PHP小白极速入门 —— 变量作用域

主讲人与课程设计：耕耕

变量法则

1. 超全局变量，任意地方都可以访问
2. 常数，一般是全局性，函数内外都可以访问
3. 全局变量（一般声明在函数外面），函数内不可以访问
4. 函数内可以通过global关键词就可以访问全局变量（其实是同一个变量）
5. 在函数中创建的静态变量，函数外无法访问，但是这个变量的值一直保留
6. 在函数中创建的局部变量，函数外是无法访问的，并且在函数执行结束后失效

内置超全局变量

不管程序的任何地方都可以访问到，也不管是函数内还是函数外，由PHP预先定义好的

$**GLOBALS**
$_SERVER
$_REQUEST
$_POST
$_GET
$_FILES
$_ENV
$_COOKIE
$_SESSION

内置超全局变量

```php
<?php
$a = 10;
$b = 20;

function addition(){
    $GLOBALS['c'] = $GLOBALS['a'] + $GLOBALS['b'];
}

addition();
echo $c;
```

全局变量

全局变量一般声明在函数外的变量，代码间可以访问，**函数内不可以访问**

```php
<?php
$a = 10;
$b = 20;

function addition(){
    echo $a; // 不可以访问，出错
}

addition();
```

全局变量

如果在函数内要访问呢？加**global**关键词

```php
<?php
$a = 10;
$b = 20;

function addition(){
    global $a;
    echo $a; // 加了global关键词，可以访问了
}

addition();
```

静态变量

静态函数只是在函数体内，在函数外无法访问。**但是执行后，其值保留，等待下次函数执行，关键词 static**

```php
<?php

function addition(){
    static $c = 100;
    $c++;
    echo "<h1>$c</h1>";
}

addition();
addition();
```

变量销毁

Php一般有自动垃圾回收机制，但是也可以通过手动销毁，用函数unset()

```php
<?php

$a = 100;
$b = &$a;

unset ($a);
echo $b;
```