

# CSS 基础

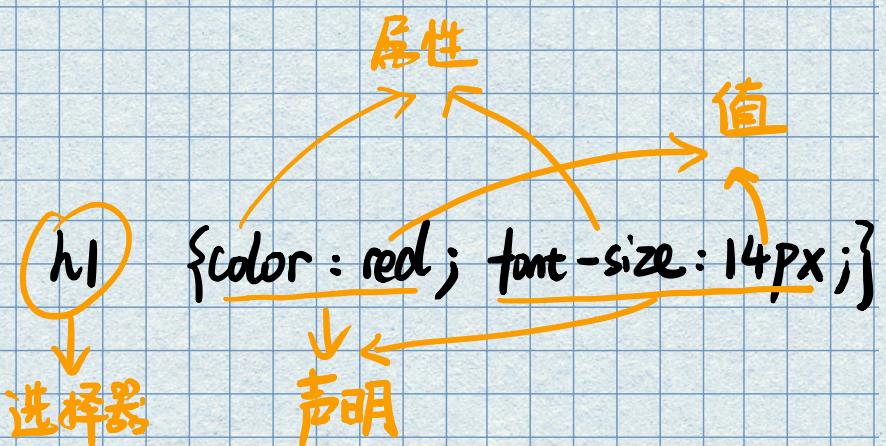
- CSS (Cascading Style Sheets)

## 层叠样式表

- 定义 HTML 内容在浏览器内的显示样式
- 内容与样式分离

## 基础语法

- CSS 规则 { 选择器 声明 }



- CSS 引用

在 <head> </head> 内

```
<style type = "text/css">
```

CSS 样式 ...

```
</style>
```

font-size: 字大小  
color  
font-family: 字体

相同样式： p, h1, h2, h3, h4 { ... }

/\* 注释 \*/

## 使用方法

⇒ 如何引用 CSS 样式

(行内样式 (内联样式)) 同时加载

在开始标签内添加 style 样式属性

<p style="color: red;"> ... </p>

(内部样式(嵌入样式)) 同时加载

<style type="text/css">

样式

</style>

! 放在 <head> 之间

(外部样式(外联样式)) 同时

把 CSS 样式代码写在一个独立文件中

扩展名：CSS 文件名.css

引入外部文件：

<link href="xx.css" rel="stylesheet" type="text/css"/>

放在 <head>

(导入式) 读完 html 文件之后加载

@ import "外部 CSS 样式";

<style type="text/css">

  @ import url( .css);

</style>

使用方法优先级

行内 > 内部 > 外部

link 与 内部 取决于位置 (就近原则)

# CSS选择器

## 标签选择器

以html标签作为选择器

`p color: ; } h1{ font-size: ; }`

使用同一标签的内容 → 同一样式

## 类选择器

- 为html添加class属性

`< h1 class="red" > ... </h1 >`

`< p > ... </p >`

`< p class="red" > ... </p >`

`⇒ .red { color: red; }`

- 对不同类型元素的同一个类设置不同的样式规则

`.red { }`

`h1.red { }`

- 同一元素设置多个类，空格隔开

`< p class="red size" > ... </p >`

文字修饰样式 `text-decoration: underline`  
属性：

## ID选择器

- 为html添加ID属性

`< p id="p1" > ... </p > ⇒ #p1 { ... }`

- id唯一 class和id值区分大小写

## 群组选择器

- 集体统一设置样式 (标签、class、id)  
`P, h1, h2, h3 { }`

## 全局选择器

`* { }`

## 后代选择器

`<P><em> CSS </em> ... </P>`

用空格隔开

`P em { ... } #P em { } P.class em { }`

## 伪类

- 定义特殊状态下的样式
- 无法用标签、id、class 及其它属性实现

⇒ 链接伪类

`:link` 未访问状态

`:visited` 已访问 `a:link { }`

`:hover` 鼠标悬停 (访问的鼠标经过某个元素时)

`:active` 激活 (-一个元素被激活)

书写顺序 `:Link > : Visited > : hover > : active`

## CSS 继承、层叠和优先级

## CSS 继承和层叠

⇒ CSS 继承

从父元素继承部分 CSS 属性

# CSS 优先级

id 选择器 > Class > 标签

同类样式多次引用、就近原则

## CSS 权值和优先级

⇒ 同一样式表中

1. 权值相同

就近原则

2. 权值不同

哪种 CSS 样式权值高，  
就用哪种

标签：1

- 统计不同选择器个数

类和伪类：10

- 每类个数 × 权值

ID：100

- 所有值相加 得出选择器权值

通配符：0

行内样式：1000

⇒ !important      `div {color: red !important;}`

调整样式规则的优先级

## CSS 命名规范

- 英文、数字
- -
- 以小写字母开头
- 命名形式：单字、连字符、下划线、驼峰
- 使用有意义命名

# CSS 文字样式

## 字体属性

⇒ font-family 字体属性

**font-family** : "1", "2", "3";

31号嵌套，外用双引，内用单引 style="‘宋体’;"

具体字体名或字体集

字体集：serif, sans-serif, Monospace, Cursive, Fantasy

```
p { font-family: "宋体", sans-serif; }
```

# 文字大小

`font-size` { 绝对单位 `xxs, xs, s, m, l ...`, `in, cm`  
相对单位 { `px 像素`  
`em %`

## 文字颜色

color : 名字 /  $rgb( , , )$ ; / 十六进制 #.....  
0 ~ 255  
0% ~ 100%

## 文字样式

## · 文字粗细

**文字粗细**   
font-weight : normal / bold / bolder / lighter / 100 ~ 900

## • 斜体

`font-style : normal / italic / oblique`

- 小型大写字母

font-variant : normal / small-caps

## font 简写

font : style variant weight size/ family  
/line-height

- 空格隔开

- 书写顺序

## 文本样式

### 文本对齐方式

- text-align (水平) 只对块级有效  
left, right, center, justify
- margin: 0 auto 使容器居中
- line-height (行高)  
长度值, 百分比
- vertical-align : 垂直 行内有效

baseline, sub, super, top, text-top  
middle, bottom, text-bottom  
长度, 百分比 上移 15px 下移 -15px

# 垂直居中

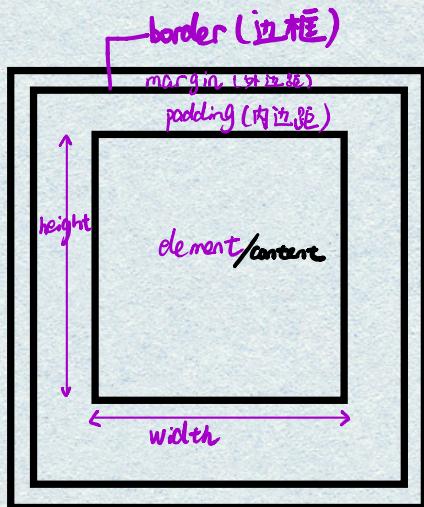
父元素: `display: table`

子元素: `display: table-cell`

## 其它样式属性

- `word-spacing` : 单词间距
- `letter-spacing` : 字母间距
- `text-transform` : 文本大小写
  - capitalize
  - uppercase
  - lowercase
  - none
- `text-decoration` : `underline` `overline` `line-through`
  - link
  - none
  - (闪烁)

## 网页中盒子模型



# 盒子模型

## 宽度

width : 长度 / 百分比 / auto

max-width

min-width

## 高度

height , max-height , min-height

哪些 HTML 元素可设置 width & height

- 块级元素
- 替换元素  
img, input, textarea
- 高和宽设置的是元素内容 (element)

## border 边框

border-width : thin, medium, thick, 长度

border-color : 颜色, transparent (透明)

border-style : none, solid, double, dotted (点状)

- 设置上下左右不同边框

border-[left, right, top, bottom] - {  
width  
color  
style}

- 边框缩写

border } 宽度、样式、颜色  
border-top }

## padding 填充

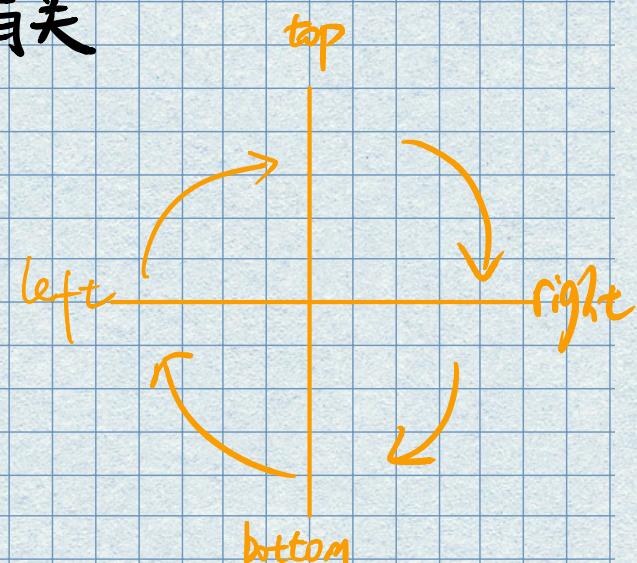
- 内边框

内容到边框之间

- padding - {  
top  
right  
bottom  
left : 长度值 / 百分比  
(不能为负)

- 盒子占的空间、还与 padding 有关

padding {  
① (上下左右)  
上 下 左 右  
① ② ③ ④  
上 上 左 右  
① ② ③ ④  
上 左 右 下  
① ② ③ ④

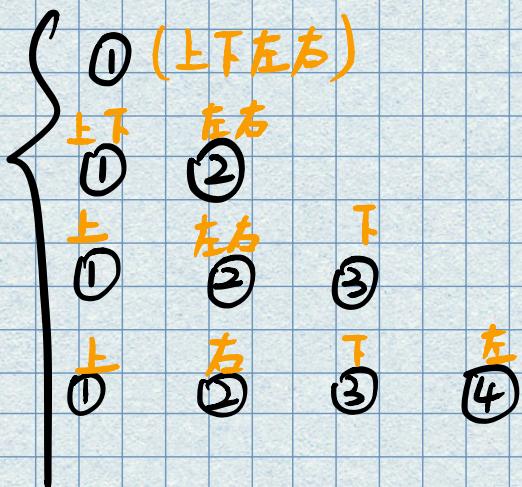


## margin 外边距

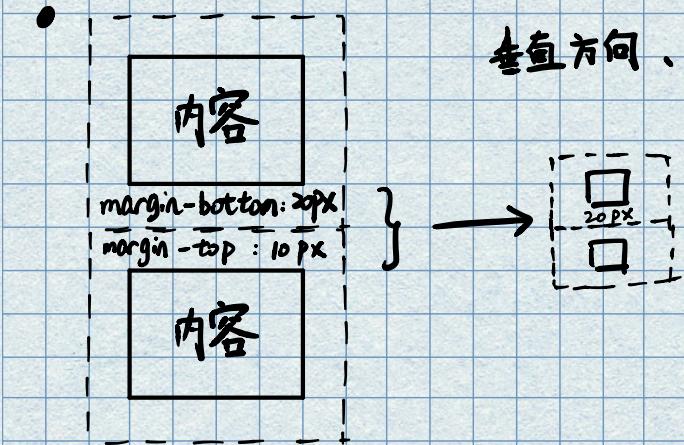
margin {  
top  
right  
bottom  
left : 长度、百分比、auto  
↓  
水平居中显示

- 可为负值

- margin



垂直方向、外边距合并，取决于大值



## display 属性

- inline (显示为行内元素) 没有换行符
- block (块级) 有换行符
- inline-block (行内元素, 具有块级元素特征)
- none

## 应用

- 为了不影响设置、初始化  
`margin: 0px, padding: 0px`
- 属性不继承  
可以设置 `div p {padding: inherit;}`

# CSS 背景

## 背景颜色

- background-color : 颜色 . transparent (透明)  
颜色 / RGB \ 六进制

背景色块 : content . padding . border  
不包含 margin

## 背景图片

- background-image : URL . none  
URL( )  
默认 : 背景图像位于左上角  
在水平和垂直方向重复  
包含 padding . border

## 背景图片重复

- background-repeat : repeat . no-repeat  
repeat-x repeat-y

## 背景图片显示方式

- background-attachment : scroll . fixed  
↓ 不移动  
默认  
随滚动条  
滚动

## 背景图片定位 设置 背景图片起始位置

- background-position

百分比、值 (x,y) 左上角(0,0)

top, right, bottom, left, center

## 缩写

- 空格、不分顺序

## CSS 列表

### 列表项标记

- list-style-type

#### 无序列表

none	
disc	●
circle	○
square	■

#### 有序列表

none	
decimal	1, 2, 3, ...
lower-roman	i, ii, ...
upper-roman	I, II, ...
lower-alpha	a, b, c
upper-alpha	A, B, C

- 使用图片

list-style-image : URL . none

### 列表项标记位置

- list-style-position : inside, outside

放置在文本内      黑默认  
环绕文本根据标记对齐      文本外  
文本不对齐

## 缩写

- 空格. 无顺序  
image 会覆盖 type

## float 属性

### CSS Float (浮动)

- 元素向左 → 右移动，不能上下
- 碰到另一个框，float 停止
- float 元素之后的元素将围绕它，之前的不受影响
- 浮动元素脱离标准流

### 基本语法

float: left 靠左浮动

float: right

float: none

### 文字环绕图片

### 清除浮动

clear: none . left . right . both

- 在浮动元素后使用一个空元素  
`<div class="clear"></div>`
- 给容器添加 `overflow: hidden`  
`* zoom: 1; /* 触发 hasLayout 兼容 IE 6.7 */`

- CSS3 “:after” 伪元素
- .clearfix : after {
   
content: ".";
   
display: block;
   
height: 0;
   
visibility: hidden;
   
clear: both; }
- .clearfix { zoom: 1; }

- 子级元素定义 height (只适合高度固定的布局)
- 父级元素一起浮动 (会产生新浮动)

## CSS定位 (Position)

Positioned Layout Module 定位布局模块

元素定位. 层叠相关

• box { position: relative; }

定位模型

static 自然模型

absolute 绝对定位

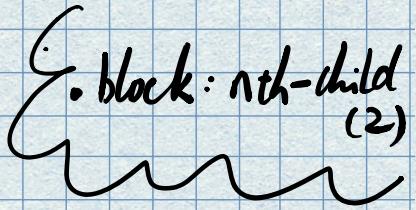
sticky 磁贴定位

relative 相对定位

fixed 固定定位

## position - static

静态、常规、自然定位



## position - relative

相对定位

- containing-block 可定位的祖先元素

position : relative ; top : 10px ; left : 10px ; →

## position - absolute

脱离常规流、绝对定位

## position - fixed

固定定位

## position - sticky

磁贴、粘性、吸附

# CSS 网页布局

## 圣杯布局

- 三列布局
- 中间栏优先

定宽	自适应	定宽
----	-----	----

展示渲染

- 允许任意列高度最高
- 用最简单的CSS、最少的HACK语句

## 双飞翼布局