Day15笔记

作业！

1.响应式页面续写就写三大块部分，输入部分如果写出来的前5人奖

2.tmooc的作业

3.下载boot的素材在今天的day15中有一个boot.zip请重新下载

9.透视效果

有钟表的建筑的塔

低可信度描述已自动生成

(1)透视

在 css 变换中如果想作出 3d 效果，一定要关注透视距离 perspective 属性。

perspective 属性值越大，元素的变形就越小。

perspective 属性值越小，元素的变形就越大。

卡通画

中度可信度描述已自动生成

- perspective 属性定义透视距离，距离为长度单位

- 模拟人眼睛到 3D 变化元素之间的距离

- 【重点】透视距离只能定义在 3D 变化的父元素上

(2) transform-style

图片包含 游戏机

描述已自动生成3D元素构建是指某个图形是由多个元素构成的，可以给这些元素的父元素设置transform-style: preserve-3d来使其变成一个真正的3D图形。

- 让盒子位于三维空间里transform-style: preserve-3d;

- 让子盒子位于此元素所在的平面内(默认)transform-style: flat;

(3)旋转函数 x 轴和 y 轴

形状

描述已自动生成2d 中旋转是 z 轴旋转，而想要感受 3d 效果，需要旋转的 x 轴和 y 轴。

- `transform:rotateX(deg);`x 轴旋转

- `transform:rotateY(deg);`y 轴旋转

图形用户界面

中度可信度描述已自动生成

图片包含 表面图

描述已自动生成(4)位移函数 z 轴

3d变换中的移动，z中的正方向面向用户,值越大，越靠近用户。

* transform: translateZ(正值) 沿z轴向前（接近模拟视觉的距离）
* transform: translateZ(负值) 沿z轴向前（远离模拟视觉的距离）

【练习】

文字图案

低可信度描述已自动生成让扑克牌旋转起来具有正反两个面

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <meta charset="UTF-8">  <style>  .box {  width: 300px;  height: 300px;  border: 1px solid black;  margin: auto;  perspective: 300px;  }  .box:hover .pk {  transform: rotateY(360deg);  }  .pk {  border: 1px solid blue;  width: 205px;  height: 289px;  margin: auto;  position: relative;  transition: 3s;  transform-style: preserve-3d;  }  .pk>div {  width: 205px;  height: 289px;  position: absolute;  }  .a {  background-image: url(./扑克a.jpg);  }  .b {  background-image: url(./扑克b.jpg);  transform: translateZ(-1px) scaleY(-1);  }  </style>  </head>  <body>  <!-- 205\*289 -->  <div class="box">  <div class="pk">  <div class="a"></div>  <div class="b"></div>  </div>  </div>  </body>  </html> |

二十四、动画

动画就是指定一组或多组成套的动作，按照指定时间，指定的方式自动完成。h5 中动画的运用效率要高于使用 js 来体现动画效果，因为动画是渲染式的。

1.关键帧

`@keyframes` 是 css 中的@规则，通过在动画序列中定义关键帧的样式，来控制 CSS 动画序列中的步骤。

以百分比来规定改变发生的时间，或者通过关键词 "from" 和 "to"，等价于 0% 和 100%。

0% 是动画的开始时间，100% 动画的结束时间。

为了获得最佳的浏览器支持，应该始终定义 0% 和 100% 选择器。

|  |
| --- |
| @keyframes move {  0% {transform: translate(0);}  100% {transform: translate(600px);}  } |

2.动画名称

`animation-name`属性就是指定动画要使用哪一个关键帧。

|  |
| --- |
| div {  width: 100px;  height: 100px;  background-color: red;  /\*这个元素使用关键帧\*/  animation-name: move;  }  @keyframes move {  0% {  transform: translate(0);  }  100% {  transform: translate(600px);  }  } |

3.动画持续时间

`animation-duration`属性代表一个动画周期的时长，单位为秒(s)或者毫秒(ms)，默认值 0 秒。

|  |
| --- |
| div {  /\*这个元素使用关键帧\*/  animation-name: move;  /\* 动画时长 \*/  animation-duration: 1s;  }  @keyframes move {  0% {  transform: translate(0);  }  100% {  transform: translate(600px);  }  } |

4.动画的运动方式

`animation-timing-function`属性跟`transition-timing-function`属性就是过渡的运动方式类似。同样也具有封装好的方式和贝塞尔曲线的方式。

- ease;默认

- ease-in;

- ease-out;

- ease-in-out;

- linear;

- steps(数值, 定位) 定位：start/end 默认 end 指逐步运动

图片包含 图形用户界面

描述已自动生成

【练习】

> 使用 steps()函数完成，逐帧完成动画效果

> loding 图标分为四段和八段完成帧动画

|  |
| --- |
| div {  width: 30px;  height: 30px;  margin: 20px auto;  animation-name: zhuan;  animation-duration: 1s;  }  @keyframes zhuan {  0% { transform: rotate(0deg);}  100% {transform: rotate(360deg);}  }  .a1 {  animation-timing-function: linear;  } /\*平滑的旋转\*/  .b1 {  animation-timing-function: linear;  } /\*平滑的旋转\*/  .a2 {  animation-timing-function: steps(4);  } /\*断续旋转分4段\*/  .b2 {  animation-timing-function: steps(8);  } /\*断续旋转分8段\*/ |

5.动画的延迟时间

`animation-delay`属性动画的延迟时间和之前过渡的延迟时间一样使用

|  |
| --- |
| div {  /\*这个元素使用关键帧\*/  animation-name: move;  /\* 动画时长 \*/  animation-duration: 1s;  /\* 运动方式 \*/  animation-timing-function: linear;  /\* 动画延迟时间 \*/  animation-delay: 2s;  } |

6.结束状态

在动画运行到某个位置的时候，动画停止，元素默认会迅速回到起始位置

- `animation-fill-mode`:设置动画结束时盒子的状态

- 属性值：`forwards`保持动画结束后的状态

- 属性值：`backwards`动画结束后回到最初的状态

7.动画化执行顺序

animation-direction 属性是动画的执行顺序

- 属性值：normal 正向，默认值

- 属性值：reverse 反向

8.动画循环次数

- `animation-iteration-count`属性主要用来定义动画的播放次数。

- 其值通常为整数，但也可以使用带有小数的数字，其默认值为 1，这意味着动画将从开始到结束只播放一次。

- 如果取值为`infinite`，动画将会无限次的播放。

9.简写方式

- animation: 动画执行时间 延迟时间 执行关键帧名称 运动方式 运动次数 结束状态;

- 最简方式 animation: 动画执行时间 执行关键帧名称;

- 执行时间和延迟时间顺序不可调整

- `animation: 2s 3s move linear 1 forwards reverse;`

10.动画停止

- animation-play-state 属性规定动画是否正在运行或暂停。

- 属性值：running 运动的，默认值

- 属性值：paused 暂停的

|  |
| --- |
| .box:hover .hsl {  animation-play-state: running;  animation-play-state: paused;  background-color: blue;  } |

【练习】

> 学子商城滚动态

> 使用学子商城的图片，文字尽量写两行以内，保持一致行数

> 鼠标移入滚动区域是停止滚动，移开恢复滚动

图形用户界面

描述已自动生成

11.开源 CSS 动画库

- animate.css 下载下载:http://www.animate.net.cn/

- 元素加入需要的类名

- 当前元素使用 animate 属性，属性值为类名和运行时间

- 需要在元素先加上 `animate\_\_animated` 类，之后在加入其他类

|  |
| --- |
| <head>  <link rel="stylesheet" href="./animate.min.css" />  <style>  div {  width: 200px;  height: 100px;  background-color: red;  animation: 1s animate\_\_rubberBand;  }  </style>  </head>  <body>  <div class="animate\_\_rubberBand">元素</div>  <div class="animate\_\_animated animate\_\_shakeX" id="div2">自动调用</div>  </body> |

二十五、响应式布局

1.多终端开发

- 现今需求从 pc 增加到移动端，移动端的发展迅速

- 多种规格的终端产生 pad，各种手机，手表，三维立体影像

- 用户在操作时的便捷，比如手机很小按钮的大小比例

2.响应式布局的产生

意在实现不同屏幕分辨率的终端上浏览网页的不同展示方式。通过响应式设计能使网站在多种终端上有更好的浏览阅读体验。（参考站长素材 https://sc.chinaz.com/moban/）

- 响应式也叫自适应网页，可以根据“浏览设备”的不同改变布局样式和文本及图片的效果。

- 同时要考虑，设备的屏幕横置等因素。

3.视口的设置

- 视口（viewport）可以理解为浏览器窗口，不包含浏览器的内容。使用视口约束浏览器大小，让内容区域完全展示在用户面前。

- 响应式和移动端页面必须设置视口，pc 看不出效果

`<meta name="viewport" content="属性名=属性值">`视口的设置通过 content 属性设置需要的值，如果 content 后面有多个属性，则属性与属性之间用英文逗号隔开

|  |
| --- |
| <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, user-scalable=0"> |

- width=device-width 视口宽度为设备宽度（device 设备）

- initial-scale=1.0 初始化视口不缩放

- maximum-scale=1.0 最大是缩放倍率（不缩放）

- user-scalable=0 用户不能缩放

- 简写方式`<meta name="viewport" content="width=device-width, initial-scale=1.0">`

4.注意事项

- 多用流式布局（文档流+浮动）和弹性布局，少用定位

- 文本，图片，尺寸尽量写相对单位，少用绝对单位

- 复杂页面不适和使用适应响应式

5.媒体查询

- media 媒体类型，浏览网页的设备

- screen：pc/pad/phone

- TV

- print： 打印机，部分手表，冰箱的显示屏等

- all： 所有设别

- 屏幕尺寸（从 bootstrap 中遍历出来的，min-width）

- 超小屏幕 (max-width:576px)

- 小屏幕 (min-width: 576px) and (max-width: 768px)

- 中屏幕 (min-width: 768px) and (max-width: 992px)

- 大屏幕 (min-width: 992px) and (max-width: 1200px)

- 超大屏幕 (min-width: 1200px)

总结语法：

```css

@media 媒体类型 and (尺寸范围) {

样式列表1

样式列表2

样式列表n

样式列表……

}

或者

@media (尺寸范围) {

样式列表1

样式列表2

样式列表n

样式列表……

}

```

根据以上的屏幕尺寸，改变 body 的颜色

|  |
| --- |
| /\* 注意空格 注意and \*/  @media screen and (min-width: 576px) and (max-width: 768px) {  body {  background-color: red;  }  } |

6.Viewport Width 和 Viewport Height

(1)vw 和 vh

- vw（Viewport Width）、vh(Viewport Height)是基于视图窗口的单位。

图片包含 图示

描述已自动生成- “视区”所指为浏览器内部的可视区域大小，不包含任务栏标题栏以及底部工具栏的浏览器区域大小。

- vw:1vw 等于视口宽度的 1%

- vh:1vh 等于视口高度的 1%

(2)兼容性

- chrome 20

- firefox 19

- IE 9.0

- opera 20.0

- safari 6.0

(3)优势

- % 是相对于父元素的大小设定的比率， vw、 vh 是视窗大小决定的。

- vw、 vh 优势在于能够直接获取高度，而用 % 在没有设置 body 高度的情况下，是无法正确获得可视区域的高度的。

- 元素会根据视口的大小设定宽高,切换手机型号查看效果

![图形用户界面, 网站

描述已自动生成]()