1. 精灵图

1.1. 为什么需要精灵图

网页中的图像过多时，服务器就会频繁的接受和发送请求图片，造成服务器的压力请求过大，这将多大降低页面的加载速度

为了有效地减少服务器接收和发送请求的次数，提高网页的加载速度，出现了css精灵技术（也称css sprite ， css雪碧）

核心原理：将网页中的一些小背景整合到一张大图中，这样服务器只需要一次请求就可以了

1.2. 精灵图（sprites）的使用

使用精灵图的核心：
精灵技术主要针对背景图片使用，就是把多个小背景图整合到一张大图中

这个大图片也成为sprites 精灵图或者雪碧图

移动背景图片位置，此时可以使用background-position

移动的距离就是这个目标图片的x和y坐标。注意网页中的坐标有所不同（盒子不动，图片动）

因为一般情况下都是往上往左移动，所以数值都是负值

2. 字体图标

2.1. 字体图标的产生

字体图标使用场景：主要用于显示网页中通用、常用的一些小图标

小图标可以用精灵图制作，但是缺点也很明显：

- 图片文件还是比较大
- 图片本身放大和缩小会失真
- 一旦图片制作完成想要更换非常复杂

此时，有一种技术的出现很好的解决了精灵图的问题，就是字体图标iconfont

字体图标可以为前端工程师提供一种方便高效的图标使用方式，展示的是图标，本质属于字体

2.2. 字体图标的优点

- 轻量级：一个图标字体要比一系列的图像要小，一旦字体加载了，图标马上渲染出来，减少了服务器请求
- 灵活性：本质其实是文字，可以随意的改变颜色，产生阴影，透明效果，旋转等
- 兼容性：几乎支持所有的浏览器

注意：
字体图标不能替代精灵技术，只是对工作中图标部分技术的提升和优化

总结：
如果遇到一些结构和样式比较简单的小图标，就用字体图标

2.3. 字体图标的下载

icomoon字库
阿里iconfont字库

2.4. 字体图标的引入在style标签中加入声明：

```
/* 字体声明 */

        @font-face {
            font-family: 'icomoon';
            src: url('fonts/icomoon.eot?p4ssmb');
            src: url('fonts/icomoon.eot?p4ssmb#iefix') format('embedded-opentype'), url('fonts/
            font-weight: normal;
            font-style: normal;
            font-display: block;
        }

        span {
            font-family: 'icomoon';
            font-size: 100px;
            color: pink;
        }
```

把下载包里面的fonts文件放到页面根目录下
html标签内添加小图标 （ 图标在vscode中看上去都一样，是因为vscode不能识别它，其实每个 都不相同， 图标从下载下来文件中的demo.html打开的页面中复制即可)

在style标签中指定样式，一定要加上一句：font-family: 'icomoon';
如：

```
span {
            font-family: 'icomoon';
            font-size: 100px;
            color: pink;
        }
```

2.5. 字体图标的追加

把压缩包里的selection.json重新上传，然后选中自己想要的新的图标，重新下载图标包，并替换原来的即可

3. css三角

网页中常见一些三角形，使用css直接画出来就可以，不必做成图片或者字体图标
当把一个盒子的高和宽的长度都设置为0，并且分别指定边框样式时，就会得到以下图形：
受此启发，可以知道三角是如何制作的（想要保留哪个三角只需把其他三个边框设置为透明即可）
例如：

```
div {
    width:0;
    height:0;
    line-height:0;
    font-size:0;
    border:50px solid transparent;
    border-left-color:pink;
}
```

4. css用户界面样式

4.1. 什么是界面样式

所谓界面样式，就是更改一些用户的操作样式，以便提高用户体验
更改用户的鼠标样式
表单轮廓
防止表单域拖拽

鼠标样式cursor

```
li {cursor:pointer}
```

设置或检索在对象上移动的鼠标指针采用荷重系统预定义的光标样式

属性值	描述
default	小白，默认
pointer	小手
move	移动
text	文本
not-allowed	禁止

4.2. 轮廓线outline

给表单添加outline:0;或者outline:none;样式之后，就可以去掉默认的蓝色边框

```
input {outline:none;}
```

4.3. 防止拖拽文本域resize

实际开发中，我们文本域右下角是不可以拖拽的

```
textarea{resize:none;}
```

小细节：文本域的HTML代码最好写到一行上，如果没有在一行上，在页面上输入时会有一个空白区域

5. vertial-align属性应用

css的vertial-align属性应用场景：经常用于设置图片或者表单（行内块元素）和文字垂直对齐

官方解释：用于设置一个元素的垂直对齐方式，但是它只针对行内元素或者行内块元素有效

语法：

```
vertial-align:baseline|top|middle|bottom
```

值	描述
baseline	默认。元素放在父元素的基线上
top	把元素的顶端与行中最高元素的顶端对齐
middle	把此元素放置在父元素的中部
bottom	把元素的顶端与行中最低的元素的顶端对齐

5.2. 解决图片底部默认空白缝隙问题

bug:图片底部会有一个空白缝隙，原因是行内块元素会和文字的基线对齐

两个解决方案：
(1)给图片添加vertical-align:middle|top|bottom等（提倡使用）
(2)把图片转换为块级元素display:block;

6. 溢出的文字省略号显示

6.1. 单行文本溢出显示省略号

单行文本溢出显示省略号，必须满足三个条件：
（1）先强制一行内显示文本
white-space:nowrap;(默认 normal自动换行)
（2）超出的部分隐藏
overflow:hidden;
（3）文字用省略号代替超出的部分
text-overflow:ellipsis;

6.2. 多行文本溢出显示省略号

多行文本溢出显示省略号，有较大的兼容性问题，适合于webkit浏览器或者移动端（移动端大多是webkit内核）

语法：

```
overflow:hidden;
text-overflow:ellipsis;
/*弹性伸缩盒子模型显示*/
display:-webkit-box;
/*限制在一个块元素显示的文本的行数*/
-webkit-line-clamp:2;
/*设置或检索伸缩盒子对象的子元素的排列方式*/
-webkit-box-orient:vertical;
```

7. 常见的布局技巧

巧妙利用一个技术更快更好的布局：
margin负值的运用

文字围绕浮动元素

行内块的运用

css三角强化

7.1. margin负值运用

让每个盒子margin往左移动-1px正好压住相邻的盒子边框
鼠标经过某个盒子的时候，提高当前盒子的层级即可（如果没有定位，则加相对定位（保留位置），如果有定位，则加z-index）

7.2. 文字围绕浮动元素

7.3. 行内块巧妙运用

7.4. css三角强化

代码：

```
widht:0;
height:0;
border-color:transparent red transparent transparent;
border-style:solid;
border-width:22px 8px 0 0;
```

8. css初始化

不同的浏览器对于有些标签的默认值是不同的，为了消除不同浏览器对于HTML文本呈现的差异，照顾浏览器兼容，需要对css初始化

简单理解：css初始化是指重设浏览器的样式（css reset）
每个网页都必须首先进行css初始化

Unicode编码字体：
把中文的字体的名称用相应的Unicode编码来代替，这样就可以有效地避免浏览器解释css代码时出现乱码的问题

黑体：\9ed1\4f53
宋体：\5b8b\8f6f\06c5\9ed1