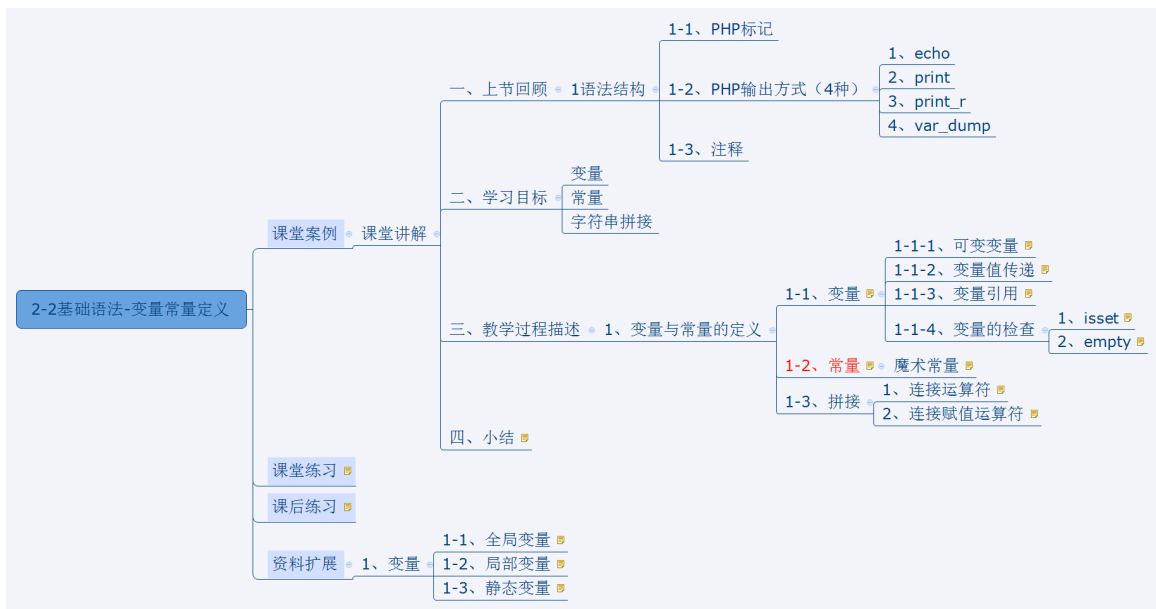


## 2-2基础语法-变量常量定义

2-2	基础语法	变量常量定义	1
1.	课堂案例		2
	课堂讲解		2
	一、上节回顾		2
	1语法结构		2
	二、学习目标		3
	变量		3
	常量		3
	字符串拼接		3
	三、教学过程描述		3
	1、变量与常量的定义		3
	四、小结		7
2.	课堂练习		7
3.	课后练习		8
4.	资料扩展		8
	1、变量		8
	1-1、全局变量		8
	1-2、局部变量		9
	1-3、静态变量		9



## 1. 课堂案例

### 课堂讲解

#### 一、上节回顾

##### 1语法结构

###### 1-1、PHP标记

###### 1-2、PHP输出方式（4种）

1、echo

2、print

3、print\_r

4、var\_dump

###### 1-3、注释

## 二、学习目标

变量

常量

字符串拼接

## 三、教学过程描述

### 1、变量与常量的定义

#### 1-1、变量

变量 可变的量

命名规则：

- 1、以\$开始
- 2、可以用中文，但是不推荐使用，不是所有版本都支持中文。
- 3、数字 可以使用数字 但是不能开头。
- 4、特殊字符不可以使用，除了\_之外 放前后都可以
- 5、变量的命名要有意义
- 6、变量是大小写敏感的

\$a 和 \$A 是不同的两个变量

```
1  <?php
2  //声明一个变量
3  $a = 'hello word';
4  //输出一个变量
5  echo $a;
```

#### 1-1-1、可变变量

所谓可变变量，即把变量的值作为变量名

```
$a = 'b';
$b = '可变变量';
echo $$a;
```

注意:使用可变变量时,被用来作为变量名的变量值一定要符号变量的命名规则才能使用!

### 1-1-2、变量值传递

// 这就是值传递

```
$val1 = 1;
```

```
$val2 = $val1;
```

### 1-1-3、变量引用

在PHP 中引用的意思是:不同的名字访问同一个变量内容。PHP

的引用允许你用两个变量来指向同一个内容。

```
$a="ABC";
$b =&$a;
echo $a;//这里输出:ABC
echo '<br>';
echo $b;//这里输出:ABC
echo '<br>';
$b="EFG";
echo $a;//这里$a的值变为EFG 所以输出EFG echo $b;//这里输出EFG
```

### 1-1-4、变量的检查

#### 1、isset

isset()只能用来判断是否为NULL和未定义

```

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

```

## 2、empty

只要数据类型是否为空或假, empty()就输出true。

```

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

```

## 1-2、常量

什么是常量: 石碑上的字刻上去就不会改了。

可以用 `define()` 函数来定义常量, 在 PHP 5.3.0 以后, 可以使用 `const` 关键字在类定义之外定义常量。一个常量一旦被定义, 就不能再改变或者取消定义。

```
define("CONSTANT", "Hello world.");  
echo CONSTANT;
```

常量的命名

- 1、常量也用英文, 可以用中文, 但是不提倡。
- 2、可以用数字, 不能放开头
- 3、特殊字符不能用, 除了 `_` 可以在任何位置
- 4、命名有意义
- 5、常量默认大小写敏感, 但是可以设置为不敏感 `define(参数, 参数, true)`

第三个参数默认为false(大小写敏感) 改为true则不区分大小写

常量的修改或删除

一旦定义 不可以删除或者修改。

常量的作用范围

常量的作用范围是全局的。

在实际开发中, 常量最常用于哪些地方?

- 1) 连接数据库的信息定义成常量, 如数据库服务器的用户名、密码、数据库名、主机名;
- 2) 将站点的部分路径定义成常量, 如web绝对路径, smarty的安装路径, model、view或者controller的文件夹路径;
- 3) 网站的公共信息, 如网站名称, 网站关键词等信息。

## 魔术常量

- 1、`__LINE__` 文件中的当前行号。
- 2、`__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。自PHP 4.0.2起, `__FILE__`总是包含一个绝对路径(如果是符号连接, 则是解析后的绝对路径), 而在此之前的版本有时会包含一个相对路径。
- 3、`__DIR__` 文件所在的目录。如果用在被包括文件中, 则返回被包括的文件所在的目录。它等价于 `dirname(__FILE__)`。除非是根目录, 否则目录中名不包括末尾的斜杠。(PHP 5.3.0中新增)
- 4、`__FUNCTION__` 函数名称 (PHP4.3.0新加)。自PHP5起本常量返回该函数被定义时的名字 (区分大小写)。在PHP4中该值总是小写字母的。
- 5、`__CLASS__` 类的名称 (PHP4.3.0新加)。自PHP5起本常量返回该类被定义时的名字 (区分大小写)。在PHP4中该值总是小写字母的。类名包括其被声明的作用区域 (例如 `Foo\Bar`)。注意PHP5.4起 `__CLASS__` 对trait也起作用。当用在trait方法中时, `__CLASS__` 是调用trait方法的类的名字。

- 6、\_\_TRAIT\_\_ Trait的名字（PHP5.4.0新加）。自PHP5.4起此常量返回trait被定义时的名字（区分大小写）。Trait 名包括其被声明的作用区域（例如 Foo\Bar）。
- 7、\_\_METHOD\_\_类的方法名（PHP5.0.0新加）。返回该方法被定义时的名字（区分大小写）。
- 8、\_\_NAMESPACE\_\_当前命名空间的名称（区分大小写）。此常量是在编译时定义的（PHP5.3.0新增）。

## 1-3、拼接

### 1、连接运算符

变量与变量、字符串之间的拼接使用拼接号“.”

```
$a = "World";  
$b = "Hello!";  
echo $a.$b;//变量的拼接  
echo $a.$b.'你好! ';//变量与字符串的拼接
```

### 2、连接赋值运算符

连接赋值运算符(“.=”)，它将右边参数附加到左边的参数之后。

```
$a = "Hello ";  
$a .= "World!";  
echo $a;
```

## 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、变量和常量的定义
- 2、字符串的拼接
- 3、数组的定义

### 3. 课后练习

将大小写不同的变量输出显示

```
name=hello world1  
NAME=hello world2  
NaME=hello world3
```

常量和变量有哪些区别？

- 1) 常量前没有\$符号；
- 2) 常量只能通过define()定义, 而不能通过赋值语句定义；
- 3) 常量可以在任何地方定义和访问, 而变量定义的访问有固定的规则；
- 4) 常量一旦定义就不能被重新定义或者取消定义, 而变量而通过赋值方式重新定义；
- 5) 常量的值只能是标量数据, 而变量的数据库类型有8中原始数据类型。

### 4. 资料扩展

#### 1、变量

##### 1-1、全局变量

作用域是指在一个脚本中某个变量在哪些地方可以使用或可见。



```

/**
 * 全局变量
 */
$a = '全局变量';

function test()
{
    global $a,$b;
    $b = $a;
}

test();
echo $b;

```

### 1-2、局部变量

任何用于函数内部的变量将被限制在局部函数范围内。

```


function test()
{
    $a = 1;
    echo $a;
}

test();
//或者
$a = 1;
echo $a;

```

### 1-3、静态变量

在函数退出时, 这个变量始终存在, 不被销毁, 但不能被其它函数使用, 当再次进入该函数时, 将保存上次  
的结果。



```
function test()
{
    static $test;
    $test .= '哈';
    echo $test.'<br>';
}
test();
test();
test();
```