**PHP是什么?**

1. PHP(Hypertext Preprocessor)超文本预处理器，是一种通用开源脚本语言。
2. 在服务器上执行，结果以纯HTML形式返回给浏览器。
3. PHP文件可包含文本、HTML、JavaScript、php代码

**基本语法**

1. php可以放在文档的任何位置
2. 以<?php开始，以?>结束
3. 文件扩展名是”.php”
4. 通常包含HTML标签和一些php脚本代码。
5. 每个代码行都必须以分号结束。分号是一种分隔符，用于把指令区分开来。
6. 有两种在浏览器输出文本的基础命令：echo和print
7. //为单行注释，/\*\*/为多行注释。

**PHP变量**

1. **变量规则**
2. 变量以$符号开头，后面跟着变量的名称。
3. 变量名必须以字母或下划线开始。
4. 变量名只能包含字母数字字符及下划线。
5. 变量名不能包含空格。
6. 变量名区分大小写。
7. php没有声明变量的命令，在第一次赋值时被创建。
8. **变量作用域**

**local、global、static、parameter**

函数内访问全局变量需要global关键字或使用$GLOBALS[“var\_str”]数组。在php函数中拥有独立的作用域，局部变量会覆盖全局变量，即使局部作用域中没有全局变量的同名变量，也会被覆盖。

**PHP EOF(heredoc)**

1. 必须后接分号，否则编译不通过。
2. EOF可以使用任意其他字符进行代替，只需保证结束标识与开始标识一致。
3. 结束标识必须顶格独自占一行。

<?php

$name="变量会被解析";

$a=<<<EOF

$name<br><a>html格式会被解析</a><br/>双引号和Html格式外的其他内容都不会被解析"双引号外所有被排列好的格式都会被保留""但是双引号内会保留转义符的转义效果,比如table:\t和换行：\n下一行"

EOF;

echo $a;?>

**PHP数据类型**

1. String 可以使用单引号或双引号表示
2. Integer 可以采用十进制、十六进制、八进制
3. Float 带小数部分的数字，或是指数形式
4. Bool
5. Array

$cars=array(“Volvo”,”BMW”,”Toyota”);

1. Object

class Car

{

var $color;

function \_\_construct($color=”green”)

{

$this->color=$color;

}

function what\_color()

{

return $this->color;

}

}

1. Null 可以用来清空变量数据

**PHP输出函数**

<?php

$x=**array**("hello",13,29.232);

**echo**($x);//Array

**print**($x);//Array

**var\_dump**($x);

/\*

array(3) {

[0]=>

string(5) "hello"

[1]=>

int(13)

[2]=>

float(29.232)

}\*/

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

/\*

Array

(

[0] => hello

[1] => 13

[2] => 29.232

)

\*/

//var\_dump返回表达式的类型与值 print\_r仅返回值，相比调试代码使用var\_dump更便于阅读

?>

**PHP常量**

1. 常量被定义后，在脚本的其他地方均不能被修改。
2. bool define (string $name,mixed $value,[,bool $case\_insensitive=false])
3. 常量在定义后，默认是全局变量，及时常量定义在函数外也可以正常使用常量。
4. 使用常量时，不能在常量名前加$，不然会将常量转为新的未定义变量使用，会导致报错。

<?php

//使用常量时，不能在常量名前加$

//区分大小写的常量名

**define**("Hello","Hello World...");

**echo** Hello;//Hello World...

**echo** **PHP\_EOL**;

**echo** hello;//hello

//不区分大小写的常量名

**define**("World","World World...",**true**);

**echo** World;

**echo** **PHP\_EOL**;

**echo** world;

//常量在定义后默认是全局的

**function** test()

{

**echo** **PHP\_EOL**;

**return** Hello;

}

**echo** test();

?>

**PHP字符串操作**

1. 并置 $txt1.$txt2
2. strlen()
3. strpos() 在字符串内查找一个字符或一段指定的文本，找到则返回匹配的字符位置，否则返回false
4. mb\_strlen()和mb\_strpos()按字处理，strlen()和strpos()按字符数处理

<?php

//字符串拼接

$str1="hello...";

$str2="world...";

$combine\_str=$str1.$str2;

**echo** $combine\_str;

**echo** **PHP\_EOL**;

//字符串长度

**echo** **strlen**("hello你好吗");

**echo** **PHP\_EOL**;

**echo** **mb\_strlen**("hello你好吗");

//字符串查找子串

**echo** **PHP\_EOL**;

**echo** **strpos**("hello你好吗","好");

**echo** **PHP\_EOL**;

**echo** **mb\_strpos**("hello你好吗","好");

?>

**PHP运算符**

<?php

//变量引用

$x="hello world...";

$y=&$x;

$y=90.232;

**var\_dump**($x);

**echo** **PHP\_EOL**;

**var\_dump**($y);

**echo** **PHP\_EOL**;

/\*

//组合比较运算符

$a=10.232;

$b=89.5454;

echo $a<=>$b;

//$a>$b 为1

//$a==$b 为0

//$a<$b 为-1

\*/

?>

**PHP三元运算符**

1. (expr1)?(expr2):(expr3)
2. $username=$test?:’nobody’ //PHP5.3+
3. $username=$\_GET[‘user’] ?? ’nobody’ //PHP7+

**PHP数组**

<?php

/\*

数值数组:带有数字ID键的数组

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

\*/

//获取数组长度

$arr1=**array**("Volvo","BMW","Toyota");

**echo** **count**($arr1);

**echo** **PHP\_EOL**;

**for**($i=0;$i<**count**($arr1);$i++)

{

**echo** $arr1[$i];

**echo** **PHP\_EOL**;

}

//遍历数组

**foreach**($arr1 **as** $key=>$value)

{

**echo** "key->".$key." "."value->".$value;

**echo** **PHP\_EOL**;

}

//关联数组

$age\_arr=**array**("Peter"=>35,"Ben"=>37,"Joe"=>90);

**echo** "Peter's age->".$age\_arr["Peter"];

**echo** **PHP\_EOL**;

//遍历数组

**foreach**($age\_arr **as** $key=>$value)

{

**echo** "key->".$key." "."value->".$value;

**echo** **PHP\_EOL**;

}

/\*

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

\*/

?>

**PHP超级全局变量**

<?php

//换行的输出

**function** echo\_eol($arg)

{

**echo** $arg;

**echo** **PHP\_EOL**;

}

//$LOBALS 是一个包含了全部变量的全局组合数组。变量的名字就是数组的键。

$x=90.3232;

$y=23.2323;

**function** add()

{

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

}

add();

echo\_eol($z);

//$\_SERVER 是一个包含了诸如头信息(header)、路径(path)、以及脚本位置(script locations)等等信息的数组。

echo\_eol($\_SERVER["PHP\_SELF"]);

echo\_eol($\_SERVER["SERVER\_NAME"]);

echo\_eol($\_SERVER["HTTP\_HOST"]);

echo\_eol($\_SERVER["HTTP\_REFERER"]);

echo\_eol($\_SERVER["HTTP\_USER\_AGENT"]);

echo\_eol($\_SERVER["SCRIPT\_NAME"]);

?>

**php魔术常量**

<?php

//\_\_LINE\_\_ 文件中的当前行号

**echo** "第".\_\_LINE\_\_."行".**PHP\_EOL**;

//\_\_FILE\_\_ 文件的完整路径和文件名。如果用在被包含文件中，则返回被包含的文件名

**echo** "该文件位于".\_\_FILE\_\_.PHP\_EOL;

//\_\_DIR\_\_ 文件所在目录，如果用在被包含文件中，则返回被包含的文件所在的目录

**echo** "该文件位于".\_\_DIR\_\_.PHP\_EOL;

//\_\_FUNCTION\_\_ 函数名称，不能表示出从属的类空间

**function** test()

{

**echo** "函数名为".\_\_FUNCTION\_\_.PHP\_EOL;

}

test();

//\_\_CLASS\_\_ 类的名称

//\_\_TRAIT\_\_ trait的名称

**trait** SayHello

{

**public** **function** say\_hello()

{

**echo** "hello...".\_\_TRAIT\_\_.PHP\_EOL;

}

}

**class** Hello

{

**use** SayHello;

}

$hello=**new** Hello();

$hello->say\_hello();

//\_\_METHOD\_\_ 类的方法名

//\_\_NAMESPACE\_\_ 当前命名空间的名称

?>