

CSS 高级

一、实验介绍

1.1 实验内容

学习完之前的课程，已经基本足够能够胜任对简单网页的样式设置了，但CSS还有其他一些相对高级的用法。本次实验我们就将学习这些用法。

1.2 实验知识点

- 定位
- 浮动
- 尺寸
- 导航栏
- 图片

二、实验步骤

2.1 CSS 定位

定位的基本思想很简单，它允许你定义元素框相对于其正常位置应该出现的位置，或者相对于父元素、另一个元素甚至浏览器窗口本身的位置。

CSS 有三种基本的定位机制：

普通流:

元素按照其在 HTML 中的位置顺序决定排布的过程

浮动:

浮动的框可以向左或向右移动，直到它的外边缘碰到包含框或另一个浮动框的边框为止。

绝对定位:

绝对定位使元素的位置与文档流无关，因此不占据空间。这一点与相对定位不同，相对定位实际上被看作普通流定位模型的一部分，因为元素的位置相对于它在普通流中的位置。

定位属性:

position, 将元素放在一个静态的, 相对的, 绝对的, 或固定的位置

通过对 top, left, right, bottom 这四个属性的赋值让元素向对应的方向偏移

overflow 设置元素溢出其区域发生的事情

clip 设置元素的显示形状, 多用于图片

vertical-align 设置元素的垂直对其方式

z-index 设置元素的堆叠顺序

接下来就着重来看看 position 属性:

为了形象, 我们先建立一个 html 文件和 CSS 文件

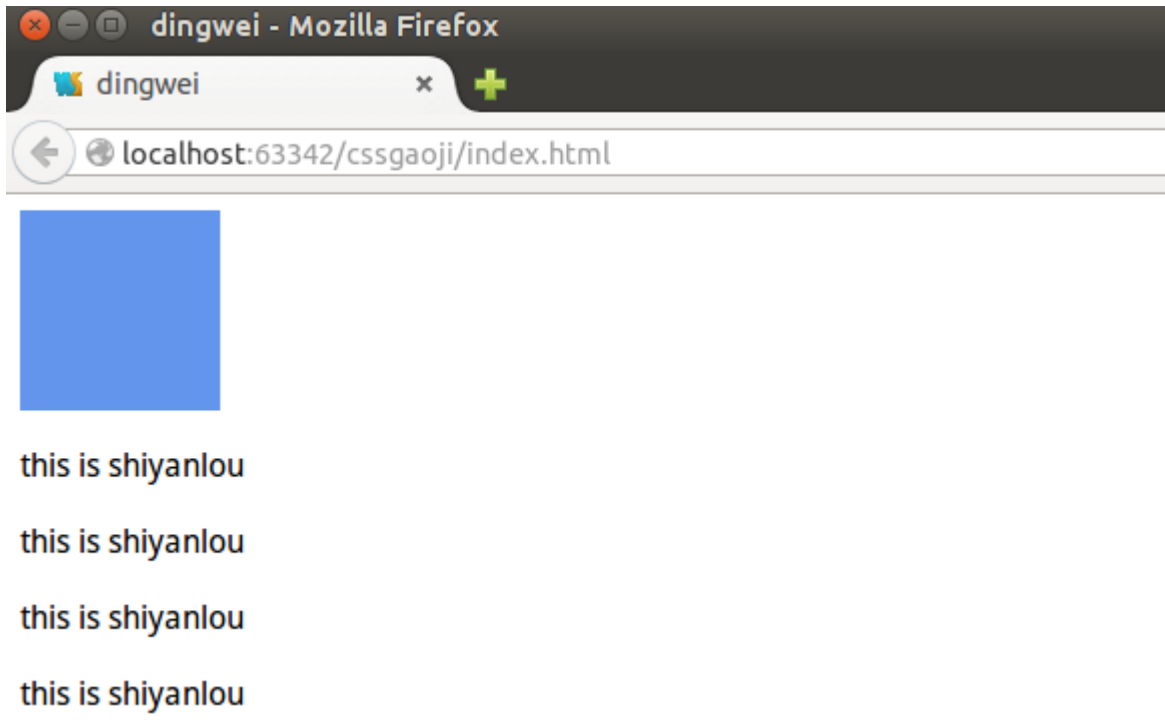
html:

```
<div class="position1"></div>
<p>this is shiyanlou</p>
<p>this is shiyanlou</p>
<p>this is shiyanlou</p>
<p>this is shiyanlou</p>
```

CSS:

```
.position1{
  width: 100px;
  height: 100px;
  background-color: cornflowerblue;
}
```

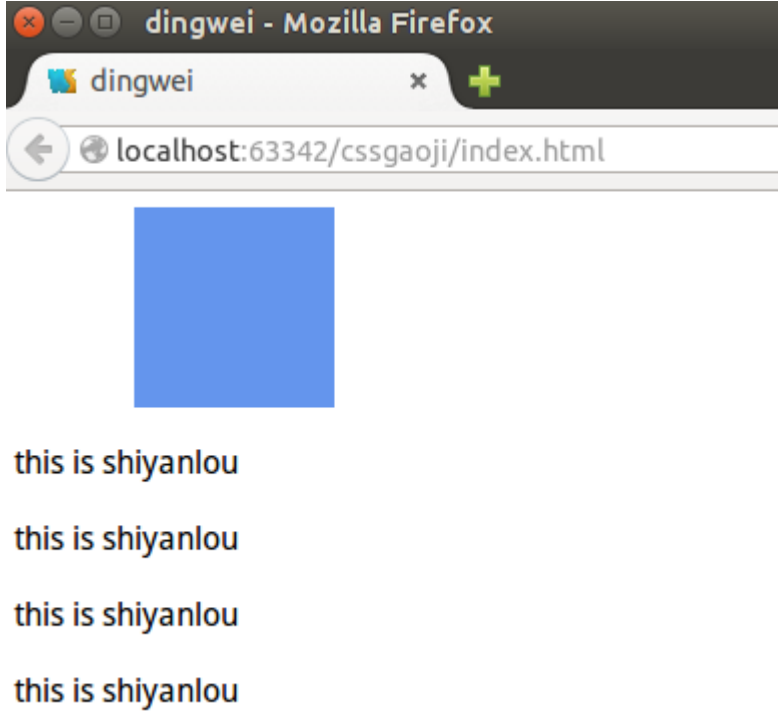
接下来我们就可以看到普通流的效果:



当我们在 CSS 中加入 position 赋值为相对的, 向左偏移 60px

```
position: relative;
left: 60px;
```

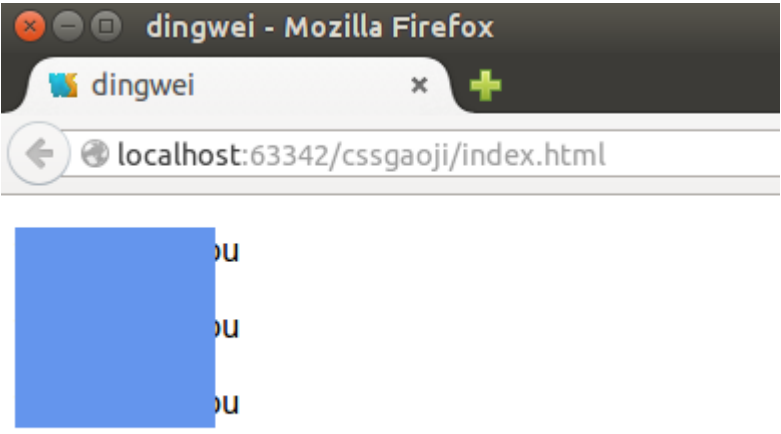
接下来我们会看见:



下面我们再将 position 设置为绝对的:

```
position: absolute;
```

效果就变成了这样:



this is shiyanlou

通过比较我么就能理解 position 这两个值的区别,还有两个属性就是 fixed和 static,fixed 是将元素固定下来,就算滚动屏幕,它也会在同一个地方不会动;而 static 设置以后,偏移量什么的就没用了.

下面我们接着来看其他的属性

当我们再加一个块在前面 div 后面的时候:

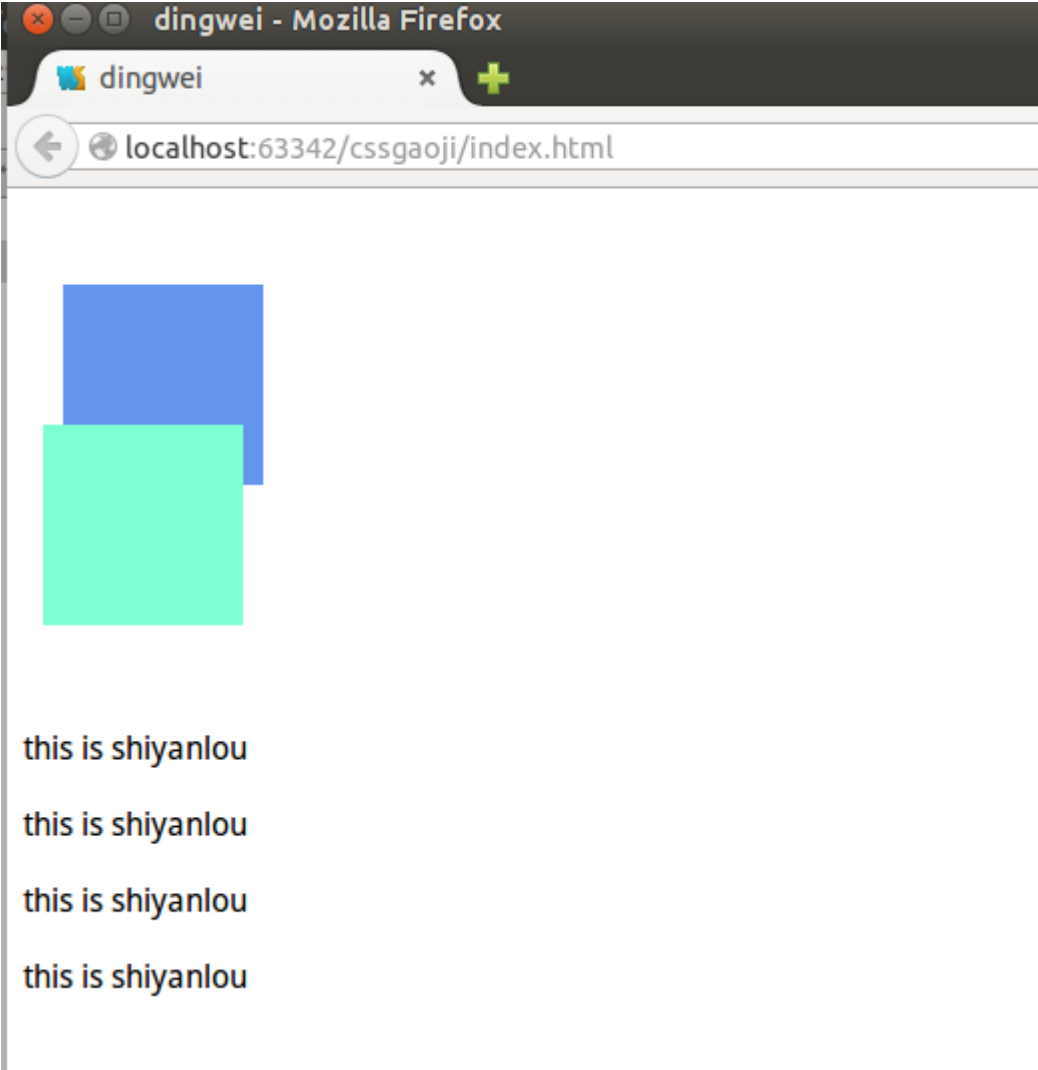
HTML

```
<div class="position1"></div>
<div class="position2"></div>
```

CSS 添加:

```
.position2{
  width: 100px;
  height: 100px;
  background-color: aquamarine;
  position: relative;
  left:10px;
  top: 10px;
}
```

就会出现下面的情况:



接下来我们就可以通过 z-index 来控制哪一块在最前面:

接下来我们就修改下 CSS 文件来交换他们的前后顺序:

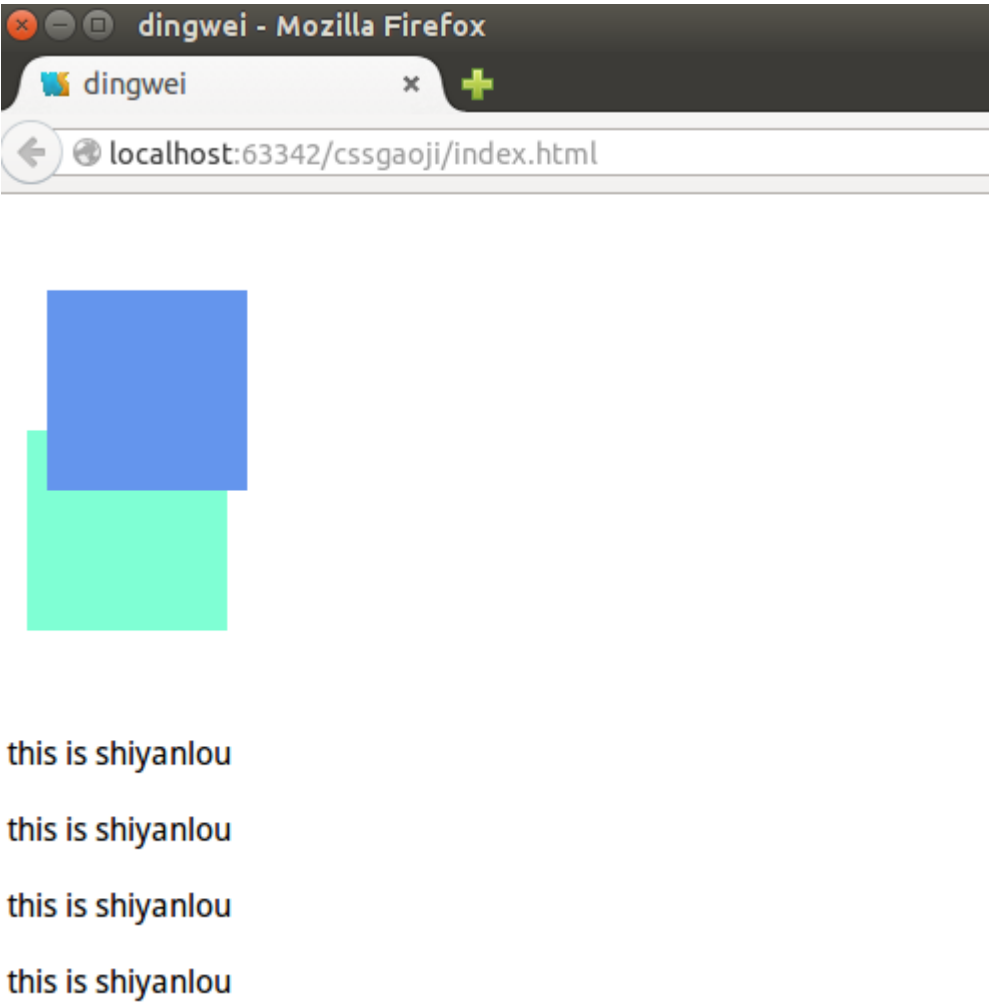
position1 中加入

```
z-index: 2;
```

position2 中加入:

```
z-index: 1;
```

就可以达到交换的效果:



2.2 CSS 浮动

这里涉及到的属性就是 float,其值可以赋值为:

left:元素向左浮动

right:元素向右浮动

none:不浮动

inherit:从父级继承浮动的属性

还有一个就 clear 属性:

主要用于去掉向各方向的浮动属性(包括继承来的属性)

下面我们就先创建一个最基础 html 和 CSS 文件,下面是基础内容:

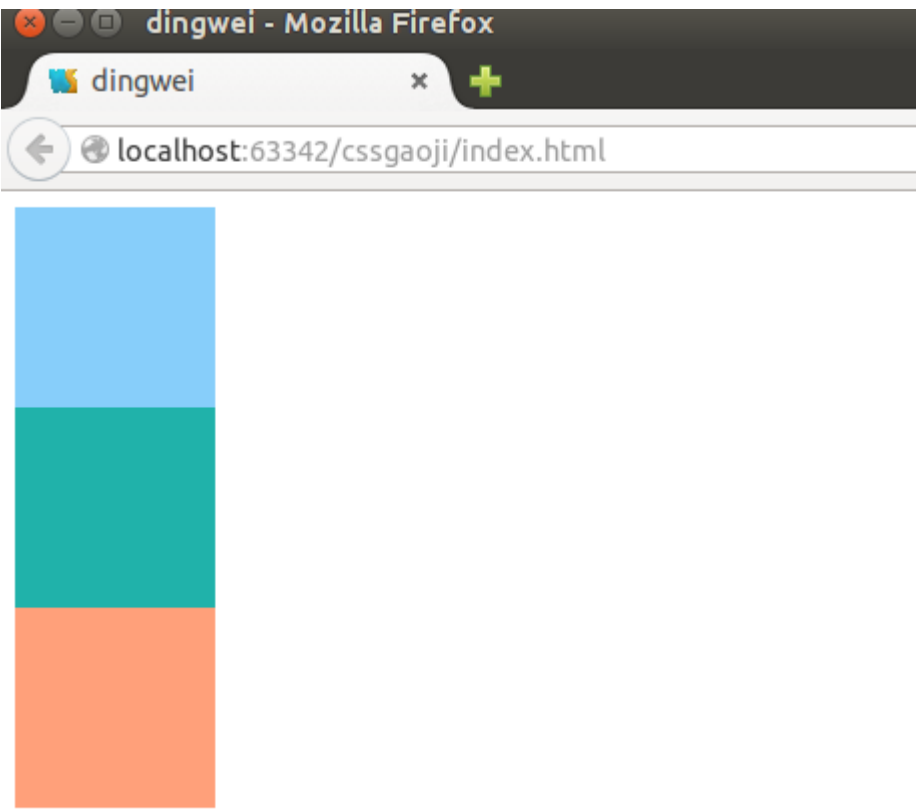
html:

```
<div class="qd"></div>
<div class="wd"></div>
<div class="ed"></div>
```

CSS

```
.qd{
    width: 100px;
    height: 100px;
    background-color: lightskyblue;
}
.wd{
    width: 100px;
    height: 100px;
    background-color: lightseagreen;
}
.ed{
    width: 100px;
    height: 100px;
    background-color: lightsalmon;
}
}
```

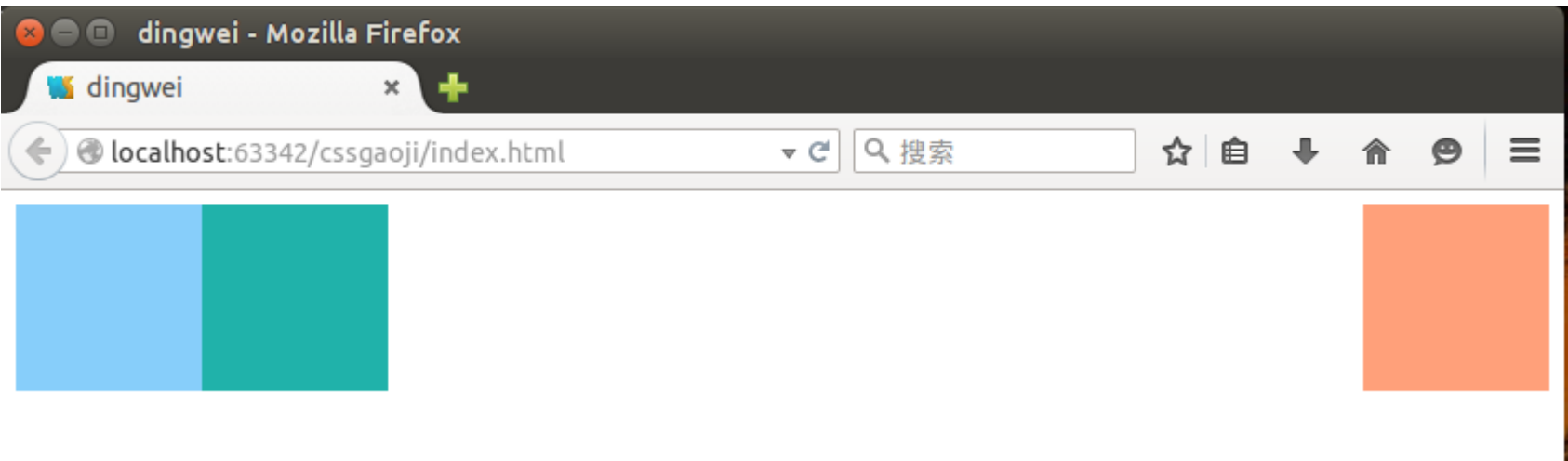
下面是显示效果:



在这个基础上我们他们全加上 float 属性,前两个往左,后一个向右,看看会有什么效果:

```
float: left;
float: right;
```

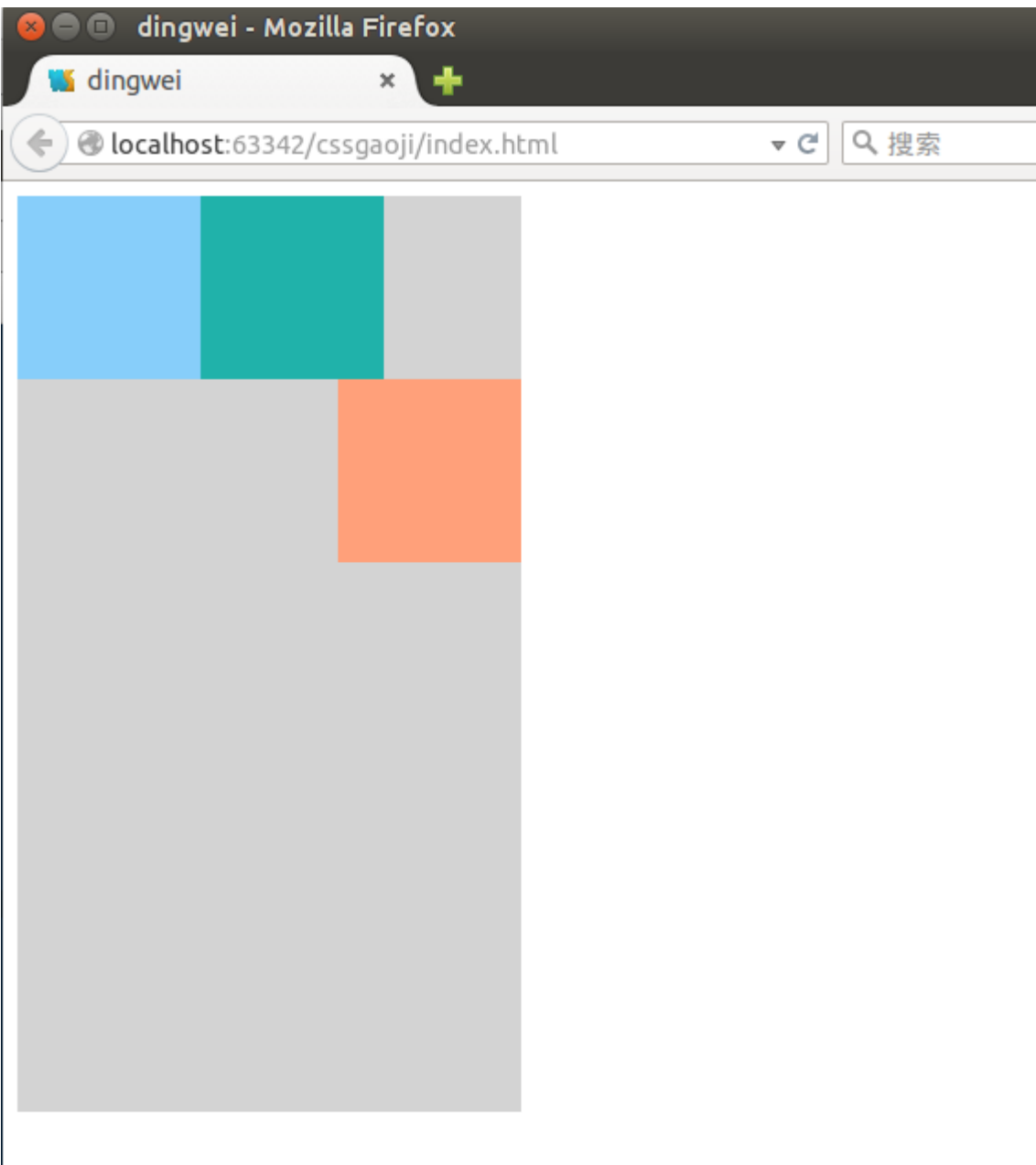
效果图



就像几个小东西在一个房间里跑,你可以规定它跑的方向,他们会跑到边框为止,为了测试,我们不妨来限定一个空间给它们(将这三个 div 全放到一个 div 中).就像这样:

```
<div class="container">
  <div class="qd"></div>
  <div class="wd"></div>
  <div class="ed"></div>
</div>
```

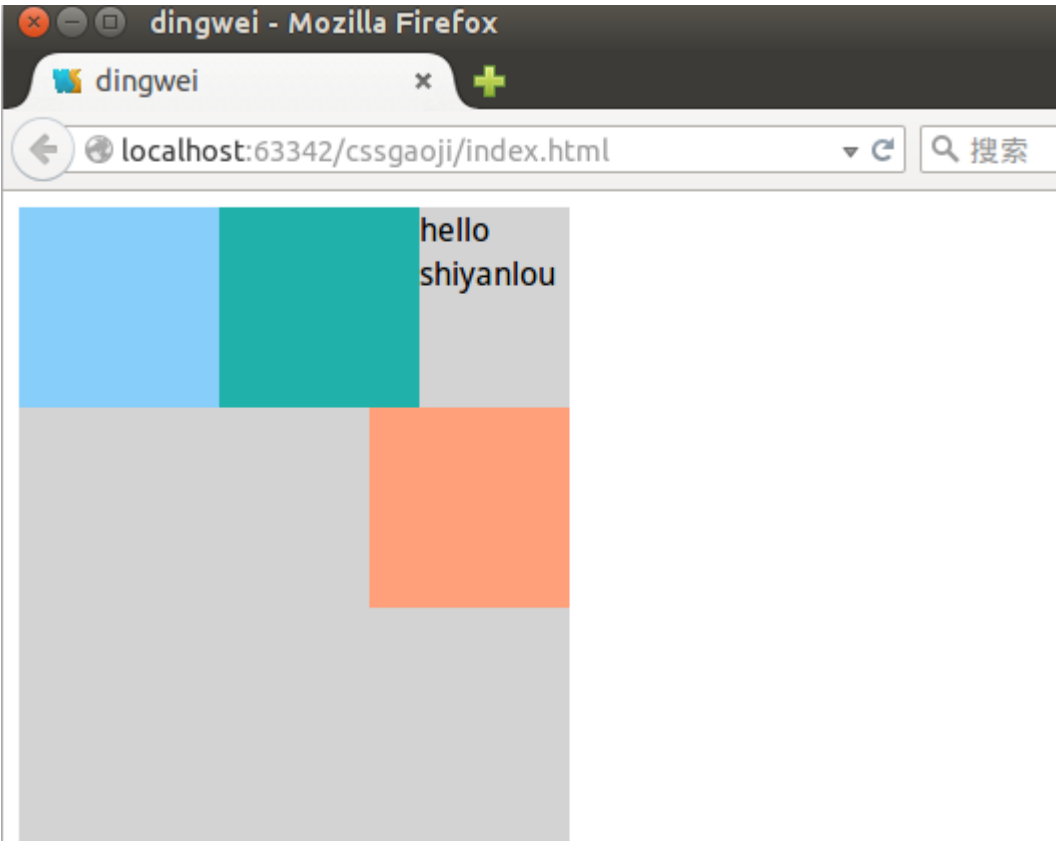
接下来你就会看见:



但是有时我们不需要浮动,就像下面这样,我们想在上面效果下面加上一句话,然后我们就直接加入了一个

```
<div class="text">hello shiyanlou</div>
```

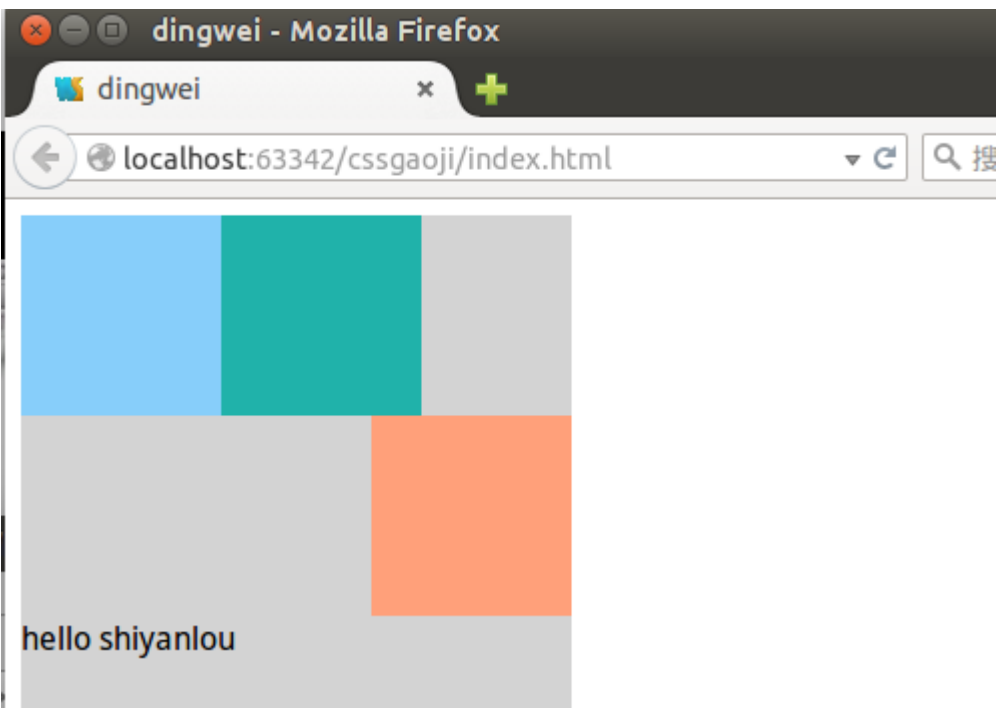
在 container 中.然后我们会看见



这说明,这个 div 也继承了浮动的属性,要想让字体到下面去,我就必须取消字体 div 浮动.那么我们就在 CSS 中添加如下如下内容:

```
.text{
    clear: both;
}
```

效果图:



2.3 CSS 尺寸

尺寸属性允许你控制元素的高度和宽度。同样，它允许你增加行间距。

涉及到的属性有

height-- 设置元素的高度。

line-height --设置行高。

max-height-- 设置元素的最大高度。

max-width --设置元素的最大宽度。

min-height --设置元素的最小高度。

min-width --设置元素的最小宽度。

width --设置元素的宽度。

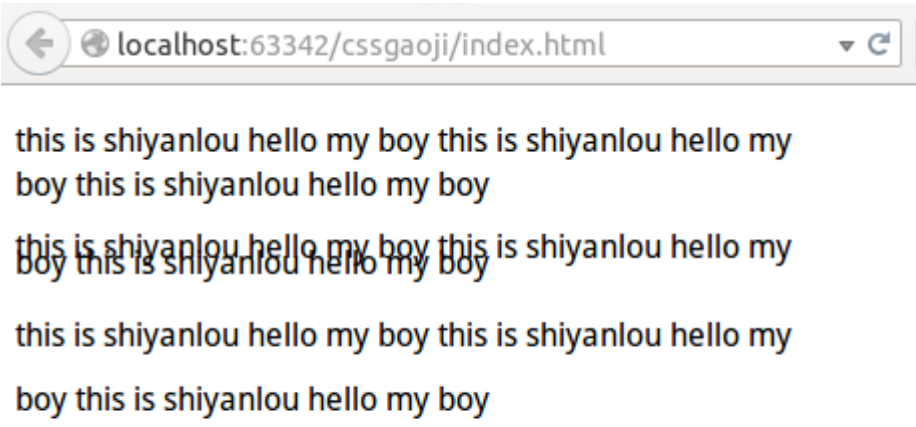
下面我们就写个 html 和 CSS 文件来具体比较下

```
.p1{
  line-height: normal;
  width: 400px;
}

.p2{
  line-height: 50%;
  width: 400px;
}

.p3{
  line-height: 200%;
  width: 400px;
}
```

效果图如下:



2.4 CSS 导航栏

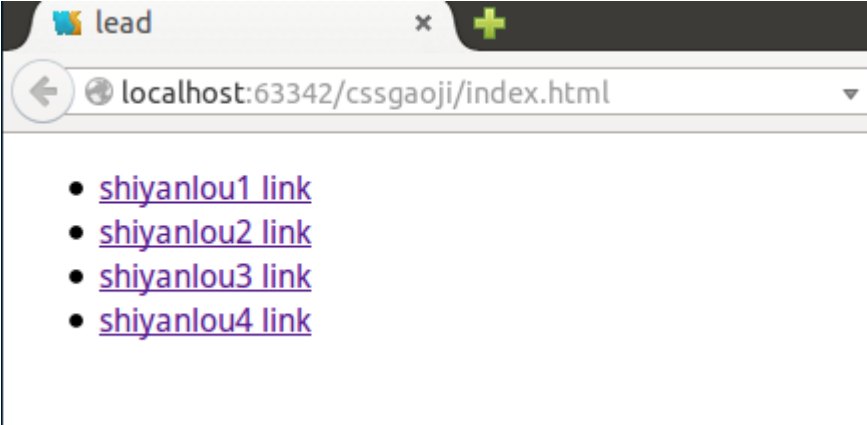
不管是什么网页,都会有导航栏来表述本网页所包含的内容,一般导航栏分为:水平导航栏,垂直导航栏.下面我们就来定制下自己的导航栏.

垂直导航栏:

首先我们以列表的形式作为最基础的承载,然后我们再其中加入本地或外部的链接,就像下面这样:

```
<ul>
<li><a href="http://www.shiyanlou.com">shiyanlou1 link</a></li>
<li><a href="http://www.shiyanlou.com">shiyanlou2 link</a></li>
<li><a href="http://www.shiyanlou.com">shiyanlou3 link</a></li>
<li><a href="http://www.shiyanlou.com">shiyanlou4 link</a></li>
</ul>
```

然后我们就会得到这样的效果:



我们一般看见的导航栏都没有下划线,和前面的带点,并且当我们的鼠标移到链接上面时链接的颜色会发生相应的变化,这就是我们现在要让 CSS 实现的效果.

首先,我们要去掉前面的点

```
ul{
  list-style: none;
}
```

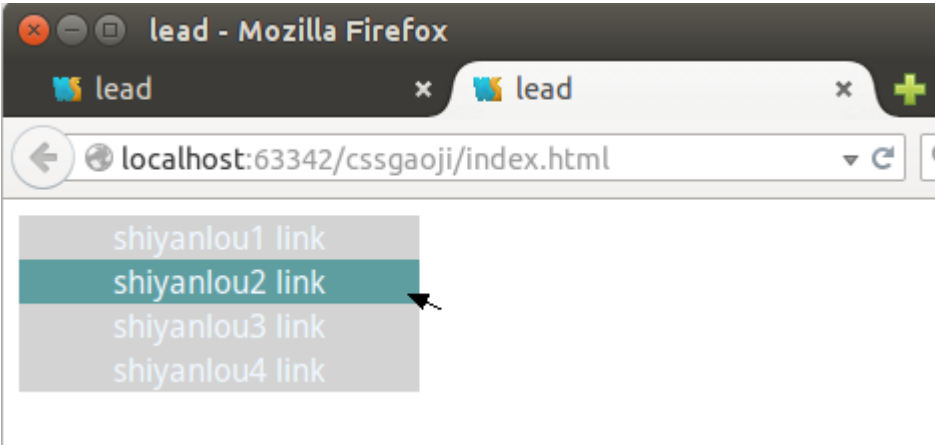
接下来我们就去掉下划线(不管是未被点击的状态还是已被点击的状态都去掉),然后加上个背景颜色,再将其显示作为块来显示:

```
a:link,a:visited{
  text-decoration: none;
  background-color: lightgray;
  display: block;
}
```

最后我们再给导航栏加个鼠标移动到上面时,改变背景颜色:

```
a:active,a:hover{
  background-color: cadetblue;
}
```

下面就是效果图



垂直的效果图讲完之后,我们再来讲讲水平的导航栏,我们就只需要修改 CSS 文件就可以了.

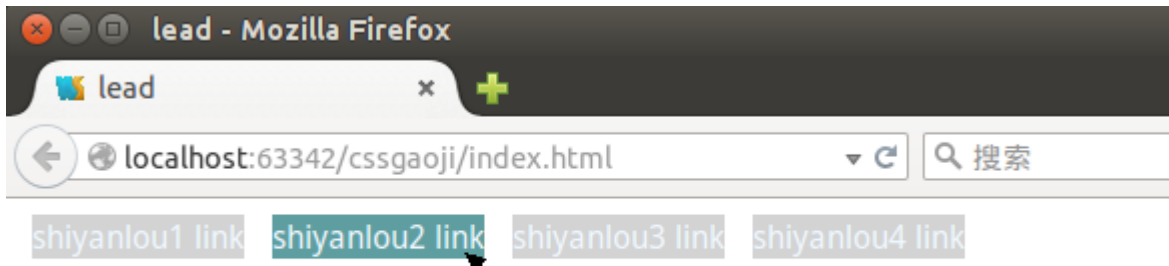
首先我们要将前面的显示效果删除,就是这句:

```
display: block;
```

然我们只需要在 li 标签中改变显示方式就可以:


```
li{
    display: inline;
}
```

这样就可以实现水平导航栏



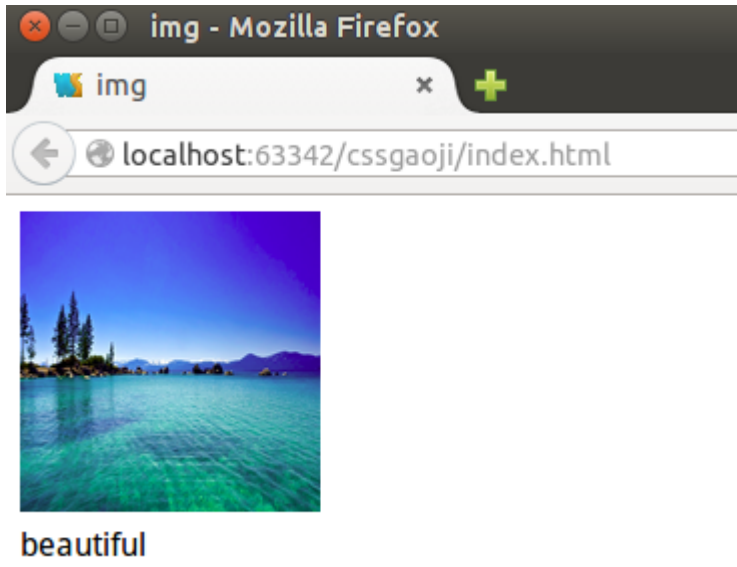
我们可以根据自己的喜好,设置边距,字体,颜色等等.这里我么不就不再一一的讲述.

2.5 CSS 图片

首先我们先引入一张图片,加上一句描述语,使用 div 承载.

```
<div class="image">
  <a href="/1348306907524.jpg" target="_self">
    
  </a>
  <div class="text">beautiful</div>
</div>
```

就是下面的效果:

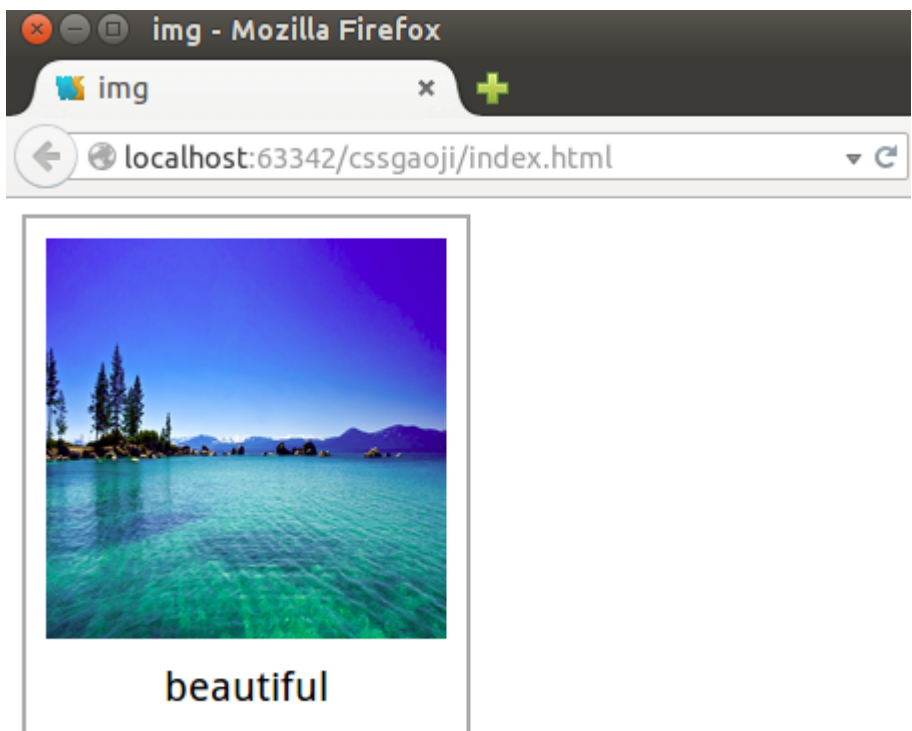


接下来我们就开始定制图片的显示:

图片加边框,修改描述字体中对其,修改字体大小,将整个 div 向左浮动,使边框与图片进行贴合:

```
.image{
    border: 2px solid darkgrey;
    width: auto;
    height: auto;
    float: left;
    text-align: center;
    padding: 5px;
}
img{
    padding: 5px;
}
.text{
    font-size: 20px;
    margin-bottom: 5px;
}
```

这就是处理过后的的效果



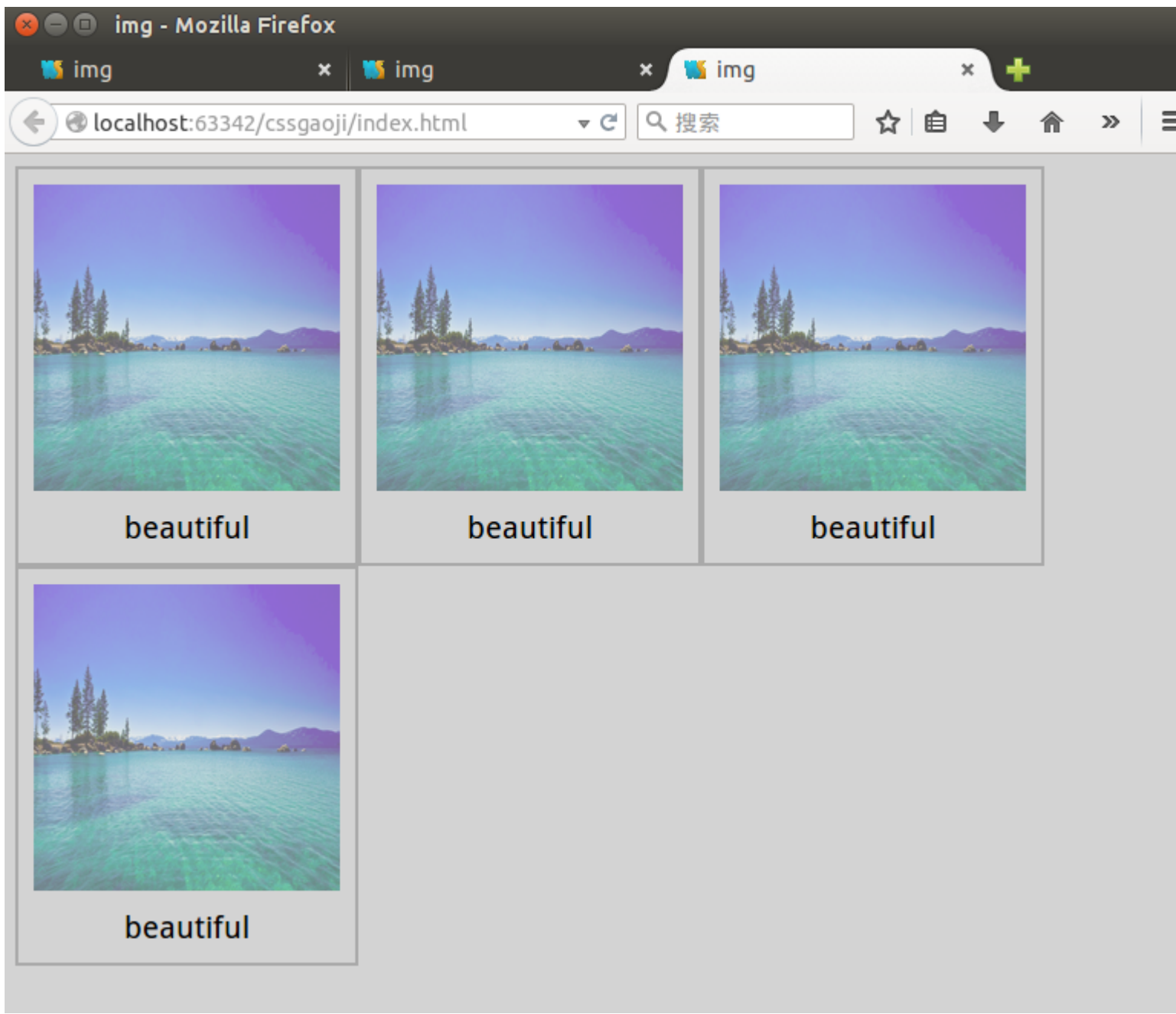
之后我么再设计图片的透明度:

这个比较简单,就只需要在图片 CSS 设置中加入:

```
opacity: 0.5;
```

这个属性的值范围为 0-1 设置透明度,0 为完全透明,1 代表完全不透明.

下面就是半透明的效果图:



三、实验总结

这一章中我们学习了 CSS 基本的高级设置,包括:定位,浮动,尺寸,导航,图片.这些都是 在平时用的很多的设置.

四、练习

根据教程练习上面讲到的高级设置及属性,熟练掌握这些讲到的设置.