

## 78. 日志处理

学习要点：

### 1. 日志处理

本节课我们来学习一下日志处理的功能，包括配置日志和写入日志等。

#### 一. 日志处理

1. 日志处理的操作由 `Log` 类完成，它记录着所有程序中运行的错误记录；
2. 在 `config` 目录下的 `log.php` 配置文件，用于设置日志信息；
3. 我们在 `runtime` 目录下后一个 `log` 文件夹，里面按照日期排好了每月的日志；
4. 使用 `record()` 方法，记录一条测试日志；

```
Log::record('测试日志!');
```

5. 我们在 `log` 日志文件夹里找到最新生成的日志，可以看到生成的日志信息；
6. 系统提供了不同日志级别，默认 `info` 级别，从低到高排列如下：
7. `debug/info/notice/warning/error/critical/alert/emergency/sql`；
8. 一般记录就是 `info` 信息，我们也可以指定我们的信息级别；

```
Log::record('测试日志!', 'error');
```

9. `record()` 方法不是实时记录，需要等待程序完毕后决定是否写入日志；
10. 如果在写入方法后添加 `close()` 关闭写入，那么 `record()` 方法则不写入；

```
Log::close();
```

11. 系统还提供了一个 `write()` 方法，进行时写入，不理睬其它限制；

```
Log::write('测试日志信息', 'error');
```

12. 系统发生异常后，会自动写入 `error` 日志，如果你想手动也可以；

```
try {
    echo 0/0;
} catch (ErrorException $e)
{
    echo '发生错误：' . $e->getMessage();
    Log::record('被除数不得为零', 'error');
}
```

13. 对于各种日志级别，系统提供了一些快捷方式和助手函数，比如：

```
Log::error('错误日志!');           //Log::record('错误日志!', 'error')
Log::info('信息日志!');            //Log::record('信息日志!', 'info')
trace('错误日志!', 'error');
trace('信息日志!', 'info');
```

14. 系统默认并不记录 HTTP 异常，因为这种异常容易遭受攻击不断写入日志；

15. 除了系统提供的几种类型，也可以自己定义日志类型；

```
Log::diy('自定义日志');
```

16. 在配置文件 `log.php` 中，可以设置限定日志文件的级别，不属于的无法写入；

```
'level' => ['error', 'info'],
```

17. 在配置文件 `log.php` 中，添加转换为 json 格式；

```
'json' => true
```

18. 使用 `::getLog()` 方法，可以获取写入到内存中的日志；

```
$logs = Log::getLog();
```

```
dump($logs);
```

19. 使用 `::clear()` 方法，可以清理掉内存中的日志；

```
Log::clear();
```

20. 在配置文件 `log.php` 中，可以设置以单独文件存储的方式；

```
'single' => true,
```