2.1

1.CSS有众多的选择器，什么时候应该用怎样的选择器才能让代码更加简洁？

主要分为四种：

（1）通配符选择器：对全文的基础属性进行设置，比如margin，padding等，修改浏览器的默认style,设置成自己熟悉且方便进行计算和排布的格式。

（2）标签选择器：对<p>,<h1>等标签进行统一修改格式，可以实现对不同区块的文字，题目，颜色等进行统一的书写，避免了在不同的<div>中的重复书写，且能使css的阅读更加清晰。

（3）类选择器：拓展性很强，在一个<div>中，不仅可以为本身这个父元素进行设置，还可以为子元素进行统一设置，只需加上子元素的标签，就可以进行修改了，而且优先级比前面两种选择器更高，可以进行覆写，修改更方便。同时还可以与伪类进行联动，对如鼠标的悬浮状态进行设置。

（4）ID选择器：与类选择器相似，但是具有最高的优先级，在优先级和层叠的计算中，往往起到决定性的影响。除此之外，ID选择器能够更好地对元素进行定位，在js中能够更精准的找到并获取元素。

2.有哪些元素居中方法，他们分别适用于怎样的场景？

（1)文本居中;设置text-align:center；

(2)区块元素居中

①单个区块：

1设置区块的maigin值，使它在父元素中处于合适的位置；

2使用绝对定位，将top,left,right,bottom设置为50%，实现居中效果；

②多个区块：

1使用Flexbox布局：设置它们的主轴和纵轴，而后使用justify-content:center和align-items:centers，设置为居中，同时还可以设置他们的宽度百分比，使他们呈现不同的大小状态且居中；

2使用Grid布局：通过display:grid和place-items:center来实现水平和垂直居中；

1. 表格居中：display:table；
2. CSS有哪些方法可以进行网页布局，请举例说明？
3. 盒子模型：css中，<div>,<li>等都是一个又一个的区块，通过设置好自己所需的几个大区块，如header,section,footer的位置，而后向其中填入自己所需的子区块，设置它们的margin,padding,width,height等属性进行网页的布局，按照从大到小的思路，先设置好不同的区块，再填入所需的文本、照片等，这样才能清晰明了的对网页进行布局设置。要达到这一点，最重要的是对自己的网页有个整体构思，明白自己要布置几个区域。
4. 浮动布局：用float来进行布局，比较常用于制作页面的导航栏，进行一些多行栏的布局
5. 位置布局：通过设置是相对位置(relative)，还是绝对位置（absolute)来进行布局。设置具体的间距，或者是相对于父元素的距离来布局，也可以使用fix来固定区块，这样可以实现导航栏在网页滑动后仍然固定在页面的最上面，方便进行页面的切换，使得浏览时有更好的体验。
6. 当然，还可以使用js来实现动态布局，在某个点击事件后对网页的相应区块进行切换。如某些注册界面可以使注册方框和登录方框进行左右滑动，给用户留下深刻印象，使得网页得交互能力增强。
7. 为什么要对不同分辨率手机进行适配？常见的适配方案有哪些？
8. 原因：分辨率指的是设备屏幕呈现的像素数量，通常是以水平像素数和垂直像素数来表示，不同的分辨率不仅影响了内容的清晰度和细节水平，而且会以不同的距离来显示网页，如果不适配，那么在编写网页的设备上浏览页面，具有较好的体验和观感，但是在其他移动设备上时，由于区块都是固定好的，会使得区块重叠，或者超出显示界面而出现滚动轴，大大降低了用户的浏览体验，同时对一些按钮、文本，也会增大使用难度。
9. 方案：

①响应式布局：对网页进行弹性布局，使得不同区块的width和height不是单一的固定数值，而是使用一些“%”值来设置，然后使用css进行媒体查询，获取手机的分辨率，在不同的像素区块去调整区块的排布，

②视口设置：使用<meta>标签中的viewport属性来设置视口，使用户可以缩放页面

③设置矢量图形，动态字体大小（rem,em)

2.2.1

1. JavaScript有哪些数据类型，与其他语言不同的地方在哪里？

①原始数据类型（Primitive Data Types）：

Number：用于表示数值，可以包括整数和浮点数。String：用于表示文本字符串。Boolean：表示布尔值，即 true 或 false。Undefined：表示未定义或未初始化的值。Null：表示一个空值或没有值。Symbol（ES6新增）：用于创建唯一的标识符。

引用数据类型（Reference Data Types）：

Object：用于表示复杂数据结构，包括对象、数组、函数等。Array：一种特殊的对象，用于存储有序的数据集合。Function：用于定义函数。Date：用于处理日期和时间。RegExp：用于处理正则表达式。

②不同的地方：

JavaScript是单线程，动态类型，解释型语言，且具有开放的生态系统，可以跨平台使用，同时在JavaScript中一切皆对象。

1. 有哪些判断数据类型的方法，他们的优缺点是什么？

①一、typeof操作符

优点：可以判断基本数据类型（返回值有：undefined,number,string,boolean,object,function）

缺点：不能判断出null，object，Array，因为它们的返回值都是object，Array属于object。typeof判断是由他们前三位的二进制决定的，null由于它的二进制都是0，object的前三位为0，所以typeof判断出它们都是object。

②instanceof 操作符

优点：用于检查对象是否是特定构造函数的实例，适用于自定义对象类型的判断。

缺点：对于原始数据类型的判断不起作用，无法区分不同全局执行上下文中创建的对象。

③Object.prototype.toString.call()

优点：可以准确判断数据类型，返回一个表示数据类型的字符串，准确，适用于原始数据类型、对象、数组和自定义对象。

缺点：代码相对冗长，需要使用额外的方法来提取类型信息。

④Array.isArray()

优点：用于检查一个值是否为数组，简单、适用于数组类型的判断。

缺点：不能判断其他数据类型，如对象或原始数据类型。

⑤typeof 和 instanceof 组合

优点：结合使用 typeof 和 instanceof 可以弥补各自的不足，实现更全面的数据类型检测，更精确地判断数据类型。

缺点：复杂，需要多次检查。

2.2.2

1. 和CSS相比，使用DOM操作对元素进行样式修饰有什么优缺点，你更喜欢哪个？
2. 优点：①具有动态性和交互性②可以改变网页原本的css属性设置③可以进行较为复杂的数学计算，获取元素信息，自定义函数进行动态功能的实现
3. 缺点：①代码较为复杂，学习和编写的成本较大②因为其功能的多样性和连续性，使得修改和维护较为困难，需要使用git来多次尝试③性能开销较大，对设备、网络的有一定的要求④由于浏览器的版本不同，因而存在一些兼容性问题
4. 个人：我较为喜欢DOM操作。因为它能够实现网页的交互，增加了趣味性，同时比css更加灵活，尽管css可以使用浮动布局等等增加网页的可适性，但是DOM具有更为强大和自由的编写方式，与之相比，css就显得有些僵硬了。但不得不承认css的简洁和可读性是非常有用的，能够清晰的对网页的设定有一个整体认知，同时在修改时也会更为便捷。
5. 接触了 JavaScript 操作 DOM 样式之后，有引发你关于 CSS 的思考吗？

在刚学习JavaScript的时候，JavaScript之余css，就如同在刚学习css的时候，html之余css。不禁觉得上一个学习内容的简单，同时新操作的新功能所带来的趣味性又是无可比拟的。JavaScript比css有更丰富的操作，但不如css简洁，如果可以为css设置一套统一的代码，将一些点击事件或者动态动画设置为固定的简洁方式加入css，那么就会使css满足大部分人的普通需求，降低一定的学习门槛，也为网页维护提供便利。

2.2.3

1. 当父元素和子元素都绑定了事件，点击子元素的时候，哪一个事件会先发生？

该问题指的使JavaScript中的事件冒泡现象，就是同时为父元素，子元素，有些时候还有为整个<body>绑定的事件，那么在点击子元素时，子元素的事件会首先被触发，而后父元素，<body>绑定的事件也都会依次被触发。因为事件冒泡现象允许事件从触发它的元素逐级向上冒泡到DOM操作中更高层的元素。

1. JS是单线程工作的，为何在我们与浏览器进行交互的过程中，网页可以**同时**进行一系列数据处理工作？

因为浏览器是多进程的，可以进行渲染，网络请求等不同的进程，进程之间相互独立，各自得到浏览器分配的CPU资源。而各自的进程由线程完成，那么就如同铺路，是有先后顺序的，肯定是先把路清理平整，再铺上沥青，不能说一边清理，一边铺路，那么js就是以一个类似的方式进行代码的运行，否则可能会出现前后矛盾的情况，使得代码紊乱，相应的功能得不到实现。

3.

1. 通过编写.ignore文件，我们可以忽略掉部分文件（文件夹），假设现在有一个已经追踪的文件，将其添加到.ignore列表中，会发生什么？

该文件仍然会被追踪。只有在文本被追踪之间就添加入.ignore文件中才会被忽略，只有通过git rm--cached以及git commit-m“”才能使得被追踪的文件不被追踪。

1. 发现一次提交漏掉了一些文件，或者 commit 内容写错了，最好怎么做？

从问题的字面上理解，很容易想出解决办法：①补交漏掉的文件；②撤回提交的操作，对所有需要提交的文件重新进行提交

但是随意撤回，如果不是在提交后就发现漏交了，那么就容易使得后面修改的数据在回溯到提交之前后消失。

最好的办法是在重新提交时先检查库里是否有新提交的内容，提前做好复制，然后说明两次提交的冲突部分，重新进行提交。

1. 生产时较少使用 reset，你认为会是什么原因呢？

reset可以用于取消提交，用于拆分提交，也可用于移动分支指针。但是它会“重置”到之前的版本，撤销之后的提交操作，造成数据丢失。在多人合作时会影响他人的工作成果，产生冲突。

从问题2，3告诉我们，在使用git时，要谨慎使用reset操作，当然还有rebase操作。所以要清楚自己的git命令会带来什么可能的后果、影响，不能片面的只去看一些操作的利处。