

3D 移动: `translate3d`

作用: 3D 移动在 2D 移动的基础上多了一个可以移动方向, 就是 Z 轴的方向。

`transform: translateX(100px)`: 仅仅是在 X 轴上移动

`transform: translateY(100px)`: 仅仅是在 Y 轴上移动;

`transform: translateZ(100px)`: 仅仅在 Z 轴上移动 Z 轴一般是用 PX 为单位

`transform: translate3d(x, y, z)`: 其中 x, y, z 分别指的是移动的轴的方向的距离

3D 旋转 `rotate3d`

作用: 3D 旋转可以让元素在三维平面沿着 X 轴, Y 轴, Z 轴或者自定义旋转;

`transform: rotateX(45deg)`: 沿 X 轴方向旋转 45 度;

`transform: rotateY(45deg)`: 沿着 Y 轴方向旋转 45 度;

`transform: rotate3d(x, y, z, 45deg)`: 沿着自定义方向旋转 45 deg 为角度。



2. 调用动画

Animation - name : 动画名称;

持续时间: Animation - duration : 持续时间;

动画简写

Animation: 动画名称 持续时间 动画曲线
何时开始 播放次数 是否反向 动画起始或
结束状态; animation: move ss linear 2s
infinite alternate;

3D 转换

X轴: 水平向右 注: X右边是正值, 左边是负值

Y轴: 垂直向下 注: Y下边是正值, 上边是负值.

Z轴: 垂直屏幕 注: 往外是正值, 往内是负值.

3D 位移: translate 3d (x, y, z);

3D 旋转: rotate 3d (x, y, z);

透视: Perspective;

3D 呈现: transform-style.



旋转点设置:

transform - Origin : X Y;

注: 后面的参数 X 和 Y 用空格隔开。

X, Y 默认转换的中心点是元素中心点 (50%, 50%)
还可以给 X, Y 设置像素或者方位名词 (top, bottom, left, right)

缩放: Scale

例: 里面写数字不跟单位, 就是倍数的意思。1 就是 1 倍, 2 就是 2 倍。

transform: Scale (X, Y); transform: Scale (2, 2);

动画:

基本使用:

1. 先定义动画: 2. 再使用动画

用 keyframes 定义动画

@ keyframes 动画名称 {

0% { width: 100px; }

100% { width: 200px; } }



keyword 关键字

作用: keyword是网页的关键字,是搜索引擎的关注点之一。

keyword最好限制为6~8个关键词,关键词之间有英文用逗号隔开,采用关键词1、关键词2的形式过渡。

transition: 过渡动画;通常与 a: hover 配合

transition: 过渡属性 (width, height) (all 同时改变宽、高) 花费时间 (s)

移动: translate.

作用: 2D与3D里面的一种功能,可以改变在页面中的位置,类似定位。

语法: transform: translate (X, Y) 或者分开写

transform: translate X (n);

transform: translate Y (n);

旋转: rotate

transform: rotate (度数 deg)

注: 1. rotate里面跟度数,单位deg. 例如 rotate (45 deg); 2. 默认的旋转中心点是元素的中心点



定位：将盒子定在某一个位置，所以定位也是在摆盒子；按照定位的方式移动盒子；定位 = 定位模式 + 边偏移

1. 定位模式：决定元素的定位方式，它通过 CSS 的 `position` 属性来设置，其值有四个：
`static` (静态定位)；`relative` (相对位置)；`absolute`：绝对定位；`fixed`：固定定位；

2. 边偏移：`top`、`bottom`、`left`、`right`；
相对定位：元素在移动位置的时候，是相对于对它原来的位置来说，但是原来位置依然占据；选择器：`position: relative`；

绝对定位：`absolute`。

特点：绝对定位元素移动位置时，是相对于它祖先元素来说：

注：1. 如果没有祖先元素或祖先元素没有定位，他会依浏览器版面来定位。

2. 如果祖先元素有定位（相对、绝对、固定定位），则以最近一级有定位的祖先元素为参考移动位置。

3. 绝对定位移动位置，原来位置不占空间



1.2.2: 浮动的元素会具有行内块的特性, 浮动盒子没有间隙。

1.2.3: 浮动元素一般配用标准流块盒子配合使用, 以更好控制浮动元素的位置, 能约束浮动的范围, 范围为父级块大小。

1.3 清除浮动

1. 额外标签法: 又名隔墙法, 在浮动元素末尾添加空标签, 例 `<div style="clear: both"> </div>`

2. 父级添加 overflow 法: hidden, auto, 或 Scroll.

3. after 伪元素法:

范例: • clearfix: after {

content: ""; display: block; height: 0;
clear: both; visibility: hidden; }

CSS 定位

特点: 绝对位置 定位是让盒子自由在某个盒子内指定位置或固定屏幕的某个位置, 并且能压住其他的盒子



1.2.2: 浮动的元素会具有行内块的特性, 浮动盒子没有间隙。

1.2.3: 浮动元素一般配用标准流块级盒子配合使用, 以更好控制浮动元素的位置。能约束浮动的范围, 范围为父级块大小。

1.3 清除浮动

1. 额外标签法: 也叫隔墙法, 在浮动元素末尾添加空标签, 例 `<div style="clear: both"> </div>`

2. 父级添加 overflow 法: hidden, auto, 或 Scroll.

3. after 伪元素法:

范例: `clearfix: after {`

`content: ""; display: block; height: 0;`
`clear: both; visibility: hidden; }`

CSS 定位

特点: ~~绝对位置~~ 定位是让盒子自由在某个盒子内移动位置或固定屏幕的某个位置, 并且能压住其他的盒子



1.8 圆角边框

语法: `border-radius: 10px / 50% (百分比);`
`radius (半径)`

1.9 盒子阴影

`box-shadow: 10px (水平阴影位置) 10px (垂直阴影的位置) 10px (模糊距离) 10px (阴影尺寸)`
`black (颜色) inset (内部阴影);`

浮动 (float)

1.1 普通流:

浮动流: 可以让多个块级元素一行内排列显示

标准流: 标签按照规定好的默认方式排列, 也是最基本的排布方式.

注: 多个块级元素用标准流来纵向排列, 多个块级元素横向排列时, 用浮动流来排列. 如果给子元素, 浮动标签给定 `margin` 值, 他不会与父级 `margin` 合并;

1.2 浮动特性

1.2.1: 浮动元素会脱离标准流

1.2.2: 浮 (脱离标准流普通流控制) 动 (移动到指定位置, 且不再保留原来位置)



1.5: 盒子居中

1.5.1: 第一步: 要设置 width 大小. 第二步: 设置 margin: auto; 但是一般设为 margin: 0 auto; 如果是行类或者是行内块元素, 就设置为 text-align: center; 居中.

1.6 外边距合并

注: 使用 margin 定义块元素的垂直外边距时, 可能会出现外边距的合并。

1. 嵌套元素外边距塌陷: 对于两个嵌套(父子关系)的块元素, 父元素有上外边距同时子元素也有上外边距, 在此时父子的外边距会相加, 父元素块会出现下塌现象.

解决方法: 1. 给父元素定义上边距框. 2. 给父元素定义上内边距; 3. 为父元素设置 overflow: hidden;

1.7 清除类外边距

如 `margin: 0; padding: 0;`

注: 行内标签尽量设置左右内外边距, 不要设置上下内外边距



(粗细) 边框样式 (border-style): solid (实线边框) / dashed (虚线边框) / dotted (点线边框)

1.2 边框简写

例: 1. border: px (边框粗细) solid (边框样式) red (边框颜色);

2. 边框位置单独设置, 例: border-top (位置): px solid red;

3. border-collapse; (边框合并)

1.3 内边距

Padding: 属于内边距

例: padding-left / top / right / bottom: px;

简写方法: padding: px; (相当于对文本距离边框的每个方位设置大小);

注: Padding 会影响盒子的大小。

1.4 外边距

margin: 属于用于设置外边距, 即控制盒子与盒子之间的距离

margin-left (左外边距) / right (右外边距) / top (上外边距) / bottom (下外边距): px;



继承性

子标签会继承父标签的某些样式、文本颜色和字号。

优先级

特点：当同一个元素指定多个选择器，就会有优先级产生。

选择器相同，则执行层叠性。选择器不同，则根据选择器权重执行。

$!important >$ 行内 style 选择器 $>$ id 选择器 $>$ 类选择器 $>$ 元素选择器

2. 权重叠加

复合选择器会有权重叠加， $ul, li \{ \}$ ， ul 和 li 选择器权重都是 0.0.0.1，叠加意思是每个位置各自相加，0.0.0.2，会比一般的基础选择器权重高。它们各位相加不会出现进位的现象：

CSS 盒子模型

特点：实质就是封装周围的 html 元素，它包括边框，外边距、内边距，以及内容。

(1) border: 边框。

1.1 border 可以设置元素的边框，边框的组成：宽度 border-width (定义边框粗细)



5. 背景复合写法

background: 背景颜色 (color) 图片路径 (url) 图片平铺 (repeat/no-repeat) 是否固定 (fixed/scroll) 位置 (left/center/right);

6. 背景色半透明

background: rgba(0, 0, 0, 0.5); 最后一个参数就是透明度改变, 取值 0-1, "0" 为全透明, "1" 为不透明, 盒子里面的内容不会发生改变。

7. CSS 有三种重要特征: 层叠性, 继承性, 优先级。层叠性: 1. 样式冲突, 遵循原则为就近原则。

例如:

<style>	在 body 的 div 会被 style 中的第
div {}	二个 div 选择器选中进行修改。反
<style>	之, 如果没有发生冲突, 就不会出
<body>	现此情况。
<div></div>	
</body>	



1. 背景图片

`background-image`: 属于描述元素的背景图像, 作用于一些常见 logo 或者装饰性图片或者超大背景图片;
范例: `background-image: none / url (图片路径);`

2. 背景平铺

`background-repeat: no-repeat` (不平铺)
`/ repeat` (平铺) `/ repeat-x` (向 X 轴平铺)
`/ repeat-y` (向 y 轴平铺);

3. 背景图片位置

`background-position`: 顺序可以改变图片在背景的位置

length: 百分数, 由浮点数组成.

注: 如果是方位名词: 如 `right`, `center` 等方位名词, 两个值前后顺序没有关系,

如果以 `x, y` (`px, px`) 为方位值, 它们位置不能更换.

4. 背景图像固定 (背景附着)

`background-attachment: scroll` (背景图像随对象一起滚动) `/ fixed` (背景固定)



行内元素:

`<a>`, ``, ``, ``, `` 等

1. 特点: 相邻的行级标签可以在同一行.
2. 高度直接设置无效.
3. 默认宽度就是它本身的宽度.
4. 行内元素只能放文本或其他的行内元素.
5. `<a>` 标签可以放块级元素, 最好转换块级模式.

行内块元素

``, `<input />`, `<td>` 同时具有块级元素和行级元素特点

1. 它可以设置宽度高度.

元素显示模式转换

作用: 一个模式的元素需要另外一种模式的特性

转换为块级元素: `display: block;`

转换为行内元素: `display: inline;`

转换为行内块元素: `display: inline-block;`

CSS 背景

特点: 可以设置颜色, 背景图片, 背景平铺。



CSS.

构成：选择器及多条申明

选择器：用于指定 CSS 的 HTML 标签，花括号内是对该对象设置具体样式。

(1)：属性和属性值以“键值对”的形式出现。

(2) 属性对指定的对象设置样式格式，例如字体、大小、颜色等。

(3) 多个键值对“之间用英文符”进行区分。

focus：伪类选择器

```
input: focus { background-color: yellow; }
```

CSS 的元素显示模式

1. HTML 分为块元素和行内元素。

常见的块元素有 `<h1>` ~ `<h6>`、`<p>`、`<div>`、``、``、`` 等

特点：

1. 自己独自站一行显示；

2. 它可以自己定义宽度、高度、外边距及内边距。

3. 宽度默认为是容器（父级宽度）的 100%；

4. 它可以允许在自己内部放行内标签或块级



10. display: line-block | float: left
transform (对象设置)

11. 格式文本标记

 定义粗体 <i></i> 定义斜体
 定义删除线 定义上标
 定义下标 定义加重语气
 无序标签.
 有序标签.
<dl><dt>名词</dt> <dd>名词</dd> </dl>
<blockquote></blockquote> 西文字符向右缩进
5字符, 中文缩进 2.5字符.

12. 计算机输出标记

<code></code> 定义计算机代码
<kbd></kbd> 定义键盘码

13. "子绝父相"

把父类中的位置设置为相对位置 position: (absolute)
子类位置设置为绝对 position: (relative)



14. 动画(animation)

animation: move 2s linear infinite)

@ keyframe move { from { } to { } }

transform: Scale(X, Y) translate (移动距离, 移动距离) rotate(旋转角度) (+ "deg") skew (拉伸距离)

15. 表单(表单域, 表单控件(表单元素), 提示信息)

<form> (表单域) [会把范围内的表单元素提交给服务器]

范例: <form form action="url地址" method="提交方式" name="表单域名称"> </form>

action: url地址, 用于处理表单数据的服务器url程序

method: get / post 用于设置表单的提交方式其取值为get或post;

name: 名称, 用于表单的名称, 以区分同一个网页的多个表单域

max length: 定义表格输入的个数.

1) 表单控件: input;

<input type="属性值"/>

type/属性值

password (密码, 用于创建密码框)

text (文本, 用于创建输入文本框)



☐ radio (单选按钮, 用于创建选择按钮) 只有加 name 属性才能多选, 相同属性, name 的属性值必须相同;

checkbox (多选按钮, 用于创建多选按钮)

submit (提交按钮, 把数据发送给后端 可以用 value 来修改: 提交'符')

reset (重置按钮, 可以用 value 来修改: 重置'符')

(2)

label 标签: `<label for = "属性"> 名词 </label>`
`<input type = "radio" value = "属性值" name = "属性值" id = "属性">`

(3)

select 标签: 用于多选项, 让用户能够选择, 同时节约空间

范例: `<select> <option> 名词 </option> </select>`

(4) textarea: 表单域

例如: `<textarea> 名词串 </textarea>`



(1) border-width border-style: solid / dashed / dotted border-color

(2) box-sizing: content-box [内容盒子, 向外扩展: width height <= 内容区] | border-box [怪异盒子: 向外扩展] width height <= border]

(3) padding (内容和边框的间距): 1值 | 2值... [padding-top...]

8.

border: dashed (虚线), 样式优先级 1 层叠
优先级: 内联 > 内部 > 外部 > 浏览器默认模式
权重: id > class 针对性越高, 优先级越强。

9. 样式格式:

(1) 选择器 { 样式名: 样式值; 样式名: 样式值; ... }

(2) .class 选择器 (类选择器)

(3) #id 选择器

(4) 伪类选择器 标记: nth-of-type (1) { ... }, :hover { }

(5) "*" 选择器, 可以设置 html 所有格式元素, 或取消 html 里面的相同元素



1. div, span 无语义标记: 块级元素, display: block.
2. 文本与段落格式化: h₁ ~ h₆
 (换行标记)
3. 嵌套 <关键字> <关键字> 文字 </关键字> </关键字>
- 4.

4.1: 属性名 = 属性值

4.2: 全局属性: style, script, id, class, meta, meta, title (注: 适用于所有标记)

5. 常用标记.

5.1: 文本与段落格式化: h₁ ~ h₆ br.

CSS -- 层叠 (继承: 子元素继承父元素的样式)

5.2: 样式的引用

(1) 行内样式 (内联)

(2) 内部样式

<style type="text/css"> body { background-color: #ccc color: red; } </style>

6. <link rel="icon" href="#"> icon: 标题栏图标标签.

7. border: 1px solid red [复合样式属性]

