PHP笔记整理

环境配置，使用wampserver

返回前端一定要返回字符串



Php里面写了参数，调用时必须传参，不然会报错



编写接口，如何把数据传给前端，动态页面

如何打开php文件：

先用wampserver创建路径：

Apache =>Alias directories =>add an alias

打开命令框后，输入一个虚拟路径，代替test（例如1702）,回车，然后输入文件路径

然后等待wampserver重启后，直接在浏览器输入localhost/1702/加路径下的文件名

创建新端口

Apache =>httpd-vhosts-conf ,打开txt之后复制里面的端口文本，直接把端口改了就可以了，端口访问localhost:端口号

利用script标签跨域



PHP是一种通用开源服务端脚本语言，将程序嵌入到HTML文档中去执行，结果以纯 HTML 形式返回给浏览器

PHP: Hypertext Preprocessor “超文本预处理器”，1994年由Rasmus Lerdorf创建，刚刚开始仅仅是为了要维护他本人个人网页而制作的一个简单程序（Perl语言编写），原名Personal Home Page（PHP由此得名），后用C语言重新编写，改名Hypertext Preprocessor

格式

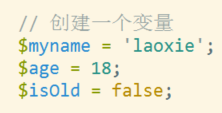
<?php

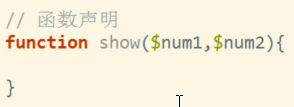
……代码

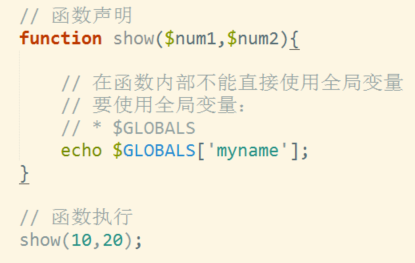
?>

输出

echo ’hello my php

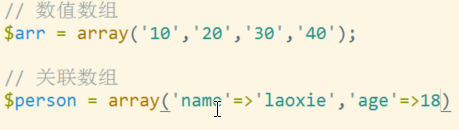


’



json\_encode（变量）

json\_decode（变量）

类似js的对象

JSON\_UNESCAPED\_UNICODE阻止中文变成unicode码

count（数组变量名）

# 分界标示符

<?php //开始

//...php代码

?> //结束

# 注释

与js一样，分单行和多行注释

- 单行注释：//

- 多行注释：/\*\*/

# 输出语句

## echo

可以输出一个或多个字符串（字符串可以包含 HTML 标签），速度较快，一般用于向前端返回数据

<?php

//输出一个字符

echo "Hello world!<br>";

//输出多个字符

echo "This", " string", " was", " made", " with multiple parameters.";

?>

## print\_r

打印关于变量的信息，适用于数组、对象的打印，一般用于测试

# 变量

## 命名规则

以 $ 符号开始，后面跟着变量的名称（$称为标识符，不属于变量组成部分）

只能包含字母数字字符以及下划线（A-z、0-9 和 \_ ）

必须以字母或者下划线字符开始

不能包含空格

区分大小写

//PHP 没有声明变量的命令。

//变量在第一次赋值时被创建：

<?php

$txt="Hello world!";

$x=5;

$y=10.5;

?>

## 全局变量

### $GLOBALS数组

PHP 将所有全局变量存储在一个名为 $GLOBALS数组中。这个数组可以任意位置访问，

格式：$GLOBALS[变量名]，其中变量名不带$。

<?php

$x='global x';

function myTest(){

//echo $x;//报错

//正确写法

echo $GLOBALS['x'];

//在函数中创建全局变量

$GLOBALS['y'] = $GLOBALS['x'] + $GLOBALS['y'];

}

myTest();

echo $y;

?>

### global关键字

用于函数内访问全局变量。

在函数内调用全局变量，除了用$GLOBALS数组，还可以在变量前加上 global 关键字

<?php

$x=5;

$y=10;

function myTest(){

global $x,$y;

$y=$x+$y;

}

myTest();

echo $y; // 输出 15

?>

### 超级全局变量

#### $GLOBALS

#### $\_POST / $\_GET

被广泛应用于收集表单数据，常用于ajax请求等操作

post和get的区别：

1get的参数会直接显示，速度更快，但是安全性低

2post更安全，但是会比较慢

3如果数据很大，一定要用post，无法用get

4jsonp只能用get

#### $\_COOKIE

用于收集前端发送过来的cookie数据



#### $\_REQUEST

变量包含了 $\_GET、$\_POST 和 $\_COOKIE 的内容

#### $\_SESSION

服务器版cookie

# 常量

命名规则与变量一致，但常量名不需要加 $ 修饰符。

常量值被定义后，在脚本的其他任何地方都不能被改变。

默认是全局变量，可以在整个运行的脚本的任何地方使用。

常量名建议全部大写

格式：

define(name,value)

name：必选参数，常量名称，即标志符。

value：必选参数，常量的值。

define("EN\_NAME", "laoxie");

# 运算符

## 算术运算符

+, -, \*, /, %

取反-，

## 赋值运算符,递增/递减运算符,比较运算符

等于：x == y

恒等于：x === y

不等于：x != y ，x <> y

不恒等于：x !== y

大于：x > y

小于：x < y

大于等于：x >= y

小于等于：x <= y

## 逻辑运算符

与：x and y，x && y

或：x or y，x || y

异或：x xor y

非：!x

## 三元运算符

格式：expr1 ? expr2 : expr3

## 条件语句

if语句

与javascript一致

switch语句

与javascript一致

## 数据类型

String（字符串）

并置运算符“.”

<?php

$txt1="你是我的小呀小苹果";

$txt2="怎么爱你都不嫌多";

echo $txt1 . ", " . $txt2;

?>

strlen()

获取字符串长度，得到的字符的字节数

strpos()

查找某个字符在字符串中的索引，如果未找到匹配，则返回 false

strpos("Hello world!","world");//=>6

Integer（整型）

Float（浮点型）

Boolean（布尔型）

Array（数组）

数组是一个能在单个变量中存储多个值的特殊变量。

类型

数值数组 - 带有数字索引值的数组

关联数组 - 带有指定的键的数组，每个键关联一个值

多维数组 - 包含一个或多个数组的数组

### array() 函数用于创建数组

//数值数组

$cars=array("Volvo","BMW","Toyota");

//关联数组

$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");

count()

获取数组长度

遍历数值数组

for

foreach()

<?php

//遍历数值数组：for循环

$cars=array("Volvo","BMW","Toyota");

$arrlength=count($cars);

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

echo $cars[$x] . "<br>";

}

//遍历关联数组：foreach..as

$age=array("Peter"=>"35","Ben"=>"37","Joe"=>"43");

foreach($age as $x=>$x\_value){

echo "Key=" . $x . ", Value=" . $x\_value;

echo "<br>";

}

?>

数组排序

sort() 对数组进行升序排列

rsort() 对数组进行降序排列

asort() 根据关联数组的值，对数组进行升序排列

ksort() 根据关联数组的键，对数组进行升序排列

arsort() 根据关联数组的值，对数组进行降序排列

krsort() 根据关联数组的键，对数组进行降序排列



Object（对象）

NULL（空值）

函数

内建函数

PHP 的真正威力源自于它的函数。在 PHP 中，提供了超过 1000 个内建的函数，请参考文档

自定义函数

参数

return返回值

function add($x,$y){

$total=$x+$y;

return $total;

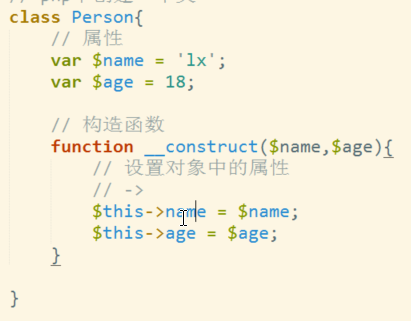
}

# 面向对象

面向对象术语

## 类

定义一个类



类使用 class 关键字后加上类名定义。

类名后的一对大括号{}内可以定义变量和方法。

类的变量使用 var 来声明, 变量也可以初始化值。

函数定义类似 PHP 函数的定义，但函数只能通过该类及其实例化的对象访问。

<?php

class Person{

// 成员属性

var $name;

var $age = 18;

//成员方法

function setName($name){

$this->name = $name;

}

function getAge(){

return $this->age;

}

}

?>

实例化对象

$p = new Person();

调用成员属性/方法

在实例化对象后，我们可以使用‘->’操作该对象调用成员属性/方法

$p->setName('老谢');

$p->name;

## 构造函数

\_\_construct构造函数是一种特殊的方法。主要用来在创建对象时初始化对象，写在创建对象的语句中。

class Web{

function \_\_construct( $par1, $par2 ) {

$this->url = $par1;

$this->title = $par2;

}

}

访问控制

PHP 对属性或方法的访问控制，是通过在属性/方法前面添加关键字来实现。

public（公有，默认）：公有的类成员可以在任何地方被访问。

private（私有）：私有的类成员只能在类本身中访问。

protected（受保护）：受保护的类成员只能在类本身、子类、父类中访问。

static（静态）：声明类属性或方法为静态，就可以不实例化类而直接访问

访问方式：类名::方法

PS：

- 类属性必须定义为公有，受保护，私有之一。如果用 var 定义，则被视为公有。

- 类中的方法可以被定义为公有，私有或受保护。如果没有设置这些关键字，则该方法默认为公有。

## 继承

使用关键字 extends 来继承一个类，继承后子类就拥有父类的属性和方法（私有除外），格式如下：

class Man extends Person {

// 添加成员函数

function setAge($age){

$this->age = $age;

}

//重写方法

function setName($name){

$this->name = $name;

return $name;

}

}

私有属性/方法不能被继承

生成动态页面

表单提交

$\_GET[‘字符串’]

$\_POST

$\_REQUREST

session

session\_start()：启动新会话

session\_destroy()：销毁一个会话中的全部数据

time()：获取当前时间，返回自1970 年 1 月 1 日 0时到当前时间的秒数

session\_start();

if(!isset($\_SESSION['last-access'])){

$\_SESSION['last-access'] = time();

}

//超过60s为超时

$overtime = (time() - $\_SESSION['last-access']) > 60;

## 接口

利用php编写数据接口

本地数据

直接返回数据

json\_encode(,常量);

把数组转成字符串

php5.4+ 使用JSON\_UNESCAPED\_UNICODE防止中文转义

json\_decode(json,assoc);

把字符串转成数组/对象

json：待解码的 json string 格式的字符串

assoc：默认false, 当该参数为 true 时，将返回 array 而非 object

文件的读取与写入

fopen(filename,mode)：打开文件

使用fopen函数打开文件时，你首先需要明确：

打开文件干什么？是读文件中的数据呢，还是将数据写入文件，亦或者读写文件

另外你需要考虑如果文件中已经存在相关数据，你是覆盖原有文件中的数据呢，还是仅仅将新数据添加至文件末尾

文件模式:

r 以只读方式打开文件，从文件头开始读

r+ 以读写方式打开文件

w 以写入方式打开文件，从文件头开始写。如果文件不存在则尝试创建，如果文件存在，则先删除文件中的内容

w+ 以读写方式打开文件，从文件头开始读写。如果文件不存在则尝试创建，如果文件存在，则先删除文件中的内容

a 以写入方式打开，从文件末尾开始追加写。如果文件不存在则尝试创建

a+ 以读写方式打开，从文件末尾开始追加写写或者读。如果文件不存在则尝试创建。

fread()：读取内容

fwrite(写到哪里，写什么)：写入内容

fclose()：关闭文件

filesize()：读取文件字符长度//数组的长度count（）//字符串长度strlen（）

//以读取模式打开文件

$myfile = fopen('./data/weibo.json', 'r');

//读取文件内容

$content = fread($myfile, filesize('./data/weibo.json'));

//关闭文件，减少资源占用

fclose($myfile);

数据库的增删改（后面课程详解）

获取远程JSON数据

API数据

curl：client Url

$url = 'http://wthrcdn.etouch.cn/weather\_mini?city=广州';

//初始化 cURL会话

$ch = curl\_init();

// 设置需要的选项

curl\_setopt($ch, CURLOPT\_URL, $url);

//执行会话

$contents = curl\_exec($ch);

//关闭会话

curl\_close($ch);

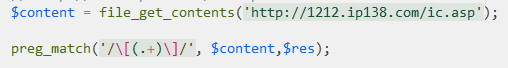
//输出内容

echo $contents;

## 爬虫

file\_get\_contents()

preg\_match\_all(正则或查找的内容，要在哪里查找，输入的接收变量，返回数组)如果只有一组就不用all



in\_array（内容,数组）判断某内容是否在数组里面