**<开发报告>**

**# 开发文档**

**## 老师要求**

>介绍网页构思 从那几个方面介绍 使用哪种技术 网页制作过程 核心代码与截图 网页细节与色彩介绍

**## 一丶准备过程**

经过三名组员讨论, 确定了网站主题. 那就是摄影, 我们准备搭建一个关于摄影的网站.

首先考虑到本次课业是小组的形式来完成的. 从效率方面来考虑, 我们使用GitHub平台来进行协同开发, 进行代码编辑, 资源整理.

综合在校的学习, 以我们现在的知识储备, 我们使用**\*\*CSS3\*\***, **\*\*HTML5\*\***和**\*\*JQ\*\***相关知识, 做到**\*\*响应式网站\*\***.

经过讨论, 根据老师的需求, 我们把网站分别从图片欣赏, 摄影后期调色预设分享, 相关摄影推荐(公众号, B站UP主, 好用的APP, Youtube频道和电脑调色程序), 摄影知识(技巧)这几个方面来介绍摄影.

然后我们使用微软的云笔记**\*\*OneNote\*\***, 我们首先使用手绘板进行网站的初始设计, 生成一个大概的设计草稿. 然后再精细讨论, 最后共享云笔记, 合理分配任务.

我们使用适合的相关技术

1. Github

2. OneNote

3. Visual Studio Code

**## 二丶开发过程**

我们确立我们的开发要求后, 我们先做导航栏, 统一不同页面的导航栏. 我们使用CSS3的 Flexbox 技术, 与少量的js代码 ,使我们CSS代码量的减少, 性能更高, 写出了一个合理的导航栏, 为了达到响应式效果, 我们在CSS中使用媒体查询技术, 在不同宽度的页面中, 达到了合适的阅读效果.

**### 2.1 导航栏**

**### 2.2 首页**

简单的一个轮播图

核心代码:

$(function(){

    //获取li的长度

    var len=$(".oLi").length;

    //把所有li的宽度算好,赋值给ul当宽度

    $("#banner").width($(".oLi").eq(0).width()\*len);

    //id是为了后面的计时器暂停设置的

    var id;

    //index用来统计滚动次数

    var index=0;

    //启动计时器

    id=setInterval(start,3000);

    //给ul(#banner)添加hover选择器事件

    $("#banner").hover(function(){

        //鼠标进入的时候暂停

        clearInterval(id);

    },function(){

        //鼠标离开的时候开始

        id=setInterval(start,3000);

    });

    //计时器中执行的代码块

    function start(){

        //执行的时候次数++

        index++;

        //用animate设置ul的left属性,与左边的距离

        $("#banner").animate({'left':-$(".oLi").eq(0).width()\*(index%len)});

        //实现无缝轮播的关键地方

        //判断index%len的值index%len=0的时候表示第一张,那么len-1表示最后一张,

        //最后一张与第一张一模一样,所有直接修改left属性

        if(index%len==(len-1)){

            $("#banner").animate({'left':0},0);

            //同时给index+1跳过第一张图的再次加载

            index++;

        }

        //实际的加载情况就是第一张,第二张,第三张,第四张,第五张(瞬间改变图片为第一张的,是一模一样的替换,不是滚动,看不出来,所以称之为无缝轮播),第二张...

    }

});

**### 2.3 图片欣赏页面**

对于图片欣赏, 会出现大量的图片, 所有决定使用瀑布流布局, 实现图片懒加载. 以保证浏览器的性能, 使用户在浏览页面时不会出现卡顿等问题.

实现响应式瀑布流, 需要在不同页面设置不同图片的列数. 使用jq, 我们先获取到当前页面的宽度, 然后根据合适的用户体验, 从而导致考虑不同列数. 使用当前页面宽度除于列数, 可得到每一列图片的宽度, 再减去每张图片之间的间隔. 从而设置图片的宽度, 实现响应式.  还是用resize()函数, 判断浏览器的大小是否变换, 变化则重新执行瀑布流函数.

实现图片懒加载, 需要获取页面滚动距离与当前浏览器的高度, 再和当前已有图片的所有高度比较, 从而判断用户是否预览到最后一张.

在开发过程中, 我们遇到了一些浏览器渲染问题, 浏览器渲染过快, 导致图片还没加载出来, 就添加到页面当中, 导致瀑布流函数获取错误的图片高度, 导致多张图片重叠在一起.

为了解决这个问题, 我们需要添加一个回调函数, 用来判断每张的图片是否加载完成. 判断图片的高度是否等于0, 如果等于0, 意味着图片还没加载, 设置等待事件, 执行回调函数, 如果不等于0, 意味着图片加载完成, 执行瀑布流函数.

核心代码 :

function waterfall(flag) {

    var w = $(".main").width(),

        box = $(".main img"),

        boxW = 0,

        allH = [],

        minH = 0,

        minHI = 0,

        col=5,

        screenW = $(window).width();

    if (screenW > 1100) {

        col = 5;

    } else if (screenW > 800) {

        col=4;

    } else if (screenW > 615) {

        col=3;

    } else {

        col=2;

    }

    boxW = (w / col) - 15;

    box.css("width", boxW);

    box.each(function (i) {

        if (i < col) {

            $(this).css("position","");

            allH[i] = $(this).height();

        } else {

            minH = Math.min.apply(null, allH);

            minHI = $.inArray(minH, allH);

            if(flag){

                //使用top动画

                $(this).css({

                    "position": "absolute",

                    // "top": minH + 15,

                    "left": box.eq(minHI).position().left,

                }).animate({

                    "top": minH + 15,

                    // "left": box.eq(minHI).position().left,

                }, 1000, "swing");

            }else{

                //不使用

                $(this).css({

                    "position": "absolute",

                    "top": minH + 15,

                    "left": box.eq(minHI).position().left,

                });

                // }).animate({

                //     "top": minH + 15,

                //     // "left": box.eq(minHI).position().left,

                // }, 300, "swing")

            }

            allH[minHI] += $(this).height()+15;

        }

    });

}

//添加后续图片的函数

function addImg(){

    var main=$(".main"),

        boxL=$(".main img").length+1,

        times=0;

    if(boxL==ImgNumber+1){

        return false;

    }

    for(boxL;times<10;boxL++){

        $("<img src='images/waterfall-flow/"+boxL+".jpg'>").appendTo(main);

        times++;

        if(boxL==ImgNumber+1){

            break;

        }

    }

    isImgLoad(function(){

        waterfall(true);

    }, 5);

}

//判断图片是否加载完成

function isImgLoad(callback, time) {

    $('.main img').each(function () {

        // 找到为0就将isLoad设为false，并退出each

        if (this.height === 0) {

            isLoad = false;

            return false;

        }

    });

    // 为true，没有发现为0的。加载完毕

    if (isLoad) {

        clearTimeout(t\_img); // 清除定时器

        // 回调函数

        callback();

        // 为false，因为找到了没有加载完成的图，将调用定时器递归

    } else {

        isLoad = true;

        t\_img = setTimeout(function () {

            isImgLoad(callback); // 递归扫描

        }, time);

    }

}

//判断是否要加载新的图片

function checkLoadNewImg(){

    var box=$(".main img"),

        scrollTop=$(window).scrollTop();

        screenH=$(window).height();

        allImgHeight=box.last().position().top+Math.floor(box.last().height()/2);

    return (allImgHeight<scrollTop+screenH)?true:false;

}

**### 2.4 调色预设分享**

通过调色前后的图片对比, 让我们看见调色包的作用. 我们先用调色后的图片在页面中, 然后再通过js检测图片的大小, 设置未调色图片的宽度, 从而达到覆盖的效果. 再利用盒子的overflow属性, left: 50%, 做到对比的效果

核心代码:

function drags(dragElement, resizeElement, container) {

  // 初始化鼠标按下拖动事件.

  dragElement

    .on("mousedown", function(e) {

      dragElement.addClass("draggable");

      resizeElement.addClass("resizable");

      // 得到初始化位置

      var dragWidth = dragElement.outerWidth(),

        posX = dragElement.offset().left + dragWidth - e.pageX,

        containerOffset = container.offset().left,

        containerWidth = container.outerWidth();

      // 设置限制

      minLeft = containerOffset + 16;

      maxLeft = containerOffset + containerWidth - dragWidth - 16;

      // 设置鼠标拖动距离的限制

      dragElement.parents().on("mousemove", function(e) {

          leftValue = e.pageX + posX - dragWidth;

          if (leftValue < minLeft) {

            leftValue = minLeft;

          } else if (leftValue > maxLeft) {

            leftValue = maxLeft;

          }

          widthValue =((leftValue + dragWidth / 2 - containerOffset) \* 100) / containerWidth +"%";

          // 设置滑块和图片的新值

          // 绑定鼠标移开事件以停止拖动

          $(".draggable").css("left", widthValue).on("mouseup", function() {

              $(this).removeClass("draggable");

              resizeElement.removeClass("resizable");

            });

          $(".resizable").css("width", widthValue);

        })

        .on("mouseup", function() {

          dragElement.removeClass("draggable");

          resizeElement.removeClass("resizable");

        });

      e.preventDefault();

    })

    .on("mouseup", function(e) {

      dragElement.removeClass("draggable");

      resizeElement.removeClass("resizable");

    });

**### 2.5 相关推荐页面**

我们使用简约的设计主题, 强调内容.

我们推荐了一些公众号, B站的一些摄影up主, 他们的视频很有趣.

还有一些Youtube频道.  点击链接时, 会跳出详细信息, 或者是二维码的对话框, 利用jq实现了对话框效果

核心代码:

$("#bup a").click(function(){

            var img=$(this).siblings(".img").children().attr("src");

            var name=$(this).siblings("h3").text();

            var p=$(this).siblings("p").text();

            var url=$(this).data("url");

            //打开对话框

            $("#dialog-container").css("background-color","rgb(0, 0, 0, 0.9)").show().children().animate({

                width: "60%",

                height: "70%",

            }, 200, "swing");

            //填充内容

            $("<div class='row row-2'><div class='col'><div class='img'><img src="+img+"></div></div><div class='col'><h3>"+name+"</h3></div></div>").appendTo("#content");

            $("<div class='row'><div class='col'><p>"+p+"</p></div></div>").appendTo("#content");

            $("<div class'row'><div class='col'><a class='btn' href='"+url+"'><span>前往主页</span></a></div></div>").appendTo("#content");

    })

**### 2.6 摄影技巧**

我们利用jq写了一个导航栏定位效果, 可以是用户能够更快的到达相关的位置, 还使用了HTML5的相关属性, data属性. 用于储存视频的地址, 已经标题和图片.然后再使用jq进行获取.  再使用animate 的 scrollTop对body进行平滑过渡

核心代码:

$(window).scroll(function(){

        var top =$(document).scrollTop();

        var menu=$(".menu");

        var item=$(".item");

        var currentId="";

        item.each(function(){

            var itemTop=$(this).offset().top-300;

            if(top>itemTop){

                currentId="#"+$(this).attr("id");

            }else{

                return false;

            }

        });

        var currentLink=menu.find(".menu-active");

        if(currentId && currentLink.children().attr("href") != currentId){

            currentLink.removeClass("menu-active");

            menu.find("[href='"+currentId+"']").parent(".menu-item").addClass("menu-active");

        }

    });

    //给锚点添加过渡动画

    $(".menu-item>a").click(function(){

        $("html,body").stop(true, true).animate({

            scrollTop: ($($(this).attr('href')).offset().top

            )},500, "swing");

    })

**### 2.7 关于我们**

我们使用jq的ajax实现了一个留言榜的功能, 以及表单验证. 通过change事件,邮箱检测使用了正则表达.

核心代码:

//检测邮箱是否符合要求

    $("#email").change(function(){

        var email=$(this).val();

        if(!/^[a-zA-Z0-9\_-]+@[a-zA-Z0-9\_-]+(\.[a-zA-Z0-9\_-]+)+$/.test(email)){

            $(this).siblings("span").text("邮箱格式错误").addClass("error-title");

            flag=false;

        }else{

            flag=true;

            $(this).siblings("span").text("输入正确").removeClass("error-title");

        }

    })

 //使用ajax传递留言, 实现无级刷新留言

    $("#submit").click(function(){

        if(flag){

            $.ajax({

                type: "POST",

                url: "about.php",

                data: {

                    name: $("#name").val(),

                    email: $("#email").val(),

                    broad: $("#broad").val(),

                },

                dataType: "json",

                success: function(data){

                    var str=`

                        <div class='text'>

                            <h4>${data.name}<span>${data.email}</span></h4>

                            <p>${data.broad}</p>

                        </div>

                    `

                    $(".text-wrap").append(str);

                    $(".text").fadeIn(300, "swing");

                },

                error: function(){

                    $(".text-wrap").append("<div class='text2'>错误</div>");

                    $(".text2").fadeOut(2000);

                }

            })

        }else{

            return false;

        }

    })