

1.4 CSS

1.4.1 CSS简介

1.4.1.1 CSS引用

引入 CSS 的三种方式：

1.外部样式表

```
<head>
  <link rel = "stylesheet" href = "a.css">
</head>
```

2.内部样式表

```
<head>
  <style>
    body {background-color: red;}
    p {margin: auto;}
  </style>
</head>
```

3.内嵌样式

```
<p style = "background-color: red;margin: auto;>
  this is a paragraph!
</p>
```

1.4.1.2 语法

1. 模版：

```
选择器 {
  属性申明1
  属性申明2
  .
  .
  .
}
```

2. 浏览器私有属性：

- **chrome , safari**
-webkit-
- **firefox**
-moz-

- IE

- ms-

- opera

- O-

3. 属性值语法

margin: [<length> | <percentage> | auto] {1,4}

一般是由 **基本元素 组合符号 数量符号** 组成

- 基本元素：

- 关键字：(auto, solid, bold...)
- 类型：
 - 基本类型 (<length>, <percentage>, <color>...)
 - 其他类型 (<'padding-width'>, <color-stop>...)
- 符号：(/ ,)
- inherit, initial

- 组合符号：

1. 空格

必须出现，要按顺序出现，例：

```
<'font-size'> <'font-family'>
```

则合法值如下：

```
- 12px arial
```

不合法值如下：

```
- 2em
```

```
- arial 14px
```

2. &&

必须出现，顺序无所谓，例：

```
<length>&&<color>
```

则合法值如下：

```
- green 2px - 1em blue
```

不合法值如下：

```
- blue
```

3. ||

至少出现一个，顺序无所谓，例：

```
underline|overline|line-through|blink
```

则合法值如下：

```
- underline
```

```
- overline underline
```

4. |

只能出现一个，例：

```
<color>|transparent
```

则合法值如下：

```
- orange - transparent
```

不合法值如下：

```
- blue transparent
```

5. []

分组作用，括号内可当做一个整体，例：

```
bold [ thin|<length> ]
```

则合法值如下：

- bold thin - bold 2em

◦ 数量符号：

1. 无

如果只是

<length>

则合法值如下：

- 1px

- 10em

不合法值如下：

- 1px 2px

2. +

可以出现一次或者多次，例：

<color-stop>[,<color-stop>]+

则合法值如下：

- #fff, red

- blue, green 50%, gray 不合法值如下： - red

3. ?

表示可以出现也可以不出现，例：

inset?&& <color>

则合法值如下：

- inset blue

- pink

4. {}

表示元素出现次数，例：

<length>{2,4}

则合法值如下：

- 1px, 2px

- 1px, 2px, 3px, 9px

不合法值如下：

- 1px

5. *

可以出现零次、一次或者多次，例：

<time> [,<time>]*

则合法值如下：

- 1s

- 1s, 4ms

6. #

可以出现一次或者多次,中间用**逗号**隔开，例：

<time> #

则合法值如下：

- 2s, 4s

不合法值如下：

- 1ms 3ms

4. @规则语法

模式：

@ 标识符 xxx;

@ 标识符 xxx {};

常用的有：

- @media 用做响应式布局，媒体查询条件
- @keyframes 主要用来描述CSS动画的中间步骤
- @font-face 主要用来引入外部字体

1.4.2 CSS选择器

1.4.2.1 简单选择器

1. 简单选择器

- 标签选择器 tag
- 类选择器 .
- ID选择器 #
- 通配符选择器 *
- 属性选择器1 [att]
- 属性选择器2 [att=val]
 - 选中值为val的属性，我们的id选择器其实就是一种特殊的属性选择器2：
 - `#nav{} == [id=nav]{}`
- 属性选择器3 [att~=val]
 - 选中包含val值的所有属性，我们的class选择器就是一种特殊的属性选择器3：
 - `.nav{} == [class~=nav]{}`
- 属性选择器4 [att|=val]
 - 选中值为val或val加中划线开头，用的不多一般用在选择en和en-**的时候
- 属性选择器5 [att^=val]
 - 以val开头的
- 属性选择器6 [att\$=val]
 - 以val结尾的
- 属性选择器7 [att*=val]
 - 包含val的
- 伪类选择器
 - :link
 - :visited
 - :hover
 - :active
 - :enabled
 - :disabled
 - :checked
 - :first-child
 - :last-child
 - :nth-child(even)
 - :nth-child(odd)
 - :nth-child(3n+1)
 - :nth-last-child(4n+1)
 - :only-child
 - :first-of-type

- :last-of-type
- :nth-of-type(3n)
- :nth-last-of-type(2n+2)
- :only-of-type
- :empty
- :root
- :not()
- :target
- :lang()
- 组合简单选择器
- 如：
 - img[src\$=jpg]{}
 - #banner:hover{}

1.4.2.2 其他选择器

1. 简单选择器（上节已说过）
 2. 伪元素选择器
- 前面是两个冒号（CSS3前只有一个冒号）

- ::first-letter{}
- ::first-line{}
- ::before{content:"abc"}
- ::after{content:"xyz"}
- ::selection 选中文字后的效果

3. 组合选择器

- 后代选择器 空格
- 子选择器 大于号
- 兄弟选择器 **都是向后看**
 - 加号 相邻兄弟选择器
 - 波浪号 通用兄弟选择器

4. 选择器分组 用逗号把选择器分开即可

1.4.2.3 继承、优先级、层叠

1. 部分子元素会默认继承父元素：
继承属性的：

- color
- font
- text-align
- list-style
- ...

非继承属性的：

- background
- border
- position
- ...

2. CSS优先级

- 计算方法：
 - a = 行内样式
 - b = ID选择器的数量
 - c = 类、伪类和属性选择器的数量
 - d = 标签选择器和伪元素选择器的数量
- `value = a*1000 + b*100 + c*10 + d` value较大的优先级高
- 如果优先级相等，后面覆盖前面的

3. CSS层叠

- 优先级
- 后面覆盖前面
- 不同的属性会合并

4. CSS改变优先级

- 改变先后顺序
- 提升选择器优先级
- !important

1.4.3 CSS文本

1.4.3.1 字体

1. **font-size:** `font-size: <length> | <percentage> | <absolute-size> | <relative-size>`

- | 表示的是只出现一个
- 例子：

文字大小：font-size:12px;

文字大小：font-size:16px;

文字大小：font-size:2em;

文字大小：font-size:200%;

2. **font-family:** `font-family: [<family-name> | <generic-family>] #`
 - `<generic-family>` = serif|sans-serif|cursive|fantasy|monospace

字体系列：font-family:arial;

字体系列：font-family:arial,Verdana,sans-serif;

字体系列：font-family:Verdana, "microsoft yahei";

字体系列：font-family:“宋体”， serif;

3. **font-weight:** `font-weight: normal | bold | bolder | lighter | 100 | 200 | 300 | ... | 900` 字体粗细：font-weight:normal;

字体粗细：font-weight: bold;

4. **font-style:** `font-style: normal | italic | oblique`

- 后两者区别在于 二是基于字体有斜体的前提下，三是字体没有，强行倾斜

字体风格：font-style:normal;

字体风格：font-style: italic;

5. **line-height**： `line-height: normal | <number> | <length> | <percentage>`

- 浏览器一般默认值为1.14左右 (normal)

行高 : line-height:40px;

行高 : line-height:3em;

行高 : line-height:300%;

行高 : line-height:3;

另一个更直接的例子 :

行高 : line-height:300%;

行高 : line-height:3em;

行高 : line-height:300%;

行高 : line-height:3em;

总结为 :

- line-height中number为直接继承number
- 而percentage需要先计算成length再继承

6. 总结 :

font: [[<font-style>||<font-variant>||<font-weight>||<font-stretch>]?<font-size>[/<line-height>]?<font-family>]]caption|icon|menu|message-box|small-caption|status-bar

如 : font:30px/2 "Consolas",monospace;

如 : font:italic bold 20px/1.5 arial,serif; 这就错了 : font:100px;

至少要有font-size和font-family

7. color :

- color:red;
- color:#f00;
- color:rgb(255,0,0);
- color:rgba(255,0,0,1);a为透明情况由0到1.