

DataBase

山东大学计算机科学与技术学院数据库课程设计

实验目标

- PHP+MySQL实现博客管理系统

使用语言

- 后台—php
- 前端—html css js
- 相关知识可以在菜鸟教程学习，链接如下[菜鸟教程](#)

实验步骤

一. 环境搭建

- windows下配置apache+php环境,具体可参考
<https://blog.csdn.net/lgx1134569285/article/details/80646965>, 使用软件为PhpStorm,可在官网下载
- 参考资源, 使用课程提供的编程小梯子, 数据放到自己的Github主页了, 链接如下:
<https://github.com/NuanBaobao/DataBase>

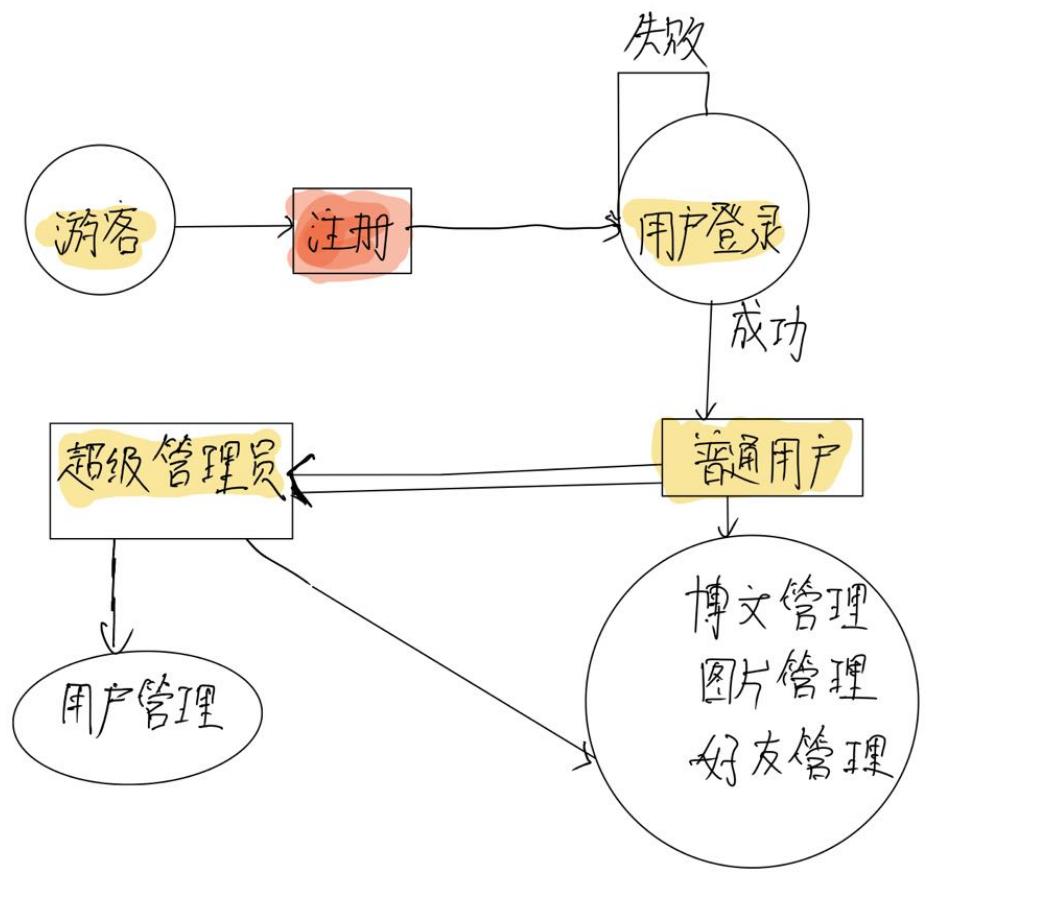
二.需求分析

- 随着Inernet的高速发展, 人与人之间的交流也在技术进步的同时发生着日新月异的变化。Blog(博客)是继Email之后出现的第4种网络交流方式, 它的出现改变了人们生活、工作和学习的方式。近些年来, 博客以不可阻挡之势高速发展, 它可以作为人们分享文章、照片和思想的绝佳平台。博客已成为新兴的网络媒介, 并扩展至销售、商业推广等主流应用。
- 如今, 博客已成为一种新的交流、沟通方式, 提供了一种可信任、实时连通的交流平台。博客充分发挥了网络开放性与交互性的特点, 使用户可在任何时间、任何地点, 通过网络方便地交流与沟通, 不仅可以实现信息的传递与获取, 还可以进行资源共享、展示自我, 为个人发展带来新的机遇。

三.数据库结构设计

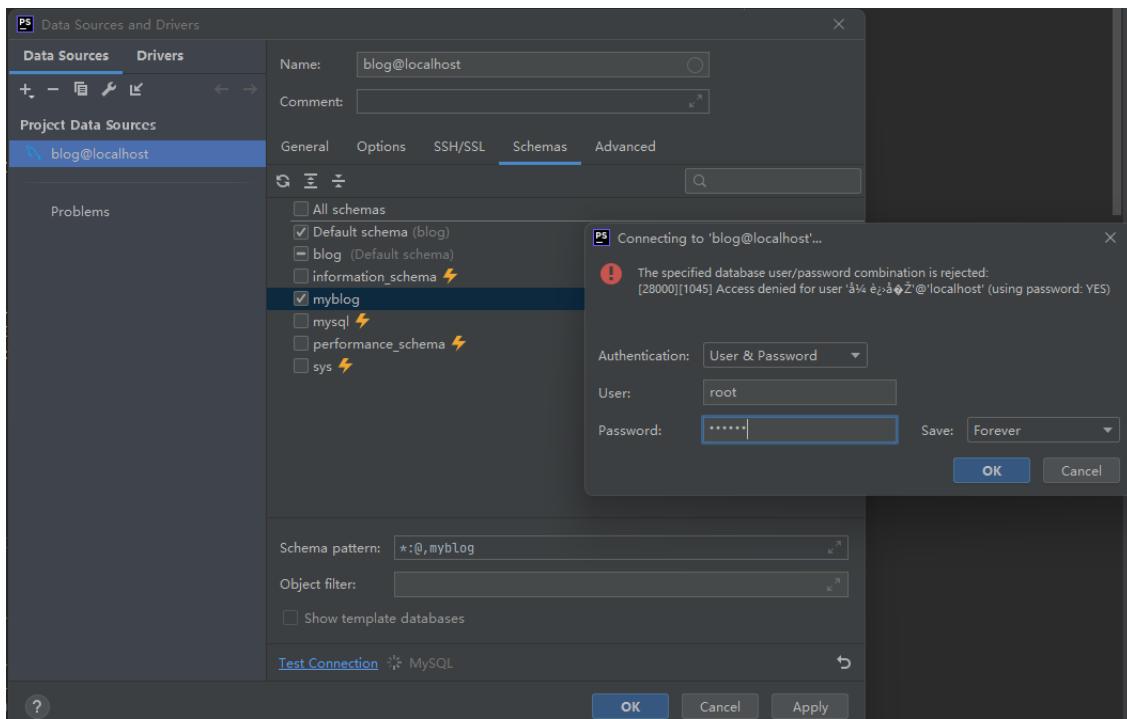
- 本系统将实现下列基本目标:
 - 普通用户可以发表删除博文、浏览博文、浏览图片、发表删除评论、图片上传删除
 - 超级管理员具备对普通用户的管理权限
- 博客管理系统主要由博文管理、图片管理、好友管理、用户管理模块组成
 - (1)博文管理模块主要由上传博文、浏览博文、查询博文、删除博文、评论查看添加、评论删除功能组成。
 - (2)图片管理模块主要由上传图片、浏览图片、删除图片功能组成。
 - (3)好友管理模块主要由添加好友、删除好友、查询好友功能组成。
 - (4)用户管理模块主要完成用户个人信息设置功能。
- 博客系统的功能结构如下
 - 图片管理
 - 上传图片 浏览图片 删除图片

- 好友管理
 - 添加好友 删除好友 查询好友
 - 用户管理
 - 个人信息设置
 - 博文管理
 - 添加评论 查看评论 删除评论 上传博文 浏览博文 删除博文 查询博文
- 大概流程如图所示



四.数据库的创建

- 在php中创建数据库比较简单，可以先用MySQL创建数据库，然后在这里引入，如图所示



五.学习 PHPMYSQL相关知识

1.ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE = utf8_general_ci ROW_FORMAT的解释

- 第一次看学长的代码，发现创建表时末尾有一行代码，看不太懂，了解如下

```
CREATE TABLE `books` (
  `isbn` int(10) NOT NULL,
  `title` varchar(50) DEFAULT NULL,
  `author` varchar(50) DEFAULT NULL,
  `publisher` varchar(50) DEFAULT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

- 1.ENGINE=InnoDB使用innodb引擎，从zhidaoMySQL 5.6开始默认使用该引擎
- 2.DEFAULT CHARSET=utf8 数据库默认编码为utf-8
- 还有其他参数，想要了解的可以看这个链接：[参考链接](#)

2.查看学长关于PHP MYSQL的代码之后，关于这部分的代码大体上有了了解，具体可以在菜鸟教程查看，附上链接

- 菜鸟教程关于PHPMysql的讲解[点击学习](#)

3.后面在做博文发表时发现我输出的时间与当前时间并不一致，这里找的资料如下

- 学长给的代码里是这样设置的，参考链接后设置如下，[mysql时区问题解决几种方法](#)

```
SET time_zone = "+00:00"; # 学长示例
set global time_zone = '+8:00'; # 修改mysql全局时区为北京时间，即我们所在的东8区
set time_zone = '+8:00'; # 修改当前会话时区
flush privileges; # 立即生效
```

六. 创建数据库模式

- 经过上面的步骤我们已经创建好数据库，现在我们建立与博客管理系统的模式，这部分我建立了五张表，分别为：
- 用户表，主要属性为注册id,注册姓名，注册密码，注册qq，注册邮箱，性别，是否登录，是否为超级管理员

```
-- 用户表adminuser
CREATE TABLE `adminuser` (
  `regid` int(11) UNSIGNED AUTO_INCREMENT NOT NULL,
  `regname` varchar(20) NOT NULL,
  `regpwd` varchar(100) NOT NULL,
  `regqq` varchar(20) DEFAULT NULL,
  `regemail` varchar(50) NOT NULL,
  `regex` varchar(10) DEFAULT NULL,
  `islock` int(11) NOT NULL DEFAULT 0,
  `fig` int(11) NOT NULL DEFAULT 0,
  PRIMARY KEY (`regid`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- 博文表，主要属性为博文id,类型，博文题目，内容，作者，提交时间

```
-- 博文表 article
CREATE TABLE `article` (
  `id` int(11) UNSIGNED AUTO_INCREMENT NOT NULL,
  `type_id` int(11) NOT NULL,
  `title` varchar(200) NOT NULL,
  `content` text NOT NULL,
  `author` varchar(20) NOT NULL,
  `pubtime` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- 评论表，主要属性为评论id，博文id,评论人，评论内容，评论时间

```
-- 评论表 comment
CREATE TABLE `comment` (
  `id` int(11) UNSIGNED AUTO_INCREMENT NOT NULL,
  `fileid` int(11) NOT NULL,
  `username` varchar(20) NOT NULL,
  `content` text NOT NULL,
  `comtime` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- 图片表，主要属性为图片id，图片名，作者,发表时间

```
-- 图片表 pictures
CREATE TABLE `pictures`(
  `id` int(11) UNSIGNED AUTO_INCREMENT NOT NULL,
  `picname` varchar(30) NOT NULL ,
  `user` varchar(50) NOT NULL ,
  `time` datetime NOT NULL,
  PRIMARY KEY (`id`)
)ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- 好友表，本人id，好友id，好友姓名

```
-- 好友表
CREATE TABLE `friends`(
  `id` int(11) NOT NULL,
  `fid` int(10) NOT NULL ,
  `fname` varchar(50) NOT NULL ,
  PRIMARY KEY (`id`)
)ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- 文章类型表，博文类型id,博文类型名称

```
-- 博文类型表 article_type
CREATE TABLE `article_type` (
  `type_id` int(11) UNSIGNED AUTO_INCREMENT NOT NULL,
  `type_name` varchar(30) NOT NULL,
  PRIMARY KEY (`type_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

- 创建好模式后执行将其真正创建，具体的执行过程如图所示：

```
myblog> CREATE TABLE `pictures`(
    `id` int(11) UNSIGNED AUTO_INCREMENT NOT NULL,
    `picname` varchar(30) NOT NULL ,
    `user` varchar(50) NOT NULL ,
    `time` datetime NOT NULL,
    PRIMARY KEY (`id`)
)ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
[2021-11-21 11:12:59] [HY000][1681] Integer display width is deprecated and will be removed in a future release.
[2021-11-21 11:12:59] completed in 16 ms
myblog> CREATE TABLE `friends`(
    `id` int(11) NOT NULL,
    `fid` int(10) NOT NULL ,
    `fname` varchar(50) NOT NULL ,
    PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4
[2021-11-21 11:12:08] [HY000][1681] Integer display width is deprecated and will be removed in a future release.
[2021-11-21 11:12:08] [HY000][1681] Integer display width is deprecated and will be removed in a future release.
[2021-11-21 11:12:08] completed in 16 ms
myblog> INSERT INTO `friends`(`id`, `fid`, `fname`) VALUES
(1,2,'zwh')
[2021-11-21 11:12:14] 1 row affected in 0 ms
```

- 创建好的关系模式如图所示

WHERE		ORDER BY				
		id	fileid	username	content	comtime
1	1	1	zjh		干的漂亮！	2021-11-17 10:23:14
2	2	1	zwh		写的挺好的，奥利给！	2021-11-18 23:39:09
3	3	2	yye		这是第二篇文章的第一条	2021-11-20 23:40:12
4	4	1	zjh		6666666666666!	0000-00-00 00:00:00
5	5	3	sq		第n次测试	2021-11-13 04:07:06
6	6	3	lbgj		第n+1次测试	2021-11-12 04:13:15
7	7	1	lyj		留下我的足迹	2021-11-10 03:44:10

七.数据库的连接

- 在自己的程序里如何连接数据库呢？这部分运用PHP MySQL实现比较简单,具体代码如下

```
<?php
const HOST = 'localhost';
const USER = 'root';
const PASS = '*****';
const DBNAME = 'myblog';
// 连接mysql
$link = @mysqli_connect(HOST, USER, PASS, DBNAME) or die("ERROR: CANNOT
CONNECT TO DATABASE!");
// 编码设置
mysqli_set_charset('utf8',$link);
```

八.学习HTML CSS相关知识

1.“#”的意思是链接当前页面,一般的格式是:

```
<a href="#"></a>
```

- href在是CSS代码的一种意思是指定超链接目标的URL。在HTML和jsp中作用一般为插入超链接，“#”是默认的当前页面，可以把#后面加上想要跳转的页面。

```
<a href="#" onclick="window.close()">关闭</a>
```

- 将href="#"是指链接到当前页面,其实是无意义的,页面也不会刷新,关键是后面的onclick="window.close()",当点击“关闭”的时候,会执行window.close () 代码。

2.使用HTML:target="_blank", 在新的页面中打开链接, 形成父子界面的关系。

- _blank -- 在新窗口中打开链接
- _parent -- 在父窗体中打开链接
- _self -- 在当前窗体打开链接,此为默认值
- _top -- 在当前窗体打开链接, 并替换当前的整个窗体(框架页)

3.nav元素并不是视觉上的导航栏，而是具有导航含义的标签,元素表示一个包含多个链接的区域，这些链接指向其他页面或本页面中的其他部分,通常一个页面导航可以这么写：

- 写法1

```
<nav>
<ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">About</a></li>
    <li><a href="#">Content</a></li>
</ul>
</nav>
```

- 写法2

```
<div class="nav">
    <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">About</a></li>
        <li><a href="#">Content</a></li>
    </ul>
</div>
```

- 写法3

```
<nav>
    <h1>Navigation</h1>
    <p>Welcome to my website.
    <a href="/blog">my blog</a>
    You can find my books and papers in the
    <a href="/publications">publication</a>
    section. If you are interested in my work...
    <a href="/contact">here</a>
    .</p>
</nav>
```

- 原文链接：<https://blog.csdn.net/ziquqileccc/article/details/103260091>

4.CSS选择器

- 元素选择器根据元素名称来选择 HTML 元素,在这里，页面上的所有元素都将居中对齐，并带有红色文本颜色：

```
p {
    text-align: center;
    color: red;
}
```

- id 选择器,使用 HTML 元素的 id 属性来选择特定元素,元素的 id 在页面中是唯一的，因此 id 选择器用于选择一个唯一的元素！要选择具有特定 id 的元素，写一个井号 (#) ，后跟该元素的 id, 这条 CSS 规则将应用于 id="para1" 的 HTML 元素：

```
#para1 {
    text-align: center;
    color: red;
}
```

- CSS 类选择器选择有特定 class 属性的 HTML 元素。如需选择拥有特定 class 的元素，写一个句点(.) 字符，后面跟类名。在此例中，所有带有 class="center" 的 HTML 元素将为红色且居中对齐：

```
.center {
    text-align: center;
    color: red;
}
```

HTML 元素也可以引用多个类,类名不能以数字开头!

- 查看表格如下，[详细信息可点击这个链接查看](#)

选择器	例子	例子描述
<u>.class</u>	.intro	选取所有 class="intro" 的元素。
<u>#id</u>	#firstname	选取 id="firstname" 的那个元素。
<u>*</u>	*	选取所有元素。
<u>element</u>	p	选取所有 <p> 元素。
<u>element,element,..</u>	div, p	选取所有 <div> 元素和所有 <p> 元素。

- CSS样式的优先级定义如下，页面中的所有样式将按照以下规则“层叠”为新的“虚拟”样式表，其中第一优先级最高：
 1. 行内样式（在 HTML 元素中）
 2. 外部和内部样式表（在 head 部分）
 3. 浏览器默认样式

5.background属性

- 其他属性很好理解，主要是background-attachment 属性指定背景图像是应该滚动还是固定的（不会随页面的其余部分一起滚动）

属性	描述
<u>background</u>	在一条声明中设置所有背景属性的简写属性。
<u>background-attachment</u>	设置背景图像是固定的还是与页面的其余部分一起滚动。
<u>background-clip</u>	规定背景的绘制区域。
<u>background-color</u>	设置元素的背景色。
<u>background-image</u>	设置元素的背景图像。
<u>background-origin</u>	规定在何处放置背景图像。
<u>background-position</u>	设置背景图像的开始位置。
<u>background-repeat</u>	设置背景图像是否及如何重复。
<u>background-size</u>	规定背景图像的尺寸。

5.CSS 外边距

- CSS margin 属性用于在任何定义的边框之外，为元素周围创建空间。通过 CSS，可以完全控制外边距。有一些属性可用于设置元素每侧（上、右、下和左）的外边距，CSS 拥有用于为元素的每一侧指定外边距的属性
 - margin-top margin-right margin-bottom margin-left
- 还可以为 margin 设置一个百分比数值，百分数是相对于父元素的 width 计算的。

6.CSS 定位属性

- CSS 定位属性允许对元素进行定位

属性	描述
position	把元素放置到一个静态的、相对的、绝对的、或固定的位置中。
top	定义了一个定位元素的上外边距边界与其包含块上边界之间的偏移。
right	定义了定位元素右外边距边界与其包含块右边界之间的偏移。
bottom	定义了定位元素下外边距边界与其包含块下边界之间的偏移。
left	定义了定位元素左外边距边界与其包含块左边界之间的偏移。
overflow	设置当元素的内容溢出其区域时发生的事情。
clip	设置元素的形状。元素被剪入这个形状之中，然后显示出来。
vertical-align	设置元素的垂直对齐方式。
z-index	设置元素的堆叠顺序。

7.display属性

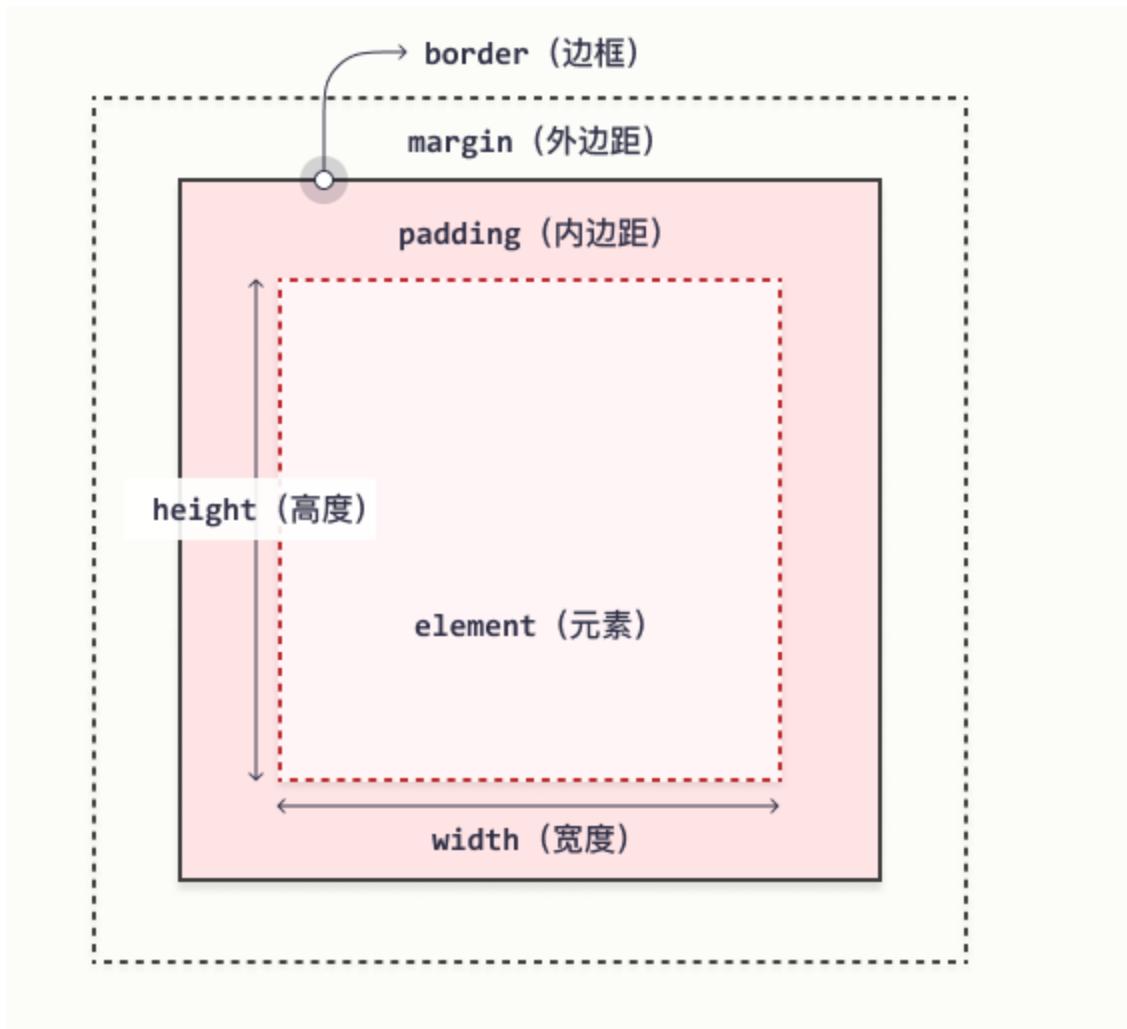
- 用于规定元素生成的框类型，影响显示方式，这个属性讲解有点多，可以参考下面的链接：[display 详解](#)

8.CSS 内边距

- CSS padding 属性用于在任何定义的边界内的元素内容周围生成空间，通过 CSS，可以完全控制内边距（填充）。有一些属性可以为元素的每一侧（上、右、下和左侧）设置内边距。CSS 拥有用于为元素的每一侧指定内边距的属性：
 - padding-top padding-right padding-bottom padding-left

9.CSS 框模型

- 所有 HTML 元素都可以视为方框。在 CSS 中，在谈论设计和布局时，会使用术语“盒模型”或“框模型”。CSS 框模型实质上是一个包围每个 HTML 元素的框。它包括：外边距、边框、内边距以及实际的内容。下图展示了框模型：



10.元素的宽度和高度

- 使用 CSS 设置元素的 width 和 height 属性时，只需设置内容区域的宽度和高度。要计算元素的完整大小，还必须把内边距、边框和外边距加起来。

11.文本属性

- 关于文本属性设置如下

属性	描述
<u>color</u>	设置文本颜色。
<u>direction</u>	指定文本的方向 / 书写方向。
<u>letter-spacing</u>	设置字符间距。
<u>line-height</u>	设置行高。
<u>text-align</u>	指定文本的水平对齐方式。
<u>text-decoration</u>	指定添加到文本的装饰效果。
<u>text-indent</u>	指定文本块中首行的缩进。
<u>text-shadow</u>	指定添加到文本的阴影效果。
<u>text-transform</u>	控制文本的大小写。
<u>text-overflow</u>	指定应如何向用户示意未显示的溢出内容。
<u>unicode-bidi</u>	与 direction 属性一起使用，设置或返回是否应重写文本来支持同一文档中的多种语言。
<u>vertical-align</u>	指定文本的垂直对齐方式。
<u>white-space</u>	指定如何处理元素内的空白。
<u>word-spacing</u>	设置单词间距。

12.CSS 字体属性

- 关于字体属性设置如下

属性	描述
<u>font</u>	简写属性。在一条声明中设置所有字体属性。
<u>font-family</u>	规定文本的字体系列（字体族）。
<u>font-size</u>	规定文本的字体大小。
<u>font-style</u>	规定文本的字体样式。
<u>font-variant</u>	规定是否以小型大写字母的字体显示文本。
<u>font-weight</u>	规定字体的粗细。

13.盒子阴影box-shadow

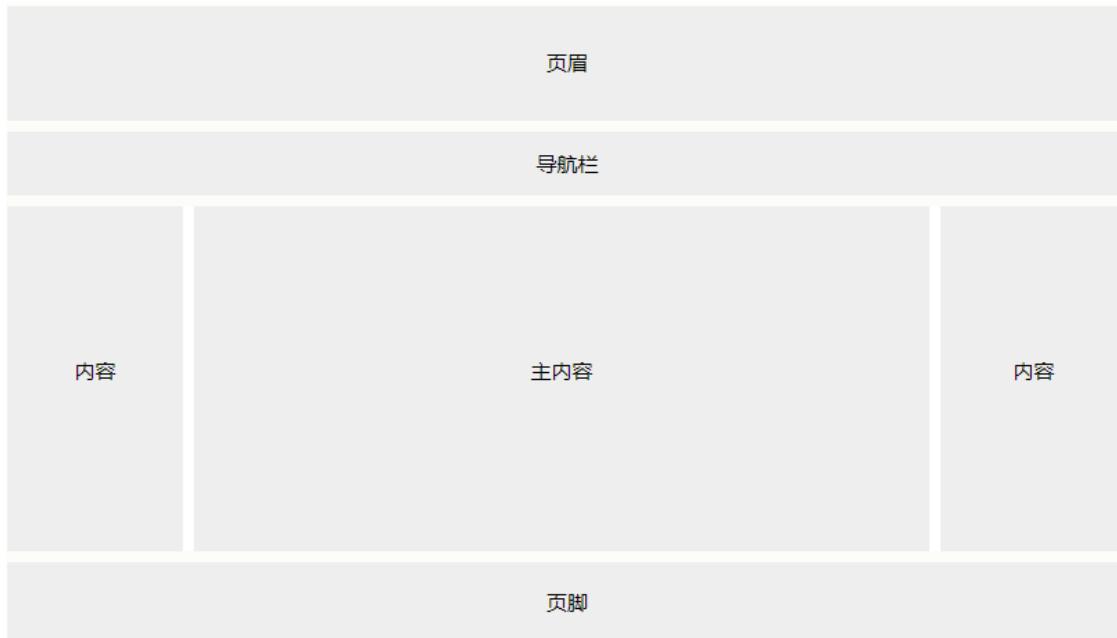
- box-shadow属性向box添加一个或多个阴影，语法：
 - box-shadow: offset-x offset-y blur spread color inset;
 - 词解释：blur:模糊 spread:伸展 inset:内凹
- 参数解释：
 - offset-x: 必需，取值正负都可。offset-x水平阴影的位置。
 - offset-y: 必需，取值正负都可。offset-y垂直阴影的位置。
 - blur:可选，只能取正值。blur-radius阴影模糊半径，0即无模糊效果，值越大阴影边缘越模糊。
 - spread: 可选，取值正负都可。spread代表阴影的周长向四周扩展的尺寸，正值，阴影扩大，负值阴影缩小。
 - color:可选。阴影的颜色。如果不设置，浏览器会取默认颜色，通常是黑色，但各浏览器默认颜色有差异，建议不要省略。可以是rgb(250,0,0)，也可以是有透明值的rgba(250,0,0,0.5)。
 - inset:可选。关键字，将外部投影(default outset)改为内部投影。inset 阴影在背景之上，内容之下。
- 原文链接：https://blog.csdn.net/qq_44607694/article/details/89605857

14.HTML中type类型常用种类

- 这里有几个注册时常用的[HTML中type类型常用种类介绍](#)

九.网页搭建

- 网页通常分为页眉、菜单、内容和页脚：



- 参考网页制作链接
- <https://www.w3cschool.cn/html/html-template.html>

1.页眉与导航栏设置

- 这里我将页眉与导航栏合并为一个，实现效果如下，这里在点击首页 登录 注册 关于时会弹出相应界面，并且已登录后还会显示个人中心以及退出按钮，但是相关的连接代码还未实现，暂时用本地页面链接实现（2021-11-19）



localhost:83342/Primitive/php-blog-master/header.php?_t=4e95cb83ccge03299vmoj5z7ed8_ll_reload+RELOAD_ON_SAVE[ER]

- 后期发现这个界面有点单调，和登陆界面一样加上背景就好看多了，效果图



- 页眉导航栏HTML实现代码如下，导航栏的CSS样式header.css借鉴了网上的代码修改而成：[参考链接1](#), [参考链接2](#) 在文末有链接可自行下载

```
<head>
    <link rel="stylesheet" type="text/css" href="./css/header.css">
</head>
<?php
    @session_start();
?>
<body>
    <div class="main-header">
        <div class="container">
            <div class="navigation">
                <div class="info">
                    <nav class="menu">
                        <ul class = "menu-one" >
                            <li><a href=".//front_page.php">首页</a> <!-- 此处
实现点击返回首页，未实现 -->
                            <?php
                                if (!empty($_SESSION["user_name"])) { # 如果
用户已登录，则显示个人中心与退出按钮
                                    echo "<li><a href='./user_center.php'>个
人中心</a></li>";
                                    echo "<li><a href='./user_exit_sql.php'>
退出</a></li>";
                                }
                                else { # 如果用户未登录，则显示登录与注册按钮
                                    echo "<li><a href='./login.php'>登陆</a>
";
                                    echo "<li><a href='./sign.php'>注册</a>
";
                                }
                            <?>
                        </ul>
                    </nav>
                </div>

                <a href="#">
                    <h2>数据无限</h2><br>
                    <span>人生只有贪心，没有动态规划</span>
                </a>
            </div>
```

```
</div>
</div>
</body>
```

2.页脚界面设置

- 页脚部分实现较为简单，我主要是加了对于山东大学和计算机学院的超链接，实现效果如下
(2021-11-19)



- 页脚部分的HTML代码如下，CSS样式footer.css在文末链接中给出，可自行下载

```
<head>
    <link rel="stylesheet" type="text/css" href=".//css/footer.css">
</head>
<div class="foot">
    <footer>
        <div id="footer" >
            <div class="copy-info">
                <p>
                    <a href="https://www.sdu.edu.cn/" target="_blank">山东大学
                </a>
                    <a href="https://www.cs.sdu.edu.cn/" target="_blank">计算
                机科学与技术学院</a>
                    <br>@数据库系统概念课程设计出品
                </p>
            </div>
        </div>
    </footer>
</div>
```

2.登录界面设置

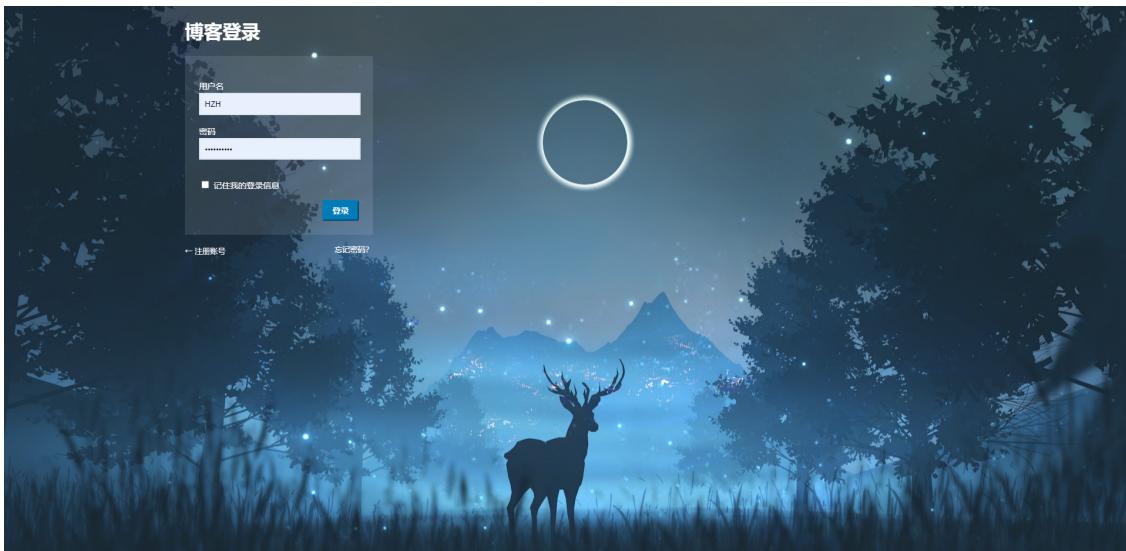
- 博客登陆界面可以按照平时所习惯的登陆界面设计，网上有很多样式，可以仿照修改，我主要参考了下面几个博客

[漂亮的登录界面中的css示意图](#)

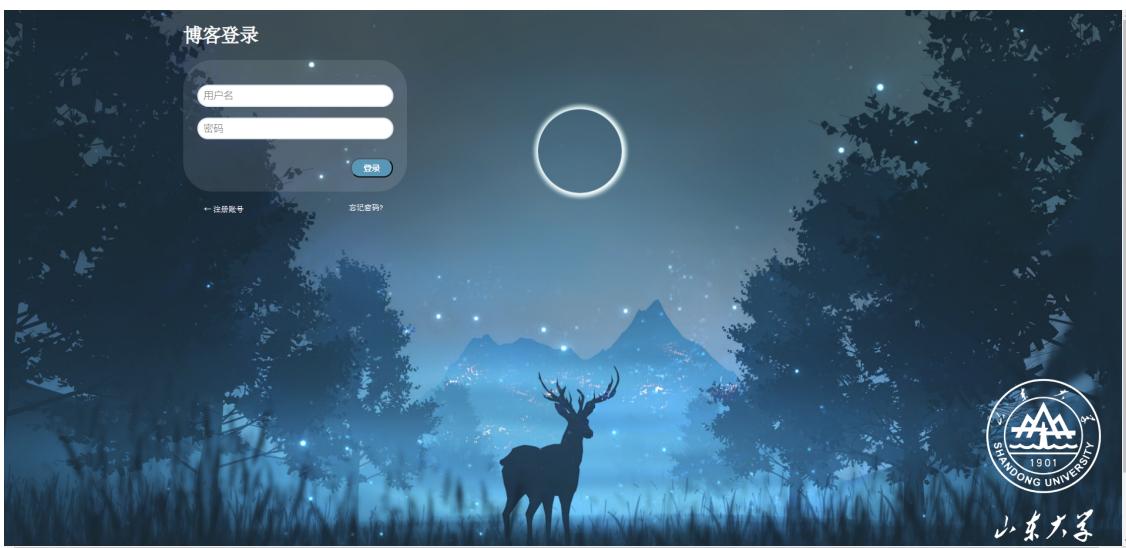
[HTML+CSS实现炫酷的登录界面](#)

[一个简约精美的登录页面\(HTML+CSS\)](#)

- 实现效果如下，用户输入用户名和密码后可以登录，若没有账号可返回注册界面，若忘记密码则执行相应操作（暂时还没想好怎么处理），可以选择记录登入信息，存入到缓存中 (2021-11-20)



- 课上看了朋友的界面，再次对界面进行修改美化，实现效果如下



- 登陆界面HTML实现代码如下，登陆界面CSS样式login.css部分文末有源代码链接，如有需要可点击下载(2021-11-19)

```
<?php
@session_start();
if (!empty($_SESSION["user_name"])) {
    header("Location:./index.php");
    exit();
}
?>
<!DOCTYPE html>
<head>
    <meta charset="UTF-8">
    <title>博客登录</title>
    <link href=".//css/login.css" rel='stylesheet' type='text/css' media="all">
</head>
<body>
<div class="bg">
    <div class = "blog">
        
    </div>
    <div class = "login">
```

```

<h1> 博客登录 </h1>
<form name="login form" action="./login_sql.php" method="POST"> <!--
-登陆后修改数据库内容-->
<p>
<!--          <label for="user_login">用户名</label>-->
<input type="text" name="user_name" id="user_login"
class="input" value="" size="20" required="required" placeholder="用户名"/>

<!--          <label for="user_pass">密码</label>-->
<input type="password" name="user_pwd" id="user_pass"
class="input" value="" size="20" required="required" placeholder="密码"/>
</p>

<p class="submit">
    <input type="submit" name="submit" class="button button-primary" value="登录" /> <!--登陆界面-->
</p>
</form>
<div class = "footer">
    <p class="nav"><a href="#">忘记密码? </a></p> <!--触法相应处理机制,
此处先用本地页面代替 -->
    <p class="backsign"><a href=".//sign.php"> &larr; 注册账号</a></p>
<!-- 触发注册账号界面 -->
</div>
</div>
</body>
</html>
<?php require("./footer.php"); ?>

```

3.注册界面设置

- 注册界面是在首页的基础上实现的,此内容界面上包括了前面实现的页眉，导航栏与页脚，实现效果如下，点击登录会跳转到登陆界面 (2021-11-21)



- 改了登陆界面后感觉这个注册界面也不是很好看，这里给输入框加了hover属性，鼠标移动到输入框时改变背景色，修改后效果图如下 (2021-11-22)



- 注册界面HTML实现代码如下，登陆界面CSS样式edit_user.css部分文末有源代码链接，如有需要可点击下载

```
<?php
require("./header.php");
?>
<!DOCTYPE html>
<html>
<head>
    <title>注册</title>
    <link rel="stylesheet" type="text/css" href="./css/sign.css">
</head>
<div class="user_information">
    <div class="appointment">
        <form action="./sign_sql.php" method="POST"> <!--接受表单变量的值，执行sign_sql.php完成相应的sql操作-->
            <div class="main">
                <div class="form-left">
                    <!--          <label for = "name">账号</label>-->
                    <input type="text" name="user_name" id="name"
placeholder="账号" required="required" autocapitalize="off"> <!--标记标签变量值-->
                </div>
                <div class="form-right">
                    <!--          <label for="email">邮 箱</label>-->
                    <input type="email" name="user_email" id="email"
placeholder="邮箱" required="required" autocapitalize="off">
                </div>
            </div>

            <div class="main">
                <div class="form-left">
                    <!--          <label for="user_pass">输入密 码</label>-->
                    <input type="password" name="user_pwd" id="user_pass"
class="input" placeholder="输入密码" required="required"
autocapitalize="off">
                </div>
                <div class="form-right ">
                    <!--          <label for="reuser_pass">确认密 码</label>-->
                    <input type="password" name="reuser_pwd"
id="reuser_pass" class="input" placeholder="确认密码" required="required"
autocapitalize="off">
                </div>
            </div>
        </form>
    </div>
</div>
```

```

        </div>
    </div>

    <div class="main">
        <div class="form-left">
            <!-- 性别 -->
            <label for="sex">性 别</label>-->
            <input type="text" name="user_sex" class="input" id="sex" placeholder="性别" required="required" autocapitalize="off">
        </div>
        <div class="form-right ">
            <!-- QQ -->
            <label for="qq">Q Q</label>-->
            <input class="bottom" type="number" id="qq" name="user_qq" placeholder="QQ" required="required" autocapitalize="off">
            <div class="clearfix"></div>
        </div>
    </div>

    <div class="bttn">
        <input type="submit" name="sign" value="注册">
    </div>
    </form>
</div>
<?php require( "./footer.php"); ?> <!-- 页脚 -->
</html>

```

4. 用户中心界面设置

- 用户中心部分界面要根据不同的用户显示不同的界面，超级管理员具备管理用户的功能，可以删除普通用户的博客，评论与图片，普通用户可以进行博文浏览上传删除，评论发表删除，图片上传删除的功能，开发中(2021-11-21)，这里是最后实现的效果



- 这里发现一个获取常见logo的网站，可以找到主题对应的图标 [Logo素材图片](#)
- 经过两天的努力心心念念的个人中心界面终于做好了，比上面的有了很大的改进，不说了，放效果图，不过上面的链接操作目前只完成了退出登录的功能，其他还是空壳子哈哈！(2021-11-23)



- 关于用户中心的实现代码如下

```
<?php
    // 处理用户中心页面
    require("./login_session_check.php");
    require("./dbConfig.php");
    require("./header.php");
    // 连接mysql
    $link = @mysqli_connect(HOST, USER, PASS, DBNAME) or die("ERROR: CANNOT
    CONNECT TO DATABASE!");
    // 编码设置
    mysqli_set_charset('utf8', $link);
    @session_start();
    $name = $_SESSION['user_name'];

    $sql = "select * from adminuser where regname = '$name' and fig = 1";
    $result = mysqli_query($link, $sql);
    $admin = mysqli_num_rows($result);

    mysqli_free_result($result);
    mysqli_close($link);
?>

<!DOCTYPE html>
<html>
<head>
    <title>个人中心</title>
    <link rel="stylesheet" type="text/css" href="./css/user_center.css">
</head>
<div class="user_center">
    <div class="big_menu">
        <a href="./post_article.php">
            <div class="center_menu">
                <h3>发布博文<br><br></h3></div>
            </a>
        <a href="./post_pictures.php"><div class="center_menu">
            <h3>图片上传<br><br></h3></div>
        </a>
```

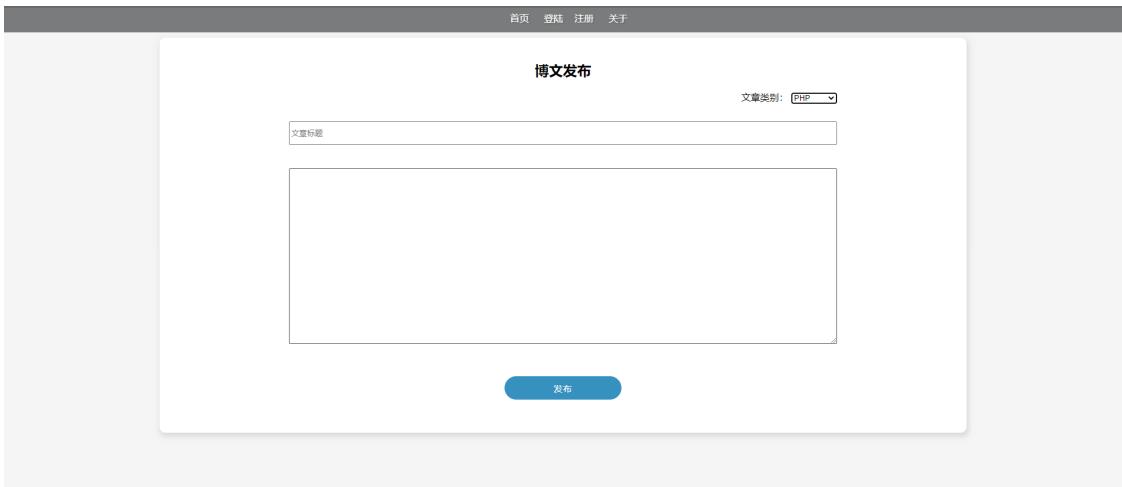
```

<a href="../manager_friend.php"><div class="center_menu">
    <h3>好友管理<br><br></h3></div>
</a>
<?php
    if ($admin > 0 ) { // 超级管理员
        echo "<a href='./manager_user.php'><div
class='center_menu'>
            <h3>管理用户<br><br><img src='images/13.png' alt='管理用户'
width='100'></h3></div>
            </a>";
    }
?
<a href="../send_comment.php"><div class="center_menu">
    <h3>发布评论<br><br></h3></div>
</a>
<a href="../user_exit_sql.php"><div class="center_menu">
    <h3>退出登录<br><br></h3></div>
</a>
</div>
</div>
<?php require("./footer.php"); ?>
</html>

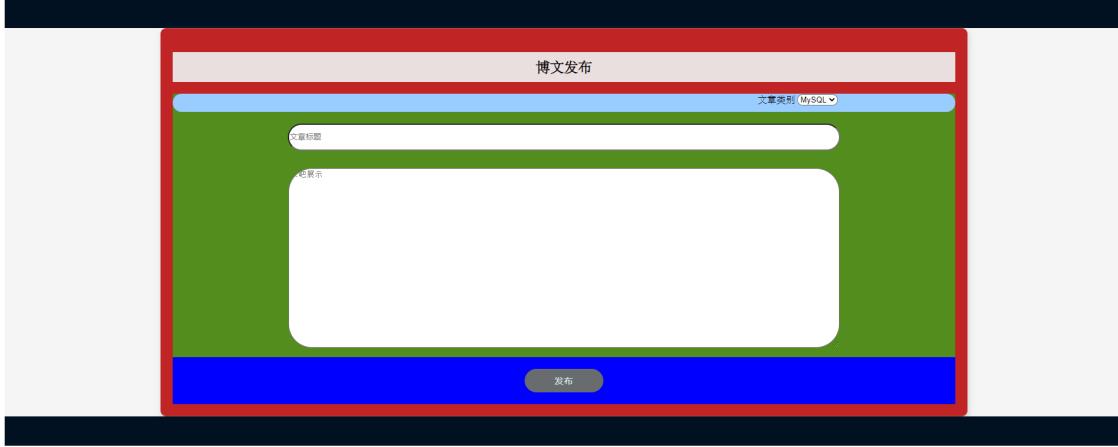
```

5. 用户发表博文界面设置

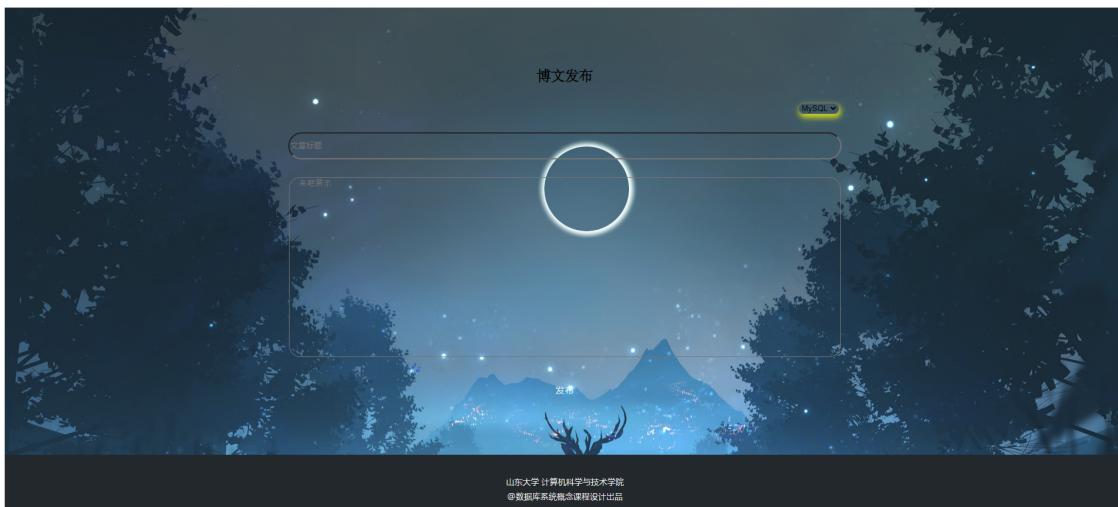
- 用户发表博文界面主要实现用户发布自己的博文，实现功能比较简单，效果如下



- 这里我觉得是一个比较重要的点。我觉得，之前写CSS样式的时候发现有些参数不好写，做出来的div元素块位置大小老是出错，这里我自己是这样写的，每次定义不同的背景色，调好后删除，实现效果如下



- 后期做好后的效果是这样的，好太多了！



6. 博文浏览界面设置

- 博文浏览界面主要实现对挑选出的博文进行展示并输出评论框和别人已经发布的评论，这部分其实是由三部分组成，第一部分为博文展示

PHP简介

发布时间：2021-11-19 13:45:59 作者：ZJH

您应当具备的基础知识在继续学习之前，您需要对以下知识有基本的了解：HTML CSS 如果您希望首先学习这些项目，请在我们的 首页 访问这些教程。 PHP 是什么？ PHP（全称：PHP: Hypertext Preprocessor，即“PHP：超文本预处理器”）是一种通用开源脚本语言。 PHP 脚本在服务器上执行。 PHP 可免费下载使用。 lamp PHP 对初学者而言简单易学。 PHP 也为专业的程序员提供了许多先进的功能。 PHP 文件是什么？ PHP 文件可包含文本、HTML、JavaScript(代码) 和 PHP 代码。 PHP 代码在服务器上执行，结果以纯 HTML 形式返回给浏览器。 PHP 文件的默认文件扩展名是 ".php"。 PHP 能做什么？ PHP 可以生成动态页面内容。 PHP 可以创建、打开、读取、写入、关闭服务器上的文件。 PHP 可以收集单数据。 PHP 可以发送和接收 cookies。 PHP 可以添加、删除、修改您的数据库中的数据。 PHP 可以限制用户访问您的网站上的一些页面。 PHP 可以加密数据。 通过 PHP，您不再限于输出 HTML，您可以输出图像、PDF 文件，甚至 Flash 电影。 您还可以输出任意的文本，比如 XHTML 和 XML。 为什么使用 PHP？ PHP 可在不同的平台上运行（Windows、Linux、Unix、Mac OS X 等）。 PHP 与目前几乎所有的正在被使用的服务器相兼容（Apache、IIS 等）。 PHP 提供了广泛的数据库支持。 PHP 是免费的。 可从官方的 PHP 资源下载它：www.php.net PHP 易于学习，并可有效地运行在服务器端。

- 初始化展示博文为id=1,之后执行相应的sql从表中查找所有博文并展现，实现代码如下

```
<?php
// 展现用户发布博文页面
require "dbConfig.php";
//require("./header.php");

// 连接mysql
$link = @mysqli_connect(HOST, USER, PASS, DBNAME) or die("ERROR: CANNOT
CONNECT TO DATABASE!");
// 编码设置
mysqli_set_charset('utf8',$link);
```

```

$sql = "select * from article";           // 选择所有的博文
$result = mysqli_query($link,$sql);
$res_num = mysqli_num_rows($result);

$blog_id = intval($_GET["blog_id"]);        // 查看博文id
if (empty($blog_id) || $blog_id > $res_num) {
    header("location:./blog_article.php?blog_id=1");
    exit();
}

$sql = "select * from article where id = $blog_id "; // 获取文章
$result = mysqli_query($link,$sql);
$article_info = mysqli_fetch_array($result);

?>
<!DOCTYPE html>
<html>
<head>
    <title>文章内容</title>
    <link rel="stylesheet" type="text/css" href=".//css/blog_article.css">
</head>
<div class="bg">
    <div class="bloginfo">
        <div class="article_title">
            <h2><?php echo $article_info['title']; ?></h2>
            <div class="author_time">
                <?php
                    echo '<a>发布时间: '.$article_info['pubtime'].'</a>';
                    echo '<a>作者: '.$article_info["author"].'</a>';
                ?>
            </div>
            <p><?php echo $article_info['content']; ?></p>
        </div>
    </div>
</div>
<?php include("./comment.php") ?> <!--评论展示-->
<?php require("./footer.php"); ?>
</html>

```

- 第二部分为发布评论区，这部分界面比较简单，主要在于执行评论的插入操作，实现界面如下

- 这部分界面实现代码如下，实现起来还是很容易

```

<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href=".//css/send_comment.css">
</head>

```

```

<?php
    @session_start();
    $_SESSION['fileid'] = $fileid;
?>



<div class="comment_board">
        <form action=".//send_comment_sql.php" method="POST" >
            <textarea id="message" name="message" required="" placeholder=" 留下你的足迹吧"></textarea>
            <input type='text' hidden='hidden' name='fileid' value="<?
php echo $fileid ;?>">
            <div class="bttn">
                <?php
                    if (!empty($_SESSION['user_name'])) {
                        echo "";
                        echo "<input type='submit' id='edit' value='发表评论'>";
                    }
                    else {
                        echo "<input type='button' ";
                        echo "onclick=\"";
                        echo 'window.open("./login.php")';
                        echo "\"";
                        echo "id='login' value='请先登陆'>";
                    }
                ?>
            </div>
        </form>
    </div>
</div>
</html>


```

- 第三部分为展示已发布的关于该博文的评论，目前实现界面很单一，只是将其展现，没有界面效果
(2021-11-23)

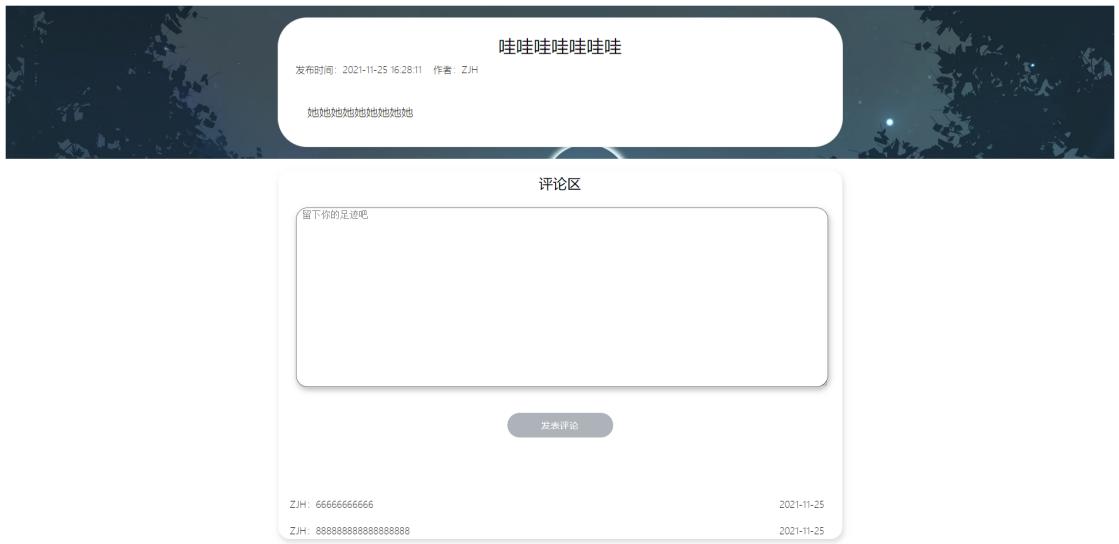
ZJH: 666666666666

2021-11-25

ZJH: 888888888888888888

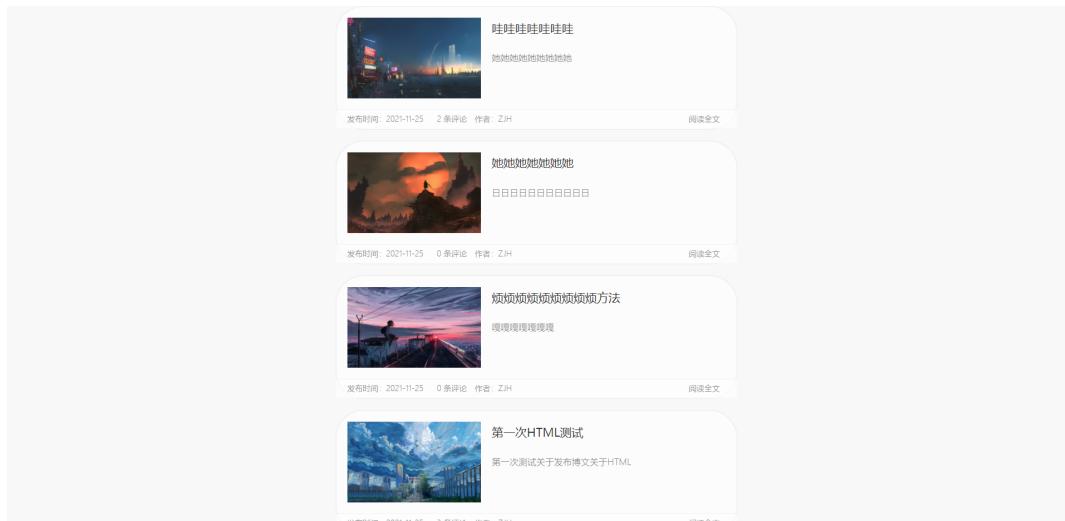
2021-11-25

- 最终添加CSS样式后的效果如图，感觉还是很可以的哈哈 (2021-11-26)



7. 主界面设置

- 目前我们已经基本实现了我们的功能，在这里添加主界面，展示最新发布的博文及其相关信息
- 主界面实现效果如下，可以将最先发布的五条博文展示出来，参考资源：[用HTML+CSS做一个漂亮简单的个人网页](#)



- 这部分实现过程要查询数据库，统计该博文的发布时间与评论次数，实现代码如下 (2021-11-25)

```

<head>
    <link rel="stylesheet" type="text/css" href=".//css/main.css">
</head>
<div class="main_top"></div>
<?php
    require "dbConfig.php";
    require ".//header.php";
    // 连接mysql
    $link = @mysqli_connect(HOST, USER, PASS, DBNAME) or die("ERROR: CANNOT
CONNECT TO DATABASE!");
    // 编码设置
    mysqli_set_charset('utf8',$link);

    $sql = "select * from article order by pubtime desc limit 0,5"; // 展现最
先发布的博文
    $result = mysqli_query($link,$sql);

```

```

while ($new_blog=mysqli_fetch_array($result)) {
?>
<div class="main-main">
    <div class="new-blog">
        <div class="cover">
            <?php echo "<a href='./blog_article.php?
blog_id=".$new_blog["id"]." '>" ?>
            
        >" height="100%;"/>
        </a>
    </div>
    <div class="inner-new">
        <header class="header-new">
            <h2 class="title-new">
                <?php echo "<a href='./blog_article.php?
blog_id=".$new_blog["id"]." '>"; ?>
                <?php echo $new_blog['title'] ?>
            </a>
            </h2>
        </header>
        <div class="content-new">
            <?php echo "<a href='./blog_article.php?
blog_id=".$new_blog["id"]." '>"; ?>
            <p> <!-- 限制输出内容 -->
            <?php
                if (strlen($new_blog['content'])>60) {
                    echo mb_substr($new_blog['content'],0,60,'UTF-
8')."...";
                }
                else { echo $new_blog['content']; }
            ?>
            </p>
            </a>
        </div>
        </div>
        <div class="meta-new">
            <span class="pull-left">
                <a class="release-date">
                    <?php echo '发布时
间: '.mb_substr($new_blog['pubtime'],0,10,'UTF-8').''; ?>
                </a>
                <a href="#">#comments

```

```

        </span>
        <span class="pull-right">
            <?php echo "<a href='./blog_article.php?
blog_id=".$new_blog["id"]." title='阅读全文'>"; ?>
                阅读全文
            </a>
        </span>
    </div>
</div>
<?php } ?>
<div class="bottom"></div>
</div>

```

十.数据库操作命令

- 上面我们已经实现了部分网页的代码，那么接下来就应该编写相应的SQL程序

1.用户注册程序

- 首先我们要创建一个用户，将其注册信息加入对应表中，用户注册完之后返回登录界面，注册过程首先要判断用户是否已存在，密码是否正确，代码如下:(2021-11-21)

```

<?php
    // 处理用户登录页面
    require "dbConfig.php";
    // 连接mysql
    $link = @mysqli_connect(HOST, USER, PASS, DBNAME) or die("ERROR: CANNOT
CONNECT TO DATABASE!");
    // 编码设置
    mysqli_set_charset('utf8',$link);

    //1.取提交的数据
    $name = $_POST["user_name"];           // 用户姓名

    $check_sql = "select * from adminuser where regname = $name;";
    $check_res = mysqli_query($link,$check_sql);
    $row_user = mysqli_num_rows($check_res);

    if ($row_user>0) {                   // 用户已注册
        echo "<script type='text/javascript'>";
        echo "alert('用户名已存在');";
        echo "location.href='sign.php';";
        echo "</script>";
    }

    $email = $_POST['user_email'];         // 用户邮箱
    $pwd = $_POST['user_pwd'];             // 用户密码
    $repwd = $_POST['reuser_pwd'];         // 用户确认密码
    $sex = $_POST['user_sex'];             // 用户性别
    $qq = $_POST['user_qq'];               // 用户QQ

```

```

if ($pwd <> $repwd) { // 用户两次密码不一致
    echo "<script type='text/javascript'>";
    echo "alert('两次输入密码不一致! ');";
    echo "location.href='sign.php';";
    echo "</script>";
}

// 插入数据
$sign_sql = "INSERT INTO adminuser (regname, regpwd, regqq, regemail,
regex, islock, fig)
VALUES ('$name', '$pwd', '$qq', '$email', '$sex', 0, 0);";

$sign = mysqli_query($link,$sign_sql);
if($sign) { // 注册成功跳转到登陆界面
    echo "<script type='text/javascript'>";
    echo "alert('注册成功');";
    echo "location.href='login.php';";
    echo "</script>";
}
else{
    echo "<script type='text/javascript'>";
    echo "alert('注册失败');";
    echo "location.href='sign.php';"; // 注册失败重新返回注册界面
    echo "</script>";
}
mysqli_close($link);
?>

```

2.用户登录程序

- 基于上面的注册信息，此时我们可以登录我们的博客系统，登陆成功后的界面还未实现（2021-11-21）

```

<?php
// 处理用户登录页面
require "dbConfig.php";
// 连接mysql
$link = @mysqli_connect(HOST, USER, PASS, DBNAME) or die("ERROR: CANNOT
CONNECT TO DATABASE!");
// 编码设置
mysqli_set_charset('utf8',$link);

// 取提交的数据
$name = $_POST["user_name"]; // 用户姓名
$pwd = $_POST["user_pwd"]; // 用户密码

$sql = "select * from adminuser where regname = '$name' and islock =
0;";
$result = mysqli_query($link,$sql);

if(mysqli_num_rows($result)>0){
    $row=mysqli_fetch_assoc($result);
    if ($pwd == $row['regpwd']){ // 判断密码是否正确
        // 登陆后修改登录状态
        $sql = "update adminuser

```

```

        set islock = 1
        where regname = '$name';";

        @session_start();
        $_SESSION["user_name"] = $name;

        echo "<script type='text/javascript'>";
        echo "alert('登录成功');";
        // 用户中心界面
        echo "location.href='user_center.php';";
        echo "</script>";
    }
    else
    {
        echo "<script type='text/javascript'>";
        echo "alert('密码不正确');";
        echo "location.href='login.php';";
        echo "</script>";
    }
}
else
{
    echo "<script type='text/javascript'>";
    echo "alert('用户名不正确或此用户已登录');";
    echo "location.href='login.php';";
    echo "</script>";
}

mysqli_free_result($result);
mysqli_close($link);
?>

```

3. 用户退出

- 用户退出系统这部分代码感觉还是很简单，将数据库中该人的是否登录选项置为0，并返回登陆界面(2021-11-22)

```

<?php
// 处理用户退出页面
require "dbConfig.php";
// 连接mysql
$link = @mysqli_connect(HOST, USER, PASS, DBNAME) or die("ERROR: CANNOT
CONNECT TO DATABASE!");
// 编码设置
mysqli_set_charset('utf8',$link);

@session_start();
// 取提交的数据
$name = $_SESSION["user_name"]; // 用户姓名

$sql = "update adminuser
        set islock = 0
        where regname = '$name';";

if(mysqli_query($link,$sql)){
    unset($_SESSION["user_name"]);
}

```

```

        echo "<script type='text/javascript'>";
        echo "location.href='login.php';";
        echo "alert('退出成功');";
        echo "</script>";
    }else{
        echo "<script type='text/javascript'>";
        echo "alert('退出失败');";
        echo "</script>";
    }

    mysqli_close($link);
?>

```

4. 用户登录检查

- 这里实现一个小功能，让所有的功能都在登录状态下实现，实现代码如下（2021-11-23）

```

<?php
@session_start();
if (empty($_SESSION["user_name"])) {
    echo "<script type='text/javascript'>";
    echo "alert('您还没有登录，请先登录');";
    echo "window.location.href='login.php'; // 用户登陆界面";
    echo "</script>";
}
?>

```

5. 用户发表博文

- 获取用户的id以及博文内容以及其他信息，插入到表中（2021-11-25）

```

<?php
// 处理用户发布博文页面
require("./login_session_check.php");
require "dbConfig.php";
// 连接mysql
$link = @mysqli_connect(HOST, USER, PASS, DBNAME) or die("ERROR: CANNOT
CONNECT TO DATABASE!");
// 编码设置
mysqli_set_charset('utf8',$link);

@session_start();
$title = $_POST['title'];
$type_id = strval($_POST['type_id']);
$content = $_POST['content'];
$author = $_SESSION['user_name'];
$pubtime=date("Y-m-d H:i:s");

$sql = "INSERT INTO article (id, type_id, title, content, author,
pubtime)
VALUES (NULL, '".$type_id."', '".$title."', '".$content."',
".$author."', '".$pubtime."');"

$res = mysqli_query($link,$sql);
if ($res>0) {

```

```

        echo "<script type='text/javascript'>";
        echo "alert('文章发表成功');";
        echo "location.href='./user_center.php';";
        echo "</script>";
        exit();
    }
else{
    echo "<script type='text/javascript'>";
    echo "alert('文章发表失败');";
    echo "location.href='./post_article.php';";
    echo "</script>";
    exit();
}

?>

```

6. 用户发表评论

- 用户发表评论实现代码很简单，就是获取博文id，将相关评论信息直接插入

```

<?php
if (!empty($_SESSION['fileid'])) {
    header("Location:./front_page.php");
    exit();
}
else{
    include("./dbConfig.php");
    include("./login_session_check.php");
    // 连接mysql
    $link = @mysqli_connect(HOST, USER, PASS, DBNAME) or die("ERROR:
CANNOT CONNECT TO DATABASE!");
    // 编码设置
    mysqli_set_charset('utf8',$link);

    $content = $_POST['message'];
    $username = $_SESSION['user_name'];
    if (empty($_SESSION['fileid'])) {
        header("Location:./front_page.php");
        exit();
    }
    $fileid = intval($_SESSION['fileid']);
    $comtime = date('Y-m-d h:i:s', time());

    $comment_sql = "INSERT INTO comment (id, fileid, username, content,
comtime)
VALUES (NULL, '".$fileid."', '".$username."', '".$content."',
".$comtime."')";

    $res = mysqli_query($link,$comment_sql);
    if ($res>0) {
        echo "<script type='text/javascript'>";
        echo "type='text/javascript'>";
        echo "alert('留言成功');";
        echo "location.href='./main.php';";
        echo "</script>";
        exit();
    }
}

```

```
    }
    else{
        echo "<script type='text/javascript'>";
        echo "type='text/javascript'>";
        echo "alert('留言失败');";
        echo "location.href='./blog_article.php?
blog_id=".$fileid."#comments';";
        echo "</script>";
        exit();
    }
}

@session_start();
unset($_SESSION["fileid"]);
?>
```