# Lesson27

目录

[Lesson27 1](#_Toc488402877)

[回顾 1](#_Toc488402878)

[命名空间 2](#_Toc488402879)

[定义： 3](#_Toc488402880)

[子命名空间的定义 5](#_Toc488402881)

[使用 5](#_Toc488402882)

[非限定名称 5](#_Toc488402883)

[限定名称 6](#_Toc488402884)

[完全限定名称 6](#_Toc488402885)

[全局空间 6](#_Toc488402886)

[OOP中常用的函数 7](#_Toc488402887)

[Class\_exists() 7](#_Toc488402888)

[method\_exists() 7](#_Toc488402889)

[Property\_exists() 8](#_Toc488402890)

[Get\_class\_methods() 9](#_Toc488402891)

[Get\_class([可选的对象名]) 9](#_Toc488402892)

[剩下的自己去看，自己补充 10](#_Toc488402893)

[OOP中的魔术常量 10](#_Toc488402894)

[\_\_CLASS\_\_ 10](#_Toc488402895)

[\_\_METHOD\_\_ 10](#_Toc488402896)

[\_\_NAMESPACE\_\_ 10](#_Toc488402897)

[小结 11](#_Toc488402898)

[验证码的封装 12](#_Toc488402899)

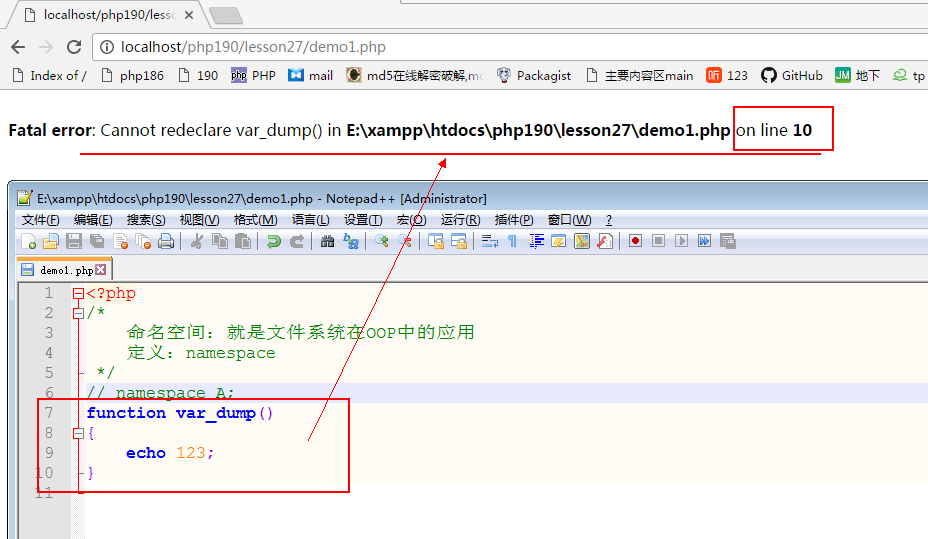
[预习 12](#_Toc488402900)

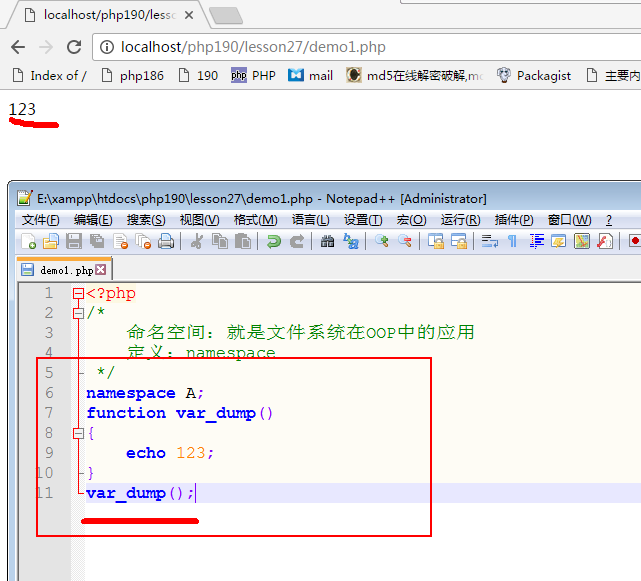
# 回顾

* 方法的重载
  + \_\_call(方法名，array 参数列表)
    - 作用：调用不存在或者不可见的方法时，自动调用
  + \_\_callStatic(方法名，array 参数列表)
    - 作用:调用不存在或不可见的静态方法时，自动调用
* 序列化
  + 作用：为了传输和保存，PHP自带的格式
  + 函数：
    - String Serialize(mixed)
    - Mixed Unserialize(string)
    - 所有的变量都可以序列化，对于对象的序列化要注意：
      * 1.序列化的结果只会保存所有的对象的属性，不包括方法
      * 2.反序列化的时候，如果结果是对象，需要在当前文件中有类的定义
  + 魔术方法：
    - \_\_sleep()
      * 序列化的时候自动调用
      * 返回需要存储的属性的名，并且是数组形式
    - \_\_wakeup()
      * 反序列化的时候自动调用
    - 这两个方法都可以对属性进行操作
* 魔术方法
  + \_\_toString()
    - 把对象当字符串输出时自动调用
    - 返回值必须是string
  + \_\_invoke()
    - 把对象当作函数调用时自动调用
  + \_\_debugInfo()
    - 把对象var\_dump()打印时自动调用
* 类型约束|访问控制
  + 就是函数中，形参前面的类型。
  + 作用：要求传入的实参必须是对应的类型，否则报错
  + PHP7之后支持所有的类型的约束
* 抽象类
  + 关键词：abstract
  + 修饰范围：类和方法，不能修饰属性
  + 特点：
    - 1.抽象类不能实例化
    - 2.抽象类中可以存在普通方法，抽象方法必须得存在抽象类中
    - 3.子类可以通过继承抽象类来实现抽象方法
    - 4.所有的抽象方法必须得实现
    - 5.抽象方法不能有函数体，并且分号结尾
* 接口
  + 关键词：interface
  + 修饰范围：只能用在类上
  + 特点：
    - 1.接口不能实例化，要想实现结果的功能，必须得通过子类实现
    - 2.实现的关键词implements
    - 3.接口中的所有方法都是抽象的，公共的方法，可以不用写public
    - 4.一个子类可以同时实现多个接口，接口名之间使用逗号隔开
    - 5.接口中的方法必须实现
    - 6.如果要实现多个接口，那么接口中不能有重名的方法
    - 7.接口中只能存在方法，常量，不能有属性

# 命名空间

* 什么是命名空间，有什么用？
  + 说白了，就好比文件系统。多个文件按照规律放在指定的目录下，访问的时候需要通过路径访问，此处的路径信息就是命名空间
* 解决的问题：
  + 1.自定义的代码和别人的代码的重名问题

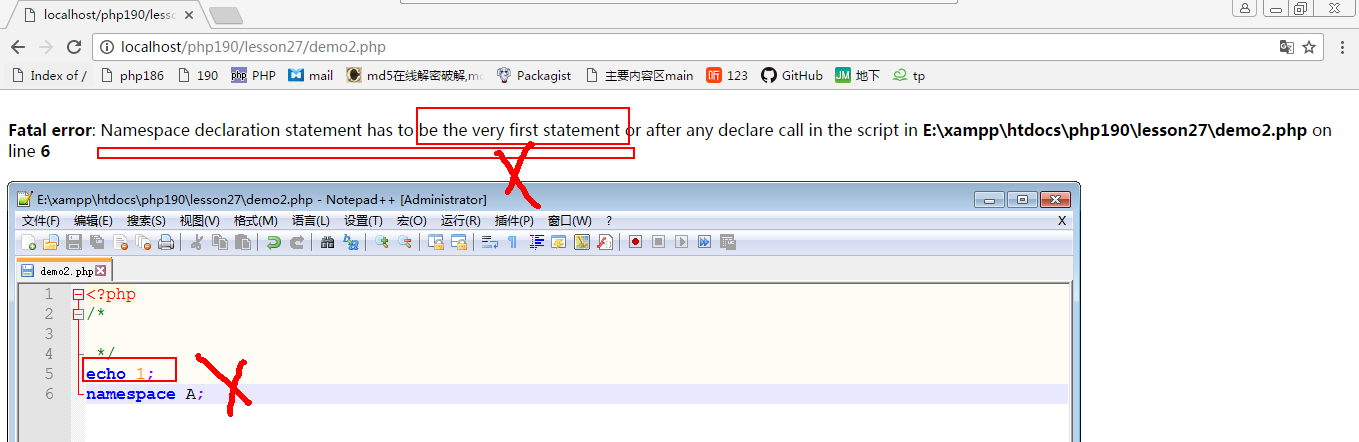




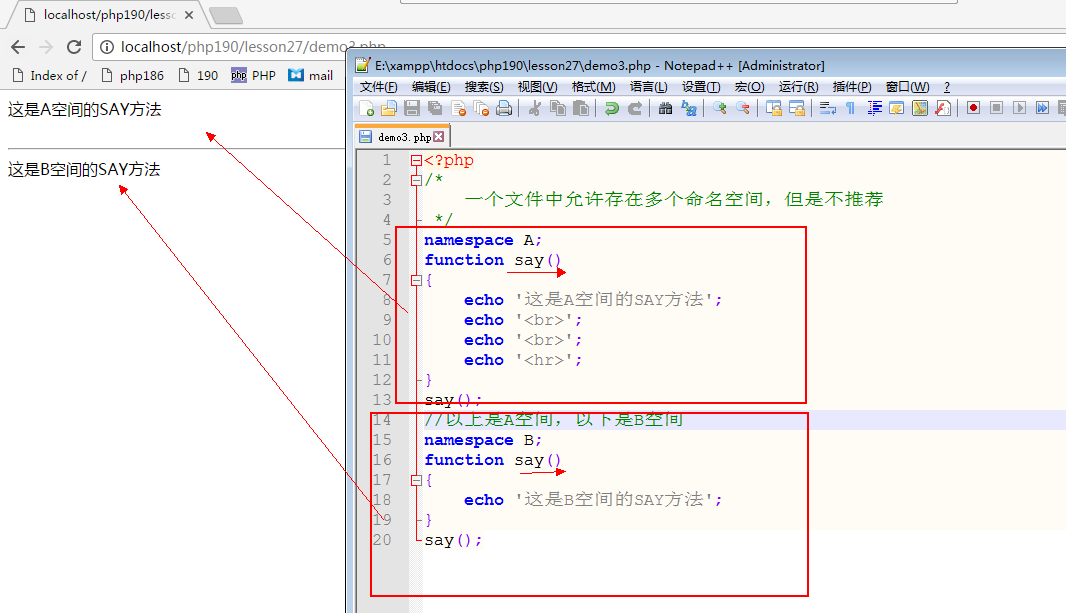
* + 2.提高代码可读性

# 定义：

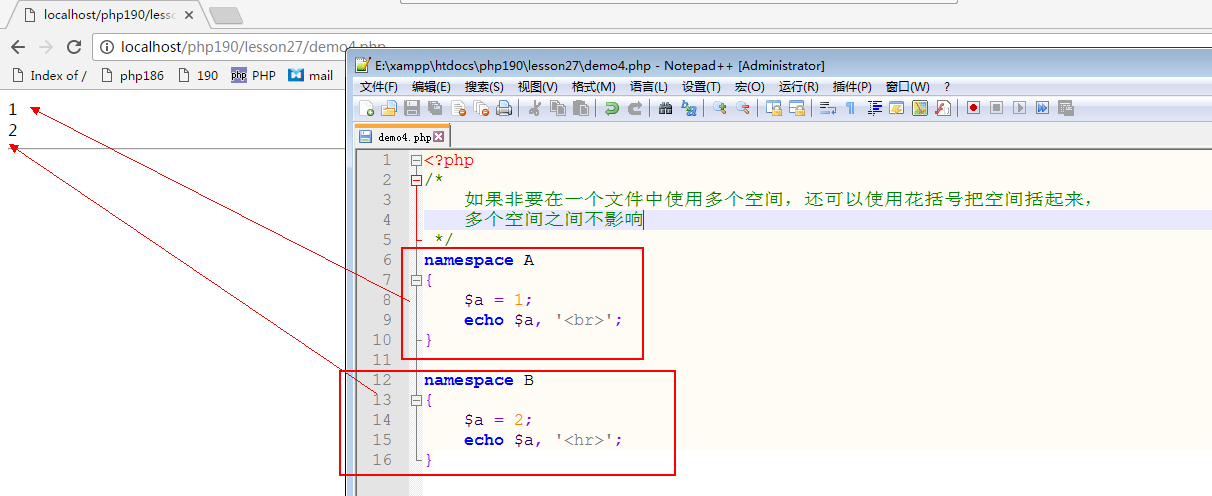
* **Namespace**
  + **Namespace XXX**
  + **注意：**
    - **1.命名空间的定义必须得在文件的首行，不能放其他位置。注释不算，也就是说命名空间定义的前面不能有代码的执行**



* **2.一个文件中允许有多个命名空间存在**

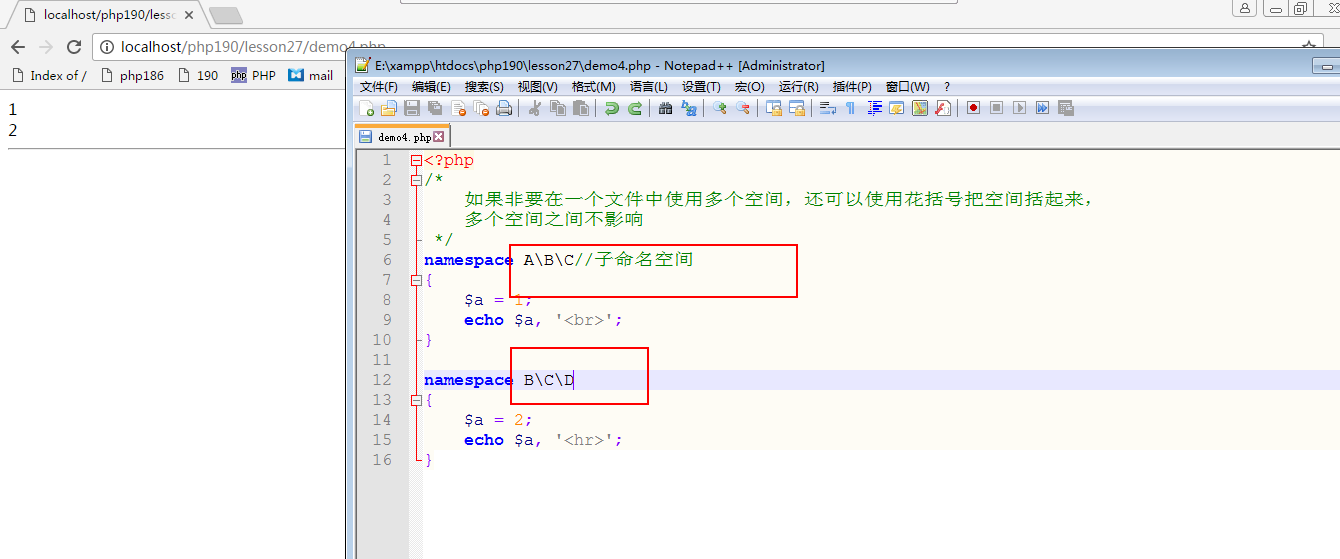


* **3如果想在一个文件中使用多个命名空间，建议，拿花括号把空间包起来**



子命名空间的定义

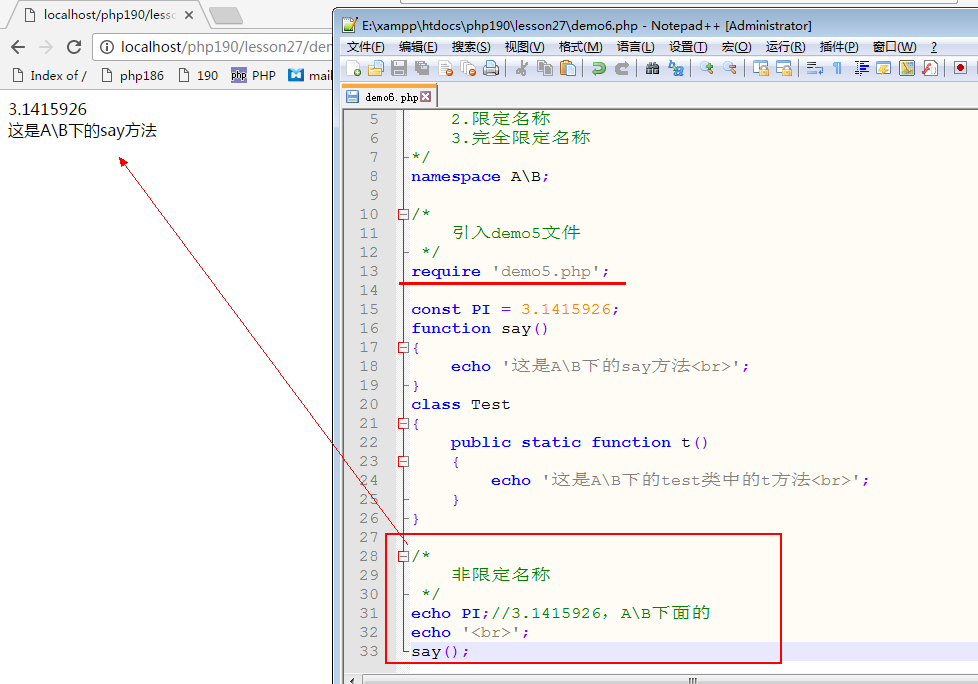
* 就是在命名空间基础上，使用\进行分隔定义的命名空间
* 说白了，就是自定义目录下，再定义目录，E:\A\B\C



## 使用

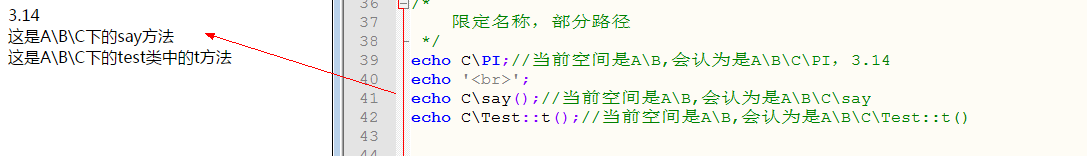
### 非限定名称

* 相当于访问文件1.jpg就表示当前目录下的1.jpg



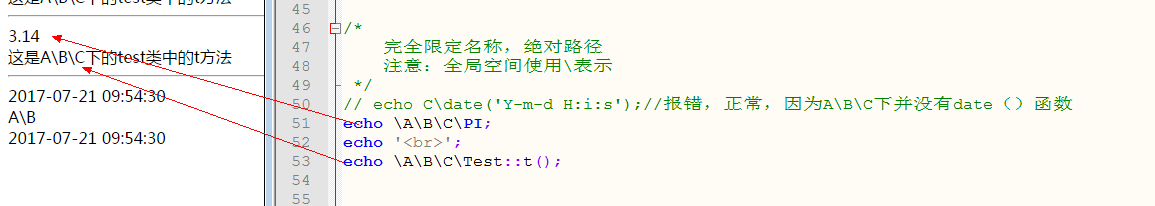
### 限定名称

* 相当于访问文件a\1.jpg，表示是当前目录下的a\1.jpg

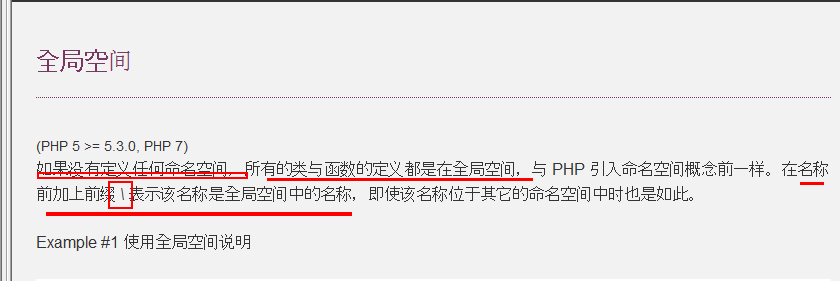


### 完全限定名称

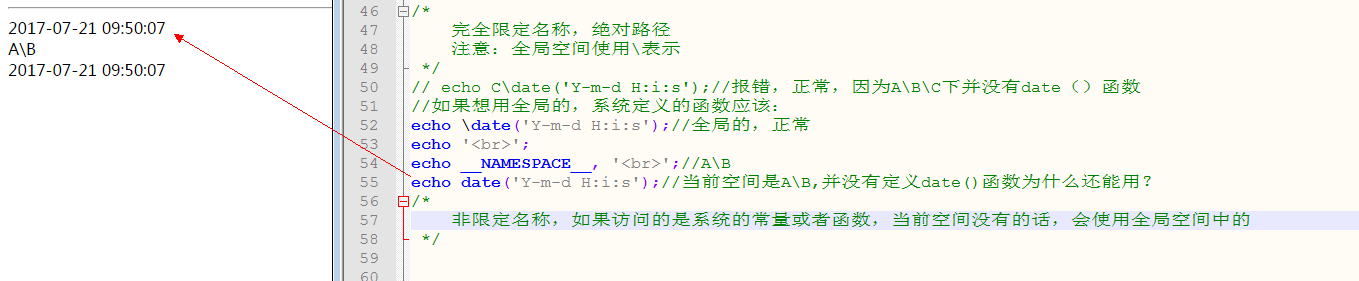
* 相当于访问文件：E：[\\a\1.jpg](file:///\\a\1.jpg)，就是绝对路径E：\a\1.jpg



### 全局空间



* **全局命名空间，表示：\，表示顶级空间，也就是全局空间，根目录**

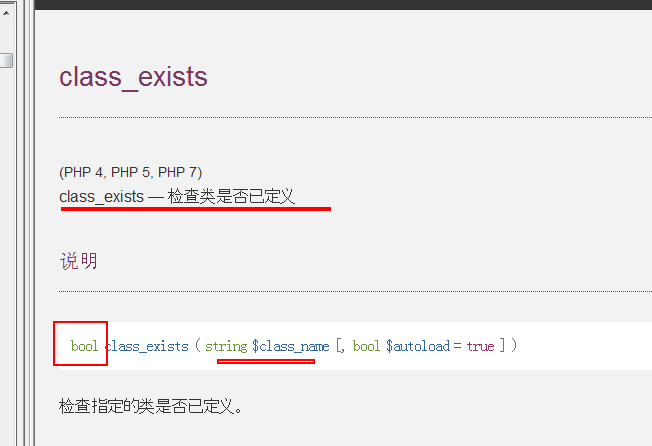


* **注意：**
  + **如果是非限定名称的方法访问系统存在的函数或者常量，如果当前空间没有，会自动去调用全局空间的**

# OOP中常用的函数

## Class\_exists()

* 判断类是否存在



## method\_exists()

* 检测方法是否存在



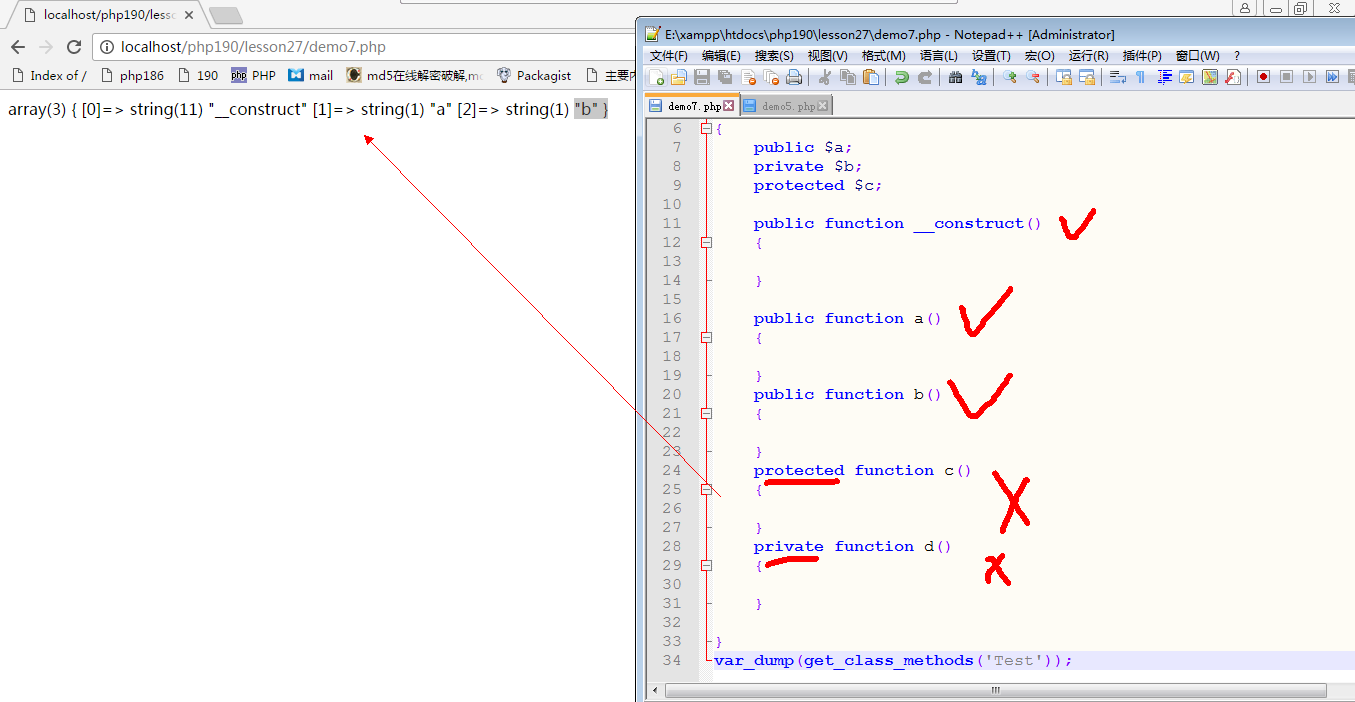
## Property\_exists()

* 检测属性是否存在或者可见



## Get\_class\_methods()

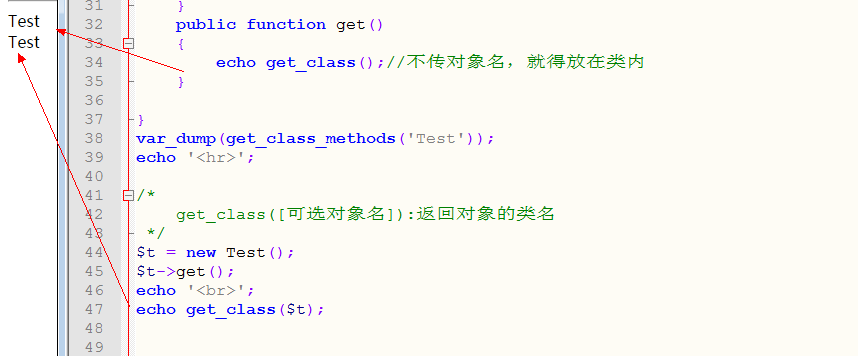
* 获取类中所有的公共方法，返回数组
* **注意：私有的，保护的方法拿不到**



## Get\_class([可选的对象名])

* 返回对象的类名





## 剩下的自己去看，自己补充

Ppt6

# OOP中的魔术常量

* 在类内部使用

## \_\_CLASS\_\_

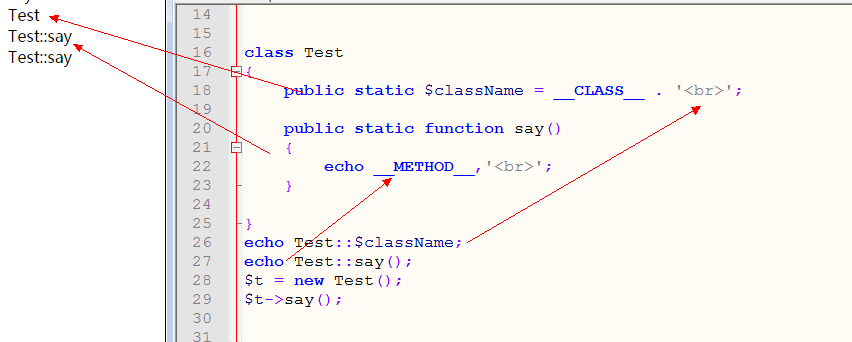
* 返回当前的类名

## \_\_METHOD\_\_

* 返回所在方法的方法名

## \_\_NAMESPACE\_\_

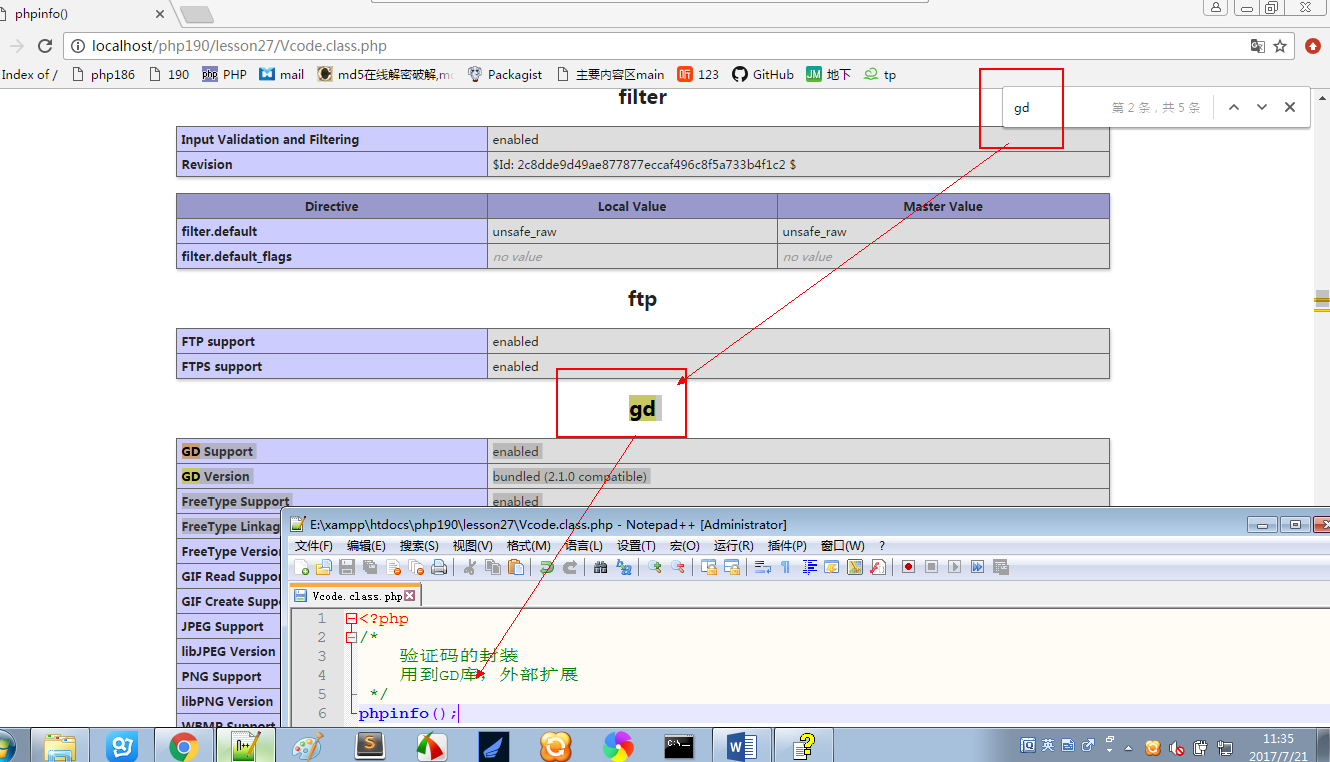
* 返回所在命名空间的命名空间名

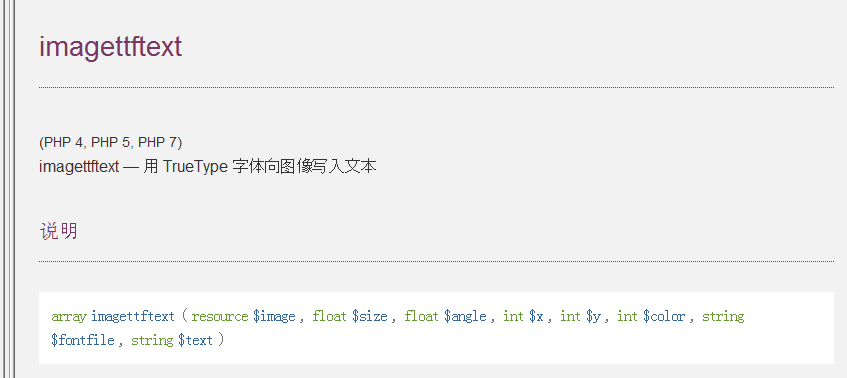


# 小结

* 基本的类和方法的定义和操作
* 封装
  + 访问控制
  + Public
  + Protected
  + Private
  + 重载
* 继承
  + Extends
  + 重写
* 关键词
  + Final
  + Static
  + Const
* 对象复制
  + 克隆
* 序列化
* 自动加载
* 魔术方法
* 抽象类
* 接口
* 命名空间

# 验证码的封装





* 具体实现详见Vcode.class.php 和test.html

# 预习

* 数据库
  + **Ppt1,2**