Atitit.跨平台预定义函数 魔术方法 魔术函数 钩子函数 api兼容性草案 v2 q216 java c# php js.docx

[1.1. 预定义函数 魔术方法 魔术函数是什么 1](#_Toc29759)

[1.2. “生成器函数Supplier 1](#_Toc30463)

[1.3. JDK预定义了很多函数接口以避免用户重复定义。最典型的是Function： 2](#_Toc3177)

[1.4. 另一个预定义函数接口叫做Consumer，跟Function的唯一不同是它没有返回值。 2](#_Toc26390)

[1.5. 还有一个Predicate，用来判断某项条件是否满足。经常用来进行筛滤操作： 2](#_Toc27263)

[1.6. Comparator 是老Java中的经典接口 2](#_Toc14698)

[1.7. Optional 接口 2](#_Toc18151)

[1.7.1. 控制机构关键字消失 3](#_Toc24226)

[1.8. 魔术函数 3](#_Toc2569)

[1.9. 1。\_\_construct()  \_\_destruct() 、 3](#_Toc22143)

[1.10. 3。\_\_call()  3](#_Toc20938)

[1.11. 4。\_\_get()  \_\_set() 3](#_Toc6098)

[1.12. 6。\_\_toString()  4](#_Toc20534)

[1.13. 7。\_\_clone()  4](#_Toc12550)

[1.14. 8。\_\_sleep()  \_\_wakeup()  4](#_Toc13389)

[1.15. 10。\_\_isset()  \_\_unset()  \_\_set\_state()  检测一个对象的属性是否存在时被调用。如：isset($c->name)。 4](#_Toc11840)

[1.16. 13。\_\_autoload()  5](#_Toc5698)

[1.17. 魔术常量 5](#_Toc31001)

[2. Ref 6](#_Toc27315)

## 预定义函数 魔术方法 魔术函数是什么

就是有着特定含义的method

易用的魔术方法，这些魔术方法可以让我们简化我们的编码，更好的设计我们的系统。

魔术函数是PHP中内置的语言特性，当程序执行到某种情况时，如果定义了这些魔术函数 (php手册中称之为“Overloading”)，则PHP会调用他们，同时也会传入相应的参数，可以认为是PHP执行过程中的钩子函数。常见的魔术函 数有\_\_construct, \_\_destruct , \_\_call, \_\_get, \_\_set, \_\_isset, \_\_unset \_\_sleep, \_\_wakeup, \_\_toString, \_\_clone 以及\_\_autoload 。它们可以用来自动加载包含文件，实现延迟执行（类似于.Net中的属性访问器）、垃圾回收、对象clone 等操作，举个\_\_autoload的例子，其他魔术函数参见 Magic Method。

作者:: 绰号:老哇的爪子 （ 全名：：Attilax Akbar Al Rapanui 阿提拉克斯 阿克巴 阿尔 拉帕努伊 ） 汉字名：艾龙，  EMAIL:1466519819@qq.com

转载请注明来源： http://blog.csdn.net/attilax

## “生成器函数Supplier

”。一个生成器函数会产生一系列元素，供给一个流。Stream.generate(Supplier<T> s)Atitit.跨平台预定义函数api兼容性草案 java c# php js

## JDK预定义了很多函数接口以避免用户重复定义。最典型的是Function：

    @FunctionalInterface  
    public interface Function<T, R> {    
        R apply(T t);  
    }

这个接口代表一个函数，接受一个T类型的参数，并返回一个R类型的返回值。

## 另一个预定义函数接口叫做Consumer，跟Function的唯一不同是它没有返回值。

    @FunctionalInterface  
    public interface Consumer<T> {  
        void accept(T t);  
    }

## 还有一个Predicate，用来判断某项条件是否满足。经常用来进行筛滤操作：

      
    @FunctionalInterface  
    public interface Predicate<T> {  
        boolean test(T t);  
    }  
    **Comparator 接口**

## Comparator 是老Java中的经典接口

## ****Optional 接口****

Optional 不是函数是接口，这是个用来防止NullPointerException异常的辅助类型，这是下一届中将要用到的重要概念，现在先简单的看看这个接口能干什么：

Optional 被定义为一个简单的容器，其值可能是null或者不是null。在Java 8之前一般某个函数应该返回非空对象但是偶尔却可能返回了null，而在Java 8中，不推荐你返回null而是返回Optional。

### 控制机构关键字消失

## 魔术函数

## 1。\_\_construct()  \_\_destruct() 、

实例化对象时被调用，   
当\_\_construct和以类名为函数名的函数同时存在时，\_\_construct将被调用，另一个不被调用。

2。\_\_destruct()   
当删除一个对象或对象操作终止时被调用。

## 3。\_\_call()

对象调用某个方法，   
若方法存在，则直接调用；   
若不存在，则会去调用\_\_call函数。

## 4。\_\_get()  \_\_set()

读取一个对象的属性时，   
若属性存在，则直接返回属性值；   
若不存在，则会调用\_\_get函数。

5。\_\_set()

设置一个对象的属性时，   
若属性存在，则直接赋值；   
若不存在，则会调用\_\_set函数。

## 6。\_\_toString()

打印一个对象的时被调用。如echo $obj;或print $obj;

## 7。\_\_clone()

克隆对象时被调用。如：$t=new Test();$t1=clone $t;

## 8。\_\_sleep()  \_\_wakeup()

serialize之前被调用。若对象比较大，想删减一点东东再序列化，可考虑一下此函数。

9。\_\_wakeup()   
unserialize时被调用，做些对象的初始化工作。

## 10。\_\_isset()  \_\_unset()  \_\_set\_state()  检测一个对象的属性是否存在时被调用。如：isset($c->name)。

11。\_\_unset()   
unset一个对象的属性时被调用。如：unset($c->name)。

12。\_\_set\_state()   
调用var\_export时，被调用。用\_\_set\_state的返回值做为var\_export的返回值。

## 13。\_\_autoload()

实例化一个对象时，如果对应的类不存在，则该方法被调用。

## 魔术常量

1。\_\_LINE\_\_   
返回文件中的当前行号。

2。\_\_FILE\_\_   
返回文件的完整路径和文件名。如果用在包含文件中，则返回包含文件名。自 PHP 4.0.2 起，\_\_FILE\_\_ 总是包含一个绝对路径，而在此之前的版本有时会包含一个相对路径。

3。\_\_FUNCTION\_\_   
返回函数名称（PHP 4.3.0 新加）。自 PHP 5 起本常量返回该函数被定义时的名字（区分大小写）。在 PHP 4 中该值总是小写字母的。

4。\_\_CLASS\_\_   
返回类的名称（PHP 4.3.0 新加）。自 PHP 5 起本常量返回该类被定义时的名字（区分大小写）。在 PHP 4 中该值总是小写字母的。

5。\_\_METHOD\_\_   
返回类的方法名（PHP 5.0.0 新加）。返回该方法被定义时的名字（区分大小写）。

（1）初识魔术方法  
Php5.0发布以来为我们提供了很多面向对象的特性,尤其是为我们提供了好多今天我们就来认识下php5.0给我们提供的魔术方法。

# Ref

Java8 Lambda表达式教程 - 自由之子 - 博客频道 - CSDN.NET.htm

php5魔术函数、魔术常量.htm

PHP魔术函数 - menglv\_1978的专栏 - 博客频道 - CSDN.NET.htm

PHP魔术函数 - 网易博客.htm