



# CSS基础

基础选择器+字体文本相关样式



黑马程序员  
[www.itheima.com](http://www.itheima.com)

传智教育旗下  
高端IT教育品牌

# 学习目标

Learning Objectives

- ◆ 能够知道 CSS 的引用方式及区别
- ◆ 能够使用基础选择器在 HTML 中选择元素
- ◆ 能够使用字体和文本相关样式修改外观样式
- ◆ 能够使用 CSS 让页面中元素水平居中
- ◆ 能够使用 Chrome 调试工具修改和检查样式



# 目录

Contents

- ◆ 基础认知
- ◆ 基础选择器
- ◆ 字体和文本样式
- ◆ Chrome调试工具
- ◆ 综合案例



# 目录

Contents

- ◆ 基础认知
- ◆ 基础选择器
- ◆ 字体和文本样式
- ◆ Chrome调试工具
- ◆ 综合案例

# 一、基础认知

**目标：理解 CSS 的作用，了解 CSS 语法规则，知道 CSS 的引入方式及其区别**

学习路径：

1. CSS初识
2. CSS引入方式

# 一、基础认知

## 1.1 CSS的介绍

- CSS: 层叠样式表 (Cascading style sheets)
- CSS作用是什么? ?
  - 给页面中的HTML标签设置样式



结构: HTML (决定了身体)



表现: CSS (决定了样式美观)



行为: JavaScript (决定了交互的动态效果)

# 一、基础认知

## 1.2 CSS 语法规则

- 写在哪里?
  - css写在style标签中，style标签一般写在head标签里面，title标签下面
- 怎么写?

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    选择器
    p {
      css属性
      color: red;
    }
  </style>
</head>
<body>
  <p>hello world 你好世界</p>
</body>
</html>
```

属性名

属性值

# 一、基础认知

## 1.3 CSS 初体验

### ➤ 常见属性:

css常见属性	作用
color	文字颜色
font-size	字体大小
background-color	背景颜色
width	宽度
height	高度

### ➤ 注意点:

1. CSS 标点符号都是英文状态下的
2. 每一个样式键值对写完之后, 最后需要写分号

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    p {
      color: red;
      font-size: 30px;
      background-color: pink;
      width: 300px;
      height: 300px;
    }
  </style>
</head>
<body>
  <p>hello world 你好世界</p>
</body>
</html>
```

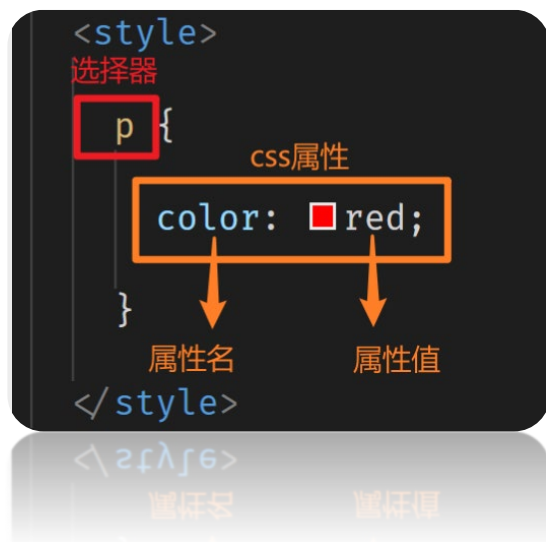
```
<\μfωj>
<\ρoqλ>
| <b>μeяfo mo1яq 你&井站<b>
<\ρoqλ>
<\μeяq>
```



# 一、基础认知

## 1.4 CSS 初识-小结

- CSS的中文名称叫做什么呢?
  - 层叠样式表
- CSS的作用是什么呢?
  - 给页面中的HTML标签设置样式
- CSS的语法规则是什么?



# 一、基础认知

**目标：理解 CSS 的作用，了解 CSS 语法规则，知道 CSS 的引入方式及其区别**

学习路径：

1. CSS初识
2. CSS引入方式

# 一、基础认知

## 2.1 CSS引入方式

- 内嵌式：CSS 写在style标签中
  - 提示：style标签虽然可以写在页面任意位置，但是通常约定写在 head 标签中
- 外联式：CSS 写在一个单独的.css文件中
  - 提示：需要通过link标签在网页中引入
- 行内式：CSS 写在标签的style属性中
  - 提示：基础班不推荐使用，之后会配合js使用

## 2.2 CSS引入方式-小结

- CSS常见三种引入方式有哪些？
  - 内嵌式
  - 外联式
  - 行内式
- CSS常见三种引入方式的特点区别有哪些（书写位置、作用范围、使用场景）？

引入方式	书写位置	作用范围	使用场景
内嵌式	CSS 写在style标签中	当前页面	小案例
外联式	CSS 写在单独的css文件中，通过link标签引入	多个页面	项目中
行内式	CSS 写在标签的style属性中	当前标签	配合js使用



# 目录

Contents

- ◆ 基础认知
- ◆ 基础选择器
- ◆ 字体和文本样式
- ◆ Chrome调试工具
- ◆ 综合案例

## 二、基础选择器

目标：理解**选择器的作用**，能够使用基础选择器在 HTML 中**选择元素**

学习路径：

1. **标签选择器**
2. 类选择器
3. id选择器
4. 通配符选择器

## 二、基础选择器

### 1.1 选择器的作用

- 选择器的作用：
  - 选择页面中对应的标签（找她），方便后续设置样式（改她）

### 1.2 标签选择器

- 结构： **标签名** { css属性名: 属性值; }
- 作用：通过标签名，找到页面中所有这类标签，设置样式
- 注意点：
  1. 标签选择器选择的是一类标签，而不是单独某一个
  2. 标签选择器无论嵌套关系有多深，都能找到对应的标签

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Document</title>
  <style>
    p {
      color: red;
    }
  </style>
</head>
<body>
  <p>标签选择器</p>
</body>
</html>
```

```
<\p\w\>
<\p\o\q\>
  <b>标签选择器</b>
<p\o\q\>
```



## 二、基础选择器

目标：理解**选择器的作用**，能够使用基础选择器在 HTML 中**选择元素**

学习路径：

1. 标签选择器
2. **类选择器**
3. id选择器
4. 通配符选择器

### 2. 类选择器

- 结构: **.类名** { css属性名: 属性值; }
- 作用: 通过类名, 找到页面中所有带有这个类名的标签, 设置样式
- 注意点:
  1. 所有标签上都有class属性, class属性的属性值称为**类名** (类似于名字)
  2. 类名可以由数字、字母、下划线、中划线组成, 但不能以数字或者中划线开头
  3. 一个标签可以同时有多个类名, 类名之间以空格隔开
  4. 类名可以重复, 一个类选择器可以同时选中多个标签

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <style>
    .one {
      color: red;
    }
  </style>
</head>
<body>
  <div class="one">类选择器</div>
</body>
</html>
```

```
<\μfwj>
<\poql>
  <q1Λ cJ922="ou6">类选择器<q1Λ>
<poql>
```

## 二、基础选择器

目标：理解**选择器的作用**，能够使用基础选择器在 HTML 中**选择元素**

学习路径：

1. 标签选择器
2. 类选择器
3. **id选择器**
4. 通配符选择器

### 3. id选择器

- 结构: **#id属性值** { css属性名: 属性值; }
- 作用: 通过id属性值, 找到页面中带有这个id属性值的标签, 设置样式
- 注意点:
  1. 所有标签上都有id属性
  2. id属性值类似于身份证号码, 在一个页面中是唯一的, 不可重复的!
  3. 一个标签上只能有一个id属性值
  4. 一个id选择器只能选中一个标签

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Document</title>
  <style>
    #two {
      color: blue;
    }
  </style>
</head>
<body>
  <div id="two">id选择器</div>
</body>
</html>
```

```
<\μfwj>
<\poql>
  <q\Λ \fq="fmo">\fq选择器<\q\Λ>
</poql>
```

### 补充：类与id的区别

- class类名与id属性值的区别
  - class类名相当于姓名，可以重复，一个标签可以同时有多个class类名
  - id属性值相当于身份证号码，不可重复，一个标签只能有一个id属性值
- 类选择器与id选择器的区别
  - 类选择器以 `.` 开头
  - id选择器以 `#` 开头
- 实际开发的情况
  - 类选择器用的最多
  - id一般配合js使用，除非特殊情况，否则不要使用id设置样式
  - 实际开发中会遇到**冗余代码的抽取**（可以将一些公共的代码抽取到一个公共的类中去）

## 二、基础选择器

目标：理解**选择器的作用**，能够使用基础选择器在 HTML 中**选择元素**

学习路径：

1. 标签选择器
2. 类选择器
3. id选择器
4. **通配符选择器**

### 4. 通配符选择器

- 结构: `*{css属性名: 属性值; }`
- 作用: 找到页面中所有的标签, 设置样式
- 注意点:
  1. 开发中使用极少, 只会在极特殊情况下才会用到
  2. 在基础班小页面中可能会用于去除标签默认的margin和padding (后续讲解)

```
* {  
  margin: 0;  
  padding: 0;  
}
```

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">  
  <meta name="viewport" content="width=device-width, initial-scale=1.0">  
  <title>Document</title>  
  <style>  
    * {  
      color: yellow;  
    }  
  </style>  
</head>  
<body>  
  <p>我是p标签</p>  
  <h1>我是h1标签</h1>  
  <div>我是div标签</div>  
  <span>我是span标签</span>  
</body>  
</html>
```

```
<\μfωγ>  
<\poql>  
<sbau>我是sbau标签</sbau>  
<qtl>我是qtl标签</qtl>  
<pt>我是pt标签</pt>
```

## 二、基础选择器

目标：理解**选择器的作用**，能够使用基础选择器在 HTML 中**选择元素**

1. 选择器作用：选中页面中对应的标签（找她），方便后续设置样式（改她）
2. 标签选择器：**标签名** { css属性名：属性值； }
3. 类选择器：**.类名** { css属性名：属性值； }
4. id选择器：**#id属性值** { css属性名：属性值； }
5. 通配符选择器：**\*** { css属性名：属性值； }





# 目录

Contents

- ◆ 基础认知
- ◆ 基础选择器
- ◆ 字体和文本样式
- ◆ Chrome调试工具
- ◆ 综合案例

## 三、字体和文本样式

目标：能够使用**字体和文本相关样式**修改元素外观样式

学习路径：

### 1. 字体样式

1. 字体大小：font-size
2. 字体粗细：font-weight
3. 字体样式：font-style
4. 字体类型：font-family
5. 字体类型：font属性连写

### 2. 文本样式

### 3. line-height行高

### 1.1 字体大小

- 属性名: **font-size**
- 取值: **数字 + px**
- 注意点:
  - 谷歌浏览器默认文字大小是16px
  - 单位需要设置, 否则无效



### 1.2 字体粗细

➤ 属性名: **font-weight**

➤ 取值:

- 关键字:

正常	normal
加粗	bold

- 纯数字: 100~900的整百数:

正常	400
加粗	700

➤ 注意点:

- 不是所有字体都提供了九种粗细, 因此部分取值页面中无变化
- 实际开发中以: 正常、加粗两种取值使用最多。



### 1.3 字体样式（是否倾斜）

- 属性名：font-style
- 取值：
  - 正常（默认值）：normal
  - 倾斜：italic

倾斜的文字

## 三、字体和文本样式

### 1.4 常见字体系列（了解）

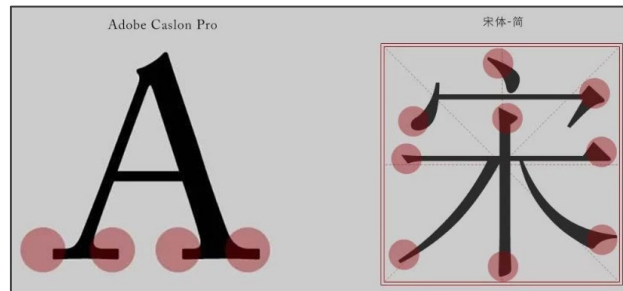
#### ➤ 无衬线字体（sans-serif）

1. 特点：文字笔画粗细均匀，并且首尾无装饰
2. 场景：网页中大多采用无衬线字体
3. 常见该系列字体：黑体、Arial



#### ➤ 衬线字体（serif）

1. 特点：文字笔画粗细不均，并且首尾有笔锋装饰
2. 场景：报刊书籍中应用广泛
3. 常见该系列字体：宋体、Times New Roman



#### ➤ 等宽字体（monospace）

1. 特点：每个字母或文字的宽度相等
2. 场景：一般用于程序代码编写，有利于代码的阅读和编写
3. 常见该系列字体：Consolas、Courier New



### 1.5 字体系列 font-family

- 属性名: **font-family**
- 常见取值: 具体字体1,具体字体2,具体字体3,具体字体4,...,字体系列
  - 具体字体: "Microsoft YaHei"、微软雅黑、黑体、宋体、楷体等.....
  - 字体系列: sans-serif、serif、monospace等.....
- 渲染规则:
  1. 从左往右按照顺序查找, 如果电脑中未安装该字体, 则显示下一个字体
  2. 如果都不支持, 此时会根据操作系统, 显示最后字体系列的默认字体
- 注意点:
  1. 如果字体名称中存在多个单词, 推荐使用引号包裹
  2. 最后一项**字体系列不需要引号包裹**
  3. 网页开发时, 尽量使用系统常见自带字体, 保证不同用户浏览网页都可以正确显示

微软雅黑  
黑体  
宋体  
楷体

系统	默认字体
Windows	微软雅黑
macOS	苹方

### 1.6 样式的层叠问题

- 问题：
  - 给同一个标签设置了相同的样式，此时浏览器会如何渲染呢？
- 结果：
  - 如果给同一个标签设置了相同的属性，此时样式会层叠（覆盖），写在最下面的会生效
- TIP：
  - CSS（Cascading style sheets）**层叠样式表**
  - 所谓的层叠即叠加的意思，表示样式可以一层一层的层叠覆盖

```
<style>
  p {
    color: red;
    color: blue;
  }
</style>
</head>
<body>
  <p>我是一个p标签</p>
</body>
</html>
```



### 1.7 字体font相关属性的连写

- 属性名: font
- 取值:
  - font : style weight size family;
- 顺序要求:
  - swsf (稍微舒服)
- 省略要求:
  - 只能省略前两个, 如果省略了相当于设置了默认值
- 注意点: 如果需要同时设置单独和连写形式
  - 要么把单独的样式写在连写的下面
  - 要么把单独的样式写在连写的里面

### 1.8 字体相关属性总结

- 字体大小: font-size
  - 数字+px
- 字体粗细: font-weight
  - 正常: normal 或 400
  - 加粗: bold 或 700
- 字体样式: font-style
  - 正常: normal
  - 倾斜: italic
- 字体系列: font-family
  - 具体字体1,具体字体2,具体字体3,具体字体4,...,字体系列
- 字体连写: font
  - font : style weight size family;

## 三、字体和文本样式

目标：能够使用**字体和文本相关样式**修改元素外观样式

学习路径：

1. 字体样式
2. **文本样式**
  1. 文本缩进：text-indent
  2. 文本水平对齐方式：text-align
  3. 文本修饰：text-decoration
3. line-height行高

### 2.1 文本缩进

- 属性名: `text-indent`
- 取值:
  - 数字+px
  - 数字+em (推荐: 1em = 当前标签的font-size的大小)

最初的现代计算机并不容易操作。当时的编程实际上是手工将电线连接成一排排电路来实现的。后来出现了机器语言和汇编语言，允许用户用代码为计算机编程，但这两种语言都需要对计算机的架构有深入的了解，使得许多科学家难以掌握。

20世纪50年代，随着符号语言的发展，特别是由约翰·巴克斯及其团队在加州圣何塞的IBM开发的“公式翻译”语言Fortran，这种情况发生了变化。利用Fortran，用户可以用人类可读的指令来编程，例如 $x = 3 + 5$ 。然后由编译器将这些指令转换成快速、高效的机器代码。

### 2.2 文本水平对齐方式

➤ 属性名: `text-align`

➤ 取值:

属性值	效果
left	左对齐
center	居中对齐
right	右对齐

➤ 注意点:

- 如果需要让文本水平居中, `text-align`属性给文本所在标签 (文本的父元素) 设置



### 2.3 文本修饰

➤ 属性名: `text-decoration`

➤ 取值:

属性值	效果
undefined	下划线 (常用)
line-through	删除线 (不常用)
overline	上划线 (几乎不用)
none	无装饰线 (常用)

➤ 注意点:

- 开发中会使用 `text-decoration : none` ; 清除a标签默认的下划线

- 最新研究: 地球的“死亡速度”远超出科学家预期
- 我们已处在第6次物种大灭绝边缘?
- 为什么早期人类要离开非洲大陆?
- 法国启动量子技术国家战略 计划5年内投资18亿欧元
- “已读不回”和“无人点赞”真的让我很焦虑
- 全球疫苗开打, 谁将率先获得群体免疫?
- 能逃避抗体攻击的新冠变异株出现, 新冠疫苗还有效吗?

### 2.4 文本样式总结

#### ➤ 文本样式

样式	属性名	常见属性值
文本缩进	text-indent	数字+px / 数字+em
文本水平对齐方式	text-align	left / center / right
文本修饰	text-decoration	underline / none

### 2.5 水平居中方法总结 `text-align : center`

- `text-align : center` 能让那些元素水平居中?
  1. 文本
  2. `span`标签、`a`标签
  3. `input`标签、`img`标签
- 注意点:
  1. 如果需要对以上元素水平居中, `text-align : center` 需要给以上元素的 **父元素** 设置





### 2.6 水平居中方法总结 `margin : 0 auto`

- 如果需要对div、p、h（大盒子）水平居中？
  - 可以通过`margin : 0 auto ;`实现
- 注意点：
  1. 如果需要对 div、p、h（大盒子） 水平居中，直接给 当前元素本身 设置即可
  2. `margin: 0 auto` 一般针对于固定宽度的盒子，如果大盒子没有设置宽度，此时会默认占满父元素的宽度

## 三、字体和文本样式

目标：能够使用**字体和文本相关样式**修改元素外观样式

学习路径：

1. 字体样式
2. 文本样式
3. **line-height**行高

### 3.1 行高

- 作用：控制一行的上下行间距
- 属性名：line-height
- 取值：
  - 数字+px
  - 倍数（当前标签font-size的倍数）
- 应用：
  1. 网页精准布局时，会设置 line-height : 1 可以取消上下间距
  2. 让单行文本垂直居中可以设置 line-height : 文字父元素高度
- 行高与font连写的注意点：
  - 如果同时设置了行高和font连写，注意覆盖问题
  - font : style weight size/line-height family ;

在感知智能上，目前机器智能正在接近和逼近人类，已经做得很不错。比方说，现在已经被广泛应用的人脸识别，在火车站等公共场所中都已开始广泛应用；再有语音识别，这方面也做得非常好，可以把语音方便地转成文字；还有光学字符识别，对于报销单等，可以直接提取出一个结构化的文档来。可以说，在感知智能这个层次上面，基于深度学习的一些技术在某种意义上已经能够跟人类媲美了。





# 目录

Contents

- ◆ 基础认知
- ◆ 基础选择器
- ◆ 字体和文本样式
- ◆ Chrome调试工具
- ◆ 综合案例

## 四、Chrome调试工具

**目标：**能够认识Chrome调试工具的基础操作，能够使用 Chrome 调试工具修改和调试样式

**学习路径：**

1. 打开方式
2. 选择元素
3. 控制样式
4. 特殊情况

### 1.1 Chrome调试工具操作

#### ➤ 打开方式

- ① 右击 → 检查 ② 看哪里

#### ➤ 选择元素

- 两种常见方法

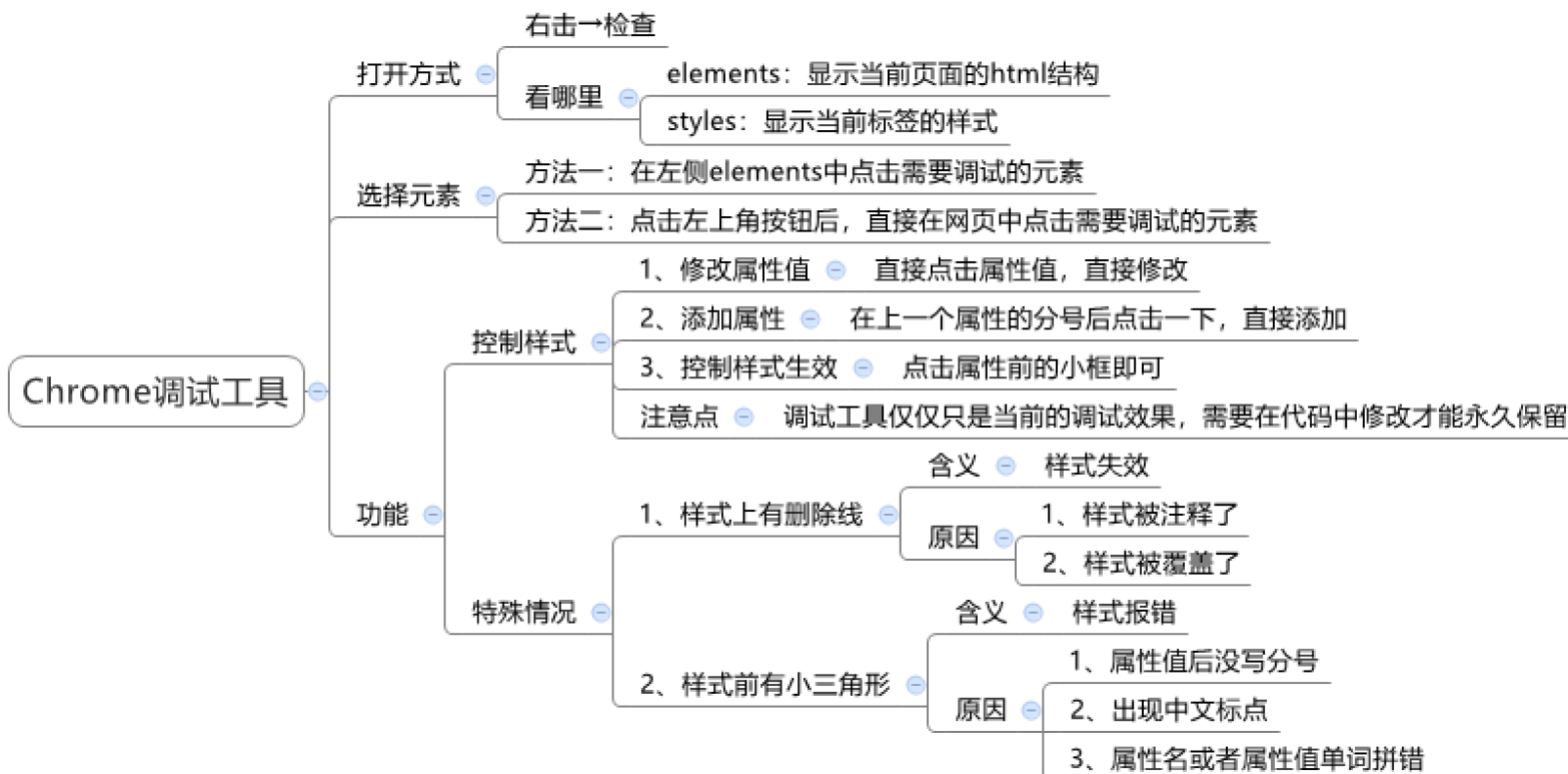
#### ➤ 控制样式

- ① 修改属性值 ② 添加属性 ③ 控制样式生效

#### ➤ 特殊情况

- ① 出现删除线 ② 出现小三角形

### 1.2 Chrome调试工具小结



### 拓展 颜色常见取值 (了解)

➤ 属性名：

- 如：文字颜色：color
- 如：背景颜色：background-color

➤ 属性值：

颜色表示方式	表示含义	属性值
关键词	预定义的颜色名	red、green、blue、yellow.....
rgb表示法	红绿蓝三原色。每项取值范围：0~255	rgb(0,0,0)、rgb(255,255,255)、rgb(255,0,0).....
rgba表示法	红绿蓝三原色+a表示透明度，取值范围是0~1	rgba(255,255,255,0.5)、rgba(255,0,0,0.3).....
十六进制表示法	#开头，将数字转换成十六进制表示	#000000、#ff0000、#e92322，简写：#000、#f00



### 拓展 颜色常见取值 (了解)

- 取值类型①：关键词
- 常见颜色取值：
  - red: 红色
  - green: 绿色
  - blue: 蓝色
  - yellow: 黄色
  - orange: 橘色
  - skyblue: 天蓝色
  - pink: 粉色
  - .....

### 拓展 颜色常见取值 (了解)

- 取值类型②: rgb表示法
- 每项取值范围: 0~255
- 常见颜色取值:
  - `rgb ( 255 , 0 , 0 )`: 红色
  - `rgb ( 0 , 255 , 0 )`: 绿色
  - `rgb ( 0 , 0 , 255 )`: 蓝色
  - `rgb ( 0 , 0 , 0 )`: 黑色
  - `rgb ( 255 , 255 , 255 )`: 白色
  - .....

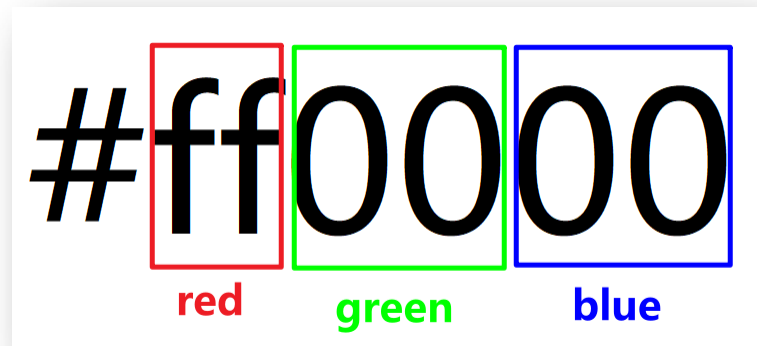


### 拓展 颜色常见取值 (了解)

- 取值类型③: **rgba**表示法
  - 其实, 比rgb表示法多一个a, a表示透明度
- a的取值范围: 0~1
  - 1: 完全不透明
  - 0: 完全透明
- 省略写法:
  - `rgba ( 0 , 0 , 0 , 0.5 )` 可以省略写成 `rgba ( 0 , 0 , 0 , .5 )`

### 拓展 颜色常见取值 (了解)

- 取值类型④：十六进制表示法
- 取值范围：
  - 两个数字为一组，每个数字的取值范围：0~9 , a , b , c , d , e , f
- 省略写法：
  - 如果三组中，**每组数字都相同**，此时可以每组可以省略只写一个数字
  - 正确写法：#ffaabb 改写成 #fab
- 常见取值：
  - #fff：黑色
  - #000：白色
- 注意点
  1. 类似于：#ffaabc 不能改写成 #fab
  2. 实际开发中会直接使用测量工具直接得到颜色，不需要前端自己设计颜色，直接复制粘贴即可。





# 目录

Contents

- ◆ 基础认知
- ◆ 基础选择器
- ◆ 字体和文本样式
- ◆ Chrome调试工具
- ◆ 综合案例

## 五、综合案例

**目标：通过今天学习的CSS相关属性，完成综合案例**

**学习路径：**

1. 综合案例1-新闻网页案例
2. 综合案例2-卡片居中案例

### 1. 新闻网页案例-效果图

#### 《自然》评选改变科学的10个计算机代码项目

2077年01月28日14:58 [新浪科技](#) [收藏本文](#)

2019年，事件视界望远镜团队让世界首次看到了黑洞的样子。不过，研究人员公布的这张发光环形物体的图像并不是传统的图片，而是经过计算获得的。利用位于美国、墨西哥、智利、西班牙和南极地区的射电望远镜所得到的数据，研究人员进行了数学转换，最终合成了这张标志性的图片。研究团队还发布了实现这一壮举所用的编程代码，并撰文记录这一发现，其他研究者也可以在此基础上进一步加以分析。

这种模式正变得越来越普遍。从天文学到动物学，在现代每一项重大科学发现的背后，都有计算机的参与。美国斯坦福大学的计算生物学家迈克尔·莱维特因“为复杂化学系统创造了多尺度模型”与另两位研究者分享了2013年诺贝尔化学奖，他指出，今天的笔记本电脑内存和时钟速度是他在1967年开始获奖工作时实验室制造的计算机的1万倍。“我们今天确实拥有相当可观的计算能力，”他说，“问题在于，我们仍然需要思考。”

如果没有能够解决研究问题的软件，以及知道如何编写并使用软件的研究人员，一台计算机无论再强大，也是毫无用处的。如今的科学研究从根本上已经与计算机软件联系在一起，后者已经渗透到研究工作的各个方面。近日，《自然》(Nature)杂志将目光投向了幕后，着眼于过去几十年来改变科学研究的关键计算机代码，并列出了其中10个关键的计算机项目。

最初的现代计算机并不容易操作。当时的编程实际上是手工将电线连接成一排排电路来实现的。后来出现了机器语言和汇编语言，允许用户用代码为计算机编程，但这两种语言都需要对计算机的架构有深入的了解，使得许多科学家难以掌握。20世纪50年代，随着符号语言的发展，特别是由约翰·巴克斯及其团队在加州圣何塞的IBM开发的“公式翻译”语言Fortran，这种情况发生了变化。利用Fortran，用户可以用人类可读的指令来编程，例如 $x = 3 + 5$ 。然后由编译器将这些指令转换成快速、高效的机器代码。

### 2. 卡片居中案例-效果图







## 总结

- ◆ 能够知道 CSS 的引用方式及特点区别
- ◆ 能够使用基础选择器在HTML中选择元素
- ◆ 能够使用字体和文本相关样式修改外观样式
- ◆ 能够使用CSS让页面中元素水平居中
- ◆ 能够使用 Chrome 调试工具修改和检查样式



传智教育旗下高端IT教育品牌