面向对象第1天

[学习目标 2](#_Toc521859187)

[什么是面向过程的编程思想 2](#_Toc521859188)

[1、面向过程概述 2](#_Toc521859189)

[2、举例说明：以上传产品为例 2](#_Toc521859190)

[什么是面向对象的编程思想？ 3](#_Toc521859191)

[1、面向对象概述 3](#_Toc521859192)

[2、举例说明：以上传产品为例 3](#_Toc521859193)

[类和对象的关系 3](#_Toc521859194)

[1、类的概念 3](#_Toc521859195)

[2、对象的概念 3](#_Toc521859196)

[类的定义 4](#_Toc521859197)

[1、类的定义语法格式 4](#_Toc521859198)

[2、举例说明 5](#_Toc521859199)

[类的成员属性定义 5](#_Toc521859200)

[1、成员属性介绍 5](#_Toc521859201)

[2、权限修饰符 5](#_Toc521859202)

[3、举例说明 6](#_Toc521859203)

[类的成员方法定义 6](#_Toc521859204)

[1、成员方法介绍 6](#_Toc521859205)

[2、举例说明 6](#_Toc521859206)

[创建类的实例对象 7](#_Toc521859207)

[1、实例化对象的含义 7](#_Toc521859208)

[2、语法格式 7](#_Toc521859209)

[3、举例说明 7](#_Toc521859210)

[对象属性操作 8](#_Toc521859211)

[1、如何访问对象的属性和方法？ 8](#_Toc521859212)

[2、对象属性的操作 9](#_Toc521859213)

[对象方法操作 9](#_Toc521859214)

[伪变量$this的使用 10](#_Toc521859215)

[1、伪变量$this的含义 10](#_Toc521859216)

[2、举例说明 10](#_Toc521859217)

[类常量的定义 11](#_Toc521859218)

[1、类常量介绍 11](#_Toc521859219)

[2、类常量定义和调用格式 11](#_Toc521859220)

[3、举例说明 12](#_Toc521859221)

[构造方法 12](#_Toc521859222)

[1、什么是构造方法 12](#_Toc521859223)

[2、举例说明 13](#_Toc521859224)

[析构方法 13](#_Toc521859225)

[1、什么是析构方法 13](#_Toc521859226)

[2、对象何时销毁 13](#_Toc521859227)

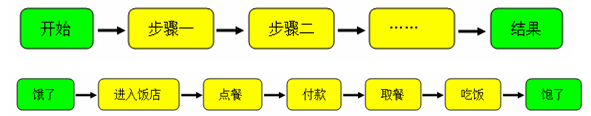
# 学习目标

* 面向过程和面向对象的编程思想
* 类和对象的关系
* 类定义、属性定义、方法定义、类常量定义
* 类的实例化
* $this关键字
* 构造方法和析构方法

# 什么是面向过程的编程思想

## 1、面向过程概述

* 面向过程是一种编程思想；
* 面向过程是一种以过程(步骤)为中心的编程思想；
* 面向过程是人类的思考问题的一种习惯；
* 面向过程的编程思想，符合我们人类解决问题的基本习惯：面对一个大问题（目标，需求），会将问题分解为若干的连续的小问题，每个小问题按顺序依次去解决，最终就自然获得大问题的结果（答案）。



## 2、举例说明：以上传产品为例

* 制作上传产品表单：产品名称、产品价格、产品数量、产品简介、产品图片等。
* 提交表单：将表单数据发送到服务器。
  + 连接MySQL服务器、选择当前数据库、设置字符集；
  + 判断表单来源是否可靠；
  + 获取表单提交数据：普通表单数据获取、上传产品图片、给图片添加水印、生成图片缩略图；
  + 构建并执行插入的SQL语句，把产品数据写入数据库；
  + 跳转到产品列表页。
* 产品列表显示页面
  + 连接MySQL服务器、选择当前数据库、设置字符集；
  + 构建并执行查询的分页的SQL语句；
  + 循环从结果集输出所有产品数据；
  + 产品数据分页。

# 什么是面向对象的编程思想？

## 1、面向对象概述

* 面向对象也是一种编程思想；
* 面向对象是一种以事物(对象)为中心的编程思想；
* 面向对象是以功能来划分问题，而不是步骤；
* 面向过程，就是着重于做什么；面向对象，就是着重于谁去做；

## 2、举例说明：以上传产品为例

* 制作上传产品表单：产品名称、产品价格、产品数量、产品简介、产品图片等。
* 提交表单：将表单数据发送到服务器。
  + 连接MySQL服务器、选择当前数据库、设置字符集(**数据库对象**)；
  + 判断表单来源是否可靠；
  + 获取表单提交数据：普通表单数据获取、上传产品图片(上传对象)、给图片添加水印(图像处理对象)、生成图片缩略图(图像处理对象)；
  + 构建并执行插入的SQL语句，把产品数据写入数据库(数据库对象)；
  + 跳转到产品列表页。
* 产品列表显示页面
  + 连接MySQL服务器、选择当前数据库、设置字符集(数据库对象)；
  + 构建并执行查询的分页的SQL语句(数据库对象)；
  + 循环从结果集输出所有产品数据；
  + 产品数据分页(分页对象)。

# 类和对象的关系

## 1、类的概念

* 类就是分类、类别、模板、概念等，类是无形的、看不见、摸不着，不实际存在的。
* 类是**具有相同属性(特征)和行为(动作)的一组对象的集合**，它为属于该类的全部对象提供了统一的抽象描述，其内部包括属性和行为两个主要部分。例如：所有在学校学习的人，具有相同的属性：学号、姓名、性别等；具有相同的行为：学习。所以，这些人构成一个类，即学生类。

## 2、对象的概念

* 客观世界中任何一个事物都可以看成一个对象，对象看得见、摸得着，实际存在的。
* 对象是构成系统的基本单位。**任何一个对象都应具有这两个要素：属性(Attribute)和行为(Behavior)**。例如：学生“张三”是一个对象，他有属性：学号、姓名、性别等；行为有：学习、吃饭等。
* 对象来自于某个类，对象不能单独存在。一个类可以产生千千万万个对象。
* 对象和类的关系是，对象是类的实例，类是对象的模板。
* 类是抽象的，几乎不占用内存，而对象是具体的，占用存储空间。

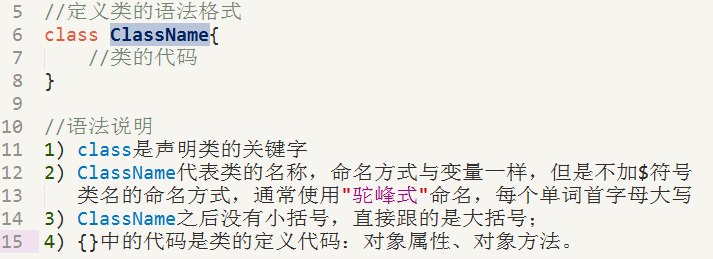
在现实中，先有对象，再有类。

在电脑中，先有类，再有对象。

# 类的定义

## 1、类的定义语法格式

类和函数一样，都是代码的封装方式。



## 2、举例说明



# 类的成员属性定义

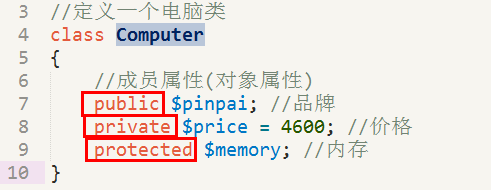
## 1、成员属性介绍

* 类的成员属性，就是指某个类具有的公共的特征、特性；
* 类中定义的变量，就是类的成员属性；
* 类的成员属性和普通变量的区别：类的成员属性必须要带权限修饰符，而普通变量不需要。

## 2、权限修饰符

* 成员属性必须要加权限修改符，否则报错。
* public(公共权限)：在任何地方都可以访问，主要包括：类外、本类中、子类中都可以访问。
* private(私有权限)：只能在本类中被访问，在类外和子类中都无权访问。
* protected(受保护的权限)：只能在本类中和子类中被访问，在类外不能被访问。

## 3、举例说明

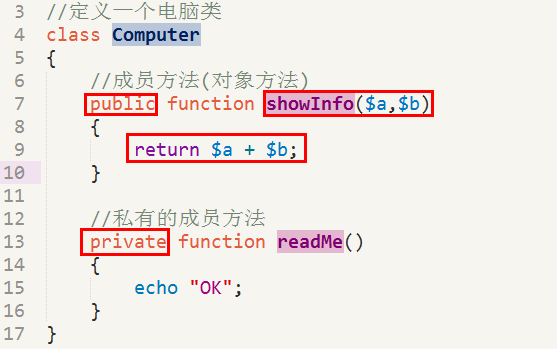


# 类的成员方法定义

## 1、成员方法介绍

* 类的方法，就是某个类的公共的的行为或动作；
* 类的成员方法，与普通函数一样，都有返回值和参数；
* 成员方法与普通函数区别：成员方法只能存在于类中，成员方法定义前可以添加访问权限修饰符；
* 提示：成员方法可以省略权限修饰符，默认访问权限是public。

## 2、举例说明



# 创建类的实例对象

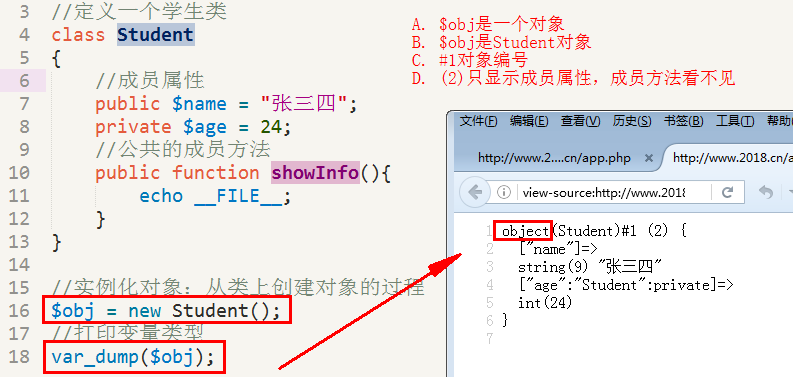
## 1、实例化对象的含义

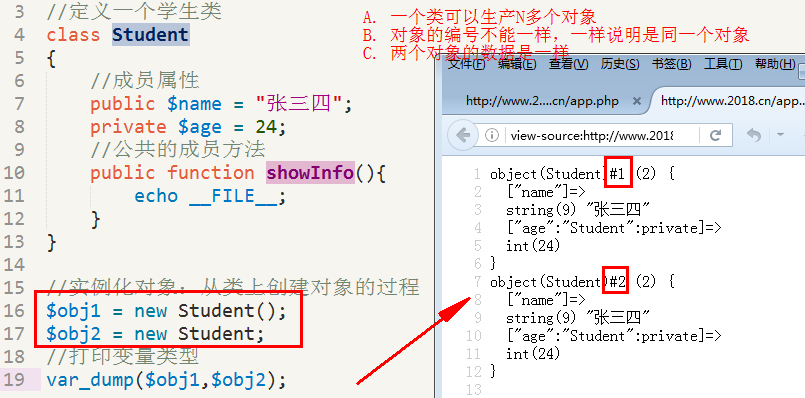
* 定义类是为了生产对象，如果不生产对象，类就没有意义。
* 一个类可以产生千千万万个对象，对象帮我们干活。
* 对象实例化：从一个类上来生产对象过程，就是类的实例化。

## 2、语法格式

* 使用new关键字来创建对象。
* JS创建对象的方法：var today = new Date()
* PHP创建对象的方法：$obj = new Student()

## 3、举例说明

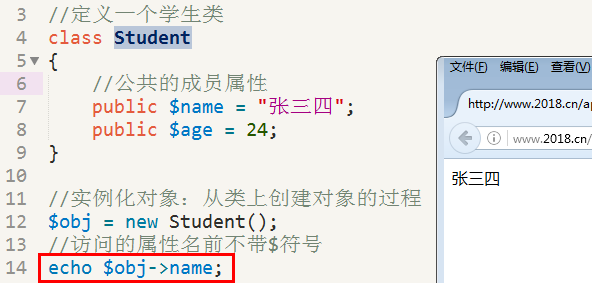




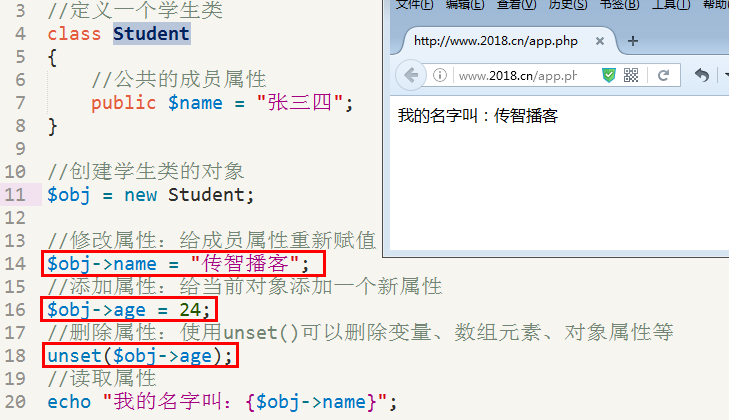
# 对象属性操作

## 1、如何访问对象的属性和方法？

* 在JS中，访问对象属性和方法，是通过"**.**"来访问的。例如：window.alert()、arrObj.length
* 在PHP中，访问对象属性和方法，是通过"**->**"来访问的。例如：$obj->name、$obj->showInfo()

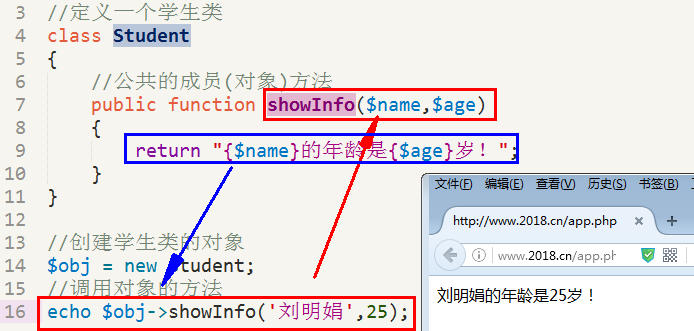


## 2、对象属性的操作



# 对象方法操作

对象方法的操作：方法定义、方法调用、传递参数、方法返回值。

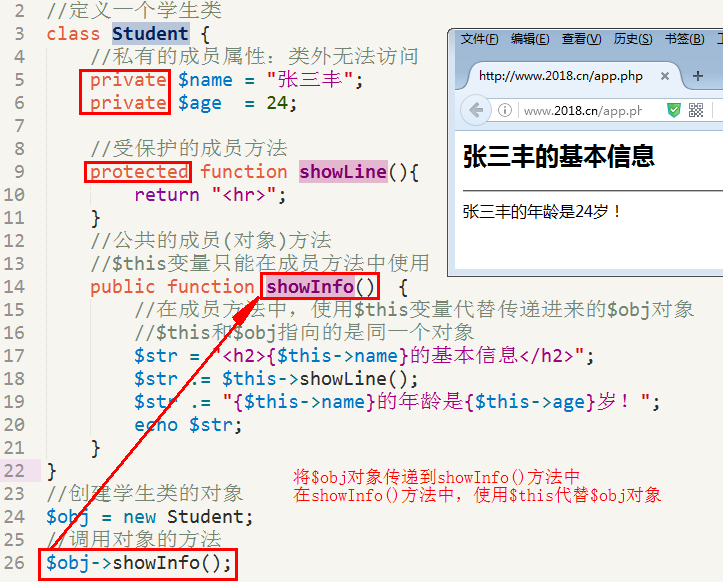


# 伪变量$this的使用

## 1、伪变量$this的含义

* 在JS中，使用this关键字来代替当前对象，例如：this.src = './a.jpg'
* 在PHP中，使用$this变量来代替当前对象，例如：$this->name = '张三'
* $this代表当前对象，是到当前对象的一个引用；
* $this更像是一个对象指针，指向当前对象；
* $this只能用在对象方法定义中，去调用对象的成员属性或成员方法。
* 只有创建对象后，$this变量才存在。类不会自动运行。

## 2、举例说明



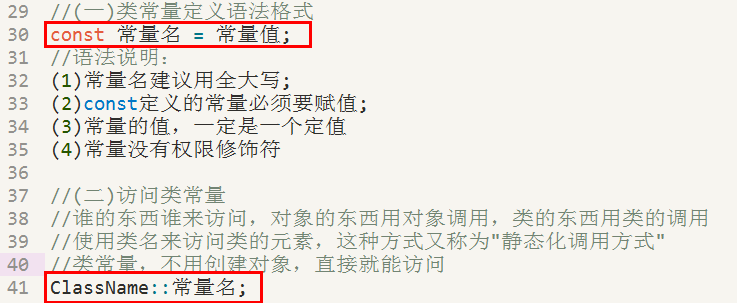
# 类常量的定义

## 1、类常量介绍

* 可以把在类中始终保持不变的值定义为常量，例如：圆周率、班级名称等。
* 常量的值必须是一个定值，不能修改，也不能删除；
* 类常量就是类的常量，是与类相关的，与对象无关。
* 类常量在内存中只有一份，不管创建多少个对象。
* 类常量可以极大节省服务器内存，可以被所有对象去共享。

## 2、类常量定义和调用格式

* 类常量没有权限，只有属性和方法才会有权限。
* 使用**const**来定义类的常量(局部常量)，只能在局部作用域下使用；define()定义常量是全局常量，在任何地方都可以使用。



## 3、举例说明

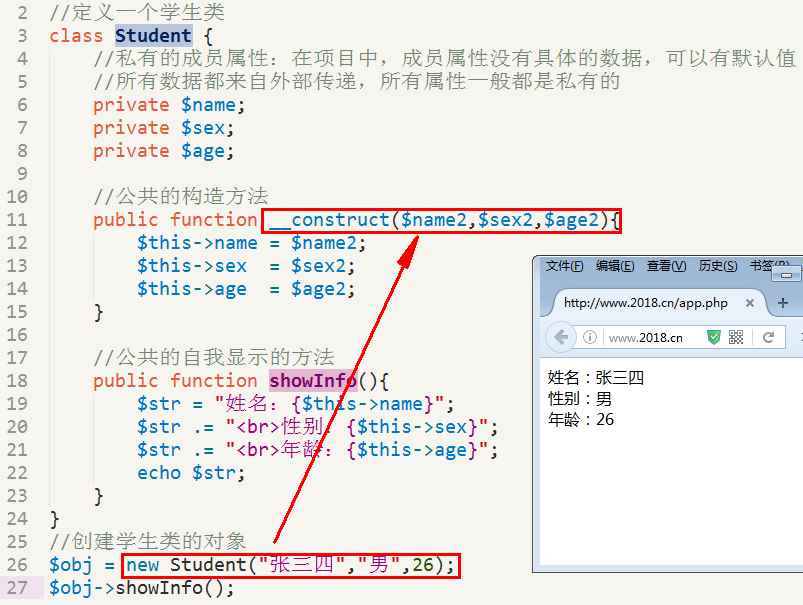


# 构造方法

## 1、什么是构造方法

* 当使用new关键字创建对象时，第1个自动调用的方法，就是构造方法；
* 构造方法的名称是固定的：**void \_\_construct ([ mixed $args [, $... ]] )**
* 构造方法可以带参数，也可以不带参数；构造方法不是必须的，是可选的；
* 构造方法的作用：对象初始化。例如：给对象属性赋值、数据库对象初始化(连接、选择数据库)
* 构造方法一定是成员方法。构造方法的权限可以自己指定。
* 构造方法一般不需要主动调用，都是自动调用的。

## 2、举例说明



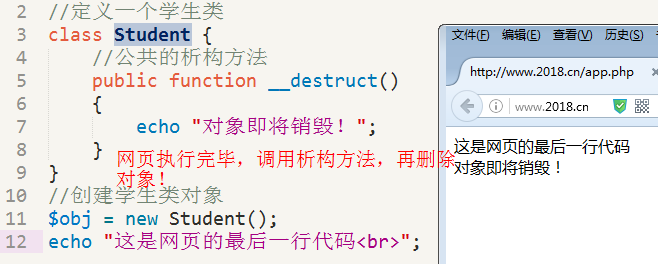
# 析构方法

## 1、什么是析构方法

* 对象销毁前自动调用的方法，就是析构方法；
* 析构方法的名称是固定的：**void \_\_destruct ( void )**
* 析构方法不带任何参数；
* 析构方法一定是成员方法。
* 析构方法的作用：垃圾回收工作，例如：断开到MySQL的连接

## 2、对象何时销毁

**网页执行完毕时，对象会自动销毁；**



**使用unset()函数手动销毁对象**

