

20230913

一、文本属性

- 1.文本阴影和描边
- 2.文本溢出
- 3.多列分栏

二、元素模糊

三、网络字体

- 1.网络字体
- 2.字体图标

四、过渡

五、变换

- 1.平移
- 2.旋转
- 3.缩放
- 4.扭曲
- 5.矩阵

六、动画

七、弹性盒子

练习题：

20230913

一、文本属性

1.文本阴影和描边

```
/* 文本阴影 */
#id_p_02{
    text-shadow: 5px 5px 2px red;
}
/* 文本描边 */
#id_p_03{
    font-size: 30px;
    -webkit-text-stroke: 1px red;
}
```

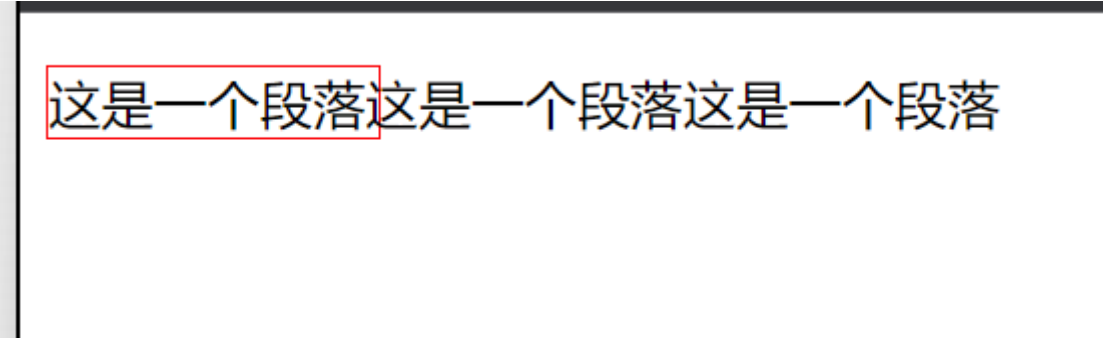
这是一个段落2

这是一个段落3

2.文本溢出

overflow

```
<!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>
    #id_p_01{
      border: red solid 1px;
      width: 100px;
      /* 当文本超出宽度时，文本会自动换行。 */
      white-space: nowrap; /* 设置为不换行 */
      overflow: hidden; /* 使溢出文本隐藏 */
      text-overflow: ellipsis; /* 将溢出文本以省略号形式展示 */
    }
    #id_p_02{
      border: gray solid 1px;
      width: 120px;
      height: 30px;
      /* 如果设定了高度限制，scroll可以产生滚动条 */
      overflow: scroll;
    }
  </style>
</head>
<body>
  <p id="id_p_01">这是一个段落这是一个段落这是一个段落</p>
  <p id="id_p_02">这是一个段落这是一个段落这是一个段落</p>
</body>
</html>
```

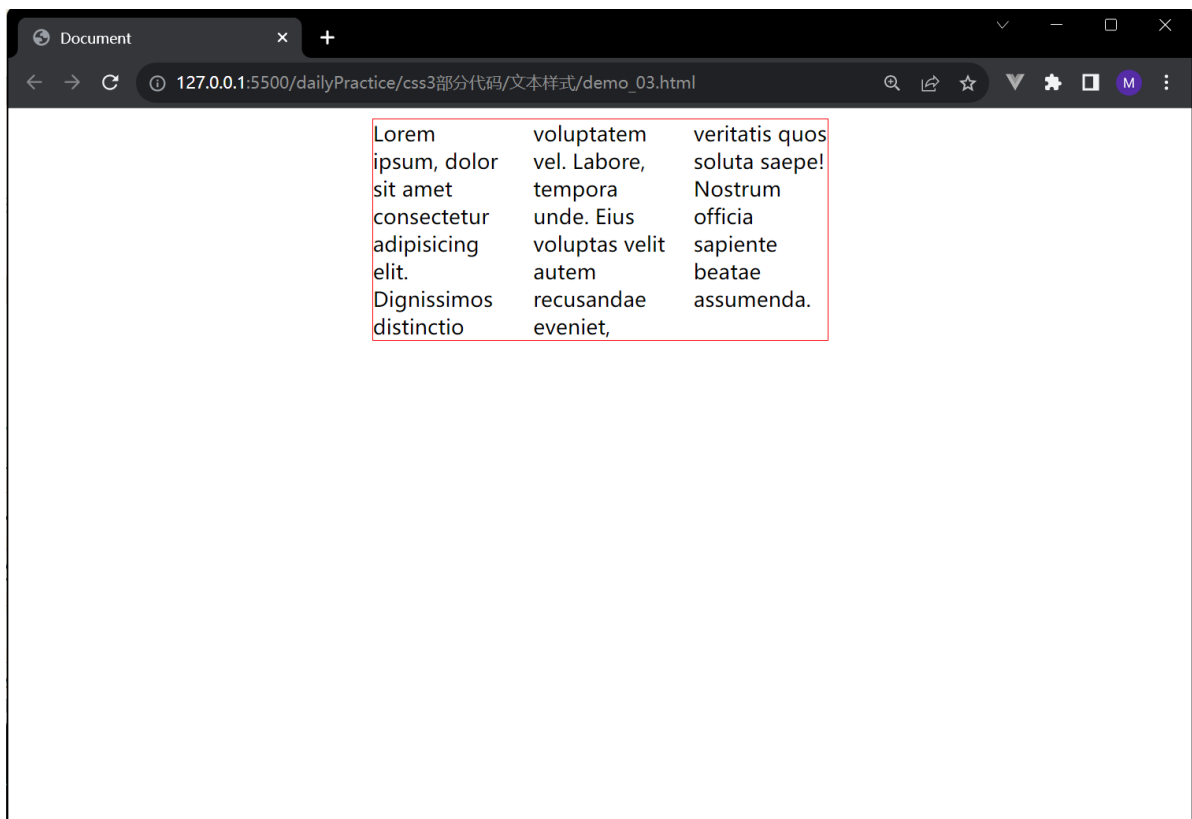


这是一个段落这是一个段落这是一个段落

3.多列分栏

```
<!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>
    #id_div_01{
      width: 350px;
      border: red solid 1px;
```

```
margin: 0 auto;
/* 分栏数量 */
column-count: 3;
/* 分栏间距 */
column-gap: 20px;
}
</style>
</head>
<body>
  <div id="id_div_01">
    Lorem ipsum, dolor sit amet consectetur adipisicing elit. Dignissimos
    distinctio voluptatem vel. Labore, tempora unde. Eius voluptas velit autem
    recusandae eveniet, veritatis quos soluta saepe! Nostrum officia sapiente beatae
    assumenda.
  </div>
</body>
</html>
```



二、元素模糊



```
<!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>
    img{
      filter: blur(5px);
    }
  </style>
</head>
<body>
  
</body>
</html>
```

三、网络字体

直接使用设计好的资源和素材



前端



> .idea

> dailyPractice



> css3部分代码



> 过渡



> 网络字体



> fonts



> icomoon



<> demo_01.html U

<> demo_02.html U

A icss.ttf U

style.css U

> 文本样式



> 新增选择器

> 元素模糊



> img



> css部分代码

> html部分代码

> img

> loginPractice



> roadProjecr

> js

> project01

> untitled



26



1.网络字体

```
<!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>
    @font-face {
      /* 自定义一种字体类型 */
      font-family: icss;
      /* 该种字体类型需要的文件 */
      src: url(icss.ttf);
    }
    p{
      font-family: icss;
    }
  </style>
</head>
<body>
  <p>这是一个段落</p>
  <p>这是一个段落2</p>
</body>
</html>
```

2.字体图标

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
  <link rel="stylesheet" type="" href="icomoon/style.css">
  <style>
    span{
      font-size: 30px;
      color: orange;
    }
  </style>
</head>
<body>
  <span class="icon-home"></span>
  <span class="icon-camera"></span>
</body>
</html>
```

四、过渡

元素从一种样式，逐渐改变为另一种样式的效果(时间)过度包含两项内容:css效果,效果持续时间

```
transition-property:all ; /* 设置参与过渡的样式,all指所有样式 */
transition-duration:5s ; /* 设置过渡时间 */
transition-delay:1s ; /* 过渡效果延迟时间 */
transition-timing-function: linear; /* 设置过渡类型 ease:加速减速 linear:匀速 */
```

五、变换

1.平移

transform translate

```
#id_div_01:hover{
/* transform 水平或垂直平移 */
/* 平移距离 translate(横轴,纵轴) */
/* transform: translate(100%,-50%); */
transform: translateY(50%);
}
```

2.旋转

rotate

变换体系三条轴

X轴: 从左至右

Y轴: 从上至下

Z轴: 从里到外

```
#id_div_01:hover{
/* 旋转 默认Z轴旋转*/
/* transform:rotate(-720deg); */
/* 可以指定哪个轴旋转 */
transform:rotateZ(-720deg);
}
```

3.缩放

scale

```
#id_div_01:hover img{
/* 缩放scale(x轴倍数,y轴倍数) */
transform:scale(0.5,0.5);
}
```

4.扭曲

skew

```
#id_div_01:hover{
  transform: skewX(20deg);
}
#id_div_02:hover{
  transform: skewY(20deg);
}
#id_div_03:hover{
  transform: skew(20deg,20deg);
}
```

5.矩阵

matrix

```
#id_div_01:hover{
  /* 矩阵变换 */
  transform:matrix(1,1,1,0,0,0);
}
```

六、动画

css3允许创建动态元素，可以部分取代网页动态图像，或js技术的动画效果

格式：

1.设置动画对象（名称可以自定义）

@keyframes动画对象名{

from{起始状态}

to{结束状态}

}

将动画对象绑定到页面元素

选择器{

...

animation: 动画对象名 时长

}

demo:

```
@keyframes act01{
  from{
    width:50px;
    border:solid 1px red;
    border-radius: 50px;
  }
  to{
    width:800px;
  }
}
```



```

background-color: orange;
font-size:50px;
border:solid 1px red;
border-radius: 400px;
}
}
#id_div_01{
width: 200px;
height:50px;
    background-color: aqua;
    animation-name:act01; /* 引入动画对象名 */
    animation-duration:5s; /* 动画执行时间 */
    animation-timing-function: ease; /* 动画执行类型 */
    /* 允许动画执行的次数,默认是1次,无限次infinite */
    animation-iteration-count: 3;
    /* 是否有反向动画 */
    animation-direction: alternate;
    /* 暂停动画 */
    animation-play-state: paused;
}

```

注意：刷新页面，自动执行动画效果

设置动画对象第二种格式

@keyframes 动画对象名{

0%{状态1}

25%{状态2}

30%{状态3}

...

100%{状态n}

}

demo:

```

<style>
@keyframes div_rect_circle{
0%{
position:absolute;
margin-left:0px ;
margin-top: 0px;
width: 100px;
height:100px;
background-color: yellow;
}
25%{
position:absolute;
margin-left:0px ;
margin-top: 0px;
width: 100px;
height:100px;
background-color: red;
border-radius: 50px;
}
}

```

```

50%{
position:absolute;
margin-left:500px ;
margin-top: 0px;
width: 100px;
height:100px;
background-color: red;
border-radius: 50px;
}
100%{
position:absolute;
margin-left:500px ;
margin-top: 300px;
width: 100px;
height:100px;
background-color: red;
border-radius: 50px;
}
}
#id_div_01{
animation: div_rect_circle 5s;
animation-timing-function: linear; /* ease 加速-减速
linear 匀速 */
}
</style>
</head>
<body>
<div id="id_div_01">这是一个div</div>
</body>

```

七、弹性盒子

css3弹性盒子（伸缩盒子/Flexible box/flexbox）弹性盒子是css3提供的一种新的布局模式。是一种当页面需要适应不同屏幕大小及设备类型时，确保元素有恰当的行为的一种布局方式 两个概念： 主轴和侧轴： 主轴为x轴， 从左至右 侧轴为y轴， 从上至下 父项和子项： 父项指外侧div 子项是内部div

demo01

```

.item1,.item2,.item3,.item4{
width: 100px;
height: 50px;
background-color: blue;
margin: 10px;
color: white;
text-align: center;
line-height: 50px;
}
.container{
width: 600px;
/* height:200px; */
height:600px;
background-color: aquamarine;
/* 添加弹性布局 */
display:flex; /* 父项添加弹性盒子属性 子项自动按主轴排列*/
/* 指定子项排列方式 四个方向 row 主轴 column 侧轴 reverse 方向 */
/* flex-direction: column-reverse; */
/* 当子项溢出,在父项外部可以看到,如果溢出页面,则无法显示 */

```

```
flex-direction: column;
/* justify-content属性 : 控制子项在主轴上所占位置(横向) */
justify-content:flex-start; /* flex-start 左对齐 center居中
flex-end右对齐 space-between 占满全部主轴空间 平均分配*/
/* align-items: 属性 : 控制子项在侧轴上所占位置(纵向) */
align-items: flex-start;
}
```

demo2:

```
.item1,.item2,.item3,.item4{
width: 100px;
height: 50px;
background-color: blue;
margin: 10px;
color: white;
text-align: center;
line-height: 50px;
}
/* order属性: 控制代码排列顺序
默认所有子项order值都为0 如果有调整,按从小到大顺序排列 */
#id_div_01{
order: 5;
}
#id_div_02{
order: 3;
}
#id_div_03{
order: -1;
}
.container{
width: 400px;
/* height:200px; */
height:400px;
background-color: aquamarine;
/* 添加弹性布局 */
display:flex; /* 父项添加弹性盒子属性 子项自动按主轴排列*/
/* flex-wrap属性默认值nowrap 自动压缩子项宽高以适应父项 wrap子项大
小不变自动换行 */
flex-wrap: wrap;
/* align-content属性 是设置了flex-wrap: wrap以后才能设置的属性,
控制换行后子项在侧轴的排列方式 flex-start/end center*/
align-content: center;
}
```

弹性盒子小节：弹性容器通过设置display属性为flex，定义为弹性容器。弹性容器包含一个或多个弹性子项！！！设置flex布局以后，子项的float，clear，vertical-align属性都将失效（弹性 浮动不共存）

练习题：

1.D 2.ABC 3.BCD 4.ABCD

HTML主要负责网页的结构和内容，而CSS则负责控制网页的外观和样式。它们一起协作，使网页开发变得更加灵活、可维护，并提供了更好的用户体验。通过将内容和样式分离，开发人员可以更容易地修改网页的外观，同时确保内容的语义结构保持不变。这两种技术是网页开发的核心组成部分，通常一起使用，以创建精美且功能丰富的网页。

`display` 属性是CSS中用来控制元素如何在页面上显示的重要属性，它可以接受不同的属性值，每个值都会影响元素的布局 and 外观。以下是一些常见的 `display` 属性值及其区别：

1. `display: block;`

- 块级元素 (block-level element) 会占据其父容器的整个宽度，独自位于一行上。
- 块级元素会在前后添加换行符，因此它们之间会有明显的垂直间距。
- 常见的块级元素包括 `<div>`, `<p>`, `<h1>` 等。

2. `display: inline;`

- 行内元素 (inline-level element) 只占据其内容所需的宽度，不会强制换行。
- 行内元素可以在同一行上并排显示，它们之间没有明显的垂直间距。
- 常见的行内元素包括 ``, `<a>`, `` 等。

3. `display: inline-block;`

- 行内块元素 (inline-block) 结合了块级元素和行内元素的特性。
- 它们占据其内容所需的宽度，但可以在同一行上并排显示，同时可以设置宽度、高度和内外边距。
- 常见的行内块元素包括 ``, `<button>` 等。

4. `display: none;`

- 元素使用 `display: none;` 不会在页面上显示，也不会占据空间。
- 这个值通常用于隐藏元素，但不会影响文档的布局。

5. `display: flex;`

- 弹性布局容器 (flex container) 用于创建弹性盒子模型，可以轻松控制内部元素的排列和对齐方式。
- 弹性容器的子元素成为弹性项，它们可以沿主轴和交叉轴排列。
- `flex` 布局通常用于响应式设计和复杂的布局需求。

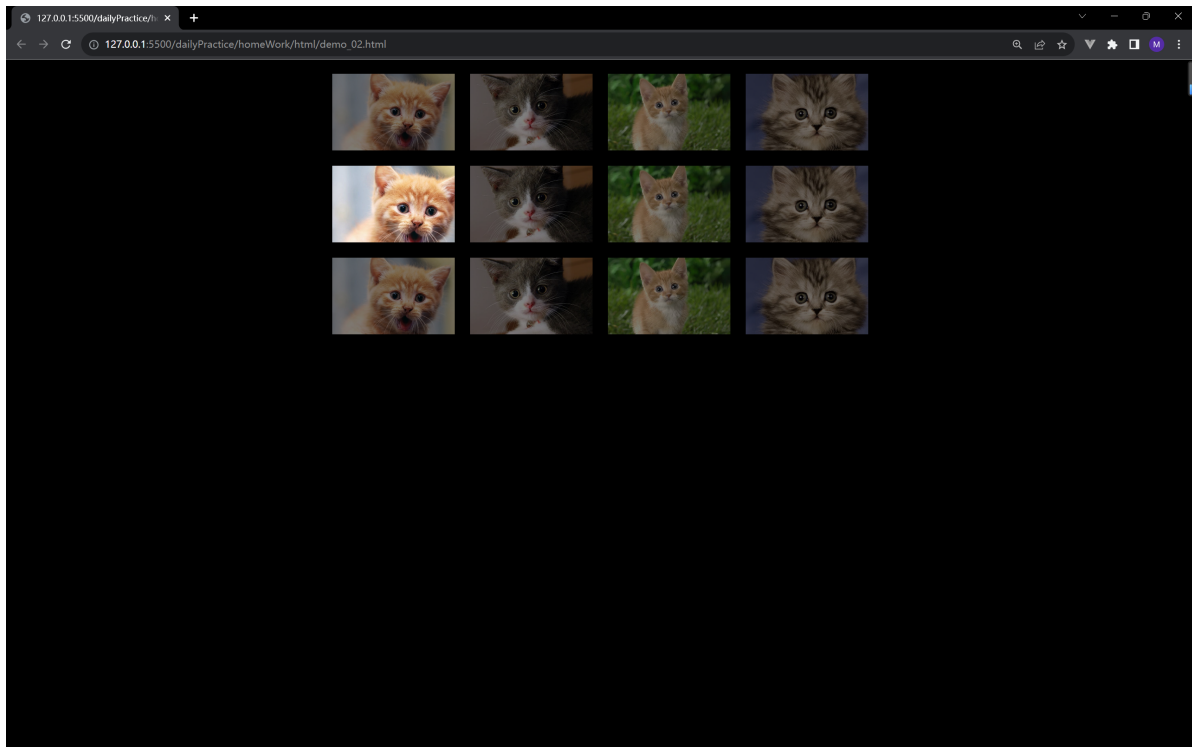
6. `display: grid;`

- 栅格布局容器 (grid container) 用于创建二维网格布局，可以将元素放置在行和列的交点上。
- 栅格布局非常适合实现复杂的网格布局，例如网页中的栏目布局。
- `grid` 布局也常用于响应式设计和复杂的布局需求。

这些 `display` 属性值具有不同的特性和用途，可以根据需要选择合适的值来实现所需的布局 and 外观效果。



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
    <style>
      @keyframes art1{
        from{
          position: absolute;bottom: 0px;
        }
        to{
          position: absolute;bottom: 15px;
        }
      }
      #id_div_1{
        background-image: url(../img/抖动的下箭头/arrow.png);
        width: 100px;
        height: 100px;
        background-repeat: no-repeat;
        position: absolute;bottom: 0px;left: 50%;
        animation: art1 1s linear;
        animation-direction: alternate;
        animation-iteration-count: infinite;
      }
    </style>
  </head>
  <body>
    <div id="id_div_1"></div>
  </body>
</html>
```



```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title></title>
    <style>
      body{
        background-color: black;
      }
      #id_div_1{
        width: 720px;
        height: 400px;
        margin: 0 auto;
      }
      #id_div_2{
        width: 160px;
        height: 100px;
        margin: 10px;
        float: left;
        opacity: 0.5;
        transition: all 2s;
      }
      .box1{
        background-image: url(../img/鼠标悬停放大清晰/img/g1.jpg);
        background-repeat: no-repeat;
        background-size: 160px;
      }
      .box2{
        background-image: url(../img/鼠标悬停放大清晰/img/g2.jpg);
        background-repeat: no-repeat;
        background-size: 160px;
      }
      .box3{
        background-image: url(../img/鼠标悬停放大清晰/img/g3.jpg);
        background-repeat: no-repeat;
        background-size: 160px;
      }
    </style>
  </head>
  <body>
    <div id="id_div_1">
      <div id="id_div_2">
        <div class="box1"></div>
        <div class="box2"></div>
        <div class="box3"></div>
      </div>
    </div>
  </body>
</html>
```

```

    }
    .box4{
        background-image: url(../img/鼠标悬停放大清晰/img/g4.jpg);
        background-repeat: no-repeat;
        background-size: 160px;
    }
    #id_div_2:hover{
        opacity: 1;
    }
    .box:hover{
        background-size: 180px;
    }
</style>
</head>
<body>
    <div id="id_div_1">
        <div id="id_div_2" class="box box1"></div>
        <div id="id_div_2" class="box box2"></div>
        <div id="id_div_2" class="box box3"></div>
        <div id="id_div_2" class="box box4"></div>
        <div id="id_div_2" class="box box1"></div>
        <div id="id_div_2" class="box box2"></div>
        <div id="id_div_2" class="box box3"></div>
        <div id="id_div_2" class="box box4"></div>
        <div id="id_div_2" class="box box1"></div>
        <div id="id_div_2" class="box box2"></div>
        <div id="id_div_2" class="box box3"></div>
        <div id="id_div_2" class="box box4"></div>
    </div>
</body>
</html>

```



```

<!DOCTYPE html>
<html lang="en">

```



```
<div class="box3"></div>
```

```
</body>
```

```
</html>
```