河北师范大学 软件学院
Software College of Hebei Normal University

Linux平台PHP服务端开发——

第十七讲 PHP HOW TOs

PHP如何从终端获取输入

```
<?php
$input = file_get_contents('php://stdin');
echo "\nYour input: ",$input,"\n";
```

- 运行后，此操作会等待用户输入，输入可以有换行，Ctrl+d结束输入。

PHP如何获取客户端信息

```
<?php
echo $_SERVER['HTTP_USER_AGENT'],'<br>'; //浏览器或其他客户端信息。
echo $_SERVER['REQUEST_METHOD'],'<br>';  //请求方法：GET，POST···
echo $_SERVER['REMOTE_ADDR'],'<br>'; //客户端IP地址，对外IP，非内网IP。
echo $_SERVER['QUERY_STRING'],'<br>'; //查询字符串,例：?a=1&b=2
echo $_SERVER['PHP_INFO'],'<br>'; //真实脚本之后的信息
echo $_SERVER['REQUEST_URI'],'<br>'; //请求页面，例：/index.html
```

- 注意：在CLI模式（命令行）下，这些都是无效的。

PHP在CLI模式如何获取参数

```
<?php
echo  $_SERVER['argc'],"\n"; //参数个数
//循环输出参数
foreach ($_SERVER['argv'] as $a) {
    echo $a,"\n";
}
```

- 直接通过$argc和$argv也可以获取参数个数和参数信息。如果文件名为args.php，在终端运行程序:

  ```
  php args.php a b c 12 23
  ```

- 则会输出参数个数，并输出args.php a b c 12 23，$argv[0]是程序文件的名称。

PHP时间处理

```
<?php
date_default_timezone_set('Asia/Shanghai'); //设置默认时区
echo strftime("%Y.%m.$d %H:%M:%S",time()); //年.月.日 时:分:秒
//把格式化字符串转换成时间数字
echo strtotime("next day"),"\n";
echo strtotime("next week"),"\n";
echo strtotime("2018-05-02 16:05:19"),"\n";
```

PHP对用户名、Email进行正则匹配

```
<?php
$user_regex = '/^[a-zA-Z][a-zA-Z0-9]{5,17}$/';
$email_regex = '/[a-z0-9A-Z\-\.\_]+\@[a-zA-Z0-9]+\.[a-zA-Z]/';
$user_test = ['abc','1233','a1293c','abqddks','a9123-23'];
$email_test = ['abcde@123.com','123801@ax.com',
            'askdh@','@ad20','abc_de@134.com'
];
regex_test($user_test, $user_regex);
regex_test($email_test,$email_regex);
```

```
function regex_test($data, $pat){
    foreach ($data as $u) {
        if (preg_match($pat,$u)){
            echo "[ok]  $u\n";
            continue;
        }
        echo "[--] $u\n";
    }
}
```

PHP对数组元素map

```
<?php
function cube($a) {
    return $a*$a*$a;
}

$ar = [1,2,3,4];
var_dump(array_map('cube', $ar));
//匿名函数形式
var_dump(array_map(function($a){return $a*($a+1);}, $ar));
```

- 如果需要对数组的每个元素进行计算后得到新的值构成新的数组，可以使用array_map，不必使用foreach循环每个元素。

检查文件、类、方法是否存在

```
<?php
$file = 'file_test.php';
if (file_exists($file)) {
    require ($file);
} else {
    echo "file not found\n";
}
if (class_exists('best') && method_exists('best','what')) {
    (new best)->what();
} else {
    echo "class or method undefine\n";
}
```

- 在相同目录下，还存在file_test.php文件，此文件中定义了类best。

获取文件的MIME类型

```
<?php
$files = [
    '../tmp/Zombie.mp3',
    '../tmp/30144-106.jpg',
    '../tmp/the_fox.mp4'
];

foreach ($files as $f) {
    echo mime_content_type($f),"\n";
}
```

- 输出测试文件的MIME类型，通过mime_content_type函数获取MIME类型。

执行正则表达式匹配替换

```
<?php
$org_text = file_get_contents('pregdata’);
$text = preg_replace(‘/abc.*/i’,’123’, $org_text);
file_put_contents('pregdata1',$text);
```

- 此操作会把所有abc开头的字符串替换为123。

计算文件的sha1值

```
<?php
echo sha1_file('zsort.php'),"\n";
echo sha1_file('zerosort.php'),"\n";
```

- 两个文件内容相同，名称不同，但是sha1_file计算的文件散列值相同，通过sha1_file可以比较两个文件是否相同，这可以保证在上传文件时，由于名称不同而重复上传。

删除文件和目录

```
<?php
if (is_dir('pt')) {
    rmdir('pt');
}

if (file_exists('zerosort.php')) {
    unlink('zerosort.php');
}
```

- 删除目录的函数要求目录必须是空目录。unlink用于删除文件，但实际上是使文件的硬链接数减一，硬链接数减为0会真正的删除文件。