页面布局

常见的 css 布局单位

- px：像素
- %: 一般认为是子元素的百分比相对于父元素
- em：当前对象内 font-size 的尺寸
- rem：rem 是 CSS3 新增的一个相对单位，相对于根元素（html 元素）的 font-size 的倍数。
- vw/wh:与视口宽度/高度

px em rem 的区别及使用场景

❤️两栏布局的实现

左边一栏宽度固定，右边一栏宽度自适应

- 利用浮动
- 利用浮动
- 利用 flex
- 利用绝对定位

❤️三栏布局的实现

- 利用绝对定位
- 利用 flex
- 利用浮动
- 圣杯布局
- 双飞翼

❤️水平垂直居中的实现

如何根据设计稿进行移动端适配

❤️对 flex 布局的理解及其使用场景

`flex:1;` 是什么?

flex:1; = flex:1 1 0n; (where n is a length unit)

- flex-grow：增长
- flex-shrink：收缩
- flex-basis：基础

响应式设计的概念及基本原理

定位与浮动

❤️为什么需要清除浮动？清除浮动的方式

- 浮动的定义：非 IE 浏览器下，容器不设高度且子元素浮动时，容器高度不能被内容撑开。 此时，内容会溢出到容器外面而影响布局。这种现象被称为浮动（溢出）。
- 浮动的原理：
  - 浮动元素脱离文档流，不占据空间（引起“高度塌陷”现象）
  - 浮动元素碰到包含它的边框或者其他浮动元素的边框停留
- 解决方法
  - 父元素 overflow:hidden
  - 父元素伪元素

```
.clearfix::after {
  content: "";
  /* display: table; */
  display: block;
  clear: both;
}
```

使用 clear 属性清除浮动的原理

❤️对 BFC 的理解，如何创建 BFC

块级格式化上下文，是 Web 页面的可视 CSS 渲染的一部分，是块级盒子的布局过程发生的区域，也是浮动元素与其他元素交互的区域。
创建 BFC 的条件

- 根元素()
- 浮动元素(float 不为 none)
- 绝对定位元素(position absolute fixed)
- 行内块元素(display inline-block) 表格单元格(display table-cell) 表格标题(display table-caption) 匿名表格单元元素 display 值为 flow-root
- overflow 值不为 visible clip
- contain 值为 layout content paint
- flex (display flex inline-flex)
- grid (display grid inline-grid)
- 多列容器（column-count 或 column-width (en-US) 值不为 auto，包括 column-count 为 1）
- column-span 值为 all 的元素始终会创建一个新的 BFC

BFC 特点

- 包含内部浮动
- 排除外部浮动
- 阻止外边距重叠
  BFC 的区域不会与浮动元素发生重叠

什么是 margin 重叠问题？如何解决？

元素的层叠顺序

❤️position 的属性有哪些，区别是什么

- absolute:相对于父元素绝对定位
- relative:相对原本的位置进行定位
- fixed:相对视口绝对定位
- static:默认值
- inherit:继承父元素

display、float、position 的关系

- 首先 display，判断是否为 none，如果是 none 则 position 和 float 属性的值不影响元素最后的表现。
- 然后 position，判断是否为 absolute 或者 fixed，如果是，则 float 属性失效，并且 display 的值应该被设置为 table 或者 block，具体转换需要看初始转换值
- 如果不是 absolute 或者 fixed，则判断 float 属性的值是否为 none，如果不是，则 display 的值则按上面的规则转换。注意，如果 position 的值为 relative 并且 float 属性的值存在，则 relative 相对于浮动后的最终位置定位。
- 如果 float 的值为 none，则判断元素是否为根元素，如果是根元素则 display 属性按照上面的规则转换，如果不是，则保持指定的 display 属性值不变。

总结：
"position:absolute"和"position:fixed"优先级最高，有它存在的时候，浮动不起作用，'display'的值也需要调整；
其次，元素的'float'特性的值不是"none"的时候或者它是根元素的时候，调整'display'的值；
最后，非根元素，并且非浮动元素，并且非绝对定位的元素，'display'特性值同设置值。

absolute 与 fixed 共同点与不同点

对 sticky 定位的理解

场景适用

❤️实现一个三角形

实现一个扇形

实现一个宽高自适的正方形

画一条 0.5px 的线

设置小于 12px 的字体

- `transform: scale(0.75);`

❤️如何解决 1px 的问题

css 基础

❤️CSS 选择器及其优先级

- 标签选择器(div)、伪元素选择器(:after)： 1
- 类选择器(#classname)、伪类选择器(:last-child)、属性选择器(a[ref="xxx"])：10
- id 选择器(#id)：100
- 内联样式：1000
- 通配选择器(*)、子选择器(>)、相邻同胞选择器(+)、后代选择器(li a)：0

注意事项

- 使用!important 声明的样式的优先级最高
- 如果优先级相同，则最后出现的样式生效
- 继承得到的样式的优先级最低
- **样式表优先级:** 内联样式>内部样式>外部样式>浏览器用户自定义样式>浏览器默认样式

css 中可继承和不可继承属性有哪些

display 的属性值及其作用

```
.container {
  display: [ <display-outside> | <display-inside> ] | <display-listitem> |
    <display-internal> | <display-box> | <display-legacy>;
}
```

❤️display 的 block、inline 和 inline-block 的区别

- **block**会独占一行，多个元素会另起一行，可以设置 width、height、margin 和 padding 属性
- **inline**元素不会独占一行，设置 width、height 属性无效。但可以设置水平方向的 margin 和 padding 属性，不能设置垂直方向的 padding 和 margin；

- **inline-block**和 inline 属性相比可以设置 width、height 属性，也可以设置垂直方向的 padding 和 margin；与 block 属性相比不会换行。

❤️隐藏元素的方法

- display:none 渲染树不会包含该渲染对象，因此该元素不会在页面中占据位置，也不会响应绑定的监听事件
- visibility:hidden 元素在页面中仍占据空间
- opacity:0 将元素的透明度设置为 0，元素在页面中仍然占据空间，并且能够响应元素绑定的监听事件。
- position:absolute 通过绝对定位将元素移出可视区
- z-index:负值 来使其他元素遮住该元素
- clip/clip-path 使用元素裁剪的方法来实现元素的隐藏，这种方法下，元素仍在页面中占据位置，但是不会响应绑定的监听事件。
- transform: scale(0,0)

link 和@import 的区别

- link 是 XHTML 标签，除了加载 CSS 外，还可以定义 RSS 等其他事务；@import 属于 CSS 范畴，只能加载 CSS。
- link 引用 CSS 时，在页面载入时同时加载；@import 需要页面网页完全载入以后加载。
- link 是 XHTML 标签，无兼容问题；@import 是在 CSS2.1 提出的，低版本的浏览器不支持。
- link 支持使用 Javascript 控制 DOM 去改变样式；而@import 不支持。

transition 和 animation 的区别

❤️display:none 和 visibility:hidden 的区别

区别	display:none	visibility:hidden
在渲染树中	从渲染树消失，渲染时不占据空间	不会从渲染树中消失， 渲染的元素还会占据相应的空间， 只是内容不可见

区别	display:none	visibility:hidden
是否继承属性	是非继承属性， 子孙节点会随着父节点从渲染树消失， 通过修改子孙节点的属性也无法显示；	hidden 是继承属性， 子孙节点消失是由于继承了 hidden， 通过设置 visibility:visible 可以让子孙节点显示
重排/重绘	重排	重绘
读屏器	内容不会被读取	内容会被读取

伪元素和伪类的区别和作用

对 requestAnimation 的理解

❤️对盒模型的理解

当对一个文档进行布局（layout）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（CSS basic box model），将所有元素表示为一个个矩形的盒子（box）。 一个盒子由四部分组成：

- content:实际内容
- padding：内边距
- border：边框
- margin：外边距

标准盒模型的 width 和 height 属性的范围只包含了 content，IE 盒模型的 width 和 height 属性的范围包含了 border、padding、conent（border 以内）

- box-sizeing: content-box 表示标准盒模型（默认值）
- box-sizeing: border-box 表示 IE 盒模型（怪异盒模型）

为什么有时候用 translate 来改变位置而不是定位

li 和 li 之间有看不见的空白间隔是什么原因引起的？如何解决

❤️CSS3 有哪些新特性

- 新增各种 CSS 选择器 （: not(.input)：所有 class 不是“input”的节点）
- 圆角 （border-radius:8px）
- 多列布局 （multi-column layout）
- 阴影和反射 （Shadoweflect）
- 文字特效 （text-shadow）
- 文字渲染 （Text-decoration）
- 线性渐变 （gradient）
- 旋转 （transform）
- 增加了旋转,缩放,定位,倾斜,动画,多背景

替换元素的概念及计算规则

常见的图片格式及使用场景

对 cssprites 的理解

什么是物理像素、逻辑像素和像素密度，为什么在移动端开发时需要用到@3x @2x 这种图片

margin 和 padding 的使用场景

对 line-height 的理解及其赋值方法

css 优化和提高性能的方法

css 预处理器/后处理器是什么？为什么要使用他们

对媒体查询的理解

对 css 工程化的理解

如何判断元素是否达到可视区域

❤️单行/多行文本溢出

- 单行文本溢出

```
overflow: hidden;
text-overflow: ellipsis; // 溢出用省略号显示
white-space: nowrap; // 规定段落中的文本不进行换行
```

- 多行文本溢出

```
overflow: hidden; // 溢出隐藏
text-overflow: ellipsis; // 溢出用省略号显示
display: -webkit-box; // 作为弹性伸缩盒子模型显示。
-webkit-box-orient: vertical; // 设置伸缩盒子的子元素排列方式：从上到下垂直排列
-webkit-line-clamp: 3; // 显示的行数
```