

Day5 学习报告

1. 概述：今天，我们主要学习的内容有：

- (1) 循环的中断
- (2) 控制脚本执行进度
- (3) 文件加载
- (4) 错误处理

以下将对今天所学内容做一一回顾：

2. 知识回顾

(1) 循环的中断：

continue：终止当次循环，进入下一次循环

break：终止当层循环，如有外层循环，则进入下层循环

理解continue 2;和break 2;的作用

(2) 循环三要素

循环变量的初始化、条件判断、变量改变

(3) 流程控制的替代语法：

如：if():else if():endif;

(4) goto语句，可以随便跳转到某标识符位置继续执行的，但是不能跳入循环、函数中，不能跳出函数、文件外

(5) 控制脚本执行速度

语法1：die(字符串)、exit(字符串)

语法2：sleep(\$n);

(6) 文件加载

1) 基本语法主要有include、require、include_once、require_once

注意它们的区别

文件加载路径问题

相对路径和绝对路径

其中绝对路径不建议用c://...，而使用以下两种形式

__DIR__.” /文件名”

\$_SERVER[“DOCUMENT_ROOT”].” /目录2/文件名”

2) 文件加载的执行过程

简单的说就是，哪个位置载入，相当于动态添加了被载入文件的代码到插入那个位置，执行完了之后，再回来继续执行下面的程序。

3) 四个载入语法的区别

4) 被载入文件的返回值以及return的作用

一个载入语句，默认返回一个值，成功为1失败为false

在被载入文件中，可以使用return，这个return就和函数中的return作用类似，如果写了return 数据，那么载入文件就能拿到那个具体的数据

(7) 错误处理

1) 错误分类

包括：语法错误、运行时错误、逻辑(如业务的)错误等

2) 错误分级

包括大概十几个错误，是一个个的二进制常量

如：系统错误：E_ERROR(致命错误)、E_WARNING(警告性错误)、E_NOTICE(提示性错误)

用户自定义错误：E_USER_ERROR(致命错误)、E_USER_WARNING(警告性错误)、E_USER_NOTICE(提示性错误)

其他错误：E_STRICT(严谨的语法检查错误)、E_ALL:代表所有错误

3) 错误的触发

系统错误的话由系统触发，如fatal error为致命错误，程序不继续执行，提示性错误和警告性错误程序会继续执行

对于自定义错误，触发方式为：

```
if(条件满足){  
    trigger(“错误说明”,E_USER_ERROR); //触发用户自定义致命错误  
}
```

4) 错误报告显示问题

是否显示错误报告设置

在php.ini中修改display_errors值为0n或php文件中代码

```
ini_set(“display_errors”,1);
```

显示哪些报告错误设置

在php.ini中修改error_reporting为E_ERROR，或php文件中代码

```
ini_set(“error_reporting”,E_ERROR);//则只会报告致命错误
```

5) 错误日志记录问题

是否记录错误日志

php.ini中设置log_errors值为0n或php文件中代码：

```
ini_set(“log_errors”,“1”);
```

记录到哪里

php文件中代码`ini_set(“error_log”, “my_error.txt”);`此时系统会在每个文件夹下建立一个`my_error.txt`文件，并且把该目录下所有php文件的错误日志都记录到该文件

php文件中代码`ini_set(“error_log”, “syslog”);`则系统会将所有的错误日志记录到同一个`syslog`文件

6) 自定义错误处理器

一旦自定义，就不会走系统的错误处理逻辑，而走自己的自定义错误处理函数

步骤1:设置用于处理的函数名

`set_error_handler(“my_error_handler”);`

步骤2:去自定义函数，主要执行错误日志输出和本地化错误日志的操作
具体查看另一篇错误处理文档