****

**实**

**验**

**报**

**告**

**课 程 Web开发**

**题 目 可交互照片墙**

**分 数**

**学院名称 智能与计算学部**

**专 业 软件工程**

**学生姓名 刘兴宇**

**学 号 3017218063**

**年 级 2017**

**班 级 1**

目录

[一、实验目的 1](#_Toc527352380)

[二、实验内容 1](#_Toc527352381)

[三、实验步骤 1](#_Toc527352382)

[四、实验结果 6](#_Toc527352383)

[五、实验结论 8](#_Toc527352384)

[六、源代码 8](#_Toc527352385)

## 实验目的

本次实验目为练习HTML+CSS的布局与样式，以及简单的交互

## 二、实验内容

制作一个可交互的照片墙，照片墙可以点击照片查看大图

## 三、实验步骤

1. 确定需求

* 照片墙是一个可以将照片按照一定的顺序排列的Web页面，基本要求是照片要合理的地排布在页面上，且照片的尺寸要合适，布局要整齐。
* 点击照片，可以在当前页面上方显示一个新的图层，其中显示这张照片的更大尺寸的图片。点击图层的其它位置可以隐藏该图层
* 可以向照片墙中添加照片
* 可以删除照片墙中的照片
* 可以替换照片墙的照片
* 页面要具有很好的过渡效果，且满足响应式设计，适应不同尺寸的设备

2. 设计页面原型

根据需求，设计界面原型。

* 上方居中显示标题和作者
* 主题部分为网格化布局照片，使用空白作为网格线。在正常计算机浏览器上每行显示4张图片，默认初始有12张图片。在平板电脑（宽度小于1024px大于690px）浏览器上每行显示3张图片。在手机（宽度小于690px）浏览器上每行显示两张图片。且在页面右下位置有一个固定不动的添加图片按钮
* 底端显示提示信息
* 大图页面为灰色蒙层，蒙层上为一张图片。在蒙层右下位置有一个删除按钮和一个替换按钮

3. 实现布局

编写index.html、index、CSS

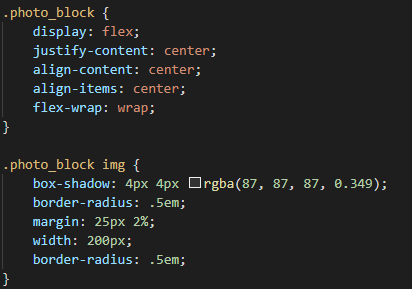
关键技术：

* 主页面



Outside\_block中为页面主题部分，其中title\_block为标题，有两个标题标签<h>组成，photo\_block是图片，由12张初始<img>组成，bottom\_block为底端提示，由两个文字标签<span>组成

其中图片布局采用flex布局，具体样式为：

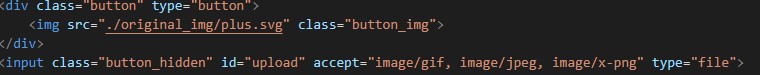


* + 对于容器整体，采用多余元素换行显示，水平、竖直以及每行行内竖直均为居中
  + 对于容器内的图片元素，采用百分比宽度20%设置图片宽度，同时确保每行显示4个图片。

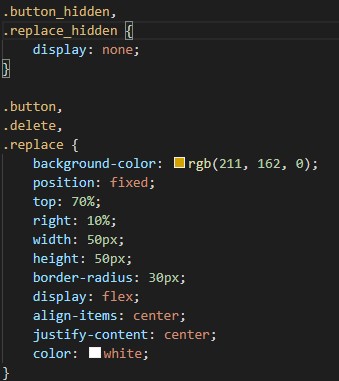
其余样式不再赘述

* 加号按钮

在outside\_block外设置一个div作为新增图片的按钮，其中放置一个加号的图片。此外还需要添加一个隐藏的<input>组件，用于触发选择文件。因为div无法单纯通过点击事件调用选择文件的接口，因此将它的点击事件绑定到可以调用选择文件事件的input上。



具体样式为：



* + display:none将input隐藏
  + position:fixed将元素设为固定布局，固定在页面的某个位置，不随页面的移动而移动。同时设置为为flex容器，使其中的“+”号可以位于按钮的中央
* 大图图层



* + Big为图层本身
  + Big\_img用于存放大图，其src为空，动态填入
  + 两个按钮使用button
  + Input作用同上
* 伪类动作
  + 对所有图片、按钮，使用:hover {cursor: pointer}使的当鼠标悬停时，变为指针
  + 对图片，使用:hover{transform.scale(1.1)}设置鼠标悬停在图片上方时，图片略微放大以指示该图片
  + 对三个按钮，使用:hover{transform: translateY(-4px)}设置鼠标悬停在按钮上方时，按钮向上浮动4px的效果
* 响应式设计

使用了flex布局，实现响应式设计

同时在<head>中设置viewport视窗，限制移动端的视窗长度

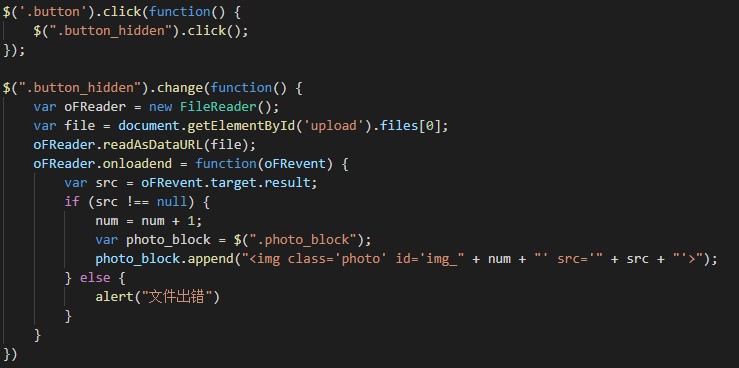
<meta name="viewport" cotent="width=device-width">

4. 实现交互

编写index.js，使用jQuery编写

核心交互功能

* 添加图片



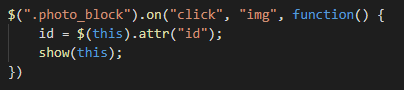
将按钮（.button）的点击事件绑定到隐藏的文件输入<input>上（.button\_hidden），并且监听选择文件改变事件change。

利用FileReader，对<Input>（id=“upload”）选择的文件file，进行读取，并且使用src接收返回的图片base64格式编码（存放在oFRevent.target.result）中。

在将或得到的src插入一个<img>标签中，添加到photo\_block最后

* 点击查看大图

为所有的图片添加点击事件show，对于新增的DOM节点，由于JS的渲染特性为在加载时一次性渲染，因此无法绑定上述事件。使用on函数，监听未知img节点的点击事件，并绑定响应函数show。并将触发点击事件的元素id存入全局变量中

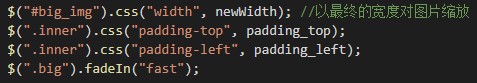
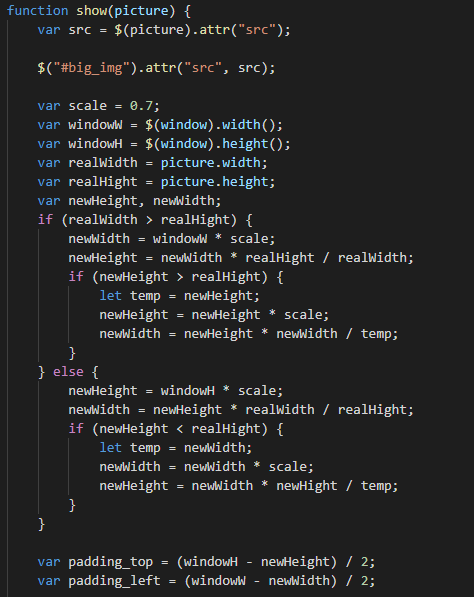


传入参数为当前被点击的DOM对象，show函数为：

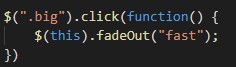
首先获取被点击图片的src，并将它赋给大图big\_img

再根据当前窗口的大小和图片的尺寸，设置大图的尺寸。若图片是宽大于高，则将宽设置为窗口宽度的scale倍，否则将高设置额为窗口高度为scale被，并等比例缩放。将新的宽度、高度赋予big\_img，并计算边距，赋予内部div（.inner）

设置完成后，使用fadeIn将大图图层显示出来（“.big”）



对大图图层，设置隐藏函数fadeOut响应点击事件



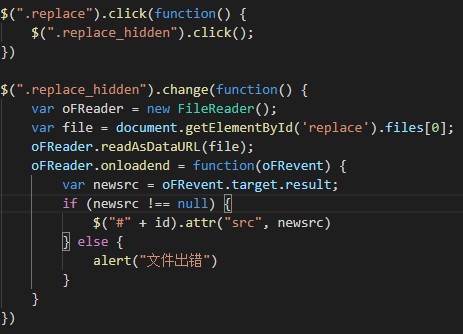
* 删除图片：

直接使用remove删除。



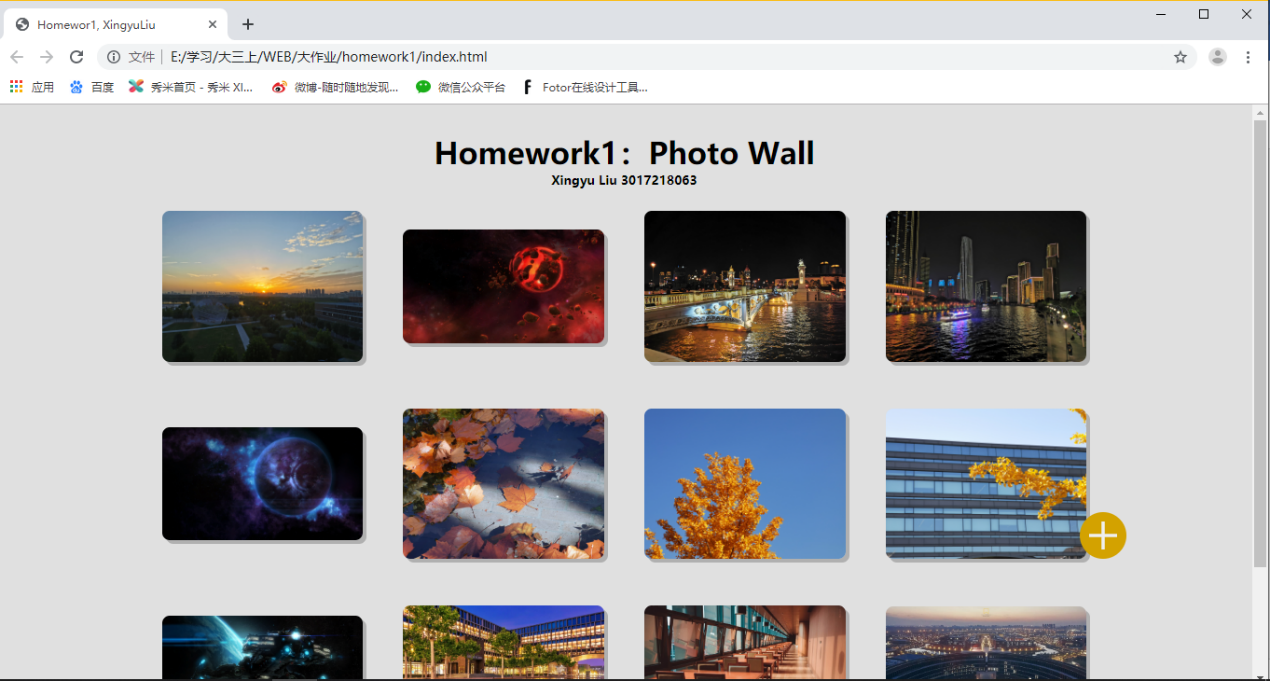
* 替换图片：

原理同新增图片，拿到src后替换原有src

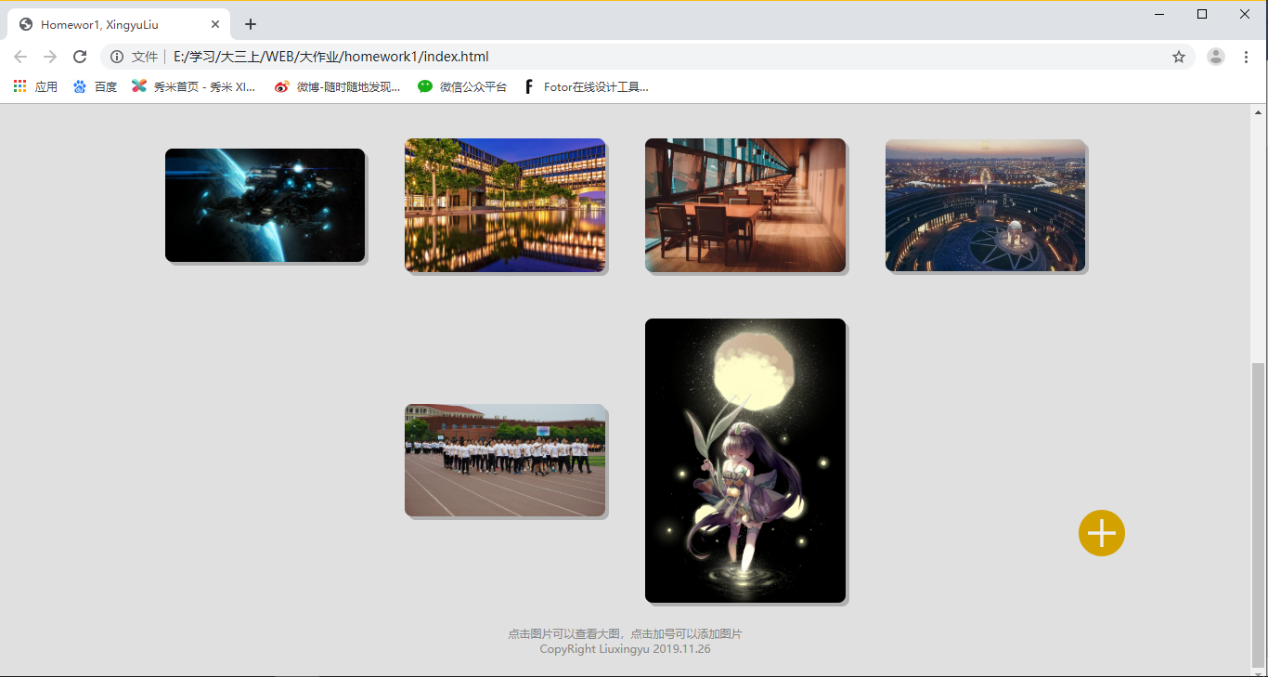


## 四、实验结果

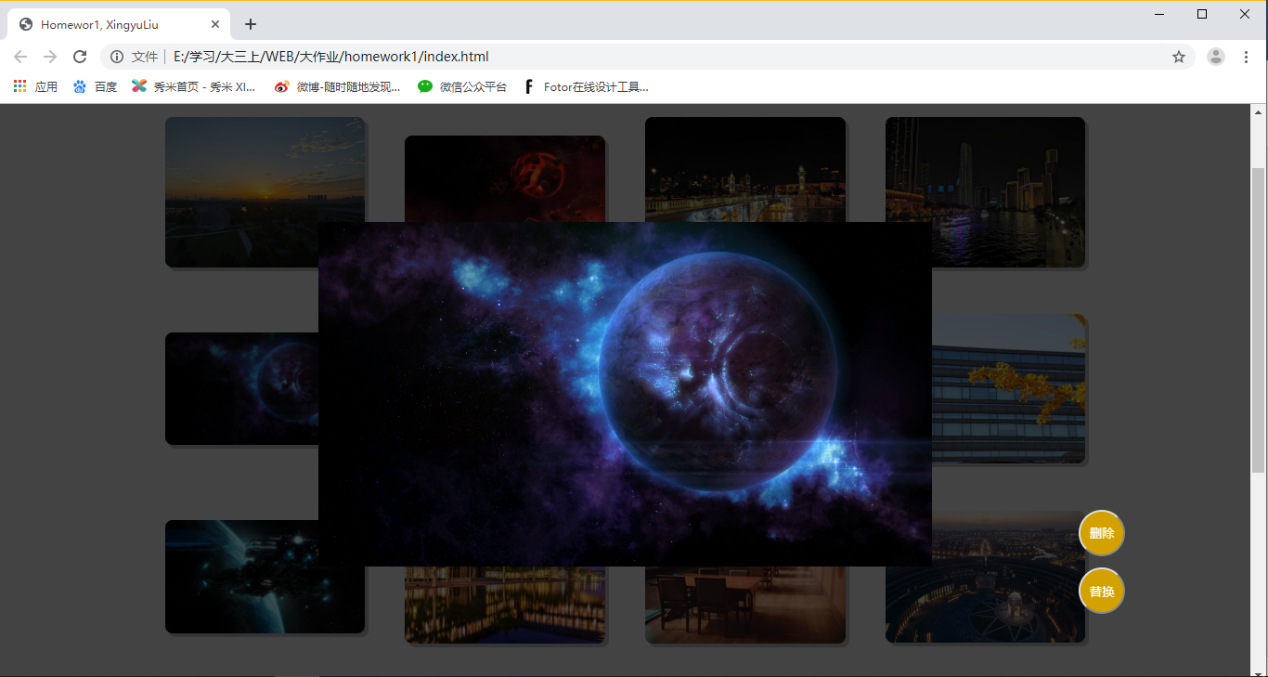
1. 界面效果



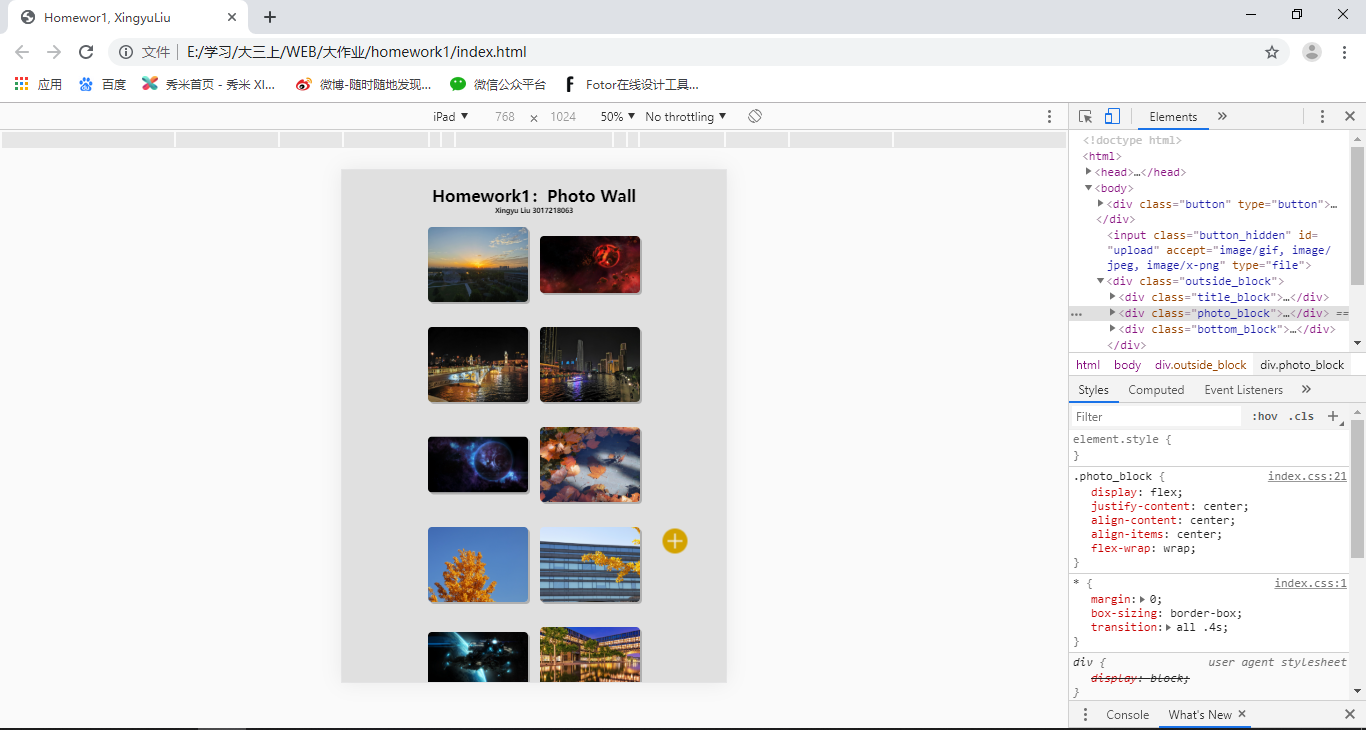
2. 添加图片后：



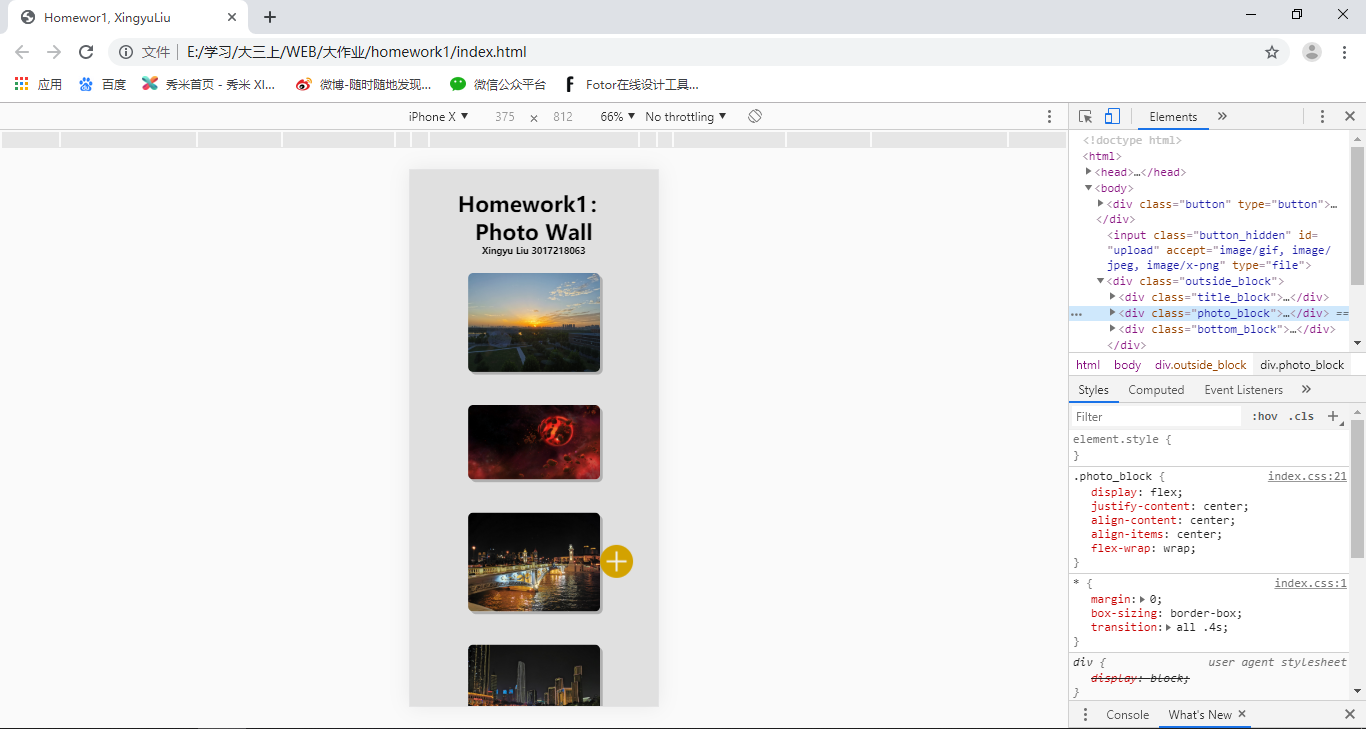
3. 查看大图



4. 平板电脑效果：



5. 手机效果



## 五、实验结论

对于Web页面的开发，设计响应式布局是十分重要的

Flex布局中align-items和align-content很容与混淆

## 六、源代码

1. index.html

<!DOCTYPE html>

<html>

<head>

    <title>Homewor1, XingyuLiu</title>

    <meta charset="utf-8">

    <meta name="viewport" content="width=980px">

    <link rel="stylesheet" type="text/css" href="./index.css">

    <script src=".\jquery-3.4.0.min.js"></script>

</head>

<body>

    <div class="button" type="button">

        <img src="./original\_img/plus.svg" class="button\_img">

    </div>

    <input class="button\_hidden" id="upload" accept="image/gif, image/jpeg, image/x-png" type="file">

    <div class="outside\_block">

        <div class="title\_block">

            <h1 class="title">Homework1：Photo Wall</h1>

            <h5 class="title"> Xingyu Liu 3017218063</h5>

        </div>

        <div class="photo\_block">

            <img id="img\_1" src="./original\_img/1.jpg">

            <img id="img\_2" src="./original\_img/2.jpg">

            <img id="img\_3" src="./original\_img/3.jpg">

            <img id="img\_4" src="./original\_img/4.jpg">

            <img id="img\_5" src="./original\_img/5.jpg">

            <img id="img\_6" src="./original\_img/6.jpg">

            <img id="img\_7" src="./original\_img/7.jpg">

            <img id="img\_8" src="./original\_img/8.jpg">

            <img id="img\_9" src="./original\_img/9.jpg">

            <img id="img\_10" src="./original\_img/10.jpg">

            <img id="img\_11" src="./original\_img/11.jpg">

            <img id="img\_12" src="./original\_img/12.jpg">

        </div>

        <div class="bottom\_block">

            <span>点击图片可以查看大图，点击加号可以添加图片</br></span>

            <span>CopyRight Liuxingyu 2019.11.26</span>

        </div>

    </div>

    <div class="big">

        <div class="inner">

            <img id="big\_img" src="" />

            <button class="delete">删除</button>

            <button class="replace">替换</button>

            <input class="replace\_hidden" id="replace" accept="image/gif, image/jpeg, image/x-png" type="file">

        </div>

    </div>

    <script type="text/javascript" src="./index.js"></script>

</body>

</html>

2. index.css

\* {

    margin: 0;

    box-sizing: border-box;

    transition: all .4s;

}

body {

    background-color: rgb(224, 224, 224);

}

.outside\_block {

    max-width: 100%;

    min-height: 100%;

    padding: 30px 10% 30px 10%;

}

.title {

    text-align: center;

}

.photo\_block {

    display: flex;

    justify-content: center;

    align-content: center;

    align-items: center;

    flex-wrap: wrap;

}

.photo\_block img {

    box-shadow: 4px 4px rgba(87, 87, 87, 0.349);

    border-radius: .5em;

    margin: 25px 2%;

    width: 200px;

    border-radius: .5em;

}

.photo\_block img:hover {

    transform: scale(1.1);

    transition: all .2s;

}

.bottom\_block {

    font-size: 12px;

    color: rgb(136, 136, 136);

    text-align: center;

}

.button\_hidden,

.replace\_hidden {

    display: none;

}

.button,

.delete,

.replace {

    background-color: rgb(211, 162, 0);

    position: fixed;

    top: 70%;

    right: 10%;

    width: 50px;

    height: 50px;

    border-radius: 30px;

    display: flex;

    align-items: center;

    justify-content: center;

    color: white;

}

.replace {

    top: 80%;

}

/\* 悬停，按钮上浮动 \*/

.button:hover {

    box-shadow: 0 10px 13px 0 rgba(1, 1, 3, 0.15);

    transform: translateY(-4px);

}

.button\_img {

    width: 30px;

    height: 30px;

}

img:hover {

    cursor: pointer;

}

.big {

    position: fixed;

    top: 0;

    left: 0;

    background: rgba(0, 0, 0, 0.7);

    z-index: 2;

    width: 100%;

    height: 100%;

    display: none;

}

.inner {

    width: 100%;

    height: 100%;

}

.img\_block {

    display: flex;

}

3. index.js

$(function() {

    var num = 12;

    var id = null;

    //将button点击事件绑定到input上，实现点击button选择文件

    $('.button').click(function() {

        $(".button\_hidden").click();

    });

    //filereader获取文件base64编码，填充到一个新<img>标签中

    $(".button\_hidden").change(function() {

        var oFReader = new FileReader();

        var file = document.getElementById('upload').files[0];

        oFReader.readAsDataURL(file);

        oFReader.onloadend = function(oFRevent) {

            var src = oFRevent.target.result;

            if (src !== null) {

                num = num + 1;

                var photo\_block = $(".photo\_block");

                photo\_block.append("<img id='img\_" + num + "' src='" + src + "'>");

            } else {

                alert("文件出错")

            }

        }

    })

    //监听所有图片以及可能尚未添加的<img>标签，并且显示大图

    $(".photo\_block").on("click", "img", function() {

        id = $(this).attr("id");

        show(this);

    })

    //隐藏大图

    $(".big").click(function() {

        $(this).fadeOut("fast");

    })

    //显示大图

    function show(picture) {

        var src = $(picture).attr("src");

        //填充大图src

        $("#big\_img").attr("src", src);

        //缩放图片。如果图片长>宽，则长边变为屏幕宽度的scale比例，如果此时高度大于屏幕高度

        //则再将高度缩放scale比例。反之同理

        var scale = 0.7;

        var windowW = $(window).width();

        var windowH = $(window).height();

        var realWidth = picture.width;

        var realHight = picture.height;

        var newHeight, newWidth;

        if (realWidth > realHight) {

            newWidth = windowW \* scale;

            newHeight = newWidth \* realHight / realWidth;

            if (newHeight > realHight) {

                let temp = newHeight;

                newHeight = newHeight \* scale;

                newWidth = newHeight \* newWidth / temp;

            }

        } else {

            newHeight = windowH \* scale;

            newWidth = newHeight \* realWidth / realHight;

            if (newHeight < realHight) {

                let temp = newWidth;

                newWidth = newWidth \* scale;

                newHeight = newWidth \* newHight / temp;

            }

        }

        var padding\_top = (windowH - newHeight) / 2;

        var padding\_left = (windowW - newWidth) / 2;

        $("#big\_img").css("width", newWidth); //以最终的宽度对图片缩放

        $(".inner").css("padding-top", padding\_top);

        $(".inner").css("padding-left", padding\_left);

        $(".big").fadeIn("fast");

    }

    //删除

    $(".delete").click(function() {

        $("#" + id).remove();

    })

    //替换

    $(".replace").click(function() {

        $(".replace\_hidden").click();

    })

    //替换

    $(".replace\_hidden").change(function() {

        var oFReader = new FileReader();

        var file = document.getElementById('replace').files[0];

        oFReader.readAsDataURL(file);

        oFReader.onloadend = function(oFRevent) {

            var newsrc = oFRevent.target.result;

            if (newsrc !== null) {

                $("#" + id).attr("src", newsrc)

            } else {

                alert("文件出错")

            }

        }

    })

})