**PHP**

1.1php语句书写

**开始和结束标记：**

****方式一：****

标准写法：<?php . . . ?>

如：

<?php

echo "Hello world";

?>

****方式二：（尽量不使用这种方式）****

<?php . . . ?>

短标签格式，不推荐，新版PHP不再支持

如：

<?

 echo “Hello world”;

?>

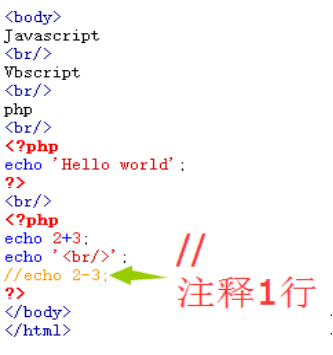
****方式三：****

<script  language=“php”>…</script>

明确指定脚本语言的方式

**注释方式：**

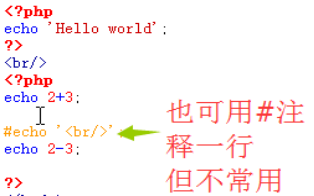
****//   可注释一行；****



****/\*  \*/   可注释多行****



****# 可注释一行，但不常用****



1.2PHP变量

**命名规则：**

PHP变量命名主要有下面几条规则：

 变量以 $ 符号开头，其后是变量的名称

 变量名称必须以字母或下划线开头

 变量名称不能以数字开头

 变量名称只能包含字母数字字符和下划线（A-z、0-9 以及 \_）

 变量名称对大小写敏感（$y 与 $Y 是两个不同的变量）

变量会在第一次赋值时被创建。这和C语言、C++、java、C#语言不同，这些语言属于强类型检查的一个语言，PHP它属于弱类型语言，对于弱类型语言，变量可以不预先声明，想用的时候直接拿过来用，但是命名要符合规范，这样使用起来会比较方便。

1.2PHP数据类型

 PHP支持的数据类型有下面这么几种：

 其中，整型、浮点型、字符型、逻辑型、NULL型为简单数据类型，也称为标量类型，这节主要介绍这几种数据类型。数组和对象属于复合数据类型。

\*\*\*\*\* 使用var\_dump函数显示变量类型和值

**字符串**

 先说一下字符串类型吧，字符串是字符序列，比如”Hello world”。字符串可以是引号内的任何文本；可以使用单引号或者双引号。

<?php

    $x = "Hello world!";

    echo $x;

    echo "<br>";

    $x = 'Hello world!';

     echo $x;

?>

**整数**

定义整数型需要注意整数的规则：

整数必须有至少一个数字（0-9）

整数不能包含逗号或空格

整数不能有小数点

整数正负均可

可以用三种格式规定整数：十进制、十六进制（前缀是 0x）或八进制（前缀是 0）

<?php

    $a=-15;

    $b=0xA;

    $c=017;

    echo $c;

?>

**浮点数**

点数是有小数点或指数形式的数字。浮点数的表现形式有两种，带小数点的形式，和指数形式

<?php

    $x = 10.365;

    var\_dump($x);

    echo "<br>";

    $x = 2.4e3;

    var\_dump($x);

?>

**逻辑**

逻辑类型又称真假类型或者布尔类型。值取true或者false。

真假型

布尔型

值： true 或 false。

<?php

    $x=true;

    var\_dump($x);

    $y=false;

    var\_dump($y);

    echo $x;

    echo $y;

?>

**Null**

null型是特殊的类型，它表示变量无没有值，还什么都不是。需要注意：null与空字符串不同，空字符串本身是一个值，而null是无值。

null表示变量无值。

null 的空，与空字符串不同

可以通过把值设置为null，将变量清空。

<?php

     $x="Hello world!";

    var\_dump($x);

    $x="";

    var\_dump($x);

     $x=null;

    var\_dump($x);

?>

**数组**

**2.1数组**

**对像**

**3.1对象**

1.3PHP运算

****赋值运算****

|  |  |  |
| --- | --- | --- |
| ****赋值**** | ****等同于**** | ****描述**** |
| x=y | x=y | 右侧表达式为左侧运算数设置值 |
| x+=y | x=x + y | 加 |
| x-=y | x=x - y | 减 |
| x\*=y | x=x \* y | 乘 |
| x/=y | x=x / y | 除 |
| x %=y | x=x % y | 模数（余数） |

****算术运算****

|  |  |  |  |
| --- | --- | --- | --- |
| ****运算符**** | ****名称**** | ****示例**** | ****结果**** |
| + | 加法 | $x + $y | $x 与 $y 求和 |
| - | 减法 | $x - $y | $x 与 $y 的差数 |
| \* | 乘法 | $x \* $y | $x 与 $y 的乘积 |
| / | 除法 | $x / $y | $x 与 $y 的商数 |
| % | 模数 | $x % $y | $x 除 $y 的余数 |

****字符串运算****



 <?php

            $a = "Hello";

            $b = $a . " world!";

            echo $b; // 输出 Hello world!

            $x="Hello";

            $x .= " world!";

            echo $x; // 输出 Hello world!

        ?>

****递增递减运算****



  <?php

            $x=10;

            echo ++$x; // 输出 11

            $y=10;

            echo $y++; // 输出 10

            $z=5;

            echo --$z; // 输出 4

            $i=5;

            echo $i--; // 输出 5

        ?>

****比较运算****

********  <?php

        $x=100;

        $y="100";

        var\_dump($x == $y); //值相等，返回true

        var\_dump($x === $y);//类型不同，返回false

?>

<?php

        $x=100;

        $y="100";

        $z=50;

        var\_dump($x != $y);//因为值相等，返回 false

        var\_dump($x <> $y);//与 != 运算结果相同

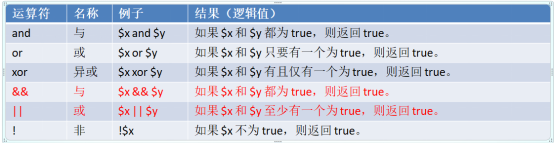
        var\_dump($x != $z); //因为值不等，返回true

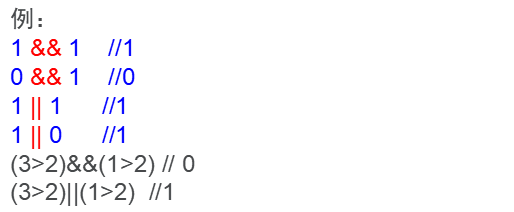
        var\_dump($x !== $y); // 因为类型不相等，返回 true

        var\_dump($x !== $z); // 因为值不相等，返回 true

     ?>

****逻辑运算****

********

********

1.4 PHP常量

****常量规则：****

**常量一旦被定义就无法更改或撤销定义**

**常量名以字母或下划线\_开头(常量名称前没有$符号)**

****定义格式：（****区分大小写****）****

define("常量名",常量值)

****（****增加参数来使得定义的常量不区分大小写****）****

define("常量名",常量值[,大小写是否不区分])

第三个参数：大小写是否不区分。

    true：不区分大小写

    false：区分大小写

    默认为false

1.5 PHP if语句

**一：**

If(条件)

        {

        当条件为true时执行的代码;

        }

<?php

$t=date("H");

if ($t<20) {

   echo "Have a good day!";

}

?>

**二:**

 if (条件) {

    条件为 true 时执行的代码;

 } else {

      条件为 false 时执行的代码;

 }

<?php

$t=date("H");

if ($t<20) {

   echo "Have a good day!";

} else {

   echo "Have a good night!";

}

?>

**三：**

if (条件) {

  条件为 true 时执行的代码;

} elseif (condition) {

  条件为 true 时执行的代码;

} else {

  条件为 false 时执行的代码;

}

 <?php

        $t=date("H");

        if ($t<10) {

           echo "Have a good morning!";

        } elseif ($t<20) {

           echo "Have a good day!";

        } else {

           echo "Have a good night!";

        }

?>

1.6 PHP switch语句

  switch(表达式)｛

        case Value1：

            code1;

            break;

        case Value2：

            code2;

            break;

        …

        default:

            code

    ｝

<?php

  $day=date("w");//获得星期几的数字

     switch($day){

    case 1:

        echo "今天是星期一";

        break;

    case 2:

        echo "今天是星期二";

        break;

    case 3:

        echo "今天是星期三";

        break;

    case 4:

        echo "今天是星期四";

        break;

    case 5:

        echo "今天是星期五";

        break;

    case 6:

        echo "今天是星期六";

        break;

    default:

        echo "今天是星期日";

    }

?>

1.7 PHP while循环

while（expr）{

  要执行的代码

}

<?php

    $sum=0;

    $i=1;  //i是计数器

    While($i<=100){

        $sum=$sum+$i;

        $i++;

    }

    echo $sum;

  ?>

1.8 PHP do while循环

****do {****

****要执行的代码;****

****} while (条件为真);****

<?php

   $sum=0;//累加和

   $i=1;//计数器

   do{

    $sum=$sum+$i;

     $i++;

    }while($i<=100);

   echo $sum;

?>

1.9 PHP for循环

****for (expr1; expr2; expr3){****

****循环体****

****}****

<?php

  $sum=0;//累加和

  //计数器

  for($i=1;$i<=100;$i++)

   {

      $sum=$sum+$i;

    };

  echo $sum;

 ?>

2.1数组

**查看数组的元素的值的两种方式：**

**第一种：**

使用Print\_r函数，来显示数组的值，这个函数把数组各个元素的值和数组的结构都显示的很清楚。

**第二种：**

采用echo语句，来显示数组的元素的值，数组名[键]

**索引数组：**

**创建方式：**

****$数组名=array（值1，值2，值3…）****

**$cars=array("Volvo","BMW","SAAB");**

**另外的创建方式：**

**$cars[0]=“Volvo”;**

**$cars[1]=“BMW”;**

**$cars[2]="SAAB";**

**Eg:**

**<?php**

**$cars=array("Volvo","BMW","SAAB");**

**Print\_r($cars);**

**?>**

**一个数组定义好后，如果要增加元素，怎么办呢？可以有两种方法：**

**（1）在指定的位置增加**

**（2）在最后增加**

**<?php**

**$cars=array("Volvo","BMW","SAAB");**

**$cars[3]="大众"; //在指定的位置增加**

**$cars[]="奥迪";  //在最后增加**

**print\_r($cars);**

**?>**

**关联数组：**

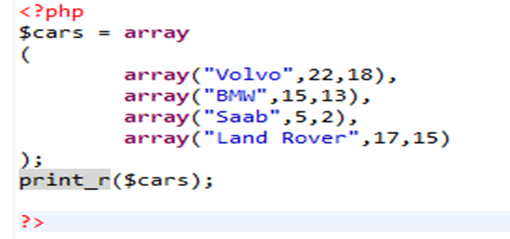
创建方式：

$cars= array("键名1"=>值1, "键名2"=>值2, ... "键名n"=>值n);

**$cars= array("Volvo"=>10, "BMW"=>5, "SAAB"=>23);**

**多维数组：**

多维数组指的是包含一个或多个数组的数组，即数组的元素又是数组。



查看值的方式：

 数组名[外层键][内层键]

数组名[被嵌套的数组名][键名/索引值]

2.2 数组的遍历

**索引数组的遍历：**

count()函数可以用来获得数组的元素个数:

count(数组名)

for循环实现遍历：

**<?php**

**$cars=array("Volvo","BMW","SAAB");**

**$arrlength=count($cars);//获取数组的元素个数**

**for($x=0;$x<$arrlength;$x++)**

**{**

**echo $cars[$x];**

**echo "<br/>";**

**}**

**?>**

foreach循环实现遍历;

foreach (数组名 as $value) //$value为储存元素的变量

    {

        循环体

    }

**<?php**

**$cars=array("Volvo","BMW","SAAB");**

**foreach($cars as $value)**

**{**

**echo $value;**

**echo "<br/>";**

**}**

**?>**

**关联数组的遍历：**

****foreach (数组名 as $key => $value) //键值对****

****{****

****循环体****

****}****

**<?php**

**$cars=array("Volvo"=>21,"BMW"=>22,"SaaB"=>23);**

**foreach($cars as $key=>$value)**

**{**

**echo "$key=>$value <br/>";**

**}**

**?>**

**多维数组的遍历：**

使用嵌套的for 或者 foreach 来遍历多维数组

****<?php****

****$cars = array****

****(****

****" Volvo"=>array("库存"=>22,"销量"=>18),****

****"BMW"=>array("库存"=>15,"销量"=>13),****

****"Saab"=>array("库存"=>5,"销量"=>2),****

****"Land Rover"=>array("库存"=>17,"销量"=>15) );****

****foreach($cars as $outKey=>$outValue){****

****echo $outKey.": ";****

****foreach($outValue as $innerKey=>$innerValue){****

****echo $innerKey."=>".$innerValue."，";****

****}****

****echo "<br/>";****

****}****

****?>****

2.3数组的排序

**索引数组的排序：**

**根据值的大小进行排序**

**sort() -以升序对数组排序**

**rsort() -以降序对数组排序**

**<?php**

**$cars=array("Volvo","BMW","SAAB");**

**echo "原数组:";**

**print\_r($cars);**

**echo "<br/>";**

**echo "升序后的数组：";**

**sort($cars);**

**print\_r($cars);**

**echo "<br/>";**

**echo "降序后的数组：";**

**rsort($cars);**

**print\_r($cars);**

**?>**

**关联数组的排序：**

**asort() 按值升序排序**

**arsort() 按值降序排序**

**ksort() 按键升序排序**

**krsort() 按键降序排序**

**<?php**

**$cars=array("Volvo"=>"35","BMW"=>"37","SAAB"=>"43");**

**echo "按值升序：";**

**asort($cars);**

**print\_r($cars);**

**echo "<br/>";**

**echo "按值降序：";**

**arsort($cars);**

**print\_r($cars);**

**echo "<br/>";**

**echo "按键升序：";**

**ksort($cars);**

**print\_r($cars);**

**echo "<br/>";**

**echo "按键降序：";**

**krsort($cars);**

**print\_r($cars);**

**echo "<br/>";**

**?>**

2.4 日期函数

**时间戳：**是指格林威治时间1970年01月01日00时00分00秒起至当下的总秒数。

**显示当前时间的函数：**

**date()函数**把时间戳格式化为更易读懂的日期和时间

****date(format,timestamp)****

第一个参数****format****是一个字符串，用于指定日期时间显示的格式，是一 个必选项。

**在date函数的format字符串中，经常会用到下面代表日期的几个字符：**

**d - 表示月里的某天（01-31）**

**m - 表示月（01-12）**

**Y - 表示年（四位数）**

**获得时间，可以用下面的指定字符：**

h - 12 小时格式

H-  24小时格式

i - 分钟

s - 秒（00 -59）

a - 午前和午后（am 或 pm）

**另外，在format格式中经常用到一些年月日的分隔符，如“-”、“/” 等。**

第二个参数****timestamp****指定所需要显示的时间是哪个时间，但是这个参 数是可以省略的，如果省略的话，date函数就会按照服务器的当前时间来处 理。

date()既可以返回日期，也可以返回时间，当然也可以两者都返回。

**<?php**

**echo "今天是 " . date("Y/m/d") . "<br>";**

**echo "今天是 " . date("Y.m.d") . "<br>";**

**echo "今天是 " . date("Y-m-d") . "<br>";**

**?>**

**<?php**

**echo "现在时间是 " . date("h:i:sa");**

**echo "<br/>";**

**echo "现在时间是 " . date("H:i:s");**

**?>**

时区设置语句：

****date\_default\_timezone\_set("时区")****

**date\_default\_timezone\_set("Asia/Shanghai");**

**显示其他(指定)时间的函数：**

用**mktime()**来创建一个时间戳。

****mktime(hour,minute,second,month,day,year)****

****这里的参数分别是时、分、秒、月、日、年。****

**<?php**

**$d=mktime(9, 12, 31, 6, 10, 2015);**

**echo "创建的时间戳是".$d;**

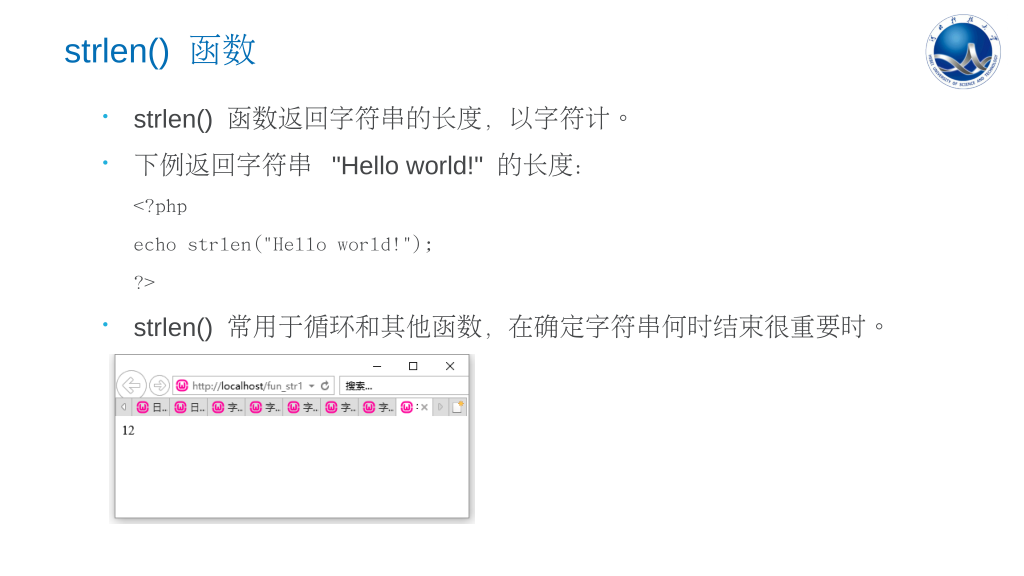
**echo "<br/>";**

**echo "创建日期是 " . date("Y-m-d h:i:sa", $d);**

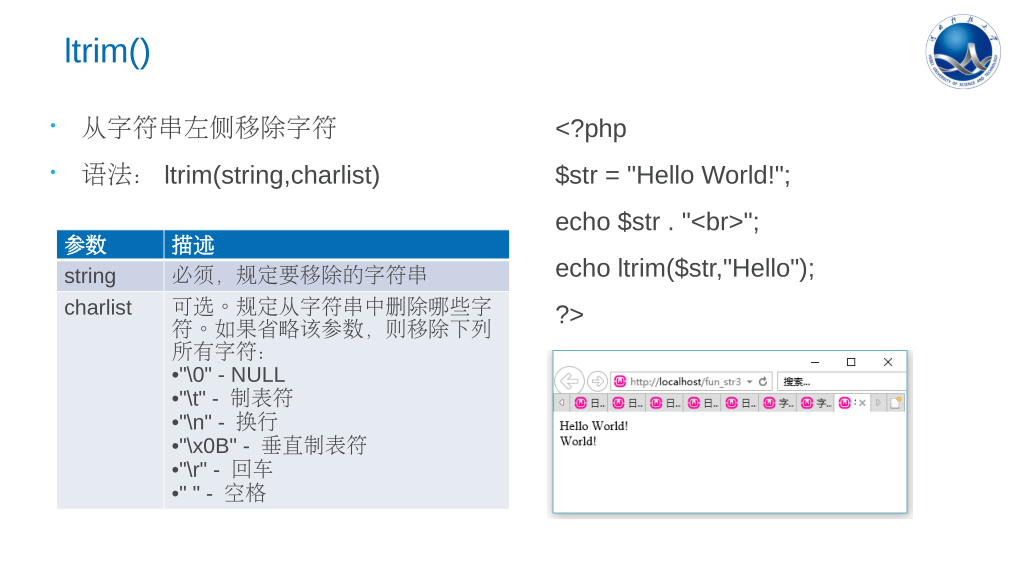
**?>**

**↓↓**

通过mktime()函数创建了时间戳，并用date()函数转化为易于识别的时间格式

2.5 字符串函数

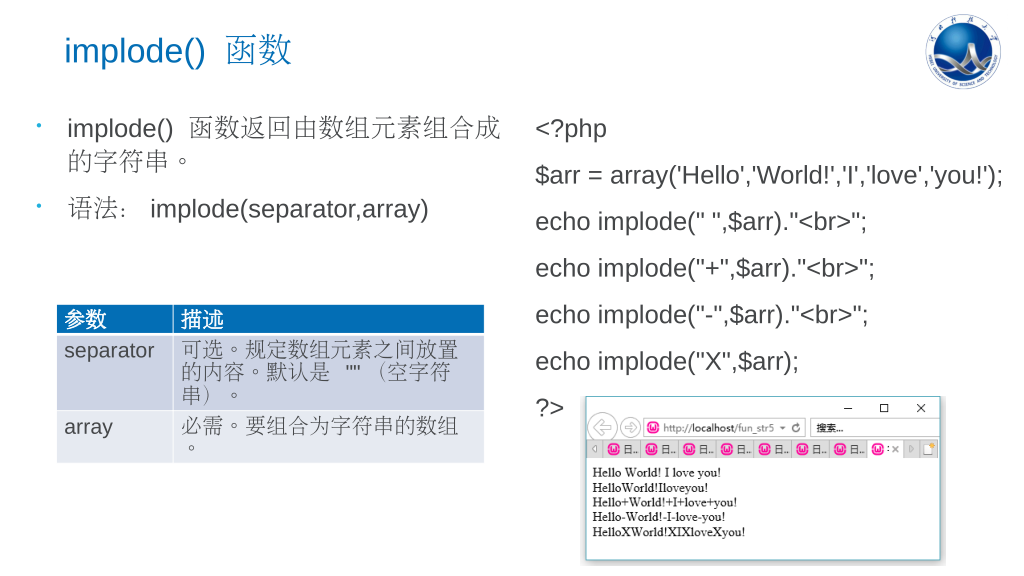




****rtrim()移除字符串右侧的空白字符或其他预定义字符。****

****trim()移除字符串两侧的空白字符或其他预定义字符。****





2.6 用户定义函数

**创建用户自定义函数格式：**

****function functionName() {****

****被执行的代码;****

****}****

****函数名不区分大小写，但变量名区分。不过还是建议坚持大小写敏感的规范.****

****参数可有可无。****

**<?php**

**function writeMsg() {**

**echo "Hello world!";**

**}**

**writeMsg(); // 调用函数**

**?>**

2.7 表单处理

****$\_POST**是一个超全局变量，是一个数组变量（键值对），适用于表单以post方式提交。**

****$GET**是一个超全局变量，是一个数组变量（键值对），适用于表单以get方式提交。**

**<html>**

**<body>**

**<form action="welcome.php"   method="post">**

**Name: <input type="text" name="name"><br>**

**E-mail: <input type="text" name="email"><br>**

**<input type="submit"  value="提交">**

**</form>**

**</body>**

****第一点，form表单定义了“action="welcome.php"”，表示当用户填写此表单并点击提交按钮后，表单数据会发送到名为 "welcome.php" 的PHP文件供处理。****

****第二点，form表单定义了method="post"，表示表单数据是通过 POST 方法发送的。****

**<?php**

**echo "欢迎".$\_POST["name"]."<br/>";**

**echo "你的邮箱是:".$\_POST["email"]**

**?>**

**<?php**

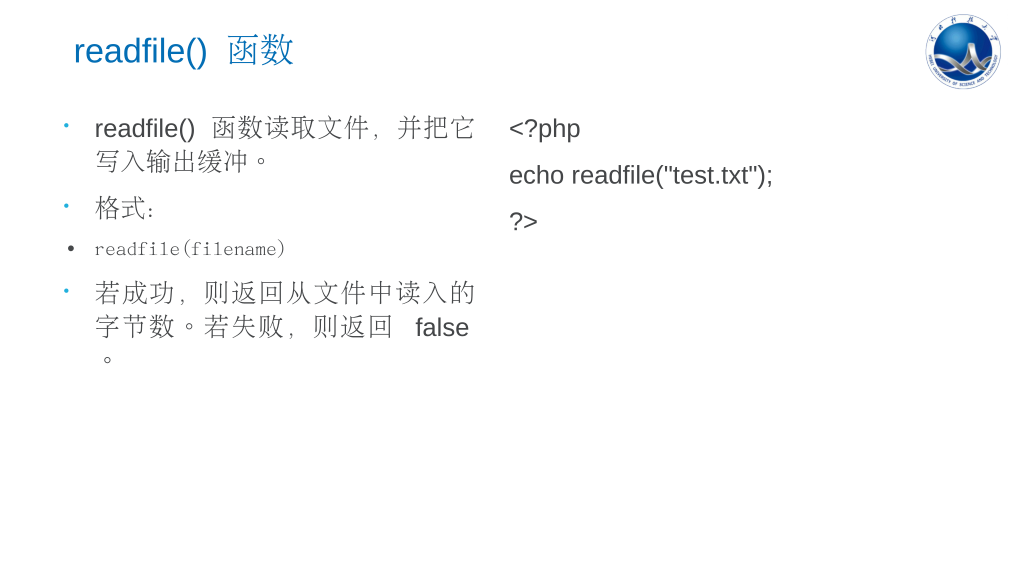
**echo "欢迎".$\_GET["name"]."<br/>";**

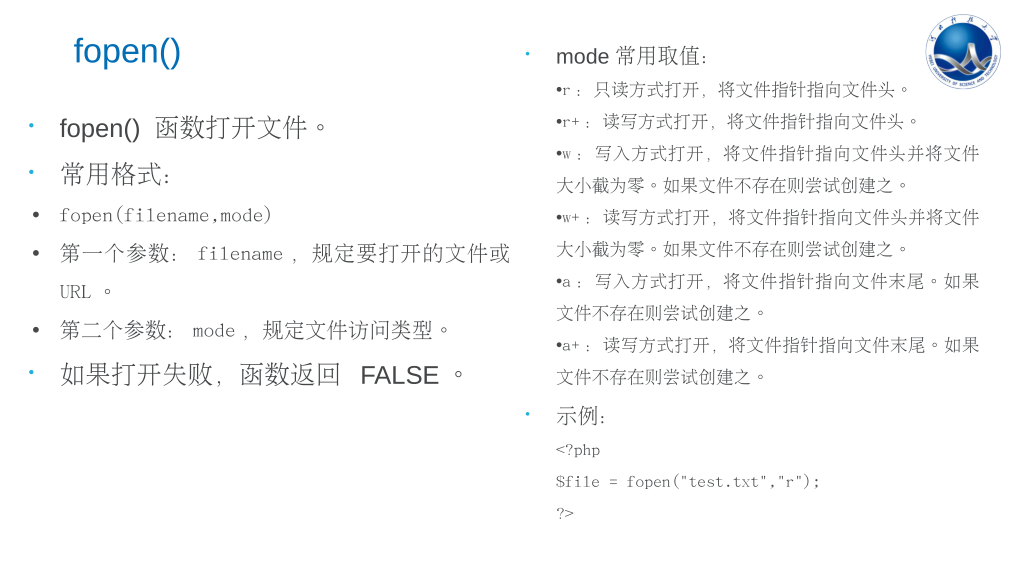
**echo "你的邮箱是:".$\_GET["email"]**

**?>**

****当表单采用post方式提交是采用$\_POST数组方式，当表单采用get方式的时候呢，采用$\_GET数组方式。****

2.8 文件打开、读取、创建、写入



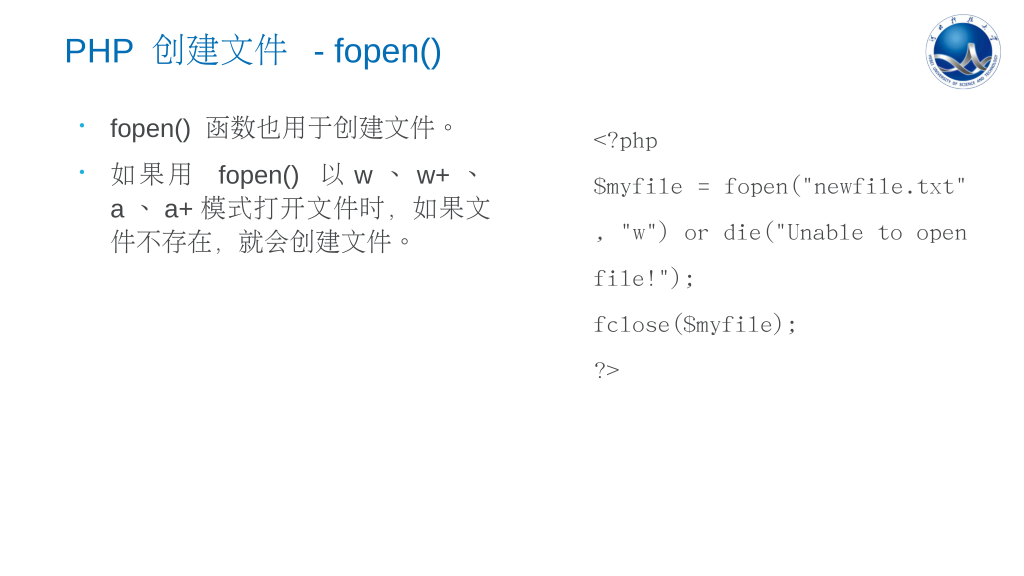


**<?php**

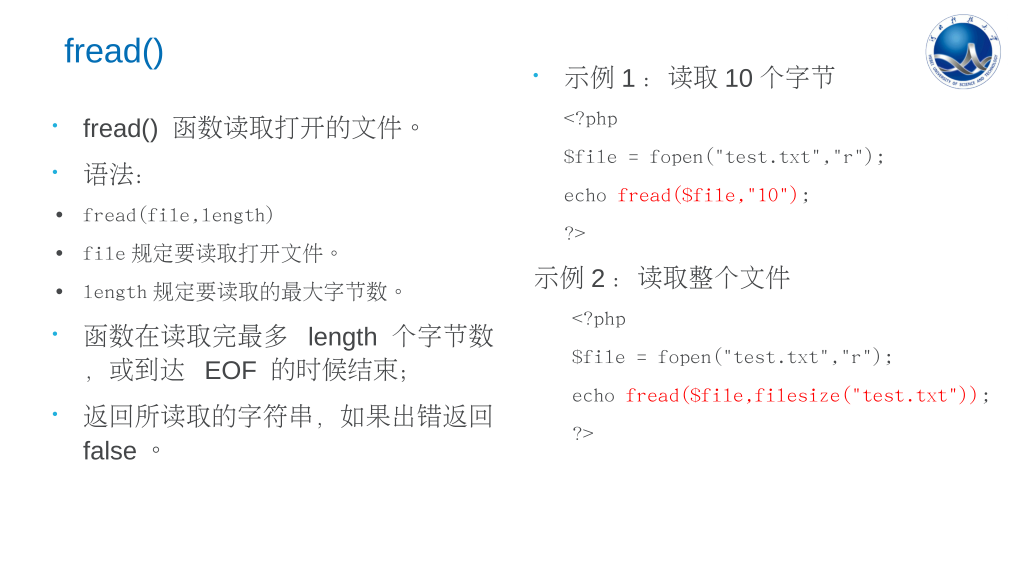
**$file = fopen("test.txt","r");**

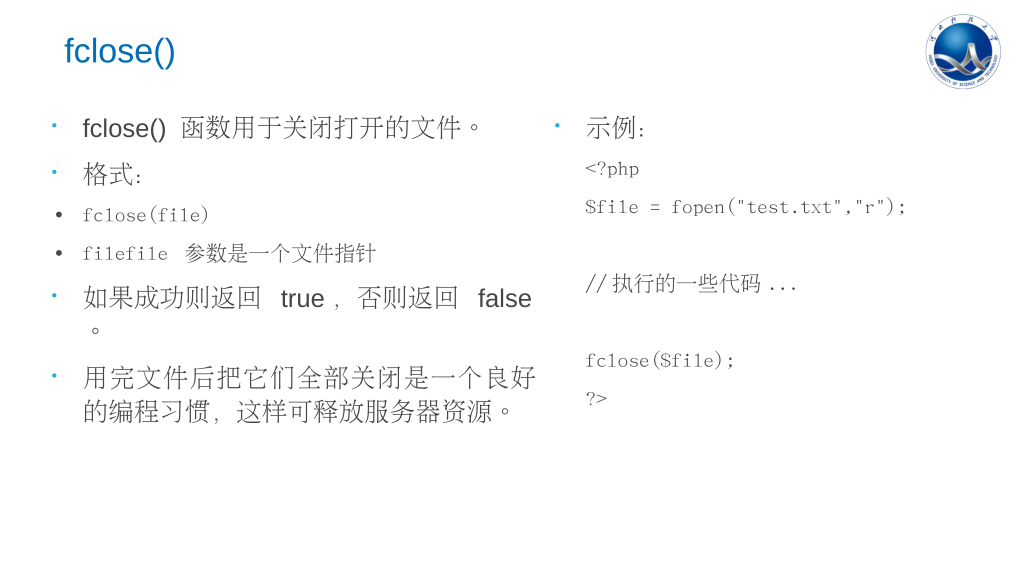
**echo $file;**

**?>**



**当文件创建失败时，fopen()将返回false,此时or前的代码返回false，因此执行后面的die()函数-输出一条消息，并退出当前脚本。**





****fread()示例1：读取10个字节****

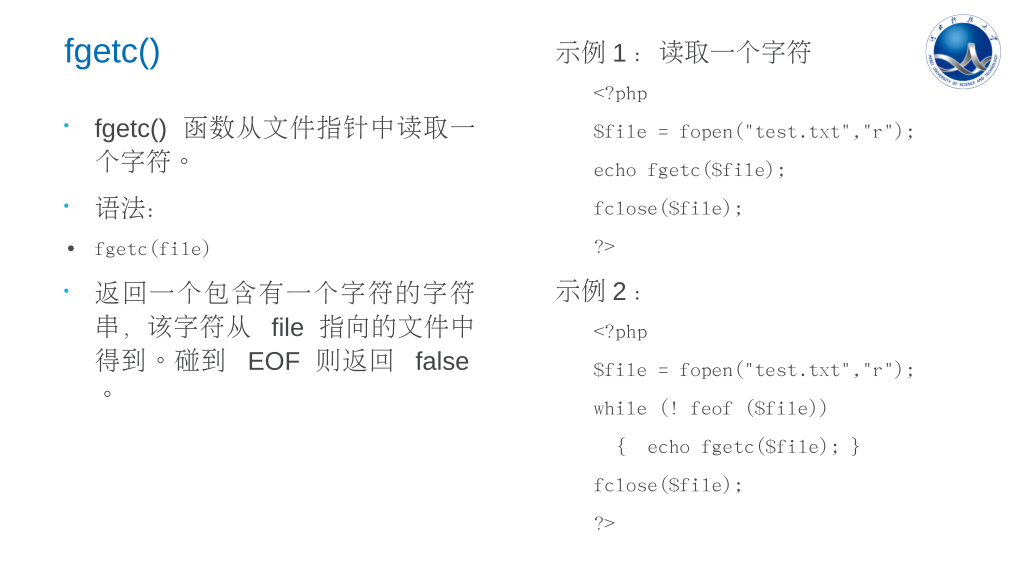
|  |
| --- |
| **<?php**  **$file = fopen("test.txt","r");**  **echo fread($file,"10");**  **fclose($file);**  **?>** |

****fread()示例2：读取整个文件****

|  |
| --- |
| **<?php**  **$file = fopen("test.txt","r");**  **echo fread($file,filesize("test.txt"));**  **fclose($file);**  **?>** |







**<?php**

**$file = fopen("test.txt","r");**

**echo fgets($file);**

**fclose($file);**

**?>**

**<?php**

**$file = fopen("test.txt", "r");**

**//输出文本中所有的行，直到文件结束为止。**

**while(! feof($file)){**

**echo fgets($file). "<br />";**

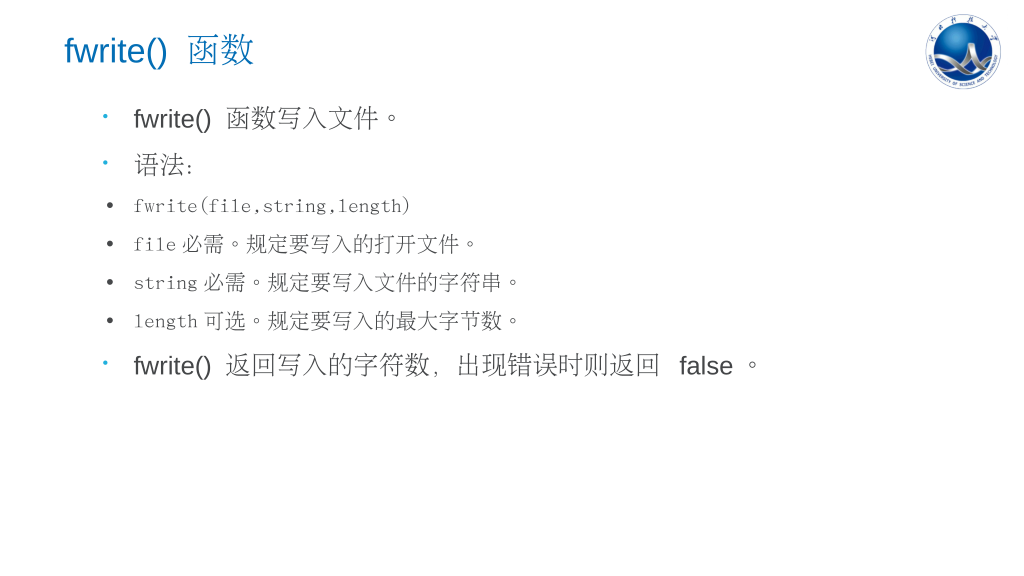
**}**

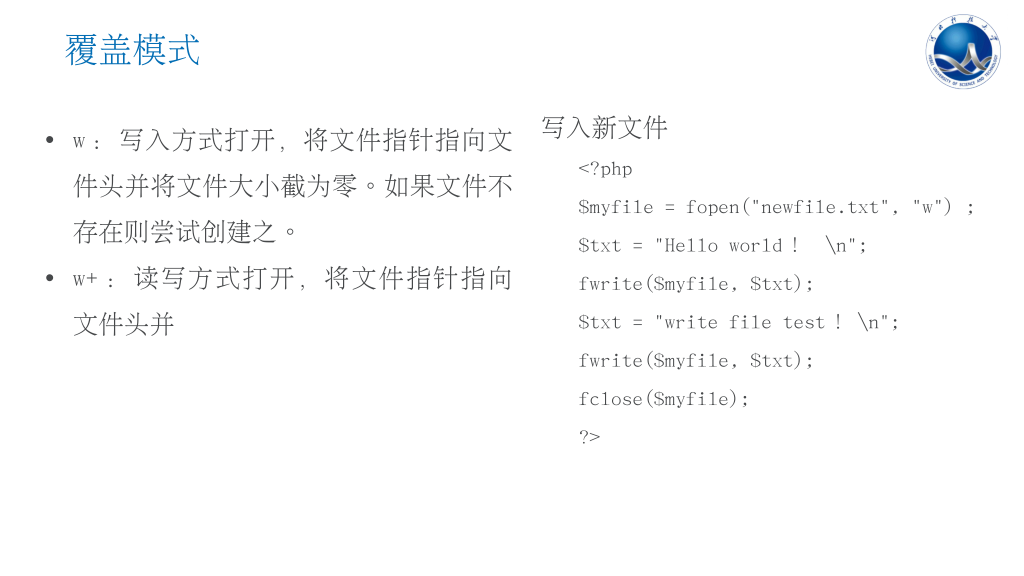
**fclose($file);**

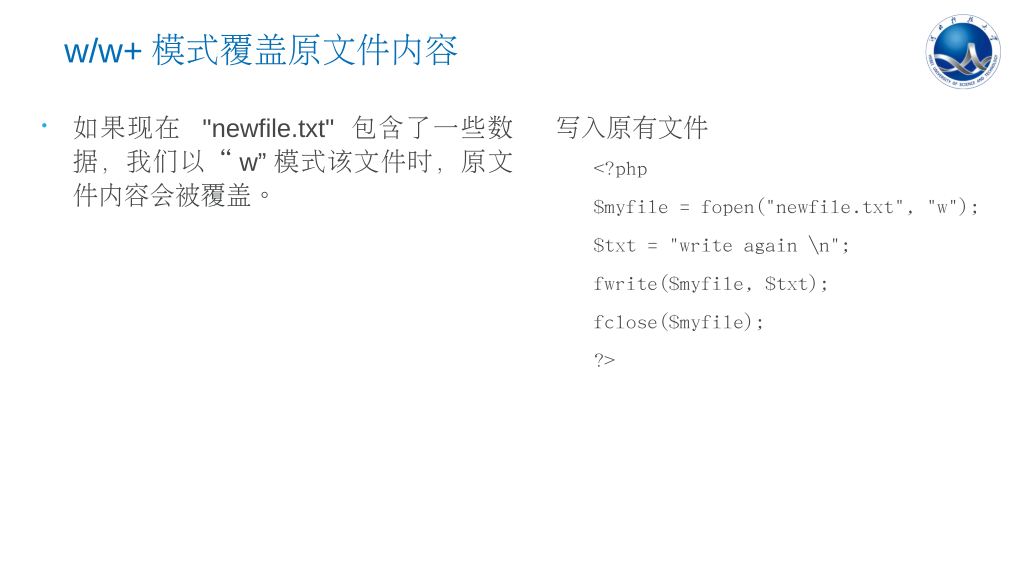
**?>**

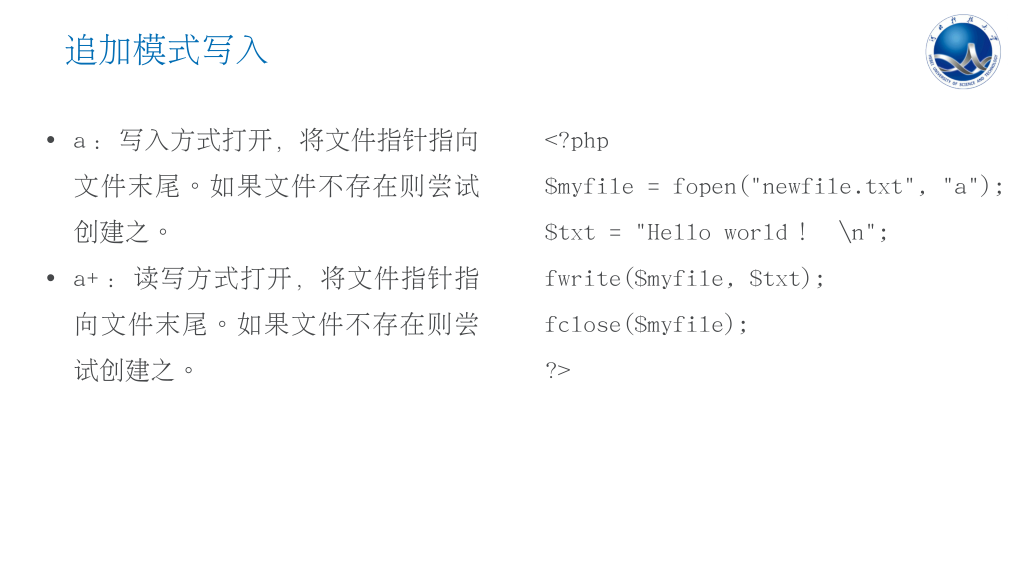
****循环读取字符:****

|  |
| --- |
| **<?php**  **$file = fopen("test.txt","r");**  **while (! feof ($file)){**  **echo fgetc($file);**  **}**  **fclose($file);**  **?>** |





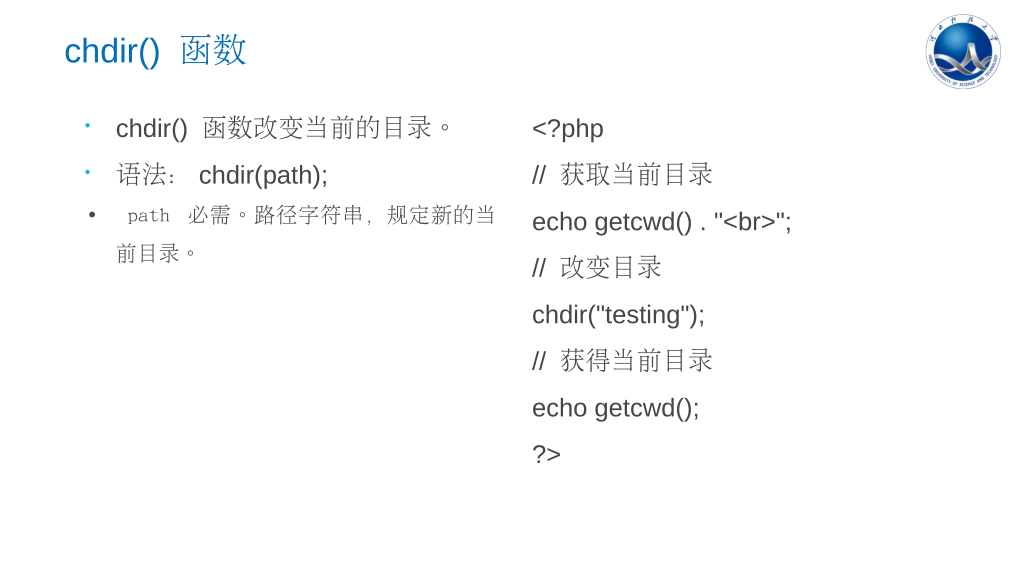


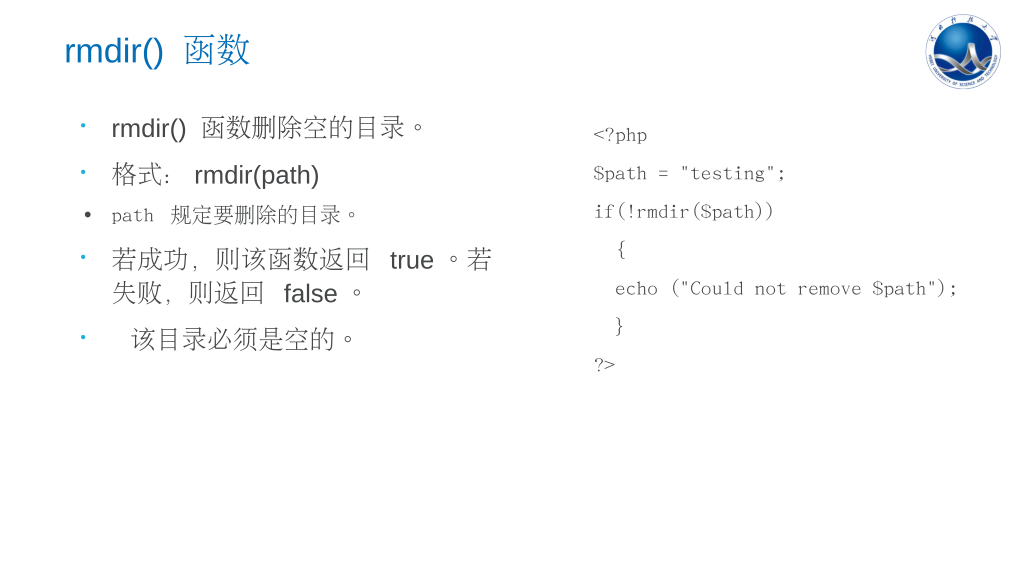


2.9 路径的创建、删除、打开、读取



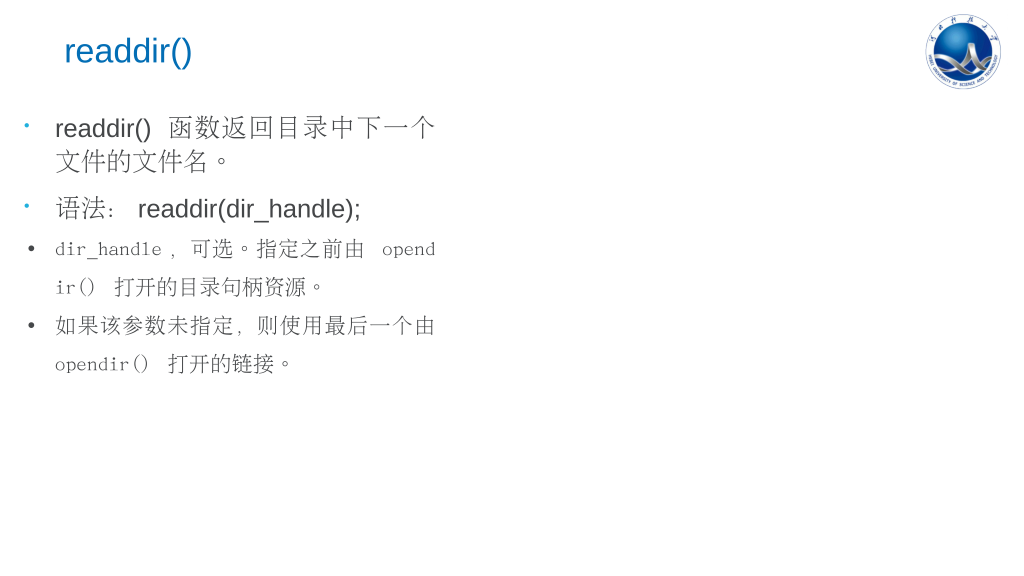










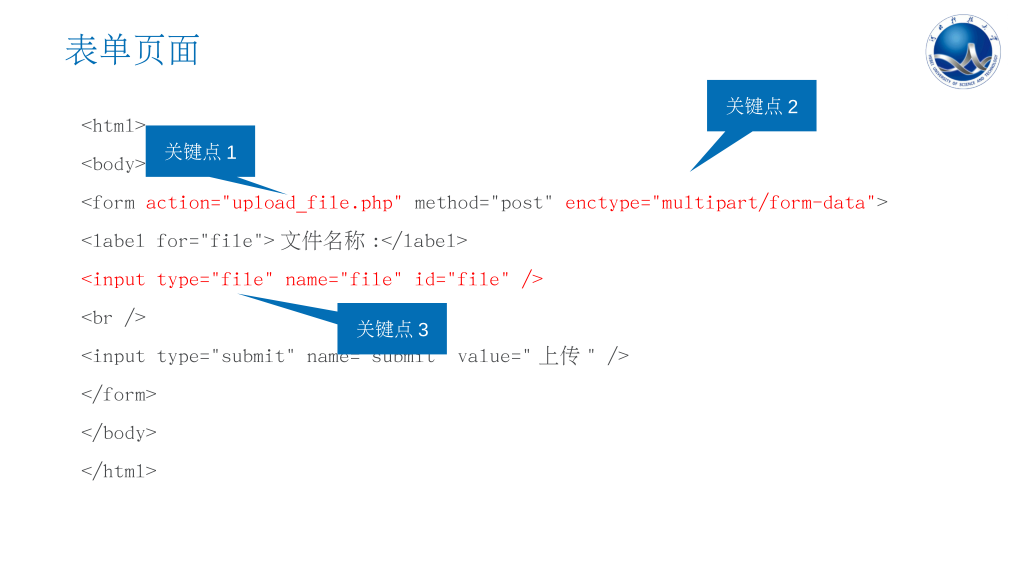


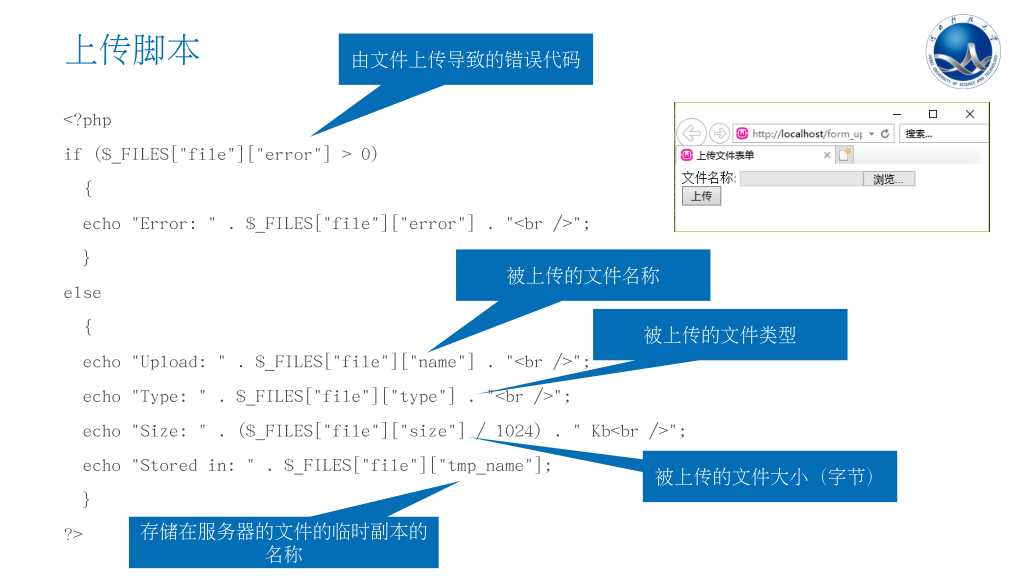


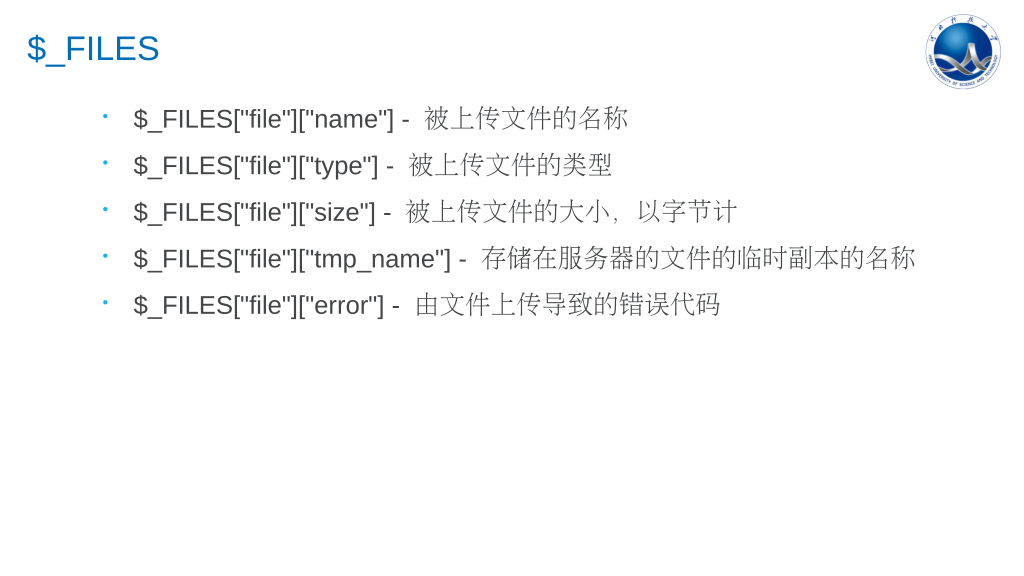


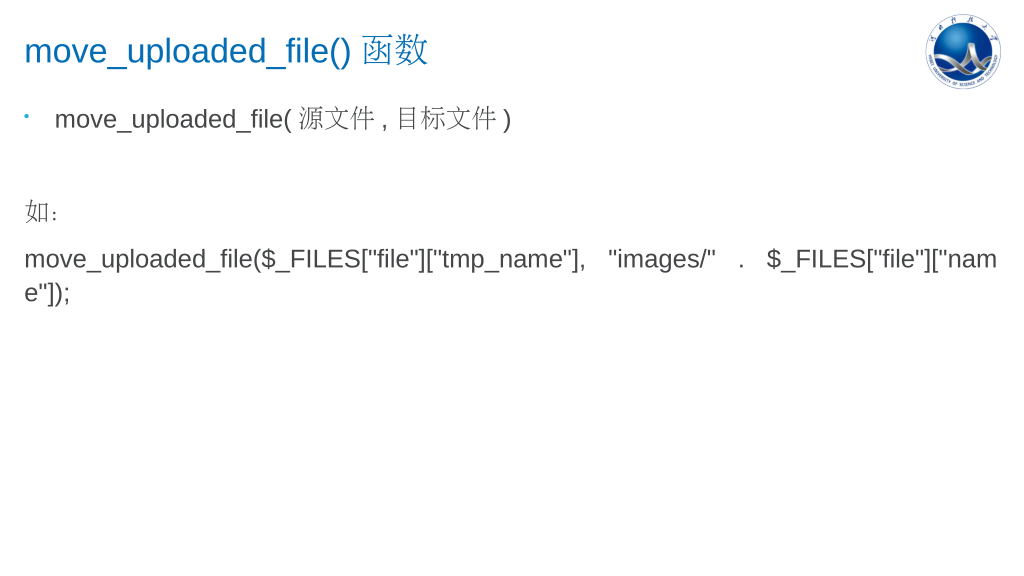
**scandir()函数是一个返回目录及文件的数组，及将目录、文件放入数组返回。**

3.0 文件上传

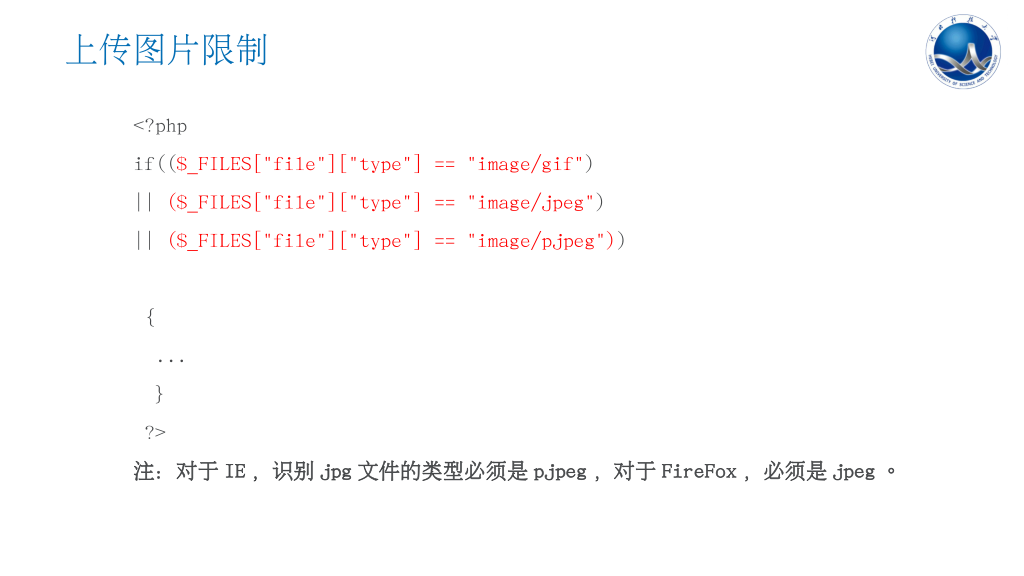


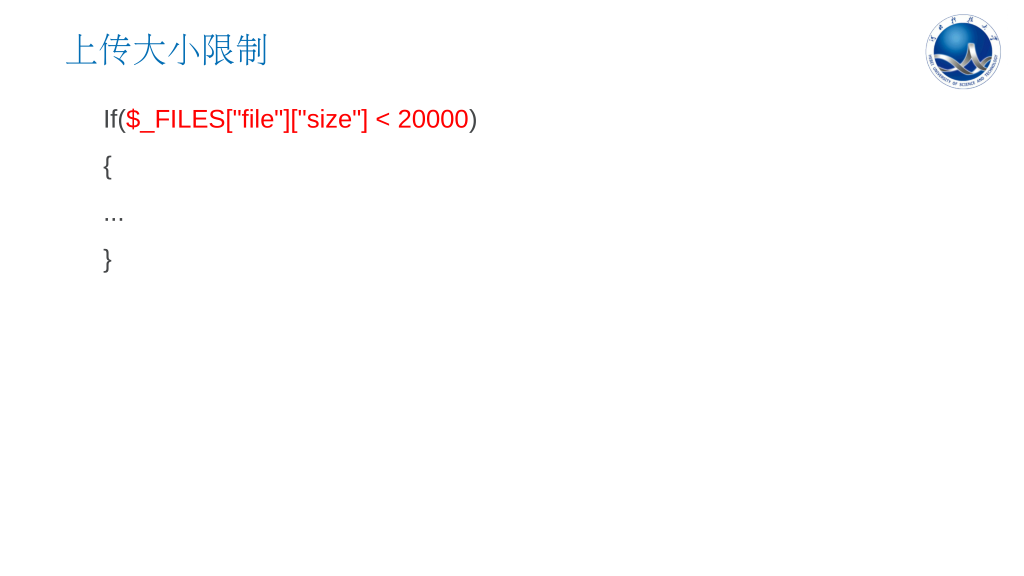




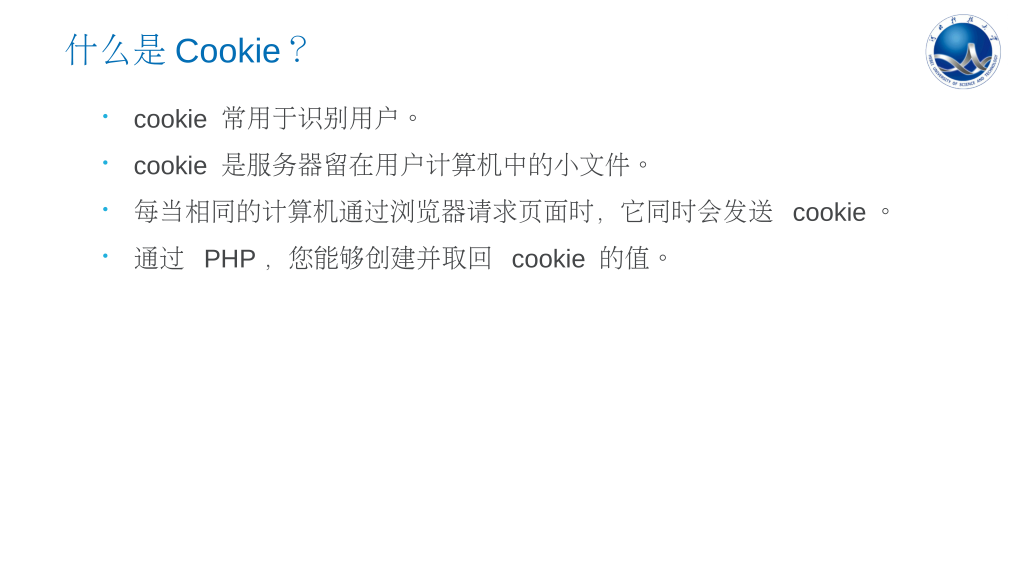


**可配合$\_FILES将上传的文件保存到指定位置。**



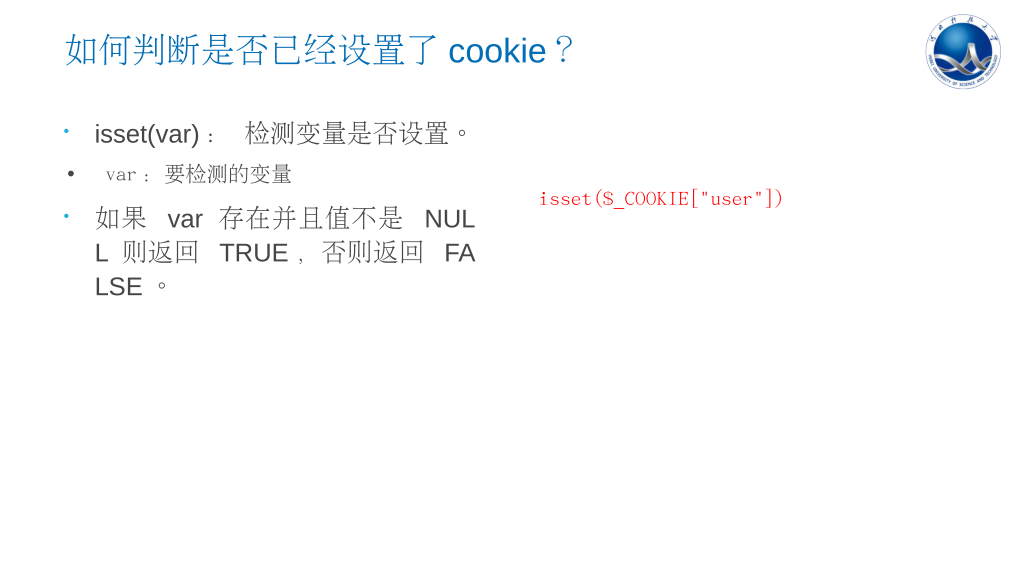


3.1 PHP cookie



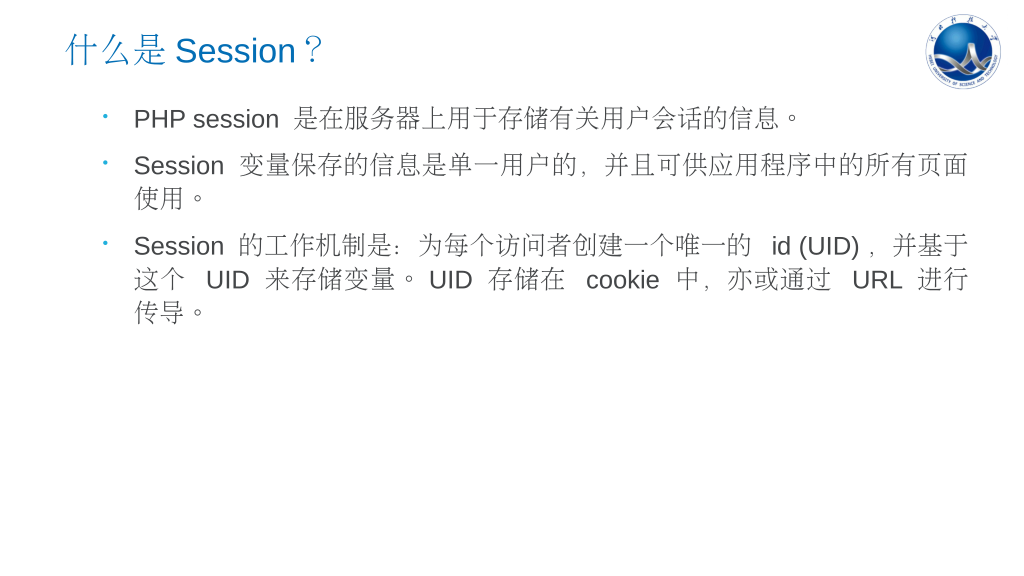








3.2 PHP session









4.0 PHP数据库访问

4.1 MySQL命令行操作





Create database 数据库名 if not exist default character set gbk;

**补充视频的主要内容：**

**1、命令行窗口的进入**

**2、查看路径下的文件**

**dir**

**dir/p**

**3、更改盘符**

**盘符：（回车）**

**4、进入或退出路径**

**cd..返回上一级**

**cd\ 返回到根路径**

**cd 路径名 如：cd users**

**cd 路径名\子路径名...**

**5、创建路径**

**md 路径名**

**6、删除路径**

**rd 路径名**

**7、快速上翻**

**上下箭头控制**

**8、复制粘贴**

**复制：右键-菜单中的标记-选择文字-回车**

**粘贴：右键-菜单-粘贴**

**9、清空屏幕**

**cls**

4.2 MySQL数据库管理工(客户端软件)具基本操作



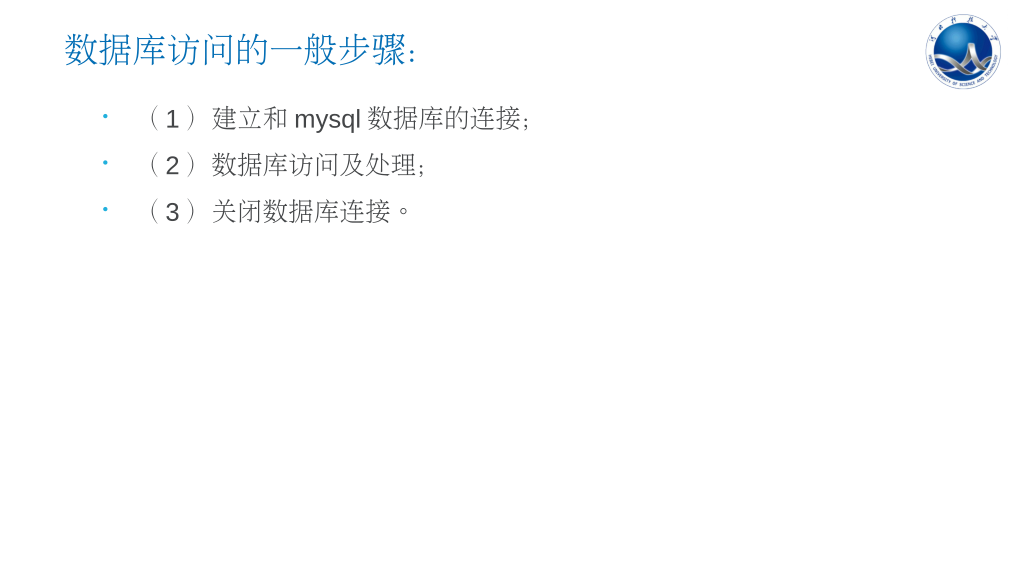
创建数据库；

创建表结构；

修改表结构；

增删改查；

4.3 PHP连接数据库





****$conn =mysqli\_connect(“localhost”,”root”,”123456”,”mydb”);****

函数如果连接成功，就返回连接标示符，如果连接失败返回false。

****$conn=mysqli\_connect("localhost","root","111111","mydb");****

****if(!$conn)****

****{****

****die("数据库连接失败".mysqli\_connect\_error());****

****}****

****echo "数据库连接成功";****

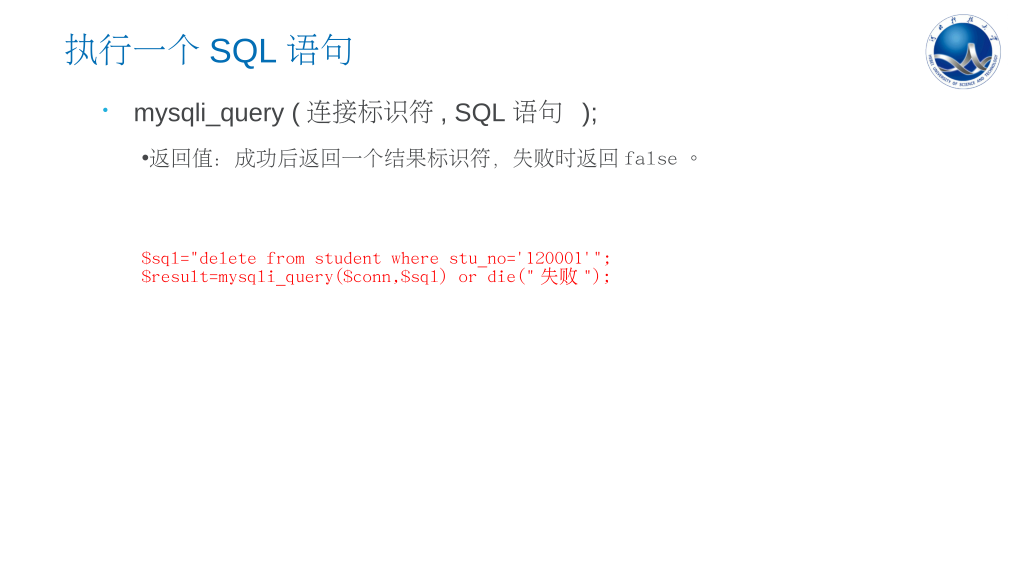
****$conn=mysqli\_connect("localhost","root","111111","mydb")or die("数据库连接失败");****

****echo "数据库连接成功";****

**关闭数据库连接：**

**mysqli\_close(连接标识符)**

4.4 PHP执行SQL语句



**使用mysqli\_query()函数执行insert、update、delete语句的操作**

**mysqli\_query()函数执行了一个insert语句**

**$conn=mysqli\_connect("localhost","root","111111","mydb") or die("数据库连接失败");**

**echo "数据库连接成功";**

**mysqli\_query($conn,'set names utf8'); **//告诉服务器，本页面的字符集是utf8****

**$sql="insert into student(stu\_no,stu\_name,sex,birthdate,telephone,email) values('17010105','王平','男','1998-05-01','18806547811','wangping@qq.com')";**

**$result=mysqli\_query($conn,$sql)or die("数据插入失败".mysqli\_error($conn));**

**echo "数据插入成功";**

**or die()的作用是，当前面的语句（...）执行失败时，即前面语句返回值为false时，执行die()。**

**mysqli\_query()函数执行了一个update语句**

|  |
| --- |
| **$conn=mysqli\_connect("localhost","root","111111","mydb") or die("数据库连接失败");**  **echo "数据库连接成功";**  **mysqli\_query($conn,'set names utf8');**  **$sql="update student set stu\_name='王平' where stu\_name='wangping'";**  **$result=mysqli\_query($conn,$sql)or die("数据更新失败".mysqli\_error($conn));**  **echo "数据更新成功";** |

**mysqli\_query()函数执行了一个delete语句**

|  |
| --- |
| **$conn=mysqli\_connect("localhost","root","111111","mydb") or die("数据库连接失败");**  **echo "数据库连接成功";**  **mysqli\_query($conn,'set names utf8');**  **$sql="update student set stu\_name='王平' where stu\_name='wangping'";**  **$result=mysqli\_query($conn,$sql)or die("数据更新失败".mysqli\_error($conn));**  **echo "数据更新成功";** |

**当然，mysqli\_query()函数也可以执行select查询语句的操作。**

4.5 PHP MySql记录集的获取

如果用mysqli\_query()函数执行一个select语句将返回一个多行多列的数据集，那么又怎么从这个数据集中获取数据呢？

PHP提供了几个函数，可供我们进一步从这个数据集中提取数据，每次都可以从这个数据集中提取一行数据，这几个函数是：

|  |
| --- |
| ****mysqli\_fetch\_row();//用来将查询结果的一行保存至索引数组;****  ****mysqli\_fetch\_assoc();//用来将查询结果的一行保存至关联数组;****  ****mysqli\_fetch\_array();//前2者的结合。**** |





<?php

    $conn=mysqli\_connect("localhost","root","111111","mydb") or die("数据库连接失败");

    mysqli\_query($conn,'set names utf8');

    $sql="select \* from student";

    $result=mysqli\_query($conn,$sql)or die("数据查询失败");

    while($row=mysqli\_fetch\_row($result))

    {

        $cols=count($row);

        for($i=0;$i<$cols;$i++)

        {

             echo $row[$i]; **//显示列的数值(垂直显示)，如：$row[0] 学号，$row[1] 姓名，$[2] 年龄...**

             if($i<$cols-1)echo ",";

        }

        echo "<br/>";

    }

?>

**mysqli\_fetch\_assoc():**

从记录集中提取数据到一个关联数组中

数组已记录集中的字段为元素的键，以字段的值为元素的值

|  |
| --- |
| **<?php**  **$conn=mysqli\_connect("localhost","root","111111","mydb") or die("数据库连接失败");**  **mysqli\_query($conn,'set names utf8');**  **$sql="select \* from student";**  **$result=mysqli\_query($conn,$sql)or die("数据查询失败");**  **while($row=mysqli\_fetch\_assoc($result))**  **{**  **echo "row[stu\_no],$row[stu\_name],$row[sex],$row[birthdate],$row[telephone],$row[email]";**  **echo "<br/>";**  **}**  **?>** |

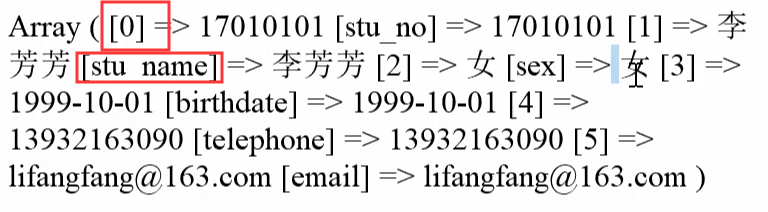
|  |
| --- |
| **<?php**  **$conn=mysqli\_connect("localhost","root","111111","mydb") or die("数据库连接失败");**  **mysqli\_query($conn,'set names utf8');**  **$sql="select \* from student";**  **$result=mysqli\_query($conn,$sql)or die("数据查询失败");**  **while($row=mysqli\_fetch\_assoc($result))**  **{**  **foreach($row as $k=>$v)**  **{**  **echo $v;**  **if($k!="email")echo ",";**  **}**  **echo "<br/>";**  **}**  **?>** |



**$row=mysqli\_fetch\_array($result);**

**print\_r($row);**

**显示结果如下，我们可以看到既有数字索引，又有关联键名索引。**



**可通过索引或者键值对方式获取数据。**

5.0 PHP数据库案例

5.1 登录界面

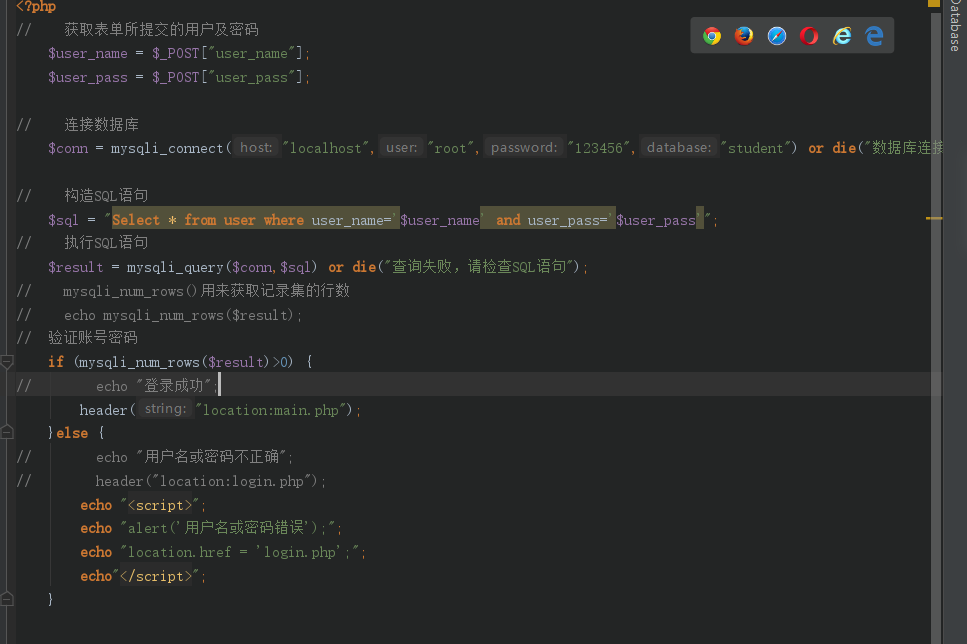


5.2 登录认证

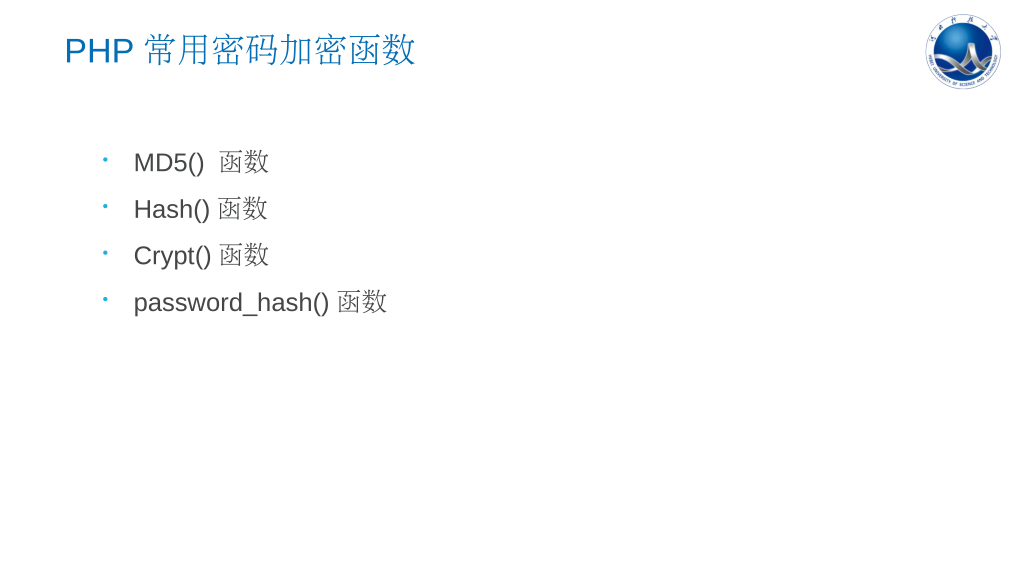
（3）会使用结果集行数函数：mysqli\_num\_rows();

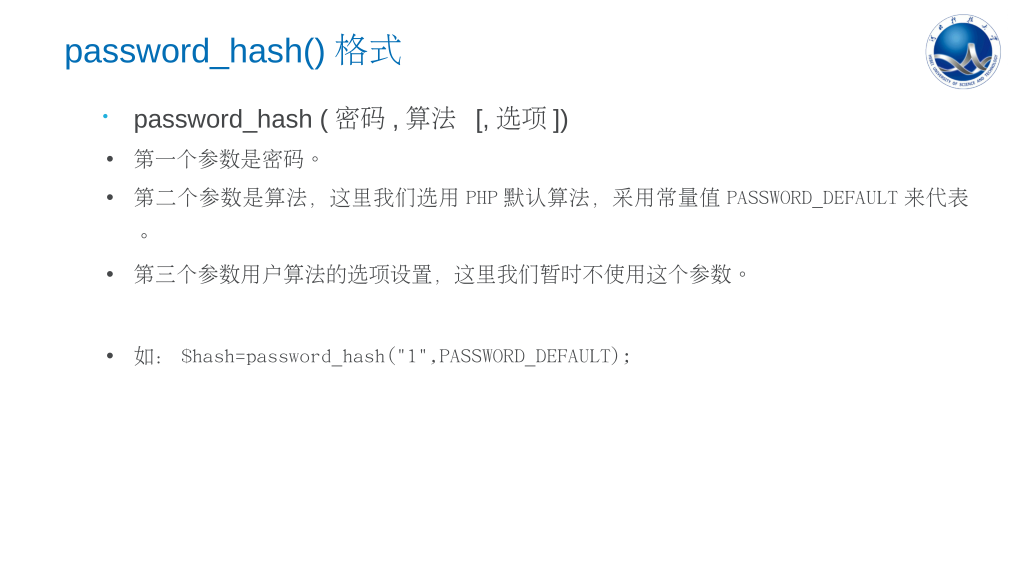
（4）会使用页面跳转函数header()

（5）会使用Javascript弹出信息框并跳转



5.3 密码加密

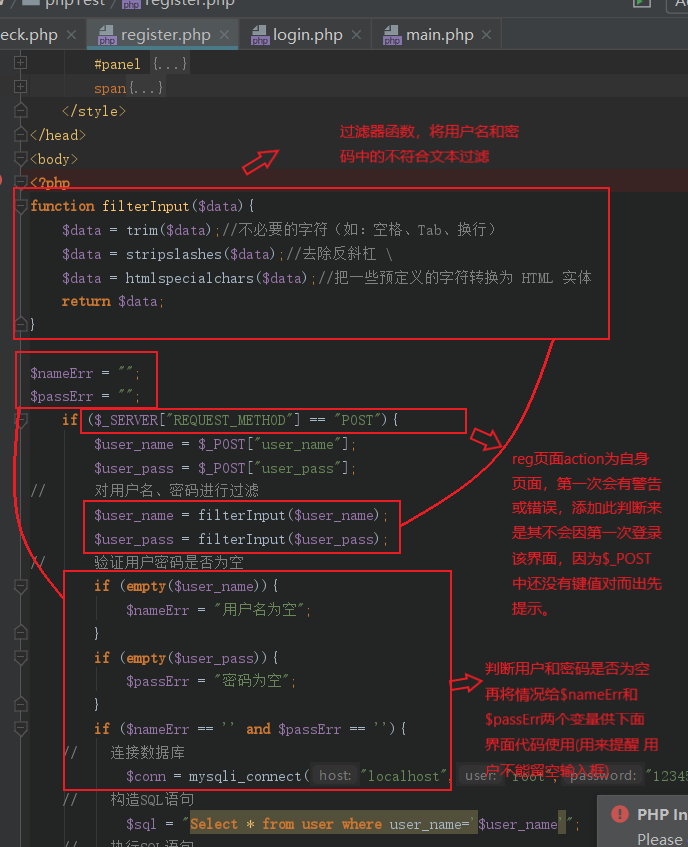


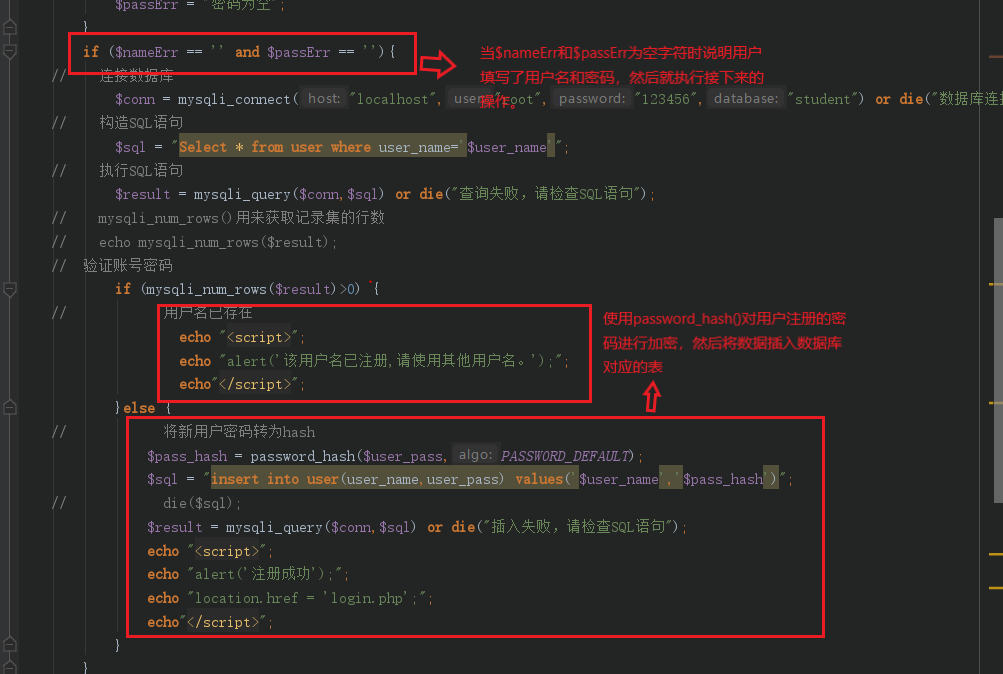


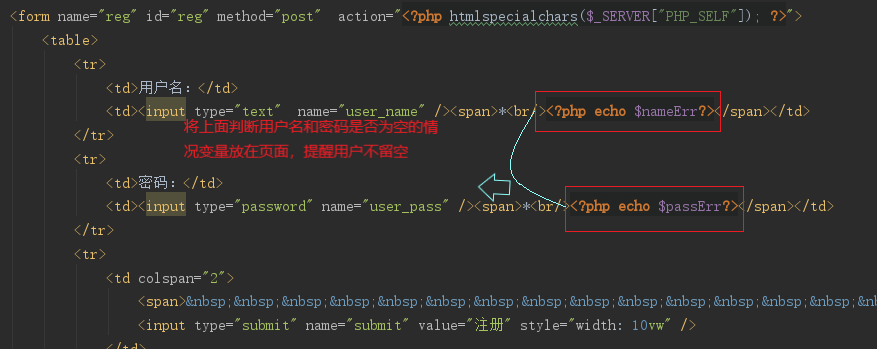




5.4 注册页面







5.5 包含文件

**PHP的文件包含有两种方式**：

**include语句**

**require 语句**

****这两种方式的使用大同小异同。****

PHP包含文件的语法是这样的：

****include 'filename';****

或者

****require 'filename';****

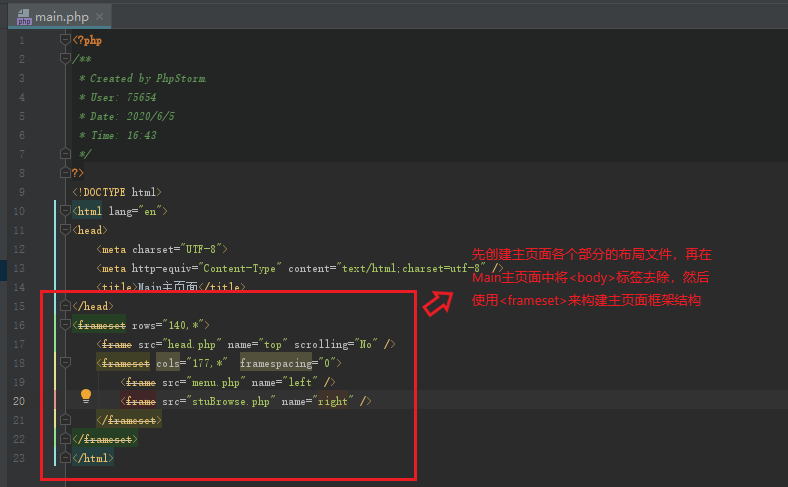
**include 和 require在包含文件不存在时的处理错误的方式方面有所不同：**

require 在出错时会生成一个致命错误（E\_COMPILE\_ERROR），在错误发生后脚本会停止执行。

include 在出错时会生成一个警告（E\_WARNING），在错误发生后脚本会继续执行。

5.6 框架集定义





5.7 当前用户的获取（session）

head.php页面如何知道当前用户是谁呢？这就需要在登录环节把当前用户记录下来，而且这个记录，其他页面是可以访问到的，同时只对本次会话有效，下一次如果再有用户登录，本次的记录就应该无效了。这种情况，我们可以采用session来解决。

**check.php中增加用户名的session保存。**

|  |
| --- |
| ...  if (password\_verify($user\_pass,$row['user\_pass']))  {      session\_start();     $\_SESSION["user"]=$user\_name;     header("location:main.php");  }  ... |

**head.php中增加用户名的显示。**

|  |
| --- |
| <body>      <h1>学生信息管理</h1>      <?php          session\_start();          echo "<h4>当前用户:".$\_SESSION["user"]."</h4>";      ?>  </body> |

5.8 浏览记录页面

stuBrowse.php ：

|  |
| --- |
| ****<htmlxmlns="http://www.w3.org/1999/xhtml">****  ****<head>****  ****<meta http-equiv="Content-Type"content="text/html; charset=utf-8" />****  ****<style>****  ****table{****  ****margin-top:50px;****  ****width:80%;****  ****border:1px#B3CDE8 solid;****  ****}****    ****th****  ****{****  ****text-align:center;****  ****background-color:#C8F4FF;****  ****}****    ****tr****  ****{        height:35px;****  ****}****    ****</style>****  ****<title>浏览记录</title>****  ****</head>****  ****<body>****  ****<?php****  ****include "conn.php";//包含连接文件****  ****mysqli\_query($conn,'set names utf8');//告诉服务器，浏览器端字符集是utf8****  ****$sql="select \* from student,major wherestudent.major\_id=major.major\_id"; //构造SQL语句，连接两个表的SQL语句。****  ****$result=mysqli\_query($conn,$sql)ordie("数据查询失败");//执行SQL语句****  ****if(mysqli\_num\_rows($result)>0){ //判断是否有查询结果****  ****echo"<table align='center' >";//输出表格标签****  ****echo"<tr><th>学号</th><th>姓名</th><th>性别</th><th>专业</th><th>生日</th><th>电话</th><th>电邮</th></tr>";//输出表头行****  ****while($row=mysqli\_fetch\_assoc($result))//从记录集获取一行数据到数组，不为false，则显示该行数据(数组中各元素****  ****{****  ****?>****  ****<tr onMouseOver="this.style.backgroundColor='#D8F4FF';"onMouseOut="this.style.backgroundColor='#ffffff';">****  ****<?php****  ****echo"<td >$row[stu\_no]</td><td>$row[stu\_name]</td><td>$row[gender]</td><td>$row[major\_name]</td><td>$row[birthdate]</td><td>$row[telephone]</td><td>$row[email]</td>";//输出获取的一行数据****  ****echo"</tr>";//输出行结束标记****  ****};****  ****echo"</table>";//输出表格结束标记****  ****}****  ****else //没有查询结果****  ****echo"尚无学生信息";****  ****?>****  ****</body>****  ****</html>**** |

5.9 添加记录页面布局

stuInsert.php:

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>添加记录</title>  
 <style>  
 #content {  
 width: 600px;  
 height: 600px;  
 border: solid 1px #6699ff;  
 margin: 50px auto;  
 padding: 30px;  
 }  
 #left {  
 float: left;  
 width: 350px;  
 height: 350px;  
 }  
 #right {  
 float: left;  
 width: 250px;  
 height: 300px;  
 }  
 #bottom {  
 clear: both;  
 width: 600px;  
 height: 300px;  
 }  
 label{  
 display: block;  
 float: left;  
 width: 55px;  
 }  
 #img{  
 width: 120px;  
 height: 160px;  
 }  
 #pic{  
 width: 120px;  
 height: 160px;  
 background: #ccc;  
 padding: 10px;  
 }  
  
 </style>  
</head>  
<body>  
<form name="form" method="post" action="<?php htmlspecialchars($\_SERVER['PHP\_SELF']);?>" enctype="multipart/form-data">  
 <div id="content">  
 <div id="left">  
 <p><label>学号:</label><span><input type="text" name="stu\_no"/></span></p>  
 <p><label>姓名:</label><span><input type="text" name="stu\_name"/></span></p>  
 <p><label>专业:</label>  
 <span>  
 <select name="major">  
 <option value="01">软件技术</option>  
 <option value="02">计算机应用</option>  
 <option value="03">嵌入式应用</option>  
 <option value="04">现代教育</option>  
 <option value="05">大数据应用</option>  
 </select>  
 </span>  
 </p>  
 <p><label>性别:</label>  
 <span>  
 男<input name="gender" type="radio" value="男" checked/>  
 女<input name="gender" type="radio" value="女"/>  
 </span>  
 </p>  
 <p><label>生日:</label><span><input type="date" name="birthdate"/></span></P>  
 <p><label>电话:</label><span><input type="text" name="telephone"/></span></p>  
 <p><label>邮箱:</label><span><input type="text" name="email"/></span></p>  
 </div>  
 <div id="right">  
 <p><label>照片</label><span>  
 <input type="file" name="file" id="file"/></span></p>  
 <div id="pic">  
 <img id="img" src="images/head.png" />  
 </div>  
 </div>  
 <div id="bottom">  
 <p>个人简介</p>  
 <textarea name="resume"rows="10" cols="80">  
 </textarea>  
 </p>  
 <p>  
 <input name="submit"type="submit" value="提交"/>  
 <input name="reset"type="reset" value="重置"/>  
 </p>  
 </div>  
  
 </div>  
 </form>  
</body>  
</html>

5.10 上传照片的预览

**1、FileReader接口的方法**

    FileReader接口有4个方法，其中3个用来读取文件，另一个用来中断读取。无论读取成功或失败，方法并不会返回读取结果，这一结果存储在result属性中。

|  |  |  |
| --- | --- | --- |
| **FileReader接口的方法** | | |
| **方法名** | **参数** | **描述** |
| readAsBinaryString | file | 将文件读取为二进制编码 |
| readAsText | file,[encoding] | 将文件读取为文本 |
| readAsDataURL | file | 将文件读取为DataURL |
| abort | (none) | 终端读取操作 |

**2、FileReader接口事件**

|  |  |
| --- | --- |
| **FileReader接口的事件** | |
| **事件** | **描述** |
| onabort | 中断 |
| onerror | 出错 |
| onloadstart | 开始 |
| onprogress | 正在读取 |
| onload | 成功读取 |
| onloadend | 读取完成，无论成功失败 |

**3、FileReader接口的使用（example）**

|  |
| --- |
| <script type="text/javascript">  var result=document.getElementById("result");  var file=document.getElementById("file");    //判断浏览器是否支持FileReader接口  if(typeof FileReader == 'undefined'){      result.InnerHTML="<p>你的浏览器不支持FileReader接口！</p>";      //使选择控件不可操作      file.setAttribute("disabled","disabled");  }    function readAsDataURL(){      //检验是否为图像文件      var file = document.getElementById("file").files[0];      if(!/image\/\w+/.test(file.type)){          alert("看清楚，这个需要图片！");          return false;      }      var reader = new FileReader();      //将文件以Data URL形式读入页面      reader.readAsDataURL(file);      reader.οnlοad=function(e){          var result=document.getElementById("result");          //显示文件          result.innerHTML='<img src="' + this.result +'" alt="" />';      }  }    function readAsBinaryString(){      var file = document.getElementById("file").files[0];      var reader = new FileReader();      //将文件以二进制形式读入页面      reader.readAsBinaryString(file);      reader.οnlοad=function(f){          var result=document.getElementById("result");          //显示文件          result.innerHTML=this.result;      }  }    function readAsText(){      var file = document.getElementById("file").files[0];      var reader = new FileReader();      //将文件以文本形式读入页面      reader.readAsText(file);      reader.οnlοad=function(f){          var result=document.getElementById("result");          //显示文件          result.innerHTML=this.result;      }  }  </script>  <p>      <label>请选择一个文件：</label>      <input type="file" id="file" />      <input type="button" value="读取图像" οnclick="readAsDataURL()" />      <input type="button" value="读取二进制数据" οnclick="readAsBinaryString()" />      <input type="button" value="读取文本文件" οnclick="readAsText()" />  </p>  <div id="result" name="result"></div>  ————————————————  版权声明：本文为CSDN博主「第2016换个昵称」的原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接及本声明。  原文链接：https://blog.csdn.net/zk437092645/article/details/8745647 |

stuInsert.php:

<script type="text/javascript">  
 window.onload = **function** () {  
 **var** fileTag = document.getElementById('file');  
 fileTag.onchange = **function** () {  
 **var** file = fileTag.files[0];  
 **var** fileReader = **new** FileReader();  
 fileReader.onloadend = **function** () {  
 **if** (fileReader.readyState == fileReader.DONE) {  
 document.getElementById('img').setAttribute('src', fileReader.result);  
 }  
 };  
 fileReader.readAsDataURL(file);  
 };  
 };  
</script>

<div id="right">  
 <p><label>照片</label><span>  
 <span class="error"><?php echo $photoErr;?></span>  
 <input type="file" name="file" id="file"/></span></p>  
 <div id="pic">  
 <img id="img" src="images/head.png" />  
 </div>  
</div>

5.11 填充下拉列表

stuInsert.php:

<body>  
**<?php  
 include** "conn.php";  
 $sql = "select \* from major";  
 $result = mysqli\_query($conn,$sql) **or die**("数据查询失败");  
**?>**

**...**

<select name="major">  
 <option value="">-- 请选择 --</option>  
 **<?php  
 if** (mysqli\_num\_rows($result)>0){  
 **while** ($row = mysqli\_fetch\_assoc($result)){  
 **?>** <option value="**<?php echo** $row['major\_id']; **?>**">**<?php echo** $row["major\_name"]; **?>**</option>  
 **<?php** }  
 }**else**{  
 **echo** "获取专业失败";  
 } **?>**

</select>

5.12 表单验证

**增加必填项验证：**

|  |
| --- |
| <head>  ...  <style>  ...  .error{     color:#F00;  }  </style>  </head>  <body>  ...  <?php  ...  //定义错误提示变量  $noErr=$nameErr=$majorErr=$photoErr=$telephoneErr=$emailErr=$birthdateErr="";  if($\_SERVER["REQUEST\_METHOD"]=="POST")//如果是提交表单  {  //获得各表单项的值  stu\_no=$\_POST["stu\_no"];//学号  $stu\_name=$\_POST["stu\_name"];//姓名  $major=$\_POST["major"];//专业     $gender=$\_POST["gender"];//性别  $birthdate=$\_POST["birthdate"];//出生日期       $telephone=$\_POST["telephone"];//手机       $email=$\_POST["email"];//邮箱       $resume=$\_POST["resume"];//个人简介  //设置必填项  if (empty($stu\_no))  {  $noErr="学号为空";  };  if (empty($stu\_name))  {  $nameErr="姓名为空";  };  if (empty($major))  {  $majorErr="专业为空";  };  ...    if(($noErr=='') &&($nameErr=='')&&($majorErr=='')&&($birthdateErr=='')&&($telephoneErr=='')&&($emailErr=='')){  ...      }  }  ?>  <form name="form1" action="<?php htmlspecialchars($\_SERVER['PHP\_SELF']) ?>"  method="post" enctype="multipart/form-data">  <div id="content">  <div id="left">  <p >  <label>学号</label>  <span><input type="text" name="stu\_no"/></span>  <span class='error'>\*<?php echo $noErr;?></span>  </p>  <p>  <label>姓名</label>  <span><input type="text" name="stu\_name"/></span>  <span class="error">\*<?php echo $nameErr;?></span>  </p>  <p>  <label>专业</label>  <span>  <select name="major">  <option value="">===请选择===</option>  <?php while($row=mysqli\_fetch\_assoc($result)){ ?>  <option value="<?php echo $row['major\_id'] ?>"><?php echo $row['major\_name']?></option>  <?php } ?>  </select>  </span>  <span class="error">\*<?php echo $majorErr;?></span>  </p>  <p><label>性别 </label>  <span>男<input name="gender" type="radio" value="男" checked />  女<input name="gender" type="radio" value="女" />  </span></p>  <p>  <label>生日</label><span><input type="date" name="birthdate"/></span>  <span class="error"><?php echo $birthdateErr;?></span>  </P>  <p>  <label>电话</label><span><input type="text" name="telephone"/></span>  <span class="error"><?php echo $telephoneErr;?></span>  </p>  <p>  <label>邮箱</label><span><input type="text" name="email"/></span>  <span class="error"><?php echo $emailErr;?></span>  </p>  </div>  <div id="right">  <p><label>照片</label><span>  <span class="error"><?php echo $photoErr;?></span>  ...  </body> |

**增加过滤函数：**

|  |
| --- |
| <?php  ...  include "conn.php";  $sql="select \* from major";  $result=mysqli\_query($conn,$sql)or die("数据查询失败");  function filterInput($data) {     $data = trim($data);//不必要的字符 (如：空格，tab，换行)。     $data = stripslashes($data);//去除反斜杠 (\)     $data = htmlspecialchars($data);//把一些预定义的字符转换为 HTML 实体     return $data;  }  //定义错误提示变量  $noErr=$nameErr=$majorErr=$photoErr=$telephoneErr=$emailErr=$birthdateErr="";  if($\_SERVER["REQUEST\_METHOD"]=="POST")//如果是提交表单  {  //获得各表单项的值  $stu\_no=$\_POST["stu\_no"];//学号  $stu\_name=$\_POST["stu\_name"];//姓名  $major=$\_POST["major"];//专业          $gender=$\_POST["gender"];//性别  $birthdate=$\_POST["birthdate"];//出生日期          $telephone=$\_POST["telephone"];//手机          $email=$\_POST["email"];//邮箱          $resume=$\_POST["resume"];//个人简介  //过滤字符         $stu\_no=filterInput($stu\_no);         $stu\_name=filterInput($stu\_name);  ...  } |

**正则表达式验证：**

**常用正则表达式：**

[https://www.cnblogs.com/longdaye/p/8001221.html](https://www.cnblogs.com/longdaye/p/8001221.html" \t "https://mooc1-1.chaoxing.com/knowledge/_blank)

**PHP为我们提供了验证正则表达式的函数：**

**preg\_match(正则表达式，待匹配的字符串)：**

preg\_match()函数的话，它的值将是0次或1次，0次是不匹配，1次是找到匹配，因为preg\_match()在第一次匹配后将会停止搜索。

**preg\_match\_all(正则表达式，待匹配的字符串)：**

preg\_match\_all()的话，可以返回0次（不匹配）或n次，0次是不匹配，preg\_match\_all()函数会一直搜索subject直到到达结尾。

这两个函数如果匹配过程中发生错误会返回 FALSE。

**图片验证：**

|  |
| --- |
| ...  <script type="text/javascript">      window.onload = function () {          var fileTag = document.getElementById('file');          fileTag.onchange = function () {              var file = fileTag.files[0];             if(!/image\/\w+/.test(file.type)){       alert("你好，选择的不是图片！");        return false;       }              var fileReader = new FileReader();              fileReader.onloadend = function () {                  if (fileReader.readyState == fileReader.DONE) {                      document.getElementById('img').setAttribute('src', fileReader.result);                  }              };              fileReader.readAsDataURL(file);          };      };  </script>  ... |