



类

❖ 定义类

```
class 类名 {  
    定义成员变量  
    定义成员方法  
}
```

❖ 定义成员变量

- var和public: 公有变量
- private: 私有变量

❖ 定义成员方法

```
function 成员方法名 ([参数表]) {  
    函数体  
}
```

```
class userInfo {  
    var $name; //定义成员变量  
    var $gender;  
    //定义成员方法  
    function getInfo() {  
        echo $this->name;  
        echo $this->gender;  
        // $this是代表类自身的伪变量  
        // -> 是成员操作符  
    }  
}
```



创建对象、访问成员

- ❖ 使用关键字new创建对象

```
$对象变量 = new 类名([实参]);
```

- ❖ 销毁对象： \$对象变量 = null;

- ❖ 使用->访问类中的成员变量或方法

```
$user->name = "王原"; //设置成员变量
```

```
$user->gender = "女";
```

```
echo $user->name;
```

```
$user->getInfo(); //调用成员方法
```

```
$user = null; //销毁对象
```

- ❖ 注意： PHP中类和对象的成员操作符是由减号后跟大于号组成->（JavaScript中是点.）。



构造函数、析构函数

❖ 构造函数

- 在创建对象时自动执行
- 名称必须是`__construct`

❖ 析构函数

- 在销毁对象时自动执行
- 名称必须是`__destruct`
- 不能带有参数



操作类

❖ ::

- 类名：访问类中的变量、常量和方法
- parent：访问父类的变量、常量和方法
- self：访问当前类的静态成员和常量

❖ instanceof

- 检测对象变量是否属于类
- 变量 instanceof 类名



类的继承

- ❖ 继承

- class 子类名 extends 父类名
- 不支持多重继承

- ❖ 支持多态



```
class Person{
    4 references
    public $name;
    2 references
    public $gender;
    2 references
    public $age;
    1 reference | 0 overrides
    function __construct($name, $gender, $age){
        $this->name = $name;      //必须使用this->
        $this->gender = $gender;
        $this->age = $age;
        $this->sayHello();
    }
    0 references | 0 overrides
    function __destruct(){
        echo $this->sayBye();
    }
    1 reference | 0 overrides
    function sayHello(){
        echo "<mark>{$this->name}说Hello!</mark><br>";
    }
    1 reference | 0 overrides
    function sayBye(){
        echo "<br><mark>{$this->name}说再见！</mark>";
    }
    4 references | 0 overrides
    function saySomething(){
        echo "我叫{$this->name}, 性别{$this->gender}, {$this->age}岁。<br>";
    }
}
```

```
class Student extends Person{
    4 references
    public $school;
    2 references | 0 overrides
    function studyAt($school){
        $this->school = $school;
    }
    1 reference | 2 overrides
    function study(){
        parent::saySomething();
        echo "我在{$this->school}上学! <br>";
    }
}
1 reference | 0 implementations
class collegeStudent extends Student{
    1 reference | 0 overrides | prototype
    function study(){
        parent::saySomething();
        echo "我在{$this->school}读大学! <br>";
    }
}
1 reference | 0 implementations
class eleStudent extends Student{
    1 reference | 0 overrides | prototype
    function study(){
        parent::saySomething();
        echo "我在{$this->school}念书! <br>";
    }
}
```

```
function intro($obj){
    if($obj instanceof Student){
        $obj->study();
    }elseif($obj instanceof Person){
        $obj->saySomething();
    }
}
$o1 = new collegeStudent('原原', '男', '21');
$o1->studyAt('吉林大学');
intro($o1);

$o2 = new eleStudent('姐姐', '女', '12');
$o2->studyAt('附属小学');
intro($o2);

$o3 = new Student('欣欣', '男', '15');
intro($o3);

$o4 = new Person('我', '女', '33');
intro($o4);
```



← → ⌛ ⌂

① localhost:3000/5-class-1.php

原原说Hello!

我叫原原, 性别男, 21岁。

我在吉林大学读大学!

妞妞说Hello!

我叫妞妞, 性别女, 12岁。

我在附属小学念书!

欣欣说Hello!

我叫欣欣, 性别男, 15岁。

我在上学!

我说Hello!

我叫我, 性别女, 33岁。

我说再见!

欣欣说再见!

妞妞说再见!

原原说再见!



遍历对象的属性

- ❖ 使用foreach语句可以遍历对象的公有属性

```
foreach($userInfo as $attr=>$value) {  
    echo "对象userInfo的属性", $attr, "的值是", $value, "<br/>";  
}
```

2、Web交互编程





- ❖ Web应用程序的基本功能就是与用户进行交互，获取和处理用户提交的数据，并将处理结果返给用户。
 - 客户端通过表单提交数据，服务器端通过超全局数组变量 `$_GET[]` 和 `$_POST[]` 来获取。
 - 服务器端可以通过 `header()` 函数向客户端输出 HTTP 协议的首部信息，达到网页重定向、发送非 HTML 文档、强制浏览器下载文件等目的。
 - 利用 `cookies` 和 `session` 技术在指定的期间内管理客户端和服务器端共享的信息。



localhost/learnTest/php/just4...

用户名: 密码: 登录

localhost/learnTest/php/just4...

用户名: 原原 密码: Morning 登录

localhost/learnTest/php/just4...

用户名: 密码: 登录

您输入的用户名是: 原原
您输入的密码是: Morning



```
<html><body>

    <form method="post" action="">
        用户名: <input type="text" name="userName" size="12">
        密码: <input type="text" name="psw" size="10">
        <input type="submit" name="login" value="登录">
    </form>

    <?php
        if (isset($_POST['login'])) {
            $userName = $_POST['userName'];
            $psw = $_POST['psw'];
            echo "您输入的用户名是: ". $userName;
            echo "<br>您输入的密码是: ". $psw;
        }
    ?>

</body></html>
```



```
<html><body>
<form method="get" action=""
用户名: <input type="text" name="userName" size="12">
密码: <input type="text" name="psw" size="10">
<input type="submit" name="login" value="登录">
</form>
<?php
if (isset($_GET['login'])){
    $userName = $_GET['userName'];
    $psw = $_GET['psw'];
    echo "您输入的用户名是: ". $userName;
    echo "<br>您输入的密码是: ". $psw;
}
?>
</body></html>
```

localhost/learnTest/

← → ⓘ loc ?»

百度 苏宁易购

» 阅读清单

用户名: 密码: 登录

您输入的用户名是: 原原

您输入的密码是: Morning



5-c2s-2.html > ...

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>a标签跳转</title>
6  </head>
7  <body>
8      <ul>
9          <li><a href="5-c2s-2.php?id=1&name=n1">GET方式到PHP, id=1, name=n1。</a></li>
10         <li><a href="5-c2s-2.php?id=2&name=n2">GET方式到PHP, id=1, name=n2。</a></li>
11         <li><a href="5-c2s-2.php?id=3&name=n3">GET方式到PHP, id=1, name=n3。</a></li>
12         <li><a href="5-c2s-2.php">没有参数的跳转。</a></li>
13     </ul>
14  </body>
15  </html>
```



5-c2s-2.php

```
1  <?php
2      if(isset($_GET['id']) && isset($_GET['name'])){
3          echo <<<_END
4          <h1>提交的id为{$_GET['id']}，name为{$_GET['name']}。</h1>
5      _END; }else{
6          echo <<<_END
7          <table border="solid 1px">
8              <tr align="center"><td>\$_SERVER环境变量</td><td>值</td></tr>
9              <tr><td>\$_SERVER['REMOTE_ADDR']</td><td>{$_SERVER['REMOTE_ADDR']}
```



← → C ⌂

① localhost:3000/5-c2s-2.html

- GET方式到PHP, id=1, name=n1。
- GET方式到PHP, id=1, name=n2。
- GET方式到PHP, id=1, name=n3。
- 没有参数的跳转。

← → C ⌂

① localhost:3000/5-c2s-2.php?id=1&name=n1

提交的id为1, name为n1。

← → C ⌂

① localhost:3000/5-c2s-2.php?id=3&name=n3

提交的id为3, name为n3。



← → C ⌂ ⓘ localhost:3000/5-c2s-2.html

- GET方式到PHP, id=1, name=n1。
- GET方式到PHP, id=1, name=n2。
- GET方式到PHP, id=1, name=n3。
- 没有参数的跳转。

← → C ⌂ ⓘ localhost:3000/5-c2s-2.php

\$_SERVER环境变量	值
\$_SERVER['REMOTE_ADDR']	::1
\$_SERVER['DOCUMENT_ROOT']	D:\phpstudy_pro\WWW\2023
\$_SERVER['SCRIPT_NAME']	/5-c2s-2.php
\$_SERVER['SCRIPT_FILENAME']	D:\phpstudy_pro\WWW\2023\5-c2s-2.php
\$_SERVER['REQUEST_METHOD']	GET
\$_SERVER['REQUEST_TIME']	1685336019
\$_SERVER['REQUEST_TIME'](1685336019)	2023年05月29日



浏览器发送请求和服务器获取请求的方法

发送请求的方法		发送请求的方式	服务器获取请求的方法
输入网址（URL）			
提交 表单	method='get'	GET方式	<code>\$_GET[]</code>
	method='post'	POST方式	<code>\$_POST[]</code>



常用的\$_SERVER[]环境变量

变量名	含义
QUERY_STRING	查询字符串信息
SCRIPT_NAME	当前文件相对于网站目录的路径和文件名
SCRIPT_FILENAME	当前文件在硬盘中的路径和文件名
PHP_SELF	当前正在执行脚本的文件名
DOCUMENT_ROOT	当前网站根目录
SERVER_SOFTWARE	Web服务器软件的名称
SERVER_PORT	服务器端的端口号
SERVER_NAME	服务器主机的名称
SERVER_SIGNATURE	包含服务器版本和虚拟主机名的字符串
REQUEST_METHOD	HTTP请求的方式，如：GET和POST
REQUEST_TIME	HTTP请求开始时的时间戳
REMOTE_ADDR	客户端IP地址



发送数据给浏览器

- ❖ 使用echo方法输出信息
- ❖ 使用header()函数
- ❖ 操作缓冲区



header函数

❖ header函数能向浏览器发送HTTP响应头信息，浏览器收到响应头信息会做出相应的反应。

- header(message [, replace [, http_response_code]]);
- message: 响应头信息, "header_name: header_value"

❖ header函数主要有以下几种用途

(1) 使用location响应头重新定向网页

- header函数要求在重定向之前，服务器不能向客户端发送任何数据（确保在HTTP首部）。

```
<?php
    header("location:http://w3school.com.cn");
    header("location:http://www.jlu.edu.cn");
    exit; /*重定向后立即执行exit，可以避免后续脚本在header前输出，带来不必要的麻烦。*/
?>
```



(2) 使用Refresh响应头使文件转向延迟

- 可以使页面延迟几秒后，重定向到url指定的页面。

```
header( header: "Refresh:3; url=http://w3school.com.cn");
```

(3) 声明文件内容

```
header( header: "content-type:text/html; charset=utf-8");
```



(4) 禁用浏览器缓存

- 不缓存请求和相应的任何内容，使用户每次都能从服务器上获取最新的网页，而不是浏览器缓存中的网页。

```
<?php
    //内容过期时间
    header("Expires: Mon, 5 Jun 2023 00:00:00 GMT");
    //标记内容最后修改时间
    header('Last-Modified:' . gmdate('D, d M Y H:i:s') . 'GMT');
    //强制不缓存
    header('Cache-Control: no-store, no-cache, must-revalidate');
    header("Pragma: no-cache"); //禁止本页被缓存
    header('Pragma: no-cache'); //兼容http1.0和https
    var_dump(headers_list());
?>
```

```
array(5) { [0]=> string(23) "X-Powered-By: PHP/7.3.4" [1]=> string(37) "Expires: Mon, 5 Jun 2023 00:00:00 GMT" [2]=> string(42) "Last-Modified:Wed, 31 May 2023 02:49:59GMT" [3]=> string(50) "Cache-Control: no-store, no-cache, must-revalidate" [4]=> string(16) "Pragma: no-cache" }
```



(5) 强制下载文件

```
$filename = "../img/img-1.jpg";
header('Content-Type: image/jpg');
header('Content-Disposition: attachment; filename = "'.$filename.'"');
header('Content-Length:'.filesize($filename));
readfile($filename);
```





(6) 发送HTTP状态

```
//200 正常状态
header('HTTP/1.1 200 OK');
// 301 永久重定向，后面加重定向地址 Location:$url
header('HTTP/1.1 301 Moved Permanently');
// 设置页面304 没有修改
header('HTTP/1.1 304 Not Modified');
//401 未经授权：访问由于凭据无效被拒绝，需要进行登录
header('HTTP/1.1 401 Unauthorized');
// 403 禁止访问
header('HTTP/1.1 403 Forbidden');
// 404 错误
header('HTTP/1.1 404 Not Found');
// 500 错误
header('HTTP/1.1 500 Internal Server Error');
```



PHP操作缓冲区

❖ 缓冲区是在服务器内存中的一块区域

- 在没有开启缓冲区时，执行文件输出的内容直接输出到浏览器。开启缓冲区后，输出内容会先存入缓冲区，直到脚本执行完毕（或遇到一些缓冲区操作指令），再将缓冲区的内容发送给浏览器。
- PHP输出→**PHP缓冲区**→Web Server缓冲区→浏览器缓冲区→浏览器显示给用户

❖ PHP提供了很多操作缓冲区的函数

- 缓冲区内容发送至下一级？
 - 缓冲区满
 - 遇到ob_flush()、ob_end_flush()或ob_get_flush()函数
 - 程序执行完
 - 遇到exit或die函数提前终止程序



PHP缓冲区操作函数

函数	功能
ob_start	打开PHP输出缓冲区
ob_get_contents	返回PHP缓冲区内容
ob_get_clean	返回PHP缓冲区内容，关闭缓冲区。
ob_get_flush	返回PHP缓冲区内容，关闭缓冲区，将缓冲区内容输出到客户端。
ob_get_length	返回PHP缓冲区内容的长度。
ob_clean	删除PHP缓冲区内容，不关闭缓冲区。
ob_flush	输出PHP缓冲区内容到浏览器，不关闭缓冲区。
ob_end_clean	删除PHP缓冲区内容，关闭缓冲区。
ob_end_flush	输出PHP缓冲区内容，不关闭缓冲区。
flush	输出Apache缓冲，将ob_flush输出的内容，以及不在PHP缓冲区的内容，全部输出至浏览器。



```
<?php

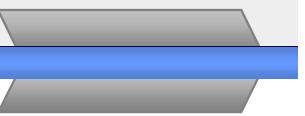
function callback($buffer){
    static $c = 0;
    return ++$c . '--' . $buffer . '<br>';
}

ob_start( callback, chunk_size: 3);
echo "1";
sleep( seconds: 1);
echo '2';
sleep( seconds: 1);
echo '33333';
sleep( seconds: 1);
echo '444444';
sleep( seconds: 1);
echo '5';
ob_end_flush();

?>
```

← → ⌂ ⌂ ⓘ localhost/php/output_buffer.php

1--112
2--33333
3--444444
4--5



```
<?php  
function callback($buffer){  
    static $c = 0;  
    return ++$c . '--' . $buffer . '<br>';  
}  
ob_start( callback, chunk_size: 3);  
echo "11";  
sleep( seconds: 1);  
echo '2';  
sleep( seconds: 1);  
echo '33333';  
sleep( seconds: 1);  
echo '44444';  
sleep( seconds: 1);  
echo '5';  
ob_end_clean();  
?>
```

← → ⌂ ⌄ ⓘ localhost/php/output_buffer.php

1--112
2--33333
3--444444



```
1 <?php
2     ob_start();
3     echo "<b>Hello World</b>"; //输出到缓冲区，但不会立即输出到页面
4     $len = ob_get_length(); // $len 保存了当前缓冲区中内容的长度
5     $out = ob_get_contents(); // $out 保存了当前缓冲区中的内容
6     ob_end_clean(); // 清空并结束缓冲区
7     echo $out, '<br>'; // 输出<b>Hello World</b>
8     var_dump($out, $len); // 将变量$out中的字符转换为小写
9 ?>
```

The screenshot shows a web browser window with the URL `localhost/learnTest/php/just4...` in the address bar. The page content displays the output of the PHP script: **Hello World** followed by the string representation of the variable `$out`, which is `string(18) "Hello World"`. Below this, the integer value of `$len` is shown as `int(18)`.



Session技术简介

- ❖ HTTP协议是无状态（Stateless）的
 - 服务器完成客户端的一次请求后，不会继续保持与该客户端浏览器的连接。
 - 因此，当用户在网站的多个页面间切换时（请求了多个页面），页面之间无法传递用户的相关信息。即，客户端每一次请求之间都是独立的。
- ❖ Web编程中，Session代表了服务器与客户端之间的会话，即一段时间内浏览器与特定服务器间的交互过程。
 - 把客户端信息存储在Session变量中，用户在网站页面之间跳转时，存储在Session变量中的信息不会丢失，而是在整个会话中一直存在下去。
 - 服务器为每个会话创建并维护一个ID，通过ID识别不同的会话，会话结束后ID自动销毁。
 - ID在会话中通过Cookie在客户端和服务器端传送



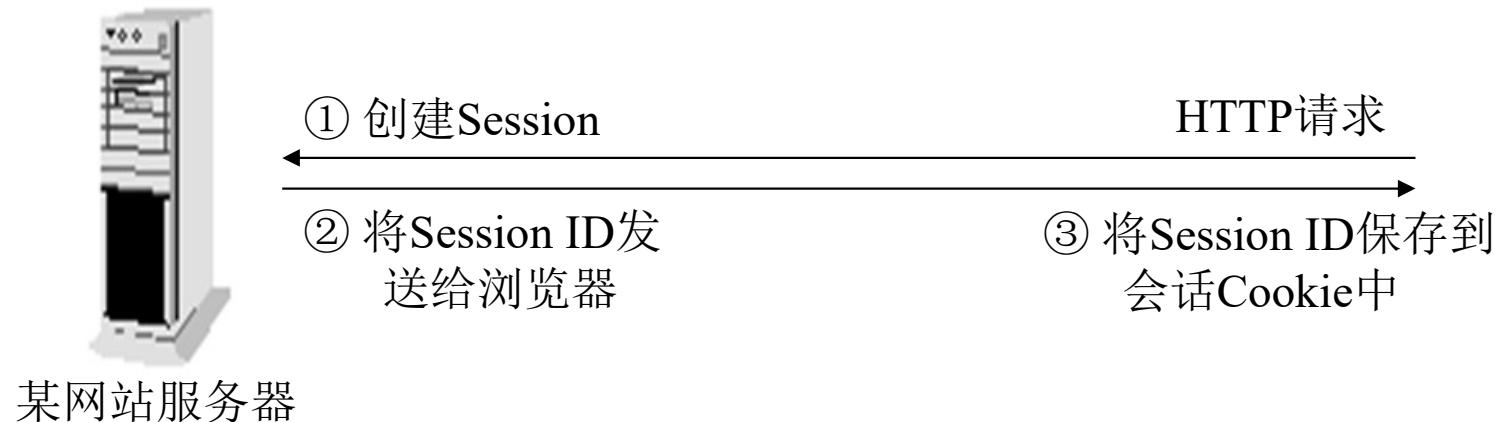
操作Session的方法

- ❖ `session_start()`: 创建或恢复一个会话
 - 必须在PHP输出之前执行！服务器自动发送HTTP头，设置客户端Cookie。
`header("Set-Cookie: session_name()=session_id(); path='/'");`
 - 在服务器上创建或寻找Session文件
- ❖ `$_SESSION[]`: 服务器上存储会话变量的超全局数组
- ❖ `unset($_SESSION["name"])` : 销毁会话变量name
- ❖ `session_unset()`: 销毁`$_SESSION`数组。会话结束时也会自动销毁。
- ❖ `session_destroy()`: 删除Session ID和Session文件

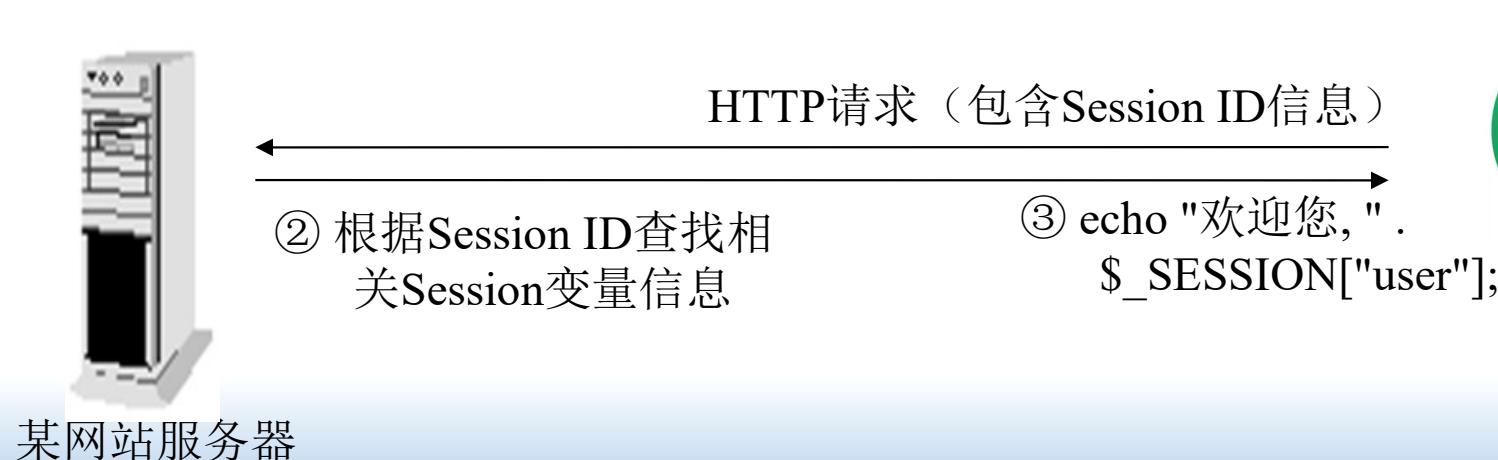


会话的创建和使用过程

第一次访问某网站



以后每次访问该网站





Session的生存期

❖ Session对象的生命周期

- 开始：用户打开网站的网页
 - 服务器自动生成Session ID和文件
- 结束：用户不再访问该网站
 - Session超时（超出有效期）
 - 主动删除Session
 - 注意：关闭浏览器 <> 不再访问该网站
 - 关闭浏览器：不会结束Session，因为服务器不知道用户关闭了浏览器；但会使Session无法被再次访问到，因为用户再次打开浏览器访问该网站时服务器会生成一个新的Session。



❖ Session对象的有效期

- 客户端超过有效期未刷新网页，或者访问其他网站，Session会自动结束。

❖ 设置服务器端Session的默认有效期

- 在PHP的配置文件php.ini中修改系统默认值 session.gc_maxlifetime (默认1440s, 24分钟)
- 使用函数ini_set()修改Session对象的默认有效期
- 不能单独对某个用户的Session设置有效期



例：禁止未登录用户访问网页

```
5-snlogin.php
1 <?php //登录页面，文件名5-snlogin.php
2     session_start();
3     $sessionName = session_name();
4     $sessionId = session_id();
5     if (isset($_POST["submit"])){
6         $userName = $_POST["userName"];
7         $password = $_POST["password"];
8         if ($userName == 'admin' && $password == '123'){
9             $_SESSION['userName'] = $userName;
10            header("location:5-snwelcome.php");
11        }else{
12            echo <<<_END
13                用户名密码错误！<br> <a href='5-snlogin.php'>返回登录页</a><br><br>
14                <b>Session信息:</b> <br><br>session_name()={$sessionName}<br>session_id()={$sessionId}
15                _END;
16            }
17        }else{
18            echo <<<_END
19                <form method="post" action=""
20                    用户名: <input type="text" name="userName" value="admin"><br>
21                    密码: <input type="password" name="password" value="123"><br>
22                    <input name="submit" type="submit" value="登录">
23                </form><br><br>
24                <b>Session信息:</b> <br><br>session_name()={$sessionName}<br>session_id()={$sessionId}
25                _END;
26        }
27    ?>
```



5-snwelcome.php

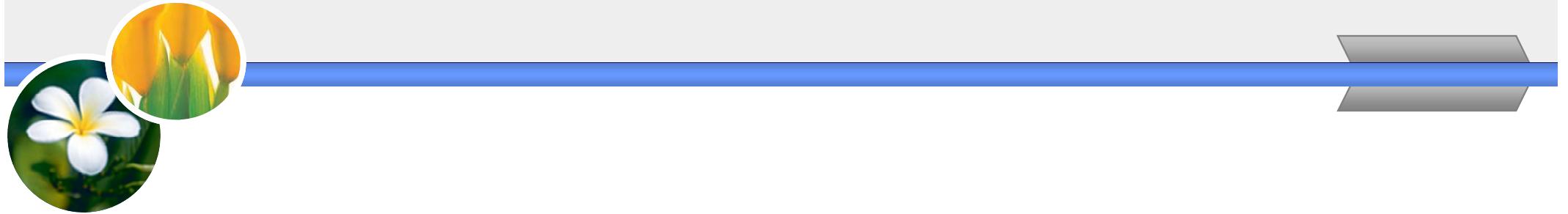
```
1 <?php //登录成功后的欢迎页面，文件名5-snwelcome.php
2     session_start();
3     $sessionName = session_name();
4     $sessionId = session_id();
5     if (isset($_SESSION['userName'])){
6         echo <<<_END
7             欢迎您，{$_SESSION['userName']}!  

8             <a href='5-snlogout.php?action=logout'>注销</a>
9             _END;
10    }else{
11        echo <<<_END
12            未登录用户！<br> <a href='5-snlogin.php'>返回登录页</a>
13            _END;
14    }
15    echo <<<_END
16        <br><br> <b>Session信息:</b> <br>
17        session_name()={$sessionName}<br>
18        session_id()={$sessionId}
19        _END;
20    ?>
```



5-snlogout.php

```
1 <?php //注销页面，文件名5-snlogout.php
2 if ($_GET['action'] == 'logout'){
3     session_start();
4     setcookie(session_name(), session_id(), time() - 60, '/');
5     session_unset();
6     session_destroy();
7     header("location:5-snlogout.php");
8 }else{
9     $sessionName = session_name();
10    $sessionId = session_id();
11    echo <<<_END
12        注销成功! <br> <a href='5-snlogin.php'>返回登录页</a>
13        <br><br> <b>Session信息:</b> <br>
14        session_name()={$sessionName}<br>
15        session_id()={$sessionId}
16    _END;
17 }
18 ?>
```



localhost/5-snlogin.php

用户名: admin
密码: ...
登录

Session信息:
session_name()=PHPSESSID
session_id()=qasq65e8pr210ue8686qpl14m7

应用 过滤

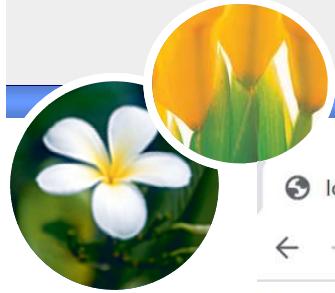
名称	值	D...	P...	E...	大	H	S...	S...	P...
Hm_lvt_d7c7037093...	1...	I...	/	2...	8.				
Hm_lpvt_d7c7037093...	1...	I...	/	会...	5.				
Phpstorm-9d4b2f6a	b...	I...	/	2...	5.	✓		S...	
PHPSESSID	q...	I...	/	会...	3.				

存储

- 本地存储空间
- 会话存储空间
- IndexedDB
- Web SQL
- Cookie**

Cookie Value 显示已解码的网址
qasq65e8pr210ue8686qpl14m7

控制台 新功能 网络请求屏蔽 搜索 问题



localhost/5-snwelcome.php

← → ⌂ ⌄ ⓘ localhost/5-snwelcome.php

欢迎您, admin!

[注销](#)

Session信息:

session_name()=PHPSESSID

session_id()=qasq65e8pr210ue8686qpl14m7

localhost/5-snlogout.php

← → ⌂ ⌄ ⓘ localhost/5-snlogout.php

注销成功!

[返回登录页](#)

Session信息:

session_name()=PHPSESSID

session_id()=

localhost/5-snlogin.php

← → ⌂ ⌄ ⓘ localhost/5-snlogin.php

用户名:

密码:

Session信息:

session_name()=PHPSESSID

session_id()=218fv1shb1u5o00ek41re8vb68



❖ php.ini文件

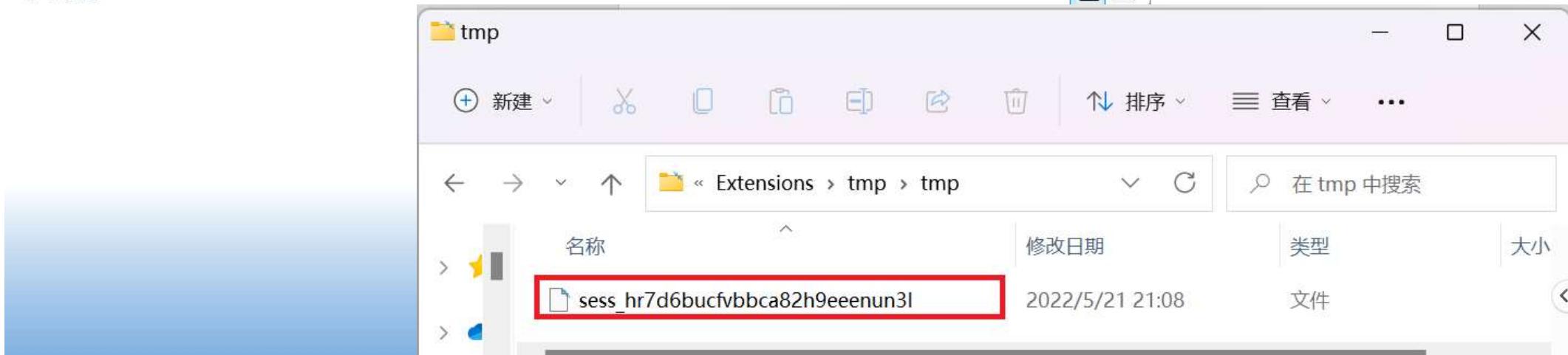
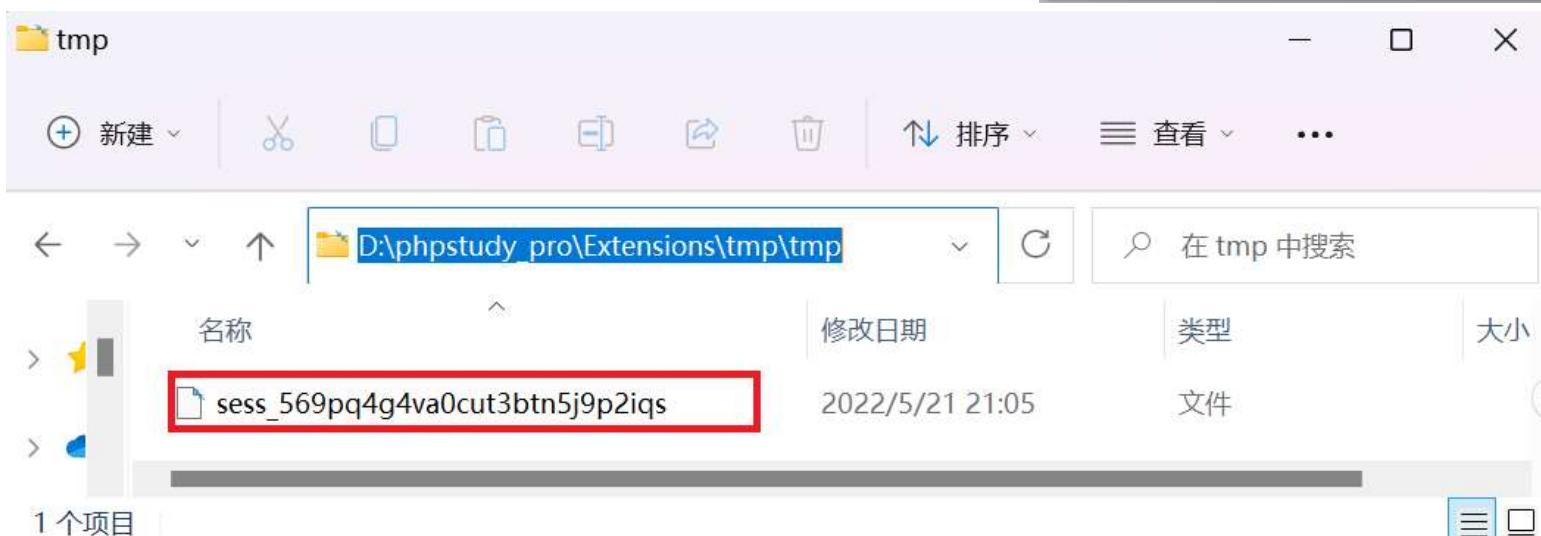
- session.save_path

php - 记事本

文件 编辑 查看

; http://php.net/session.save-path
session.save_path='D:\phpstudy_pro\Extensions\tmp\tmp'
; Whether to use strict session mode.

行 1365, 列 55 | 100% | Windows (CRLF) | UTF-8





Cookie技术简介

- ❖ Cookie也是在网页间共享信息的重要方法
- ❖ Cookie通过HTTP报头在服务器端与浏览器间交换信息
- ❖ Cookie创建后，以文件形式保存在客户机磁盘。
- ❖ 用户下次访问网站时，浏览器自动发送Cookie给服务器。
- ❖ 每个Cookie不能大于4KB，每个网站在一个客户端最多能保存20个Cookie。



使用Cookie的方法

❖ 服务器端创建（或更改）并发送Cookie的方法

- `setcookie(name, value, expire, path);`
- `setcookie`要在PHP输出之前执行，因为要在HTTP协议头部发送Cookie，类似`header("Set-Cookie: name=value");`。
- `expire`为Cookie过期的时间戳，`path`为Cookie有效的服务器路径。超过有效期的Cookie在客户端会被自动删除。

❖ 服务器端获取Cookie的方法

- 超全局数组 `$_COOKIE`
`$voted = $_COOKIE["voted"];`

❖ 从服务器端删除客户端Cookie的方法

- 把有效期更改成过去的时间点

```
setcookie("voted", "", time()-3600, "/");
```

用户名: admin

密码: ...

登录

Session信息:

session_name()=PHPSESSID

session_id()=qasq65e8pr210ue8686qpl14m7

应用 过滤

名称	值	D...	P...	E...	大...	H...	S...	S...	P...
Hm_lvt_d7c7037093...	1...	I...	/	2...	8.				
Hm_lpvt_d7c7037093...	1...	I...	/	会...	5.				
Phpstorm-9d4b2f6a	b...	I...	/	2...	5.	✓		S...	
PHPSESSID	q...	I...	/	会...	3.				

Cookie Value 显示已解码的网址
qasq65e8pr210ue8686qpl14m7

应用
清单
Service Workers
存储
存储
本地存储空间
会话存储空间
IndexedDB
Web SQL
Cookie
http://localhost
私密状态令牌

控制台 新功能 网络请求屏蔽 搜索 问题



```
<?php //记录客户端访问网站的次数
if (!isset($_COOKIE['visnum'])){
    setcookie('visnum', 1, time() + 86400); //24小时
    echo "您是第一次访问本网站。<br>";
} else{
    $visnum = $_COOKIE['visnum'];
    $visnum++;
    setcookie('visnum', $visnum, time() + 86400);
    echo "您是第{$visnum}次访问本网站。<br>";
}
```

← → ⌂ ⌂ ① localhost/php/cookie.php

您是第2次访问本网站。

元素 控制台 源代码 网络 性能 内存 应用 安全 Lighthouse Recorder »

IndexedDB
Web SQL
Cookie http://localhost 信任令牌

名称	值	Domain	Path	Expires / Max-Age	大小
visnum	2	localhost	/php	2022-05-22T13:50:36.431Z	7



Session和Cookie的比较

	Session	Cookie
相似点		
功能	存储用户信息，在站点所有页面都可以访问。	
不同点		
建立方式	访问网页时自动建立	需要通过代码建立
存储位置	服务器端	客户端



JSON

- ❖ JSON (JavaScript Object Notation, JS对象表示法) 是一种轻量级的数据交换格式。既易于人类阅读和编写，也易于机器生成和解析。
- ❖ JSON构建于两种结构
 - 键:值对的集合描述对象。
 - 一个对象以{开始，以}结束。每个键后跟一个冒号:，键:值对之间使用逗号,分隔。
 - 值的有序列表描述数组。
 - 一个数组以[开始，以]结束。值之间使用逗号,分隔。
- ❖ JSON的值可以是数值、字符串、逻辑值、数组、对象、null。值可以嵌套。



❖ PHP对象的JSON字符串样例

```
{  
    "firstname": "Brendan",  
    "lastname": "Eich"  
}
```

❖ PHP对象数组的JSON字符串样例

```
[  
    {"firstName": "Brendan", "lastName": "Eich"},  
    {"firstName": "Bill", "lastName": "Gates"},  
    {"firstName": "Steve", "lastName": "Jobs"}  
]
```



PHP处理JSON数据的函数

- ❖ PHP提供了两个处理JSON数据的函数，只接受utf-8编码的字符。
 - `json_encode()`: 将PHP数据转换成JSON格式数据。
 - `json_decode()`: 将JSON格式数据转换成PHP数据。



json_encode

- ❖ json_encode: 将PHP数据转换为JSON格式数据。

将PHP数组转换成JSON格式的字符串：

```
1 <?php
2     $arr = array('a'=>1, 'b'=>2, 'c'=>3, 'd'=>4, 'e'=>5);
3     echo json_encode($arr);
4 ?>
```

结果为：

```
{"a":1,"b":2,"c":3,"d":4,"e":5}
```



将PHP对象转换成JSON格式的字符串：

```
1 <?php  
2     $obj->body = 'another post';  
3     $obj->id = 21;  
4     $obj->approved = true;  
5     $obj->favorite_count = 1;  
6     $obj->status = null;  
7     var_dump(json_encode($obj));  
8 ?>
```

结果为：

```
| string(80) "{"body":"another post","id":21,"approved":true,"favorite_count":1,"status":null}"
```



json_decode

- ❖ json_decode: 将JSON格式数据转换为PHP数据。

将JSON格式的字符串转换成PHP对象：

```
1 <?php
2     $json = '{"a":1,"b":2,"c":3}';
3     var_dump(json_decode($json)); //输出变量结构
4 ?>
```

结果为：

```
| object(stdClass)#1 (3) { ["a"]=> int(1) ["b"]=> int(2) ["c"]=> int(3) }
```



将JSON格式的字符串转换成PHP对象数组：

```
1 <?php
2
3 $strjson = '[{"firstName": "Brendan", "lastName": "Eich"}, {"firstName": "Bill", "lastName": "Gates"}, {"firstName": "Steve", "lastName": "Jobs"}]';
4
5 var_dump($strjson);
6
7 echo "<br>";
8
9 $objjson = json_decode($strjson);
10
11 var_dump($objjson);
```



将JSON格式的字符串转换成PHP对象数组：
结果为：

```
string(174) "[ { \"firstName\":\"Brendan\", \"lastName\": \"Eich\" }, {  
\"firstName\":\"Bill\", \"lastName\":\"Gates\" }, { \"firstName\": \"Steve\", \"lastName\":  
\"Jobs\" } ]"  
array(3) { [0]=> object(stdClass)#1 (2) { ["firstName"]=> string(7) "Brendan"  
["lastName"]=> string(4) "Eich" } [1]=> object(stdClass)#2 (2) {  
["firstName"]=> string(4) "Bill" ["lastName"]=> string(5) "Gates" } [2]=>  
object(stdClass)#3 (2) { ["firstName"]=> string(5) "Steve" ["lastName"]=>  
string(4) "Jobs" } }
```



常用文件操作函数

- ❖ 文件打开关闭函数
- ❖ 文件读取函数
- ❖ 移动文件指针函数
- ❖ 文本文件写入和追加函数



文件打开关闭函数

❖ fopen(filename, mode)

- 以mode方式打开filename文件。如果打开文件成功，返回指向该文件的（resource类型）指针；如果打开失败，返回false。
- mode: r 只读， w 覆盖写入， a 追加写入， b 二进制模式打开。

❖ fclose(fp)

- 关闭文件指针fp指向的通过fopen打开的文件。关闭成功返回true；失败返回false。

❖ feof(fp)

- 判断文件指针fp是否到达文件末尾EOF（最后一个字符后），已到达或出错时返回true，否则返回false。



文件读取函数

❖ fread(fp, length)

- 从文件指针fp处读取length个字节。如果要读取整个文件，length可通过filesize函数获取文件大小。可用于二进制文件读。
- fp是fopen打开且尚未被fclose关闭的有效文件指针。

```
5-fread.php
1 <?php
2     $fp = fopen("./5-info.txt", "rb");
3     echo $fp . "<hr>";
4     $inStr = fread($fp, filesize("./5-info.txt"));
5     echo $inStr . "<hr>";
6     echo nl2br($inStr); //将换行符转换成<br/>标记
7     fclose($fp);
8 ?>
```



← → C ⌂

① localhost/5-fread.php

Resource id #3

测试用文本文件 第一行 第二行 第三行

测试用文本文件
第一行
第二行
第三行

≡ 5-info.txt

- 1 测试用文本文件
- 2 第一行
- 3 第二行
- 4 第三行



❖ fgets(fp[, length])

- 从文件指针fp处读取一行，返回长度最多为length-1个字节的字符串。读到回车符或length-1个字符时停止。未指定length时默认为1024字节。

```
5 fgets.php
1 <?php
2     $fp = fopen("5-fread.php", "rb");
3     echo $fp . "<hr>";
4     while(($inStr = fgets($fp)) !== false){
5         echo $inStr . '<hr>';
6     }
7     if(!feof($fp)){
8         echo "fgets文件读取错误!";
9     }
10    fclose($fp);
11 ?>
```



← → C ⌂ ⓘ localhost/5-fgets.php

Resource id #3

```
$fp = fopen("./5-info.txt", "rb");
```

```
echo $fp . "
```

```
";
```

```
$inStr = fread($fp, filesize("./5-info.txt"));
```

```
echo $inStr . "
```

```
";
```

```
echo nl2br($inStr); //将换行符转换成  
标记
```

```
fclose($fp);
```

```
?>
```



❖ file_get_contents(filename)

- 无须打开和关闭操作，直接读取，返回字符串，失败时返回 false。

```
5-file_get_contents.php
1 <?php
2     $inStr = file_get_contents("http://www.baidu.com");
3     echo $inStr;
4 ?>
```

The screenshot shows a web browser window with the address bar containing 'localhost/5-file_get_contents.php'. The main content area of the browser displays the Baidu search engine homepage, featuring the Baidu logo and a search bar with the placeholder '百度一下，你就知道'.

At the bottom of the browser window, there is a footer bar with links: '关于百度' (About Baidu), 'About Baidu', '使用百度前必读' (Must Read Before Using Baidu), '帮助中心' (Help Center), '京公网安备11000002000001号' (Beijing Public Security Bureau No. 11000002000001), '京ICP证030173号' (ICP License No. 030173), '©2023 Baidu', '互联网药品信息服务资格证书 (京)-经营性-2017-0020', '信息网', and a speaker icon indicating '扬声器 (Realtek(R) Audio): 已静音' (Speaker (Realtek(R) Audio): Muted).



❖ fgetc(fp)

- 从文件指针fp处读取一个字节。可读取二进制文件。

```
file fgetc.php
1 <?php
2 $fp = fopen("5-info.txt", "rb");
3 if($inChar = fgetc($fp)){
4     var_dump($inChar);
5 }
6 ?>
```

← → ⌂ ⌂ ⓘ localhost/fgetc.php

string(1) "◆"



❖ file(filename)

- 无须打开和关闭操作，直接读取文件filename，返回数组。返回数组的每个元素对应文件的一行，失败时返回false。可读取二进制文件。

```
5-file.php
1 <?php
2     if ($inArr = file("5-info.txt")) {
3         var_dump($inArr);
4     }
5 ?>
```

← → ⌂ ⌂ ⓘ localhost/5-file.php ⌂ ☆ ⌂ ⌂ ⌂ :

array(5) { [0]=> string(23) "测试用文本文件 " [1]=> string(11) "第一行 " [2]=> string(11) "第二行 " [3]=> string(11) "第三行 " [4]=> string(2) " " }

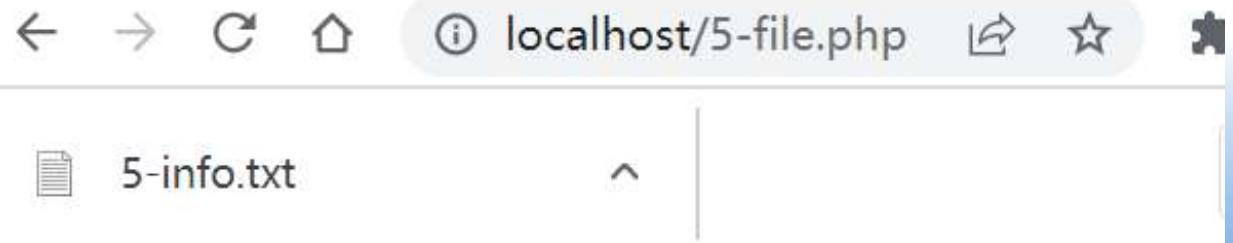


❖ readfile(filename)

- 无须打开和关闭操作，直接读取文件filename的内容到输出缓冲区，成功读取后返回读取的字节数，失败返回false。

5-readfile.php

```
1 <?php
2     $filename = "5-info.txt";
3     if(file_exists($filename)){
4         header('Content-Description: File Transfer');
5         header('Content-Type: text/txt');
6         header('Content-Disposition: attachment; filename="' . basename($filename) . "'");
7         header('Expires: 0');
8         header('Cache-Control: must-revalidate');
9         header('Pragma: public');
10        header('Content-Length: ' . filesize($filename));
11        readfile($filename); //读到输出缓冲区
12        exit;
13    }
14 ?>
```





移动文件指针函数

❖ rewind(fp)

- 移动文件指针fp到文件头。

❖ ftell(fp)

- 返回文件指针fp的位置。

❖ fseek(fp, offset[, origin])

- 移动文件指针fp到origin+offset处。origin为SEEK_SET(0)表示文件头, SEEK_CUR(1)表示当前位置; SEEK_END(2)表示文件末尾。
- 执行成功返回0; 失败返回-1。



≡ 5-chars.txt

1 abcdefghijklmnopqrstuvwxyz

5-ftell.php

```
1 <?php
2     $fp = fopen("5-chars.txt", "rb") or die("文件打开失败");
3     echo ftell($fp)."<br>";
4     echo fread($fp, 5)."<br>";
5     echo ftell($fp)."<br>";
6     fseek($fp, 10, 1);
7     echo ftell($fp)."<br>";
8     echo fread($fp, 5)."<br>";
9     fseek($fp, -10, 2);
10    echo fread($fp, 10)."<br>";
11    echo ftell($fp)."<br>";
12    rewind($fp);
13    echo ftell($fp)."<br>";
14    fclose($fp)
15 ?>
```



localhost/5-ftell.php

0

abcde

5

15

pqrst

qrstuvwxyz

26

0



文本文件写入和追加函数

❖ `fwrite(fp, string[, length])`

- 向文件指针fp位置写入字符串string。length为最多写入的字节数。写入成功返回写入字节数；否则返回false。

❖ `file_put_contents(filename, data[, mode])`

- 无须打开和关闭操作，以mode方式将data（可以是字符串、数组或数据流）写入文件filename。
- mode默认是覆盖写，FILE_APPEND追加写入。



fwrite.php

```
1 <?php  
2     $fp = fopen("new.txt", "wb");  
3     fwrite($fp, "这是写入的第一行\n");  
4     fwrite($fp, "最多写入23个字符不写不写\n", 23);  
5     fclose($fp);  
6 ?>
```

≡ new.txt

```
1 这是写入的第一行  
2 最多写入23个字符
```

fwrite.php

```
1 <?php  
2     $fp = fopen("new.txt", "ab");  
3     fwrite($fp, "这是追加写入的一行\n");  
4     fclose($fp);  
5 ?>
```

≡ new.txt

```
1 这是写入的第一行  
2 最多写入23个字符这是追加写入的一行
```



5-fwrite.php

```
1 <?php
2     $fp = fopen("new.txt", "w+");
3     for($i = 0; $i < 10; $i++)
4         fwrite($fp, "这是写入的一行话\n");
5     rewind($fp);
6     $str = fread($fp, filesize("new.txt"));
7     echo nl2br($str);
8     fclose($fp)
9 ?>
```

← → ⌂ ⌂ ⓘ localhost/5-fwrite.php



```
5-fwrite.php
1 <?php
2     echo file_put_contents('new.txt', '第一次'); //写入字符串
3     $data = '要写入的数据';
4     $num = file_put_contents('new.txt', $data, FILE_APPEND); //追加方式写入
5     echo $num; //返回写入的字节数, 18
6 ?>
```

← → ⌂ ⌂ ⓘ localhost:3000/5-fwrite.php

918

≡ new.txt
1 第一次要写入的数据



文件及目录的基本操作函数

- ❖ 复制、移动和删除文件
- ❖ 获取文件属性
- ❖ 目录的基本操作
- ❖ 统计目录和磁盘大小



复制、移动和删除文件

❖ copy(source, dest)

- 将文件source的内容复制到文件dest。成功返回true，失败返回false。

❖ unlink(filename)

- 删除文件filename。成功返回true，失败返回false。

❖ rename(oldname, newname)

- 文件或目录oldname 重命名为newname。
- 当newname与oldname的路径不同时，进行文件移动。
- 文件newname存在，覆盖。
- 目录newname存在，警告。



❖ ftruncate(filename, length)

- 将文件filename截断到长度length。成功返回true，失败返回false。

❖ file_exists(filename)

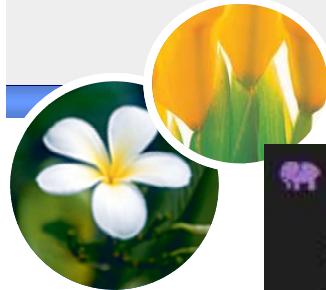
- 判断文件或目录filename是否存在。成功返回true，失败返回false。

❖ is_file(filename)

- 判断路径filename是否存在且为文件。成功返回true，失败返回false。

❖ is_dir(filename)

- 判断路径filename是否存在且为目录。成功返回true，失败返回false。



5-file-copy-del-mov.php

```
1 <?php  
2     //复制、删除、移动文件  
3     if (file_exists('test.txt')){  
4         if(copy('test.txt', '../testbak.txt')){  
5             echo '文件复制成功！';  
6             if(unlink('test.txt'))  
7                 echo '文件删除成功！';  
8             else  
9                 echo '文件删除失败！';  
10            if (file_exists('../testbak.txt')){  
11                if(rename('../testbak.txt', 'testnew.txt'))  
12                    echo '文件移动并重命名成功！';  
13                else  
14                    echo '文件移动失败！';  
15            }  
16        }else  
17            echo '文件复制失败，源文件可能不存在！';  
18    }else  
19        echo "文件不存在！";  
20  
21 ?>
```



获取文件属性

❖ filesize(filename)

- 只读，返回文件filename的大小。失败时返回false。

❖ filetype(filename)

- 只读，返回文件filename的类型，如文件或文件夹。

❖ filectime(filename)

- 返回文件filename创建的时间戳

❖ filemtime(filename)

- 只读，返回文件filename最近一次修改的时间戳。

❖ fileatime(filename)

- 只读，返回文件filename最近一次访问的时间戳。



- ❖ **realpath(filename)**
 - 返回文件filename的规范化绝对路径
- ❖ **pathinfo(filename)**
 - 返回有关filename的关联数组，包含文件filename的路径和文件名信息。
- ❖ **dirname(filename)**
 - 返回文件filename与当前路径的关系
- ❖ **basename(filename)**
 - 返回文件filename的文件名
- ❖ **stat(filename)**
 - 返回文件filename统计信息的关联数组



5-fileattr.php

```
1 <?php
2     $filename = "5-info.txt";
3     echo "绝对路径为: " . realpath($filename) . "<br>";
4     echo "所在目录为: " . dirname($filename) . "<br>";
5     echo "文件名为: " . basename($filename) . "<br>";
6     echo "文件大小为: " . filesize($filename) . "字节<br>";
7     echo "文件类型为: " . filetype($filename) . "<br>";
8     echo "文件创建时间为: " . date("Y年m月d日 H:m:s", filectime($filename)) . "<br>";
9     echo "文件最后修改时间为: " . date("Y年m月d日 H:m:s", filemtime($filename)) . "<br>";
10    echo "文件最后访问时间为: " . date("Y年m月d日 H:m:s", fileatime($filename)) . "<br>";
11    echo "pathinfo(\$filename)为: <br>";
12    var_dump(pathinfo($filename));
13    echo "<br>stat(\$filename)为: <br>";
14    var_dump(stat($filename));
15 ?>
```



← → ⌂ ⌂

① [localhost/5-fileattr.php](#)



绝对路径为: D:\phpstudy_pro\WWW\2023\5-info.txt

所在目录为：.

文件名为：5-info.txt

文件大小为：58字节

文件类型为：file

文件创建时间为：2023年06月02日 15:06:30

文件最后修改时间为：2023年06月02日 20:06:16

文件最后访问时间为：2023年06月02日 20:06:02

`pathinfo($filename)`为：

```
array(4) { ["dirname"]=> string(1) "." ["basename"]=> string(10) "5-info.txt"  
["extension"]=> string(3) "txt" ["filename"]=> string(6) "5-info" }
```

`stat($filename)`为：

```
array(26) { [0]=> int(4294967252) [1]=> int(0) [2]=> int(33206) [3]=> int(1)
[4]=> int(0) [5]=> int(0) [6]=> int(-44) [7]=> int(58) [8]=> int(1685708762)
[9]=> int(1685707876) [10]=> int(1685692470) [11]=> int(-1) [12]=> int(-1)
["dev"]=> int(4294967252) ["ino"]=> int(0) ["mode"]=> int(33206) ["nlink"]=>
int(1) ["uid"]=> int(0) ["gid"]=> int(0) ["rdev"]=> int(-44) ["size"]=> int(58)
["atime"]=> int(1685708762) ["mtime"]=> int(1685707876) ["ctime"]=>
int(1685692470) ["blksize"]=> int(-1) ["blocks"]=> int(-1) }
```



目录的基本操作

❖ mkdir(dirname)

- 创建目录dirname。成功返回true，失败返回false。

❖ rmdir(dirname)

- 删除目录dirname， 目录必须为空。

❖ getcwd()

- 返回当前工作目录

❖ chdir(dirname)

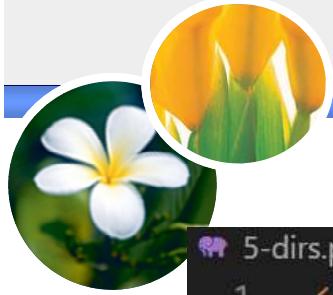
- 改变当前目录为dirname

❖ opendir(dirname)

- 打开目录dirname， 返回目录的指针。



- ❖ **closedir(dirh)**
 - 关闭（opendir返回的）目录指针dirh指向的目录
- ❖ **readdir(dirh)**
 - 遍历（opendir返回的）目录指针dirh，读取成功返回dirh当前指向的文件名，然后将dirh向后移一位，dirh指向目录结尾时，返回false。
- ❖ **scandir(dirname, sort)**
 - 以数组形式返回目录dirname中的文件和目录，sort默认字母升序排列，为1时降序排列。
- ❖ **rewinddir(dirh)**
 - 将（opendir返回的）目录指针dirh重置到目录开头



5-dirs.php

```
1 <?php
2     $num = 0;
3     $dirname = "../";
4     $dirh = opendir($dirname);
5     $bgcolor = '#cccccc';
6 ?>
7 <table border="1" width="800">
8     <caption><b>目录<?=$dir?>中的内容</b></caption>
9     <tr align="left" bgcolor="<?= $bgcolor?>">
10        <th>文件名</th><th>大小</th>
11        <th>类型</th><th>修改时间</th>
12    </tr>
13 <?php
14     while ($file = readdir($dirh)){
15         if($file != "." && $file != ".."){
16             $filename = $dirname . "/" . $file;
17             $num++;
18             echo "<tr bgcolor={$bgcolor}>";
19             echo "<td>{$filename}</td>";
20             echo "<td>". filesize($filename) . "</td>";
21             echo "<td>". filetype($filename) . "</td>";
22             echo "<td>" . date("Y年m月d日 H:i:s"), filemtime($filename) . "</td></tr>";
23         }
24     }
25     closedir($dirh);
26 ?>
27 </table>
28 在<b><?=$dirname?></b>目录下的子目录和文件共有<b><?=$num?></b>个<b>
```



← → C ⌂ ⓘ localhost/5-dirs.php

目录中的内容

文件名	大小	类型	修改时间
..//2023	20480	dir	2023年06月03日 10:08:05)
..//error	0	dir	2022年03月07日 13:43:56)
..//index.html	2307	file	2019年09月03日 14:30:48)

在..//目录下的子目录和文件共有3个



5-dir.php

```
1 <?php
2     if(!file_exists('temp'))
3         mkdir('temp');
4     else
5         echo "目录已存在，不能创建! <br>";
6     if(file_exists('data'))
7         rmdir('data');
8     else
9         echo "该目录不存在，不能删除! <br>";
10    echo getcwd() . "<br>";
11    chdir("../");
12    echo getcwd() . "<br>";
13 ?>
```

← → ⌂ ⌂ ⓘ localhost/5-dir.php

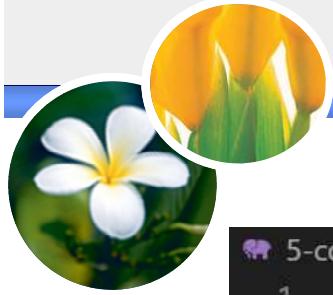
该目录不存在，不能删除！

D:\phpstudy_pro\WWW\2023
D:\phpstudy_pro\WWW

← → ⌂ ⌂ ⓘ localhost/5-dir.php

目录已存在，不能创建！

该目录不存在，不能删除！
D:\phpstudy_pro\WWW\2023
D:\phpstudy_pro\WWW



5-copydir.php > ...

```
1  <?php
2      4 references
3      function copyDir($dirFrom, $dirTo){
4          if (is_file($dirTo)){
5              echo "目标位置不是目录，不能创建！";
6              return false;
7          }
8          if (!file_exists($dirTo)){
9              mkdir($dirTo);
10             echo "创建目录: " . $dirTo . "<br>";
11         }
12         if ($dirh = opendir($dirFrom)){
13             while ($filename = readdir($dirh)){
14                 if ($filename != "." && $filename != ".."){
15                     $subFileFrom = $dirFrom . "/" . $filename;
16                     $subFileTo = $dirTo . "/" . $filename;
17                     if (is_dir($subFileFrom))
18                         copyDir($subFileFrom, $subFileTo);
19                     if (is_file($subFileFrom)){
20                         copy($subFileFrom, $subFileTo);
21                         echo "拷贝文件: " . $subFileFrom . " --> " . $subFileTo . "<br>";
22                     }
23                 }
24             }
25         }
26     }
27     copyDir("./img", "./temp");
28 ?>
```



```
5-deldir.php > ...
1  <?php
2      4 references
3      function delDir($dirname){
4          if(file_exists($dirname)){
5              if($dirh = opendir($dirname)){
6                  while($filename = readdir($dirh)){
7                      if ($filename != "." && $filename != "..") {
8                          $subFile = $dirname . '/' . $filename;
9                          if (is_dir($subFile))
10                             delDir($subFile);
11                          if (is_file($subFile)){
12                              unlink($subFile);
13                              echo "删除文件: " . $subFile . "<br>";
14                          }
15                      }
16                  }
17                  closedir($dirh);
18                  rmdir($dirname);
19                  echo "目录: " . $subFile . "<br>";
20              }
21          }
22      }
23      delDir("./temp");
24  ?>
```



← → C ⌂ ⓘ localhost/5-copydir.php

拷贝文件: ./img/gif.gif --> ./temp/gif.gif
拷贝文件: ./img/jpeg.jpeg --> ./temp/jpeg.jpeg
拷贝文件: ./img/mail.gif --> ./temp/mail.gif

← → C ⌂ ⓘ localhost/5-deldir.php

删除文件: ./temp/gif.gif
删除文件: ./temp/jpeg.jpeg
删除文件: ./temp/mail.gif
目录: ./temp/mail.gif



统计目录和磁盘大小

❖ disk_free_space(dirname)

- 返回目录/磁盘dirname可用空间的字节数，不存在时返回false。

❖ disk_total_space(dirname)

- 返回目录/磁盘dirname所在文件系统或磁盘分区空间的字节数，不存在时返回false。

5-dirsize.php > ...

```
1  <?php
2      4 references
3      function dirSize($dirname){
4          $dirSize = 0;
5          if ($dirh = opendir($dirname)){
6              while($filename = readdir($dirh)){
7                  if($filename != "." && $filename != ".."){
8                      $subFile = $dirname . "/" . $filename;
9                      if (is_dir($subFile))
10                         $dirSize += dirSize($subFile);
11                      if (is_file($subFile))
12                          $dirSize += filesize($subFile);
13                  }
14              }
15              closedir($dirh);
16              return $dirSize;
17          }
18      }
19      $dirSize = dirSize("../");
20      var_dump($dirSize);
21      echo "<br>" . round($dirSize/pow(2, 10), 3) . "KB";
22  ?>
```

← → ⌂ ⌄

localhost/5-dirsize.php

int(16543199)
16155.468KB



5-jsonfile.php X reginfo.txt

5-jsonfile.php > ...

```
1 <?php //JSON文件输入输出处理样例
2     define('HEAD', '<!DOCTYPE html><head><meta charset="utf-8"/><title>JSON文件的输入输出处理</title></head><body>');
3     define('TAIL', '</body></html>');
4     define('REGINFO_FILE', 'reginfo.txt');
5     date_default_timezone_set("Asia/Shanghai"); //设置时区为中国
6     2 references | 0 implementations
7     class RegInfo{
8         6 references
9         var $userName;
10        6 references
11         var $password;
12        6 references
13         var $regTime;
14         2 references | 0 overrides
15         function outInfo(){
16             echo "用户名: " . $this->userName . "<br>";
17             echo "密码: " . $this->password . "<br>";
18             echo "注册时间: " . $this->regTime . "<br>";
19         }
20     }
```



```
5-jsonfile.php X  reginfo.txt

5-jsonfile.php > ...

16     $regUser = new RegInfo();
17     $regUser->userName = $_GET['userName'];
18     $regUser->password = $_GET['password'];
19     $regUser->regTime = date('Y-m-d H:i:s');
20     $jsonStr = json_encode($regUser);
21     file_put_contents(REGINFO_FILE, $jsonStr, FILE_APPEND);
22     file_put_contents(REGINFO_FILE, "\n", FILE_APPEND);
23     $regInfos = file(REGINFO_FILE);
24     echo HEAD;
25     foreach ($regInfos as $jsonStr){
26         $regInfo = json_decode($jsonStr);
27         $regUser->userName = $regInfo->userName;
28         $regUser->password = $regInfo->password;
29         $regUser->regTime = $regInfo->regTime;
30         $regUser->outInfo();
31     }
32     echo TAIL;
33 ?>
```



← → C ⌂

① localhost/5-jsonfile.php?userName=badman&password=<hr>

用户名: **badman**

密码:

注册时间: 2023-06-03 10:45:09

← → C ⌂

① localhost/5-jsonfile.php?userName=admin&password=Don%27t%20tell%20you

用户名: **badman**

密码:

注册时间: 2023-06-03 10:45:09

用户名: admin

密码: Don't tell you

注册时间: 2023-06-03 10:49:10

☰ reginfo.txt

```
1 {"userName":<b>badman</b>,"password":<hr>,"regTime":"2023-06-03 10:45:09"}  
2 {"userName":<b>admin</b>,"password":"Don't tell you","regTime":"2023-06-03 10:49:10"}
```



文件上传功能的实现

❖ PHP中文件上传功能的实现步骤

- (1) 选择并添加文件到上传表单中，单击提交按钮后选择的文件数据将被发送到服务器；
- (2) 用全局变量 **\$_FILES** 获取与上传文件有关的各种信息；
- (3) 用文件上传处理函数对上传文件进行后续处理。



操作上传文件

- ❖ 若文件上传标签的name属性值为upfile， 可以使用 `$_FILES['upfile']` 来获取上传文件的有关信息。
 - `$_FILES['upfile']['name']`: 上传文件的原名称
 - `$_FILES['upfile']['type']`: 上传文件的类型
 - `$_FILES['upfile']['size']`: 上传文件的大小
 - `$_FILES['upfile']['tmp_name']`: 上传文件在服务器端保存的临时文件名（包含路径）
 - `$_FILES['upfile']['error']`: 错误代码。0表示没有错误（内置常量：`UPLOAD_ERR_OK`）。
- ❖ `move_uploaded_file(tempfile, destfile)`
 - 将上传时保存的临时文件`tempfile`移动到指定`destfile`



← → C ⌂ ⓘ localhost:3000/5-uploadfile.html

文件上传示例

请选择要上传的文件:

选择文件 未选择任何文件

上传

← → C ⌂ ⓘ localhost:3000/5-uploadfile.html

文件上传示例

请选择要上传的文件:

选择文件 gif.gif

上传

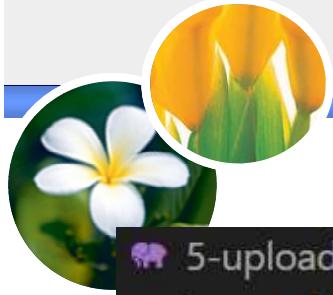
← → C ⌂ ⓘ localhost:3000/5-uploadfile.php

文件上传成功!



5-uploadfile.html > ...

```
1  <!DOCTYPE html>
2  <head>
3      <meta charset="UTF-8">
4      <title>文件上传</title>
5  </head>
6  <body>
7      <h2 align="center">文件上传示例</h2>
8      <p>请选择要上传的文件: </p>
9      <form action="5-uploadfile.php" method="post" enctype="multipart/form-data">
10         <input type="file" name="upfile"><br><br>
11         <input type="submit" value="上传">
12     </form>
13  </body>
14  </html>
```



5-uploadfile.php

```
1  <!DOCTYPE html>
2  <head>
3      <meta charset="UTF-8">
4      <title>文件上传结果</title>
5  </head>
6  <body>
7  <?php
8      $errno = $_FILES['upfile']['error'];
9      if ($errno != UPLOAD_ERR_OK)
10         echo "<h4>文件上传失败，错误类型: {$errno}</h4>";
11     else{
12         $tmpName = $_FILES['upfile']['tmp_name'];
13         $upfileName = $_FILES['upfile']['name'];
14         $upfileName = mb_convert_encoding($upfileName, 'GB18030', 'utf-8');
15         if (file_exists($tmpName)){
16             if (move_uploaded_file($tmpName, './files/'.$upfileName))
17                 echo '<h4>文件上传成功! </h4>';
18         }
19     }
20 ?>
21 </body>
22 </html>
```