

static关键字

静态属性

- 语法：public static \$property;
- 访问：类名::\$property; (如果在类的内部，还可以用self访问。即self::\$property)

静态方法

- 语法：public static function funcName(){}
- 访问：类名::funcName();
- ****注意：伪变量\$this在静态方法中不可用。根据定义，静态属性和静态方法是在类上被调用的，而不是在对象上。**

静态变量

- 语法：static \$var;
- 特点：静态变量只存在于函数作用域内，也就是说，静态变量只存活在栈中。一般的函数内的变量在函数结束后会释放，比如局部变量，但是静态变量却不会，就是说，下次在调用这个函数的时候，该变量的值会保存下来。
- 例子：

```
function test()
{
    static $count = 0;

    $count++;
    echo $count;
    echo '<br/>';
    if ($count < 100) {
        test();
    }
    echo $count;
    $count--;
    echo '<br/>';
}
```

```
test();
// 如果变量$count前没有static关键字，这个函数会陷入死循环。因为变量$count的值不会保留下来。即每次调用test()函数时，$count
```

延迟静态绑定

- 语法：
- 功能：用于在继承范围内引用静态调用的类。

- 例子:

```
<?php
// 例1:
class A{
    public static function create(){
        return new static();
    }
}
class B extends A{
}
echo get_class(A::create()); //A
echo '<br/>';
echo get_class(B::create()); //B
/*
    1、static::不再被解析为当前方法所在的类，而是实际运行时计算的。
    2、new static()生成的对象由调用者决定。
*/
/*-----*/
// 例2:
class A{
    public static function who(){
        echo __CLASS__;
    }
    public static function test(){
        echo static::who();
    }
}
class B extends A{
    public static function who(){
        echo __CLASS__;
    }
}
B::test(); //B
```

note路径:

- D:\phpStudy\PHPTutorial\WWW\index\aaa\static\static.md