

## 1、谈谈对盒子模型的理解

### 1. 是什么

当对一个文档进行布局的时候，浏览器的渲染引擎会根据标准之一的 **CSS** 基础框盒模型，将所有元素表示为一个个矩形的盒子

一个盒子由四个部分组成：**content**、**padding**、**border**、**margin**

### 2. 分类

1) 标准盒子模型：是浏览器默认的盒子模型，**width/height** 只是内容高度，不包含 **padding** 和 **border**值

2) **IE** 怪异盒子模型：**width/height** 包含了 **padding**和 **border**值

## 2、H5和HTML5区别

1. **H5**是一个产品名词，包含了最新的**HTML5**、**CSS3**、**ES6**等新技术来制作的应用

2. **HTML5**是一个技术名词，指的就是第五代**HTML**

## 3、flex的基本使用？

### 1.是什么

1) 浏览器推荐的一种布局模型

2) 组成：弹性容器/弹性盒子/主轴/侧轴

### 2. 特点

特点：主轴默认水平侧轴默认垂直/弹性盒子沿主轴排列/行内元素变成弹性盒子可设置宽高/可以控制主轴和侧轴对齐方式

### 3. 容器有以下属性：

**flex-direction**属性决定主轴的方向

**flex-wrap**属性定义，如果一条轴线排不下，换行

**justify-content**属性定义了项目在主轴上的对齐方式

**align-items**属性定义项目在交叉轴上如何对齐。

**align-content**属性定义了对多根轴线的对齐方式。

### 3. flex使用

#### 1) 主轴对齐方式

*/\* 水平居中对齐 重要\*/*

**justify-content: center;**

*/\* 间距在子盒子的两侧 \*/*

**justify-content: space-around;**

*/\* 两端对齐 空间在子盒子之间 重要\*/*

**justify-content: space-between;**

*/\* 盒子的所有间距都相等 \*/*

**justify-content: space-evenly;**

#### 2) 侧轴对齐方式

*/\* 居中对齐 重点\*/*

**align-items: center;**

## 4、行内元素有哪些？块级元素有哪些？空元素（void）有哪些？

- 1、行内元素：`a`, `span`, `strong`, `em`
- 2、块级元素：`div`, `ul`, `li`, `p`, `h1-h6`
- 3、行内块元素：`img`, `input`, `label`, `button`, `select`, `textarea`
- 4、空元素（`void`）：即没有内容的HTML元素。`br`（换行），`hr`（水平分割线），`meta`, `link`, `input`, `img`

## 5、BFC（Block Formatting Context，块级格式化上下文）

1. 是什么  
所谓 **BFC**，指的是一个独立的布局环境，**BFC** 内部的元素布局与外部互不影响
2. 触发 **BFC** 的方式：
  - 1) 设置浮动
  - 2) `overflow` 设置为 `auto`、`scroll`、`hidden`
  - 3) `position` 设置为 `absolute`、`fixed`
3. 常见的 **BFC** 应用有：
  - 1) 解决浮动元素令父元素高度坍塌的问题
  - 2) 解决非浮动元素被浮动元素覆盖问题
  - 3) 解决外边距垂直方向重合的问题

## 6、CSS选择器的优先级

1. 是什么  
**CSS**选择器是**CSS**规则的第一部分，他是标签和样式的纽带，选择器所选择的元素，也叫做“选择器的对象”
2. 优先级规则  
`!important` > 行内样式 > **ID**选择器 > 类选择器 > 标签 > 通配符 > 继承

## 7、初始化CSS样式的意义

1. 为什么要初始化**CSS**样式：因为浏览器的兼容问题，不同浏览器对有些标签的默认值是不同的，如果没有对**CSS**初始化往往会出现浏览器之间的页面显示差异。

## 8、HTML5新特性

1. 语义化标签：`header` `nav` `section` `article` `aside` `footer`
2. 多媒体标签：`video` `audio`
3. `input`类型：`number` `search` `email` `date` `file` `time` `url`
4. 本地离线存储
  - 1) `localStorage` 长期存储数据，关闭浏览器数据不会丢失
  - 2) `sessionStorage` 临时存储，浏览器关闭数据会丢失
5. 自定义属性 `data-*`
6. 画布 `Canvas`
7. `websocket` 双向通信协议

## 9、CSS3新特性

1. 圆角 `border-radius`
2. 盒模型 `box-sizing`
3. 阴影: `box-shadow` 盒子阴影 `text-shadow` 文字阴影
4. 过渡 `transition`
5. 2D转换 `transform` `translate`(平移) `scale`(缩放) `rotate`(旋转) `transform-origin` 控制转换中心点
6. 3D转换 `perspective`(透视距) `transform-style`(3D控件效果)
7. 渐变 `linear-gradient`
8. 弹性布局 `flex`
9. 媒体查询 `@media screen and () {...}`
11. 自定义动画 `@keyframes` `animation`

## 10、对HTML语义化的理解

1. 搜索引擎的爬虫也依赖于HTML标记来确定上下文和各个关键字的权重，利于SEO
2. HTML语义化让页面的内容结构化，结构更清晰，便于对浏览器、搜索引擎解析
3. 使阅读源代码的人更容易将网站分块，便于阅读维护理解

## 11、你知道SEO中的TDK吗

1. 是什么  
seo搜索引擎优化,可以提高网站的排名
2. 优化方式  
在SEO中,TDK其实就是title、description、keywords这三个标签, title表示标题标签, description是描述标签, keywords是关键词标签

## 12、如何垂直水平居中盒子

1. 水平垂直居中一: `margin + position`

```
div {
    position: absolute;
    width:500px;
    height:300px;
    top: 50%;
    left: 50%;
    margin: -150px 0 0 -250px;
    background-color: pink;
}
```
2. 水平垂直居中二: `transform + position`

```
div {
    position: absolute;
    width:500px;
    height:300px;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    background-color: pink;
}
```
3. 水平垂直居中三: `flex`

```
.father {
    display: flex;
    align-items: center;
```

```
        justify-content: center;

    }
    .father .son {
        width: 100px;
        height: 100px;
        background-color: pink;
    }
}
```

## 13、什么是重绘和重排

### 1. 是什么

1) 重排(回流): 当DOM元素影响了元素的几何属性(例如宽和高), 浏览器需要重新计算元素的几何属性, 同样其它元素的几何属性也会和位置也会因此受到影响。浏览器会使渲染树中受到影响的部分失效, 并重新构造渲染树。这个过程称为“重排”。

2) 重绘: 完成重排后, 浏览器会重新绘制受影响的部分到屏幕上中, 该过程称为“重绘”。

### 2. 重排重绘的因素

- 1) 当我们改变DOM的大小和位置, 增加删除, 浏览器窗口改变都会导致重排(回流)
- 2) 当给DOM元素改变颜色的时候, 会导致重绘

## 14、px/em/rem/vw有什么区别

### 1. 是什么

页面布局中常用的一种单位

### 2. 区别

- 1) px 固定的像素, 一旦设置了就无法因为适应页面大小而改变
- 3) em 是相对单位, 相对自身font-size, 没有则相对于父元素
- 4) rem 是相对单位, 相对于根元素的font-size
- 5) vw 是相对单位, 相对视口的宽度

## 15、rem布局的原理

1. rem 是css的相对单位, rem缩放是相对根元素字体大小。
2. rem 布局的本质是等比缩放, 一般是基于宽度。
3. rem 会配合媒体查询(或js动态获取屏幕宽度)来一起使用, 来实现屏幕的适配。

## 16、怎么用CSS画一个三角形

### 1. 实现原理

- 1) 利用盒模型的 border 属性上下左右边框交界处会呈现出平滑的斜线这个特点, 通过设置不同的上下左右边框宽度或者颜色即可得到三角形或者梯形。
- 2) 如果想实现其中的任一个三角形, 把其他方向上的 border-color 都设置成透明即可

### 2. 示例代码

```
.box {
    width: 0;
    height: 0;
    border-left: 50px solid transparent;
    border-right: 50px solid transparent;
    border-top: 50px solid transparent;
    border-bottom: 100px solid red;
}
```

## 17、使元素消失的方法有哪些

1. **opacity: 0**, 该元素隐藏起来了, 但不会改变页面布局, 并且, 如果该元素已经绑定 一些事件, 如 **click** 事件, 那么点击该区域, 也能触发点击事件的
2. **visibility: hidden**, 该元素隐藏起来了, 但不会改变页面布局, 还占据位置
3. **display: none**, 把元素隐藏起来, 并且会改变页面布局, 可以理解成在页面中把该元素删掉, 不占位置。

## 18、单行元素的文本省略号

```
.box {  
  overflow: hidden;           // 溢出隐藏  
  text-overflow: ellipsis;    // 溢出用省略号显示  
  white-space: nowrap;        // 文本不进行换行  
}
```

注意: 前提是此元素要设置具体的宽度

## 19、rgba() 和 opacity 的透明效果有什么不同?

1. **opacity** 作用于元素, 以及元素内的所有内容的透明度, **rgba()** 只作用于元素的颜色或其背景色
2. 设置**rgba**透明的元素的子元素不会继承透明效果

## 20、如何设置比12px还要小的字体

1. 为什么

谷歌默认可设置最小字体 **12px**, 特殊场景需要设置更小字体, 例如: 盒子放不下

2. 方式一

```
p {  
  font-size: 12px;  
  transform: scale(0.8);  
}
```

注意: 行内元素没有效果

2. 方式二

```
p {  
  font-size: 12px;  
  zoom: 0.5;  
}
```

