

2-3数据类型

2-3数据类型	1
1. 课堂案例	2
课堂讲解	2
一、上节回顾	2
变量	2
常量	2
拼接字符串	2
二、学习目标	2
数据类型	2
引号的区别	2
三、教学过程描述	2
变量与常量的定义	2
四、小结	9
2. 课堂练习	9
3. 课后练习	9
4. 资料扩展	9
1、数据类型强制转换	9
2、数据类型的判断	10
2-1、var_dump	10
2-2、gettype(变量)	10
2-3、is系列的类型判断函数	11
2-4、is_null	11
2-5、unset	12
3、count() 函数	12
4、php 手册查看	12

1. 课堂案例

课堂讲解

一、上节回顾

变量

常量

拼接字符串

二、学习目标

数据类型

引号的区别

三、教学过程描述

变量与常量的定义

数据类型

1、标量

1-1、整型 (int)

整型是没有小数的数字。整型值可以使用十进制，十六进制，八进制或二进制表示，前面可以加上可选的符号(- 或者 +)。

```
<?php
$a = 1234; // 十进制数
$a = -123; // 负数
$a = 0123; // 八进制数 (等于十进制 83)
$a = 0x1A; // 十六进制数 (等于十进制 26)
```

1-2、浮点型 (float)

浮点数是有小数点或指数形式的数字。(也叫浮点数 float, 双精度数 double 或实数 real)

```
$a = 1.234;  
$b = 1.2e3;  
$c = 7E-10;
```

1-3、字符串 (string)

1-3-1、单引号赋值

```
$a='老虎老鼠傻傻分不清楚';  
echo $a;
```

1-3-2、双引号赋值

```
$b="我是一只小老虎";  
echo $b;
```

1-3-3、单引号与双引号的区别

- 1).单引号不能解析变量,双引号可以解析变量。
- 2).单引号不解析转义字符。双引号可以解析转义字符。单引号可以解析'\'
- 3).单引号中可以插入双引号, 双引号中可以插入单引号,相互插入的时候,对应的引号只是单纯的字符而已。
- 4).单引号使用变量需要用字符连接符.双引号也可以用。
- 5).双引号使用变量时候用{}定界符/"\${a} is a variable!"
- 6).使用的时候用单引号节省效率。

定界符的功能和双引号一样。

一般情况下都用单引号。。

SQL语句的时候用双引号, 提高开发效率

定界符大文章的使用定界符方便

SQL语句的时候用双引号

```
$name = 'hello';  
echo '$name word';  
echo "$name word";
```

1-3-4、定界符赋值

1、(Heredoc 结构)

第三种表达字符串的方法是用 `heredoc` 句法结构:`<<<`。在该运算符之后要提供一个标识符, 然后换行。接下来是字符串本身, 最后要用前面定义的标识符作为结束标志。

结束时所引用的标识符必须在该行的第一列, 而且, 标识符的命名也要像其它标签一样遵守 PHP 的规则: 只能包含字母、数字和下划线, 并且必须以字母和下划线作为开头。
注意: 可能有一个分号 (;) 外, 绝对不能包含其它字符。这意味着标识符不能缩进, 分号的前后也不能有任何空白或制表符

`[EOD]`

2、Nowdoc 结构

就象 `heredoc` 结构类似于双引号字符串, `Nowdoc` 结构是类似于单引号字符串的。`Nowdoc` 结构很象 `heredoc` 结构, 但是 `nowdoc` 中不进行解析操作。这种结构很适合用于嵌入 PHP 代码或其它大段文本而无需对其中的特殊字符进行转义。

一个 `nowdoc` 结构也用和 `heredocs` 结构一样的标记 `<<<`, 但是跟在后面的标识符要用单引号括起来, 即 `<<<'EOT'`。Heredoc 结构的所有规则也同样适用于 `nowdoc` 结构, 尤其是结束标识符的规则。

```
$str = <<<'EOD'
喜欢全世界的风景
EOD;
echo $str;exit;
```

1-4、布尔值

这是最简单的类型。`boolean` 表达了真值, 可以为 `TRUE` 或 `FALSE`。两个都不区分大小写。

布尔值为假的情况:

- 1、整形0
- 2、浮点型的0.0
- 3、空字符串和字符串'0'
- 4、布尔false
- 5、空数组array()
- 6、没有成员属性的对象(PHP4)
- 7、资源都是真
- 8、null

2、复合数据类型

2-1、数组(array();)

array 为数组变量, 可以是二维、三维或者多维数组, 其中的元素也很自由, 可以是 string、integer, 甚至是 array。

可以用 array() 语言结构来新建一个 array。它接受一定数量用逗号分隔的 key => value 参数对。

key 只能是整数和字符串

```
array( key => value
```

```
, ...
```

```
)
```

// 键(key) 也叫下标, 可是一个整数 integer 或字符串 string

// 值(value) 可以是任意类型的值

//PHP5.4起

```
$array = [
```

```
    key => value ,
```

```
    ...
```

```
];
```

2-1-1数组的分类

按照下标分类:

索引数组: 下标为整型

关联数组: 下标为字符串

混合数组: 下标可以为整型和字符串

按照元素的值是否是数组类型分类

一维数组: 元素的值都是标量类型

二维数组: 元素的值有数组类型

多维数组: 元素嵌套了多层数组

1、索引数组

索引是自动分配的(索引从 0 开始)

```
//索引是自动分配的（索引从 0 开始）|
$array2 = array(
    'one',
    'two',
    'three'
);
```

2、关联数组

关联数组是使用您分配给数组的指定键的数组。

下标(键)如果是字符串必须用引号建议下标最好不要使用中文，键不能重复

```
//关联数组
$array2 = array(
    '1'    => 'one',
    '2'    => 'two',
    'name' => 'three'
);
//或者
$array2[1]    = 'one';
$array2[2]    = 'two';
$array2['name'] = 'three';
```

3、多维数组

多维数组指的是包含一个或多个数组的数组。

```
(
    array("Volvo",22,18),
    array("BMW",15,13),
    array("Saab",5,2),
    array("Land Rover",17,15)
);
```

4、混合数组

下标可以为整型和字符串

2-1-2、定义数组方式

1、直接赋值法

```
/**
 *使用直接赋值法
 */
//定义一个不指定下标的数组
$arr1[] = 10;
$arr1[] = 20;
$arr1[] = 30;
var_dump($arr1);

//指定下标为整数的数组
$arr2[1] = 10;
$arr2[2] = 20;
$arr2[3] = 30;
var_dump($arr2);

//定义下标为字符串的数组
$arr3['name'] = '刘备';
$arr3['age'] = 25;
$arr3['sex'] = '男';
var_dump($arr3);
```

2、使用array()

```
/**
 *array():语言结构
 */
//定义一个不指定下标的数组
$arr1 = array(10, 20, 30); //元素直接用逗号隔开
var_dump($arr1);

//指定整数下标的
$arr2 = array(1 => 10, 2 => 20, 3 => 30);
var_dump($arr2);

//指定字符串下标
```

```
$arr3 = array('name' => '刘备', 'age' => 25, 'sex' => '男');  
var_dump($arr3);
```

3、快捷定义

```
/**  
 *快捷方式:[]  
 */  
  
//定义个不指定下标的数组  
$arr1 = [10, 20, 30];  
var_dump($arr1);  
  
//定义指定整数下标的数组  
$arr2 = [1 => 10, 2 => 20, 3 => 30];  
var_dump($arr2);  
  
//定义字符串下标的数组  
$arr3 = ['name' => '刘备', 'age' => 25, 'sex' => '男'];  
var_dump($arr3);
```

2-2、对象(object)

要创建一个新的对象 object, 使用 new 语句实例化一个类:



```
class foo  
{  
    function do_foo()  
    {  
        echo "Doing foo.";  
    }  
}  
  
$bar = new foo;  
$bar->do_foo();
```

3、特殊数据类型

3-1、资源(resource)

资源

resource

是一种特殊变量, 保存了到外部资源的一个引用。资源是通过专门的函数来建立和使用的。

3-2、null(空类型)

1、没有声明的变量

2、直接赋值为 null

3、被声明但是被 unset() 掉

注意: 空字符串和0都不是空类型注意区分!! 只有以上三种情况才为空!

四、小结

1、单双引号的区别

2、数组的格式

3、4种标量

5、变量和常量的定义

6、定界符的区别

2. 课堂练习

- 1、 定义一个数组, 下标是从1开始的
- 2、 定义个数组下标分别是:name, sex

3. 课后练习

- 1、定义一个数组下标从0开始的
- 2、定义一个数组下标是name键值是西游记的人物名
- 3、在第二题中插入一个name等于卡卡罗特的人

4. 资料扩展

1、数据类型强制转换

```

/**
 * 类型的强制转换
 */
$foo = 10;
$bar = (bool)$foo;
var_dump($foo);
var_dump($bar);

```

(int), (integer) - 转换成整型

(bool), (boolean) - 转换成布尔型

(float), (double), (real) - 转换成浮点型

(string) - 转换成字符串

(array) - 转换成数组

(object) - 转换成对象

2、数据类型的判断

2-1、var_dump

此函数显示关于一个或多个表达式的结构信息，包括表达式的类型与值。数组将递归展开值，通过缩进显示其结构。

```

$a = 123;
$b = '大王叫我来巡山';
var_dump($a); // 输出 int 123
var_dump($b); // 输出 string '大王叫我来巡山' (length=21)

```

2-2、gettype(变量)

可以查询所有的8种类型【返回值】：代表该类型的字符串

```
$a = 123;  
$b = 123.12;  
$c = '小心猴子变苍蝇';  
$arr = array(1,2,3);  
echo gettype($a). '<br>'; //输出integer  
echo gettype($b). '<br>'; //输出double  
echo gettype($c). '<br>'; //输出string  
echo gettype($arr). '<br>'; //输出array
```

2-3、is系列的类型判断函数

标 量: is_int() is_float() is_string() is_bool()

复合类型: is_array() is_object()

特殊类型: is_resource() is_null()

is_numeric() 是不是数值类型

is_scalar() 是不是标量

```
$a=123;  
$jg=is_float($a);  
var_dump($jg); //false
```

2-4、is_null

检测变量是否为 NULL

```

$a;
$b = false;
$c = '';
$d = 0;
$e = null;
$f = array();
//...
var_dump(is_null($a)); //boolean true
var_dump(is_null($b)); //boolean false
var_dump(is_null($c)); //boolean false
var_dump(is_null($d)); //boolean false
var_dump(is_null($e)); //boolean true
var_dump(is_null($f)); //boolean false

```

2-5、unset

unset(变量) 删除的是变量名和变量值之间的联系而已 不删除值

```

$a = 100;
$b = &$a;
unset($a);
echo $b;

```

3、count() 函数

获取数组的长度 - count() 函数

4、php 手册查看