# １　HTML和CSS

1、

|  |  |
| --- | --- |
| **常见选择器** | **适用场景** |
| 元素选择器 | 某一类型元素需要用同一些样式 |
| id选择器 | 某个特定元素需要用同一些样式 |
| 类选择器 | 多个不同类型元素需要用同一些样式 |
| 通配选择器 | 需要选中文档所有元素 |
| 复合选择器（交集选择器、并集选择器） | 将某些类型的样式“压缩”在一起 |
| 关系选择器（子元素选择器、后代元素选择器、兄弟元素选择器） | 需要选中某一元素的指定子元素/后代元素/兄弟元素时 |
| 属性选择器 | 需要选中含有指定属性的元素时 |
| 伪类选择器 | 需要选中一个或多个含有特殊状态的元素第一个子元素、被点击的元素、鼠标移入的元素···）或去除某些符合条件的元素（否定伪类） |
| 伪元素选择器 | 需要选中右特殊位置（第一个字母、第一行···）的元素时 |

通常可以将多种选择器结合使用，以使代码更简洁。

2、

|  |  |  |
| --- | --- | --- |
| **居中方法（垂直居中：假设父元素是盒子容器且高度已设定）** | **操作** | **应用场景** |
| 行内元素水平居中 | 在父元素中设置text-align: center; | 文字水平居中、图片水平居中 |
| 块元素水平居中（定宽） | 设置该元素左右margin值为auto | 已知元素宽度时 |
| 块元素水平居中（不定宽） | 在该元素/父元素中设置text-align: center;  当不定宽块级元素的宽度不要占一行时，可以设置display 为 inline类型或inline-block | 未知元素宽度/元素宽度可能改变 |
| 垂直居中：子元素是行内元素 | 设置父元素，  使line-height==height | 文字垂直居中 |
| 垂直居中：子元素是块元素但高度未设定 | 给父元素设定display:table-cell;vertical-align:middle; |  |
| 垂直居中：子元素是块元素且高度已设定 | 计算子元素的margin-top或margin-bottom，计算方法为(父元素高度-子元素高度)/2 |  |
| 水平垂直居中 | 结合以上两类 |  |

3、

|  |  |  |
| --- | --- | --- |
| **布局方法** | **作用** | **应用场景** |
| 流动模型（flow） | 默认的网页布局模式，块元素在所处的包含元素内自上而下排列，行内元素在所处的包含元素内从左到右排列 | 默认网页布局、一列布局 |
| 浮动模型（float） | 使元素脱离文档流进行浮动 | 二列布局、三列布局、去除排列间的空格 |
| 层模型（layer） | 相对定位（position:relative） 相对于偏移前的位置移动（设置相对定位后元素仍在文档流内） | 利用z-index改变元素覆盖顺序 |
| 绝对定位（position:absolute） 若父元素具有定位属性，则相对于父元素进行定位；若父元素不具有定位属性，则相对于body元素定位（设置绝对定位后，元素脱离文档流） |
| 固定定位（position:fixed） 设置固定定位的元素始终位于浏览器窗口内视图的某个位置，不受文档流动影响 |

4、在网页开发过程中，由于手机屏幕尺寸、分辨率不同，或是需要考虑横竖屏问题，为了使web页面在不同移动设备上具有相适应的展示效果，需要对不同分辨率的手机进行适配。

常见方案：百分比方案、rem方案、vh/vm方案、rem+vw/vh方案和基于媒体查询的响应式设计。通常将几种方法结合使用。

# ２　JavaScript

## ２.１数据类型

1. JavaScript数据类型：原始类型（undefined,null,boolean,number,string）、引用类型（function,array```）

和其他语言的区别：js拥有动态类型（相同变量可用作不同类型），且只有一种数值类型

1. 常用判断数据类型的方法

·typeof：放在起单个操作数的前面，操作数可以是任意类型，返回值是js中的基本数据类型（number、boolean、string、object、undefined、function）。优点：操作简单方便；缺点：无法分辨object具体类型，且只能判断JavaScript已有类型。同时，如果操作数是null，会返回object。

·instanceof：左操作数是待检测其类的对象，右操作数是对象的类。如果左侧的对象是右侧的实例，则返回true，否则返回false。优点：可以判断对象是由哪个类实例化出来的。缺点：左操作数必须是对象（object），若不是，直接返回false。

## ２.２DOM操作

1. 优点：更加灵活，可以根据浏览器窗口实际的大小来定义元素的大小，同时可以改变元素样式，营造动画效果等。

缺点：代码冗长，而且速度稍慢

相对而言我更喜欢用DOM操作样式修饰，因为这种方式比较灵活，方便调整。

## ２.３事件

1. 哪个事件先发生取决于事件传播的方法是冒泡还是捕获。（默认使用冒泡传播，如果将useCapture值设置为true则使用捕获传播）

·在冒泡中，最内侧元素的事件会首先被处理，然后是更外侧的。也就是子元素绑定的事件先发生。

·在捕获中，最外侧元素的事件会首先被处理，然后是更内侧的。也就是首先处理父元素绑定的事件。

1. 虽然js是单线程语言，但js的宿主环境（如浏览器）是多线程的，宿主环境通过事件驱动使js具备了异步的属性。

# Git

1、将已经追踪的文件添加到.ignore列表中，该文件依然处于跟踪状态，不会被忽略。