Switch 语句

原理:

使用 switch 表达式和 case 后面的表示进行比较,相等则执行 case 里面的代码,如果 switch 表达式和 case 后面的表达式都不相等,执行 default 代码块;

如果 case 里面没有 break 语句,会一直执行相等 case 后面的所有的代码块(包括下一个 case 中的代码块);

If 与 switch 语句的区别

If 条件判断(条件比较适合范围的判断)

Switch 语句 :一般情况下,进行等值判断

循环结构

While

原理: 当满足条件(条件成立),循环执行相应的循环体操作,直到条件不成立,则循环终止;

Do-while

原理: 首先执行循环体,再进行判断,满足条件,继续执行循环体,不满足退出循环;

While 和 do-while 的区别

While 循环,首先判断,满足条件,执行循环体;

Do while: 先执行一次,满足条件继续执行循环体,不满足结束循环.最少执行一次循环体.

对于判断条件基于循环体的执行时,必须使用 do while;

Break 和 continue

Break: 跳出当前循环,当前循环中 break 语句后面的代码不执行,

使用场合:switch 语句和循环语句;

Continue: 跳出当前循环,执行下一次循环

使用场合: 只能在循环语句中使用;

流程控制语句的替代语法

将 if,while,for 左大括号替换为: (冒号)

右大括号替换为: endif; endwhile; endfor;

文件载入

多个 php 文件可能使用相同功能的代码,可以将相同功能的代码在公共的文件中,需要使用的地方引入即可.

语法: include “php 文件路径”; // 引用当前脚本同一目录下的 php 文件.

相对路径:

./ 表示当前 php 文件(脚本文件)的目录

../ 表示当前 php 文件(脚本文件)的上一级目录;

绝对路径: 磁盘上的文件的真实路径

- 建议使用相对路径;

Include 与 require 区别

Include / include_once / require / require_once 四个结构,不是函数;

Include 和 require 区别

- Include 引入文件失败,返回警告,后续代码继续执行;
- Require 引入文件失败,返回 error,后续代码不执行;

_once 的作用

对于已经载入的文件,不能进行重复载入文件,避免重复载入文件,使用_once ;

脚本执行控制

停止脚本执行

语法: die(提示信息);

终止后续代码的执行,可以输出提示信息;

同功能函数:

语法:ext(提示);

延迟脚本执行

语法: sleep(秒数);

延迟代码执行指定的秒数,用于查看程序执行过程中产生的临时文件;

函数

函数的五要素:

1, function 关键字

2, 函数名

3, 参数列表

4, 函数体

5, 返回值, return 语句,将数据返回到调用的地方