

css基础知识

css基础知识

1. 认识css
 - 概念
 - 写在哪儿
 - 如何写
2. 选择器初级
 - id与class命名
 - 选择器分类（初级版）
 - 选择器权重
3. 基础知识介绍

1. 认识css

概念

- CSS(英文全称: Cascading Style Sheets), 层叠样式表。它是网页的装饰者, 用来修饰各标签排版(大小、边距、背景、位置等)、改变字体的样式(字体大小、字体颜色、对齐方式等)、设置图片(宽高、位置等)等。
- CSS以HTML为基础, 提供了丰富的功能, 如字体、颜色、背景的控制及整体排版等, 而且还可以针对不同的浏览器设置不同的样式。
- **CSS可以帮助调整排版布局的展现 美化标签和内容 提高版面的信息密度 加强信息的冲击力和直观性**

写在哪儿

css代码分为**内部样式**、**外部样式**、**行内样式**三种写法。

- 内部样式

写在HTML文档的 `style` 标签中, `style`通常放置于头部:

```
<head>
  <style>
    /*这里写css代码*/
  </style>
</head>
```

`style`标签可以加上 `type="text/css"` 这条属性, 用于告诉浏览器这里是什么格式的代码, 但是在HTML5规范中可以不用写。

- 外部样式

单独创建 `.css` 后缀的文件, 然后通过`link`标签引入, `link`通常放置于头部:

```
<head>
  <link href="CSS文件的路径" rel="stylesheet">
</head>
```

- 行内样式/内联样式

直接写在标签属性中：

```
<body>
  <p style="css代码">段落文字</p>
</body>
```

- 三种方式比较

样式表	优点	缺点	使用情况	使用场景
行内样式表	书写方便，权重高	没有实现结构与样式分离，不便于维护，不可以重复利用	极少	某个标签需要单独的样式设置时使用
内部样式表	部分结构与样式分离，较便于维护	没有彻底实现结构与样式分离，不可以重复利用	一般	css代码量不多，且和当前页面联系紧密不需要复用时使用
外部样式表	完全实现结构与样式分离，可重复利用	如果代码量较少情况下，引入法更麻烦	最多，推荐	css代码量大时，或者需要重复利用时使用

如何写

```
/*选择器{属性:值;属性:值;}*/
p{
  color: red;
  font-size: 20px;
}
/*
规范：
  值以分号结束
  推荐换行书写，可读性更强
  推荐 值 与 : 号之间加空格，可读性更强
*/
```

2. 选择器初级

id与class命名

- id命名唯一，单个标签不允许拥有多个id名，同页面中不存出现重复的id名（id重名会影响js代码获取标签）
- class命名不唯一，单个标签可以拥有多个class名，不同标签允许使用相同class名
- 命名规范

- 允许 大写字母 小写字母 数字-，必须以字母开头，单词一般小写；
- 见名知意，不允许单个字母做名字，各命名词汇英文参考：[命名词汇](#)；
- 单词之间使用-号连词或者驼峰连词，具体看公司整体规范；
- 单词不要简写过渡，可以不影响理解的前提下简写。

选择器分类（初级版）

- id选择器
eg: `#wrap`，选中id名为wrap的标签。
- class选择器
eg: `.nav`，选中所有class名为nav的标签。
- 标签选择器
eg: `p`，选中所有p标签。
- 后代选择器
eg: `#wrap .nav`，选中#wrap标签中所有拥有nav类名的**后代**标签。
- 通配选择符
eg: `*`，选中页面中所有元素

选择器权重

如果 同一个标签 被 多个选择器 赋予了 相同 的属性且值不一样时，会按照选择器的优先级来决定以哪个样式为准，比如：

```
/*css代码*/
.info{color: green;}
p{color: red;}
```

```
<!--HTML代码-->
<p class="info">段落文字</p>
```

此时 `.info` 和 `p` 都选中了这个p标签，并且两者规定了不同的字体颜色，那么最终显示出的颜色就要根据选择器的优先级来决定。

- !important > 行内 > id > class > 标签 > 通配符 | 兄弟 | 相邻兄弟 | 子代 > 继承
- 先比较级别，级别一样比较各级别选择器出现的次数
- 当两个选择器权重一样时，以最后出现的为准

3. 基础知识介绍

- 常用的颜色表示法
 1. 直接使用颜色英文名（缺点颜色不好描述出来）
 - 比如red, orange yellow 等等.....
 2. 十六进制的rgb值
 - 语法。#红色绿色蓝色
 - 颜色浓度通过 00 - ff 来设置
 - 如果颜色两位两位重复可以简写
 - #aabbcc --- #abc
- px 像素的简单介绍

