作 业

1. 填空题
2. 在 HTML 中，用来创建标题的标签是 \_h1~h6\_\_\_\_\_\_。
3. 要创建一个超链接，使用的 HTML 标签是 \_\_\_\_\_<a href=”URL”/>\_\_\_。
4. 要插入图片，使用的 HTML 标签是 \_\_img\_\_\_\_\_\_。
5. HTML 文档的主体内容放在 \_\_body\_\_\_\_\_\_ 标签内。
6. 用于定义表格行的标签是 \_\_\_tr\_\_\_\_\_。
7. 定义无序列表的标签是 \_\_\_ul，ol\_\_\_\_\_。
8. HTML 注释的语法是 \_\_<!-- TODO -->\_\_\_\_\_\_。
9. 用于嵌入视频和音频的标签分别是 \_\_video\_\_\_\_\_\_、\_\_\_audio\_\_\_\_\_。
10. 用于定义表格数据单元的标签是 \_\_td\_\_\_\_\_\_。
11. 用于定义文档头部信息的标签是 \_\_head\_\_\_\_\_\_。
12. 用于定义段落的标签是 \_\_\_p\_\_\_\_\_。
13. HTML 中用于定义样式表链接的标签是 \_\_link\_\_\_\_\_\_。
14. 判断题
15. HTML 中的标签都是大小写敏感的。 F
16. <div> 标签用于定义一个块级的容器，可以用来组织页面上的内容。T
17. HTML 中，无论在哪里添加空格、换行符或制表符，浏览器都会将它们合并为一个空格。T
18. 在 HTML 中，表格的每一行都用 <row> 标签定义。F
19. 在 HTML 中，可以通过设置 <a> 标签的 target 属性为 "\_blank" 来在新标签页打开链接。T
20. 编程题

用表格等标签创建一个个人简历模板，模板实例如下图：

表格

描述已自动生成

<!DOCTYPE html>

<html lang="zh-CN">

<head>

<meta charset="UTF-8">

<title>个人简历</title>

<style>

table {

width: 80%;

border-collapse: collapse;

margin: 0 auto;

font-family: Arial, sans-serif;

}

table, th, td {

border: 1px solid black;

}

th, td {

padding: 10px;

text-align: center;

}

.photo {

width: 100px;

height: 100px;

}

.photo img {

width: 100%;

height: 100%;

}

</style>

</head>

<body>

<h2 style="text-align: center;">个人简历</h2>

<table>

<tr>

<td>姓名</td>

<td>张建安</td>

<td>年龄</td>

<td>21</td>

<td>性别</td>

<td>男</td>

<td rowspan="6" class="photo"><img src="./图片.jpg" alt="个人照片"></td>

</tr>

<tr>

<td>籍贯</td>

<td>山西</td>

<td>民族</td>

<td>汉族</td>

<td>学历</td>

<td>本科</td>

</tr>

<tr>

<td>毕业院校</td>

<td colspan="3">山西农大</td>

<td>专业</td>

<td>软件工程</td>

</tr>

<tr>

<td>联系方式</td>

<td colspan="5">13033458033</td>

</tr>

<tr>

<td>现居住地</td>

<td colspan="5">山西省晋中市榆次区乌金山路258号山西工程科技职业大学</td>

</tr>

<tr>

<td>个人技能</td>

<td colspan="5">Vue、Java、SpringBoot</td>

</tr>

<tr>

<td>主修课程</td>

<td colspan="6">计算机组成原理、操作系统、计算机网络、Vue框架设计、Linux操作系统、操作系统</td>

</tr>

<tr>

<td>兴趣爱好</td>

<td colspan="6">写代码</td>

</tr>

<tr>

<th>学习经历</th>

<td>起止日期</td>

<td colspan="3">学校名称</td>

<td>专业</td>

<td>学历</td>

</tr>

<tr>

<td>本科</td>

<td>2021.9-2024.7</td>

<td colspan="3">山西农业大学</td>

<td>软件学院</td>

<td>本科</td>

</tr>

<tr>

<td>本科</td>

<td>2021.9-2024.7</td>

<td colspan="3">山西农业大学</td>

<td>软件学院</td>

<td>本科</td>

</tr>

<tr>

<th>工作实践经历</th>

<td>起止日期</td>

<td>单位名称及职位</td>

<td>主要职责</td>

<td colspan="3">备注</td>

</tr>

<tr>

<td>无</td>

<td>xxx</td>

<td>xxx</td>

<td>xxx</td>

<td colspan="3">xxx</td>

</tr>

<tr>

<td></td>

<td>xxx</td>

<td>xxx</td>

<td>xxx</td>

<td colspan="3">xxx</td>

</tr>

<tr>

<th rowspan="2">自我评价</th>

<td colspan="6">aaaaaaaaa</td>

</tr>

<tr>

<td colspan="6">aaaaaaa</td>

</tr>

<tr>

<th rowspan="2">其他说明</th>

<td colspan="6">xxxxxxxxxxxxxxxxxx</td>

</tr>

<tr>

<td colspan="6">xxxxxxxxxxxxxxxxxx</td>

</tr>

</table>

</body>

</html>

1. 简答题
2. 什么是 alt 属性？它在 HTML 中的作用是什么？

答：

alt 属性是 HTML 中用于为图像元素（<img>）提供替代文本的属性。它的主要作用包括：

图像无法加载时的替代文本：当图像由于某种原因无法加载时，浏览器会显示 alt 属性中提供的文本。这对于用户体验非常重要，使得页面内容仍然可以理解。

提高可访问性：屏幕阅读器等辅助技术会读取 alt 属性中的文本，帮助有视觉障碍的用户理解图像内容。因此，使用合适的 alt 文本可以大大提高网页的可访问性。

SEO 优化：搜索引擎会抓取 alt 属性中的内容，帮助他们更好地理解图像的内容，从而提高网页在搜索结果中的排名。

一个简单的例子如下：

<img src="image.jpg" alt="描述图像内容的替代文本">

在这个例子中，如果图像 image.jpg 无法加载，或者用户使用屏幕阅读器访问网页，他们会看到 alt 属性中的文本 “描述图像内容的替代文本”。

1. 请解释 HTML 中的 id 和 class 属性的区别。

答：id：

唯一性：每个 id 在同一个 HTML 文档中必须是唯一的。这意味着一个 id 值只能被一个元素使用。

用法：主要用于需要唯一标识的元素，比如DOM 操作，或者 CSS 中的样式。

语法：id 属性的值可以包含字母、数字、连字符（-）、下划线（\_），但不能以数字开头。

选择器：在 CSS 中，用#来选择具有特定 id 的元素，例如 #aaa。

Class：

复用性：class 属性的值可以在同一个 HTML 文档中的多个元素中重复使用。这意味着多个元素可以共享同一个 class。

用法：主要用于分组和分类元素，以便对多个元素应用相同的样式或进行相同的操作。

语法：class 属性的值可以包含字母、数字、连字符（-）和下划线（\_），可以通过空格分隔多个类名。

选择器：在 CSS 中，用点.来选择具有特定 class 的元素，例如 .javaclass。

区别：

唯一性：id 是唯一的，一个 id 只能用于一个元素；而 class 是可以复用的，可以用于多个元素。

选择器：CSS 中使用 # 来选择 id，使用 . 来选择 class。

用途：id 主要用于唯一元素的标识和操作，class 主要用于分组和应用共享样式或功能。

作 业

1. 填空题
2. 用于将样式应用于所有元素的CSS选择器是 \*(通配符选择器) 。
3. 用于设置元素背景颜色的属性是 background-color 。
4. 如何选择所有 <section> 元素内部的所有 <div> 元素section div(交集选择器)。
5. 用于在元素的边框和内容之间创建空间的属性是 padding 。
6. text-align 属性值 right 用于实现什么效果 文本右对齐 。
7. 如何在CSS中指定加粗字体？ font-weight: bold; 。
8. 用于控制字体大小的CSS属性是 font-size 。
9. 如何选择一个类名为 "intro" 的元素 .intro 。
10. 用于将文本转换为大写的属性是 text-transform: uppercase 。
11. CSS属性 line-height 用于控制什么？ 行高 。
12. 编程题

根据下列要求，给index.html文件中的内容添加样式。

给body和container设置背景色，并使得容器居中显示。

给大标题“第一卷 一个正直的人”设置颜色，字体为斜体，文字居中。

作者“维克多·雨果”设置文字居中，并在文字附近添加作者图标。

给正文文本添加浅红色阴影，并且每一段落首行缩进两个字符。放大段落第一个文字，将文字颜色设置为红色，段落首行字体加粗。

为第一段文本中的人名添加下划线，并且鼠标悬浮改变字体颜色，光标样式显示为一只手。

为注释部分的文本添加样式：缩小字体大小，改变字体颜色（为灰色）。第一条注释的英文全部转换为大写。为奇数条注释添加红色双实线类型的下划线。

给全文设置适当的行高和边距。

文本

描述已自动生成

注：具体颜色可以用工具拾取，或者颜色相近即可。页面效果如上图。

<!--

\* @Author: GianZhang-SXAU 90677658+gianzhang@users.noreply.github.com

\* @Date: 2024-07-01 21:59:54

\* @LastEditors: GianZhang-SXAU 90677658+gianzhang@users.noreply.github.com

\* @LastEditTime: 2024-07-02 11:18:36

\* @FilePath: \7.2作业\index.html

\* @Description: 这是默认设置,请设置`customMade`, 打开koroFileHeader查看配置 进行设置: https://github.com/OBKoro1/koro1FileHeader/wiki/%E9%85%8D%E7%BD%AE

-->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Document</title>

<style>

body {

background-color: #E7E4D5;

margin: 0;

padding: 0;

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

line-height: 1.6;

padding: 20px;

font-family: "宋体";

}

.container {

background-color: #F6F4EC;

padding: 20px;

box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);

width: 60%;

}

h1 {

color: #9D7E5D;

font-style: italic;

text-align: center;

}

h4 {

text-align: center;

position: relative;

}

h4::before {

position: absolute;

left: -25px;

}

p {

text-indent: 2em;

text-shadow: 1px 1px 2px rgba(255, 0, 0, 0.1);

margin: 10px 0;

position: relative;

}

p::first-letter {

font-size: 1.5em;

font-weight: bold;

color: red;

}

p::first-line{

font-weight: bolder;

}

span {

text-decoration: underline;

}

span:hover {

color: blue;

cursor: pointer;

}

.note p {

font-size: 0.5em;

color: gray;

}

.note p:nth-of-type(1) {

text-transform: uppercase;

}

.note p:nth-of-type(odd) {

border-bottom: 2.5px double red;

}

sup{

color: #F9CDB0;

border: 0.5px solid #F9CDB0;

border-radius: 1px;

}

</style>

</head>

<body>

<div class="container">

<h1>第一卷　一个正直的人</h1>

<h4>维克多·雨果<sup>作者</sup></h4>

<p>一八一五年，迪涅①的主教是<span>查理·佛朗沙·卞福汝·米里哀</span>先生。他是个七十五岁左右的老人；从一八○六年起，他已就任迪涅区主教的职位。</p>

<p>虽然这些小事绝不触及我们将要叙述的故事的本题，但为了全面精确起见，在此地提一提在他就任之初，人们所传播的有关他的一些风闻与传说也并不是无用的。大众关于某些人的传说，无论是真是假，在他们的生活中，尤其是在他们的命运中所占的地位，往往和他们亲身所作的事是同等重要的。米里哀先生是艾克斯法院的一个参议的儿子，所谓的司法界的贵族。据说他的父亲因为要他继承②那职位，很早，十八岁或二十岁，就按照司法界贵族家庭间相当普遍的习惯，为他完了婚。米里哀先生虽已结婚，据说仍常常惹起别人的谈论。他品貌不凡，虽然身材颇小，但是生得俊秀，风度翩翩，谈吐隽逸；他一生的最初阶段完全消磨在交际场所和与妇女们的厮混中。革命③爆发了，事变叠出，司法界贵族家庭因受到摧毁，驱逐，追捕而东奔西散了。米里哀先生，当革命刚开始时便出亡到意大利。他的妻，因早已害肺病，死了。他们一个孩子也没有。此后，他的一生有些什么遭遇呢？法国旧社会的崩溃，他自己家庭的破落，一般流亡者可能因远道传闻和恐怖的夸大而显得更加可怕的九三年①的种种悲剧，是否使他在思想上产生过消沉和孤独的意念呢？一个人在生活上或财产上遭了大难还可能不为所动，但有时有一种神秘可怕的打击，打在人的心上，却能使人一蹶不振；一向在欢乐和温情中度日的他，是否受过那种突如其来的打击呢？没有谁那样说，我们所知道的只是：他从意大利回来，就已经当了教士了。</p>

<div class="note">

<p>①迪涅（Digne）在法国南部，是下阿尔卑斯省的省会。</p>

<p>②当时法院的官职是可以买的，并可传给儿孙。</p>

<p>③指一七八九年法国资产阶级革命。</p>

<p>④一七九三年是革命达到高潮的一年。</p>

</div>

</div>

</body>

</html>

作 业

1. 选择题
2. 下列哪些CSS属性可以被子元素继承？AC
3. **color** B. border **C. margin** D. padding
4. 下列哪个不是可以继承的属性？D
5. font-size B. text-decoration C. font-weight **D. border-width**
6. 在CSS中，以下哪个属性用于控制文本的行间距？C
7. letter-spacing B. word-spacing **C. line-height** D. text-align
8. 如果希望文本在超出容器范围时显示省略号，应该使用哪个CSS属性?B
9. overflow: hidden;
10. **text-overflow: ellipsis;**
11. white-space: nowrap;
12. text-wrap: nowrap;
13. 如何实现两个div并排显示？A
14. **float: left;或者float: right;**
15. ttext-align: left;或者text-align: right;
16. display: inline-block;
17. display: block;
18. 简答题
19. 什么是 CSS 盒模型？请举例说明盒模型的计算方式。

CSS 盒模型描述了页面中的每个元素被表示为一个矩形的盒子。它包括内容区域、内边距、边框和外边距。计算盒模型大小的方式是：元素的总宽度 = width + padding-left + padding-right + border-left-width + border-right-width + margin-left + margin-right。

1. 简述元素之间的外边距合并问题，父子元素外边距合并问题如何解决？

外边距合并指相邻元素的外边距会合并为一个外边距的现象。父子元素的外边距合并可以通过以下方法解决：

给父元素添加 overflow: hidden; 或者 overflow: auto;

给父元素添加边框或内边距

1. 什么叫做高度塌陷？如何解决？

高度塌陷是指当父元素只包含浮动子元素时，父元素的高度会塌陷为零的情况。解决方法包括：

清除浮动：在父元素的最后一个浮动子元素后面添加一个空的块元素，并清除其浮动状态。

使用overflow属性：给父元素添加 overflow: hidden; 或者 overflow: auto;，强制父元素包含浮动子元素。

**作 业**

1. **选择题**
2. 以下哪种是CSS中的默认定位方式？

A. relative

B. absolute

**C. static**

D. fixed

2. 当一个元素使用position: absolute;时，它的定位基准是？

A. 窗口视口

**B. 父元素的位置**

C. 浏览器窗口的左上角

D. 文档的左上角

3. 使用position: fixed;时，元素相对于什么进行定位？

A. 元素本身的位置

B. 父元素的位置

**C. 浏览器窗口的左上角**

D. 文档的左上角

4. 以下哪种定位方式会使元素脱离文档流？

A. static

B. relative

**C. absolute**

**D. fixed**

5. 使用position: relative;时，元素的初始位置保留在文档流中，但它的偏移量是相对于？

A. 窗口视口

B. 父元素的位置

C. 浏览器窗口的左上角

**D. 自身原本在文档流中的位置**

6. 一个绝对定位的元素如果没有设置top、bottom、left、right属性，它会？

**A. 保持在原来的位置**

B. 垂直居中

C. 自动贴边

D. 不可见

7. 如何使一个元素固定在页面右下角？

**A. position: fixed; bottom: 0; right: 0;**

B. position: absolute; bottom: 0; right: 0;

C. position: relative; bottom: 0; right: 0;

D. position: static; bottom: 0; right: 0;

1. **简答题**
2. 什么是z-index？

**CSS 中的一个属性，用于控制元素的堆叠顺序（stacking order）。元素的堆叠顺序决定了元素在 z 轴上的显示顺序，即哪个元素会覆盖在其他元素的上方。z-index 只能应用于定位元素（position 值为 relative, absolute, fixed, 或 sticky）。**

1. 在一个父元素div中，如何让一个子元素div水平居中，垂直居中？

方法一：使用 Flexbox

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

.parent {

display: flex;

justify-content: center;

align-items: center;

height: 100vh; /\* 让父元素的高度充满视口高度 \*/

}

.child {

width: 100px;

height: 100px;

background-color: lightblue;

}

</style>

<title>aaa</title>

</head>

<body>

<div class="parent">

<div class="child"></div>

</div>

</body>

</html>

方法二：使用 Grid

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

.parent {

display: grid;

place-items: center;

height: 100vh; /\* 让父元素的高度充满视口高度 \*/

}

.child {

width: 100px;

height: 100px;

background-color: lightcoral;

}

</style>

<title>aaa</title>

</head>

<body>

<div class="parent">

<div class="child"></div>

</div>

</body>

</html>

方法三：使用绝对定位和变换

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

.parent {

position: relative;

height: 100vh; /\* 让父元素的高度充满视口高度 \*/

}

.child {

position: absolute;

top: 50%;

left: 50%;

transform: translate(-50%, -50%);

width: 100px;

height: 100px;

background-color: lightgreen;

}

</style>

<title>aaa</title>

</head>

<body>

<div class="parent">

<div class="child"></div>

</div>

</body>

</html>

1. **编程题**

重构网易严选(https://you.163.com/)顶部导航。



<!DOCTYPE html>

<html lang="zh-CN">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>网易严选</title>

    <link rel="stylesheet" href="styles.css">

</head>

<body>

    <nav class="navbar">

        <div class="navbar-brand">

            <a href="#">网易严选</a>

        </div>

        <div class="navbar-menu">

            <a href="#">首页</a>

            <a href="#">新品</a>

            <a href="#">热卖</a>

            <a href="#">分类</a>

            <a href="#">专题</a>

            <a href="#">

                <svg t="1720150291156" class="icon" viewBox="0 0 1024 1024" version="1.1" xmlns="http://www.w3.org/2000/svg" p-id="1578" width="16" height="16"><path d="M341.333333 896H256V128h512v768H341.333333z m0-85.333333h341.333334V213.333333H341.333333v597.333334z m85.333334-42.666667v-85.333333h170.666666v85.333333h-170.666666z" fill="#444444" p-id="1579"></path></svg>

                APP

            </a>

        </div>

    </nav>

</body>

</html>

**作 业**

1. **简答题**
2. 什么是弹性布局？容器属性有哪些？项目属性有哪些？

弹性布局是一种用于布局用户界面元素的 CSS 布局模型，旨在提供一种更加高效、简洁和一致的方法来排列和分布容器内的子元素。

容器属性

display: 定义 flex 容器。

display: flex; 表示将元素设为块级 flex 容器。

display: inline-flex; 表示将元素设为行内级 flex 容器。

flex-direction: 定义主轴的方向（即项目的排列方向）。

row (默认): 从左到右排列项目。

row-reverse: 从右到左排列项目。

column: 从上到下排列项目。

column-reverse: 从下到上排列项目。

flex-wrap: 定义项目是否换行。

nowrap (默认): 不换行，所有项目在一行内排列。

wrap: 换行，项目在必要时会换行。

wrap-reverse: 换行，但行的排列方向为反向。

flex-flow: 是 flex-direction 和 flex-wrap 的简写。

例如：flex-flow: row wrap;

justify-content: 定义主轴上的对齐方式。

flex-start (默认): 项目从主轴的起点开始。

flex-end: 项目从主轴的终点开始。

center: 项目在主轴上居中。

space-between: 项目之间的间隔相等。

space-around: 项目周围的间隔相等。

align-items: 定义交叉轴上的对齐方式。

stretch (默认): 项目被拉伸以适应容器。

flex-start: 项目在交叉轴的起点对齐。

flex-end: 项目在交叉轴的终点对齐。

center: 项目在交叉轴上居中对齐。

baseline: 项目的基线对齐。

align-content: 定义多根轴线的对齐方式。仅在有多根轴线时使用。

stretch (默认): 轴线被拉伸以适应容器。

flex-start: 轴线在交叉轴的起点对齐。

flex-end: 轴线在交叉轴的终点对齐。

center: 轴线在交叉轴上居中对齐。

space-between: 轴线之间的间隔相等。

space-around: 轴线周围的间隔相等。

**项目是 flex 容器的直接子元素。**

order: 定义项目的排列顺序。数值越小，项目排列越靠前。默认值为 0。

例如：order: 1;

flex-grow: 定义项目的放大比例。默认值为 0 (即如果存在剩余空间，也不放大)。

例如：flex-grow: 1; 表示项目会占据剩余空间。

flex-shrink: 定义项目的缩小比例。默认值为 1 (即如果空间不足，项目会缩小)。

例如：flex-shrink: 0; 表示项目不会缩小。

flex-basis: 定义项目的初始大小。可以是长度单位（如 px, em, %）或关键字 auto。默认值为 auto。

例如：flex-basis: 100px;

flex: 是 flex-grow, flex-shrink 和 flex-basis 的简写。

例如：flex: 1 1 auto;

align-self: 定义单个项目在交叉轴上的对齐方式，覆盖 align-items 属性。

auto (默认): 继承父容器的 align-items 属性。

flex-start: 项目在交叉轴的起点对齐。

flex-end: 项目在交叉轴的终点对齐。

center: 项目在交叉轴上居中对齐。

baseline: 项目的基线对齐。

stretch: 项目被拉伸以适应容器。

1. 在一个父元素div中，如何用flex布局让一个子元素div水平居中，垂直居中？

父元素 div 的 CSS 设置：

将 display 属性设置为 flex，使父元素成为一个 flex 容器。

使用 justify-content 属性将子元素水平居中。

使用 align-items 属性将子元素垂直居中。

子元素 div 的 CSS 设置：

应用父元素的 flex 布局属性。

比如：

.parent {

display: flex; /\* 将父元素设为flex容器 \*/

justify-content: center;/\* 水平居中 \*/

align-items: center; /\* 垂直居中 \*/

height: 100vh; /\* 父元素高度设为视口高度，便于观察效果 \*/

}

.child {

/\* 子元素可以有自己的样式设置 \*/

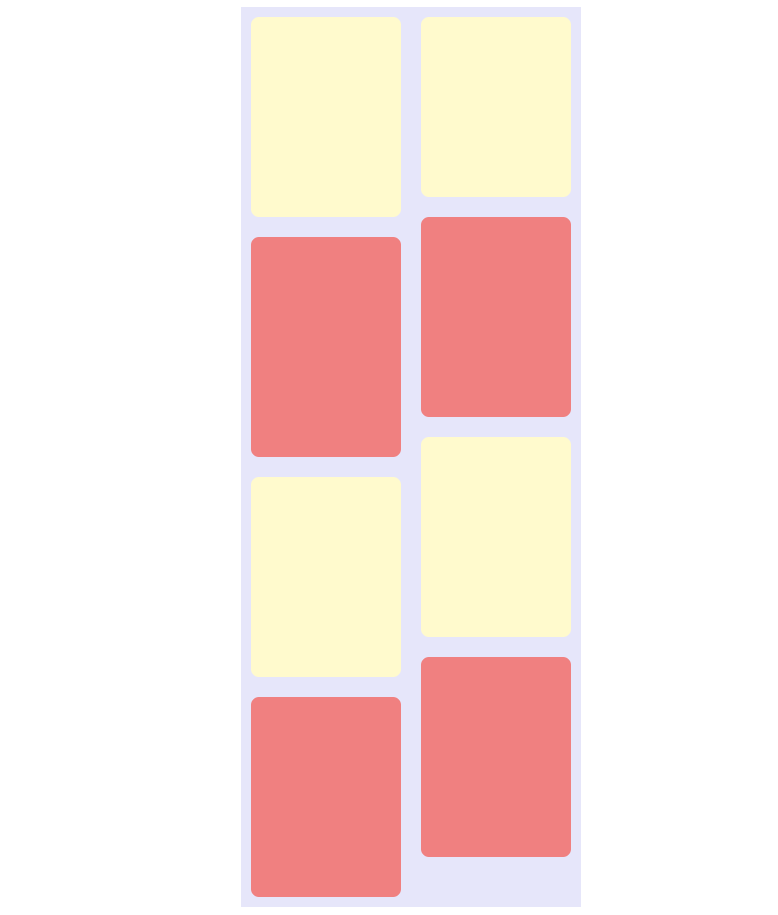
width: 100px;

height: 100px;

background-color: lightblue;

}

1. **编程题**
2. 使用弹性布局实现以下效果。



<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<style>

body {

display: flex;

justify-content: center;

align-items: center;

height: 100vh;

background-color: #E6E6FA;

margin: 0;

}

.container {

display: grid;

grid-template-columns: repeat(2, 1fr);

grid-auto-rows: 100px;

gap: 10px;

width: 250px;

}

.item {

background-color: #FFFACD;

border-radius: 5%;

}

.item:nth-child(even) {

background-color: #F08080;

}

.item:nth-child(3),

.item:nth-child(6) {

grid-row: span 2;

}

</style>

</head>

<body>

<div class="container">

<div class="item"></div>

<div class="item"></div>

<div class="item"></div>

<div class="item"></div>

<div class="item"></div>

<div class="item"></div>

<div class="item"></div>

<div class="item"></div>

</div>

</body>

</html>

1. 使用border属性，绘制出一个直角三角形。

形状

中度可信度描述已自动生成

<!--

\* @Author: 张建安 337587856@qq.com

\* @Date: 2024-07-08 21:40:11

\* @LastEditors: 张建安 337587856@qq.com

\* @LastEditTime: 2024-07-08 23:06:50

\* @FilePath: \实训\7.8作业\T2.html

\* @Description: 这是默认设置,请设置`customMade`, 打开koroFileHeader查看配置 进行设置: https://github.com/OBKoro1/koro1FileHeader/wiki/%E9%85%8D%E7%BD%AE

-->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>T2</title>

<style>

.right-angle-triangle {

width: 0;

height: 0;

border-bottom: 100px solid #F62AF6;

border-right: 100px solid transparent;

}

</style>

</head>

<body>

<div class="right-angle-triangle"></div>

</body>

</html>

1. 使用flex布局，实现骰子的六个面。（1-6）

例如： 5

形状

描述已自动生成

<!--

\* @Author: 张建安 337587856@qq.com

\* @Date: 2024-07-08 21:40:11

\* @LastEditors: 张建安 337587856@qq.com

\* @LastEditTime: 2024-07-08 23:06:50

\* @FilePath: \实训\7.8作业\T3.html

\* @Description: 这是默认设置,请设置`customMade`, 打开koroFileHeader查看配置 进行设置: https://github.com/OBKoro1/koro1FileHeader/wiki/%E9%85%8D%E7%BD%AE

-->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>骰子</title>

<style>

body {

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

height: 100vh;

background-color: #E6E6FA;

}

.dice {

display: flex;

flex-wrap: wrap;

width: 200px;

height: 200px;

gap: 20px;

}

.dot {

width: 50px;

height: 50px;

background-color: #F08080;

border-radius: 0%;

}

.hidden {

visibility: hidden;

}

.slider {

margin-top: 30px;

}

</style>

</head>

<body>

<header>这是一个骰子</header>

<div class="dice">

<div class="dot hidden"></div>

<div class="dot hidden"></div>

<div class="dot hidden"></div>

<div class="dot hidden"></div>

<div class="dot hidden"></div>

<div class="dot hidden"></div>

<div class="dot hidden"></div>

<div class="dot hidden"></div>

<div class="dot hidden"></div>

</div>

<a>1<input type="range" min="1" max="6" value="1" class="slider" id="diceSlider">6</a>

<script>

const slider = document.getElementById('diceSlider');

const dots = document.querySelectorAll('.dot');

const patterns = [

[4], // 1

[0, 8], // 2

[0, 4, 8], // 3

[0, 2, 6, 8], // 4

[0, 2, 4, 6, 8], // 5

[0, 1, 2, 6, 7, 8] // 6

];

slider.addEventListener('input', () => {

const value = slider.value;

dots.forEach(dot => dot.classList.add('hidden'));

patterns[value - 1].forEach(index => dots[index].classList.remove('hidden'));

});

// 初始化

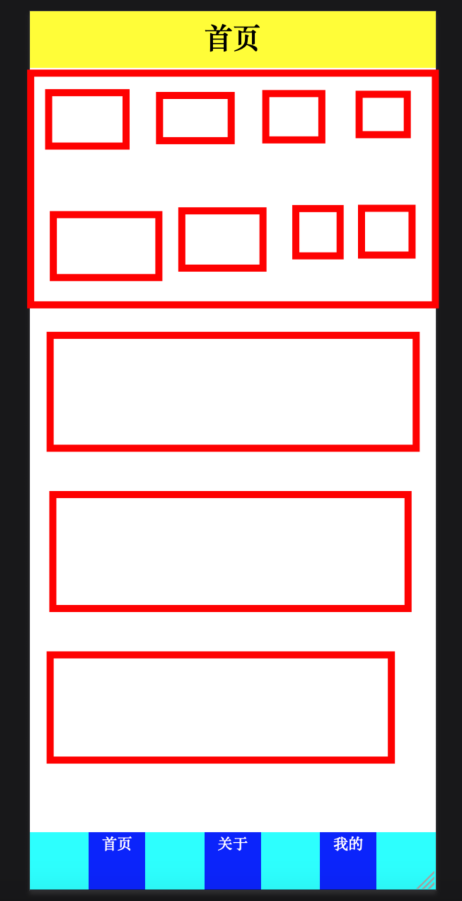
slider.dispatchEvent(new Event('input'));

</script>

</body>

</html>

1. （优先）实现手机app首页布局，如下图。



<!--

\* @Author: 张建安 337587856@qq.com

\* @Date: 2024-07-08 21:33:29

\* @LastEditors: 张建安 337587856@qq.com

\* @LastEditTime: 2024-07-08 21:34:59

\* @FilePath: \实训\7.8作业\index.html

\* @Description: 这是默认设置,请设置`customMade`, 打开koroFileHeader查看配置 进行设置: https://github.com/OBKoro1/koro1FileHeader/wiki/%E9%85%8D%E7%BD%AE

-->

<!DOCTYPE html>

<html lang="zh-CN">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>首页</title>

<style>

body {

margin: 0;

font-family: Arial, sans-serif;

}

.header {

background-color: yellow;

text-align: center;

padding: 10px;

font-size: 24px;

}

.container {

padding: 10px;

}

.grid-container {

display: grid;

grid-template-columns: repeat(5, 1fr);

gap: 10px;

}

.grid-container .item {

background-color: white;

border: 2px solid red;

height: 80px;

}

.grid-row {

display: grid;

grid-template-columns: repeat(3, 1fr);

gap: 10px;

margin-top: 10px;

}

.grid-row .item {

background-color: white;

border: 2px solid red;

height: 100px;

}

.box {

background-color: white;

border: 2px solid red;

height: 150px;

margin-top: 10px;

}

.footer {

display: flex;

justify-content: space-around;

background-color: cyan;

padding: 10px 0;

position: fixed;

width: 100%;

bottom: 0;

}

.footer div {

flex: 1;

text-align: center;

padding: 10px 0;

background-color: blue;

color: white;

}

</style>

</head>

<body>

<div class="header">首页</div>

<div class="container">

<div class="grid-container">

<div class="item"></div>

<div class="item"></div>

<div class="item"></div>

<div class="item"></div>

<div class="item"></div>

</div>

<div class="grid-row">

<div class="item"></div>

<div class="item"></div>

<div class="item"></div>

</div>

<div class="box"></div>

<div class="box"></div>

<div class="box"></div>

</div>

<div class="footer">

<div>首页</div>

<div>关于</div>

<div>我的</div>

</div>

</body>

</html>