); /\* Safari 5.1 - 6.0 \*/

background: -o-linear-gradient(red, blue); /\* Opera 11.1 - 12.0 \*/

background: -moz-linear-gradient(red, blue); /\* Firefox 3.6 - 15 \*/

background: linear-gradient(red, blue); /\* 标准的语法（必须放在最后） \*/

}

/\*默认从上到下\*/

* background: linear-gradient(to right, red , blue);、

从左到右

* background: linear-gradient(to bottom right, red , blue);

从左上角到右下角

* background: linear-gradient(red, green, blue);

多个颜色节点均匀分布

* background: linear-gradient(to right, rgba(255,0,0,0.2), rgba(255,255,0,1));

最好使用透明度，否则看的有点难受

* background: repeating-linear-gradient(red, yellow 10%, green 40%);

效果：

green在40%的位置，red在一开始，yellow在10%的位置。后面重复上面的

* 径向渐变Radial Gradients

#grad {

background: -webkit-radial-gradient(red, green, blue); /\* Safari 5.1 - 6.0 \*/

background: -o-radial-gradient(red, green, blue); /\* Opera 11.6 - 12.0 \*/

background: -moz-radial-gradient(red, green, blue); /\* Firefox 3.6 - 15 \*/

background: radial-gradient(red, green, blue); /\* 标准的语法 \*/

}

最中间红，最外边蓝，一层一层向外变换

* background: radial-gradient(circle, red, yellow, green)

circle 表示圆形，ellipse 表示椭圆形。默认值是 ellipse

* background: radial-gradient(60% 55%, closest-side,blue,green,yellow,black);

closest-side

farthest-side

closest-corner

farthest-corner

* user-select

.box{

-webkit-user-select: none;

-moz-user-select: none;

-ms-user-select: none;

user-select: none;

}

是否可以被选中

可能取值：

* none:文本不能被选择
* text:可以选择文本
* all：当所有内容作为一个整体时可以被选择。如果双击或者在上下文上点击子元素，那么被选择的部分将是以该子元素向上回溯的最高祖先元素。
* Element:可以选择文本，但选择范围受元素边界的约束
* outline: none;

border外边一圈轮廓

* text-indent:50px;

首行缩进

max-width:

max-height:

10%: 定义基于包含它的块级对象的百分比最大宽度

100px: 最大宽度值

* background-position:

默认值：0% 0%。

* top left

top center

top right

center left

center center

center right

bottom left

bottom center

bottom right

如果仅规定了一个关键词，那么第二个值将是"center"

* x% y%

第一个值是水平位置，第二个值是垂直位置。

左上角是 0% 0%。右下角是 100% 100%。

如果仅规定了一个值，另一个值将是 50%。

* xpos ypos

第一个值是水平位置，第二个值是垂直位置。

左上角是 0 0。单位是像素 (0px 0px) 或任何其他的 CSS 单位。

如果您仅规定了一个值，另一个值将是50%。

可以混合使用 % 和 position 值。

* background-attachment: fixed;

默认值scroll，背景图片随滚动条滚动

fixed不随屏幕滚动条滚动

* background-size:80px 60px;
* length

设置背景图像的高度和宽度

* percentage

以父元素的百分比来设置背景图像的宽度和高度

如果只设置一个值，则第二个值会被设置为 "auto"，按比例缩放

* cover

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。按比例缩放的

背景图像的某些部分也许无法显示在背景定位区域中。

* contain

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

我觉得这个很好用，按比例缩放，还尽量充满容器

* background-repeat: repeat-y;

repeat-x

no-repeat

让图片铺满整个屏幕，网上很多教程都是给body设置背景图片，这我觉得这样不好，既不符合元素的逻辑关系（铺满屏幕的照片在所有元素之上），又是在body上做文章，不具有可移植性

我这样实现：在body末尾加一个div.popBigPhoto

.popBigPhoto{

width: 100%;

height: 100%;

background-image: url('../images/pic1.jpg');

background-attachment: fixed;

background-position: center center;

background-size: contain;

background-color: rgba(0, 0, 0, 0.6);/\*设置背景灰色蒙版\*/

background-repeat: no-repeat;

position: fixed;

z-index: 100;

top: 0;

left: 0;

display: none;

}

document.querySelector("p");

参数是CSS选择器，只选择符合条件的第一个，如果要选择所有，用querySelectorAll()方法

获取文档中 class="example" 的第一个 <p> 元素:

document.querySelector("p.example");

获取文档中有 "target" 属性的第一个 <a> 元素：

document.querySelector("a[target]");

width: 50vw;

浏览器100vw表示的就是浏览器的视窗宽度(Viewport)

# HTML5

**dataset**属性存取data-\*自定义属性的值

* dataset 属性是HTML5 JavaScript API的一部分，用来返回一个所有选择元素 data- 属性的DOMStringMap对象
* 使用这种方法时，去掉data- 前缀
* data- 属性名如果包含了连字符，例如data-date-of-birth ，连字符将被去掉，并转换为驼峰式的命名dataOfBirth

<div id="user" data-id="123" data-name="helloWorld" data-date-of-birth>你好世界</div>

var a = document.querySelector('#user');

a返回

DOMStringMap {

dateOfBirth:""

id:"123"

name:"helloWorld"

\_\_proto\_\_:DOMStringMap

}

* dataset.id

返回'123'

* dataset.dataOfBirth

返回""

* 设置data-date-of-birth的值.

a.dataset.dateOfBirth = '1985-01-05';

* 增加

a.dataset.attr1='15522'

* 删除

delete a.dataset.id

配合querySelectorAll使用更佳

选择所有包含 'data-flowering' 属性的元素

document . querySelectorAll ( '[data-flowering]' ) ;

选择所有包含 'data-text-colour' 属性值为red的元素

document . querySelectorAll ( '[data-text-colour="red"]' ) ;