**php解析url并得到url中的参数及****获取url参数的四种方式**

下面一段代码是php解析url并得到url中的**参数**，代码如下所示：

<?php

$url = 'http://www.baidu.com/index.php?m=content&c=index&a=lists&catid=6&area=0&author=0&h=0®ion=0&s=1&page=1';

$arr = parse\_url($url);

var\_dump($arr);

$arr\_query = convertUrlQuery($arr['query']);

var\_dump($arr\_query);

var\_dump(getUrlQuery($arr\_query));

/\*\*

\* 将字符串**参数**变为数组

\* @param $query

\* @return array array (size=10)

'm' => string 'content' (length=7)

'c' => string 'index' (length=5)

'a' => string 'lists' (length=5)

'catid' => string '6' (length=1)

'area' => string '0' (length=1)

'author' => string '0' (length=1)

'h' => string '0' (length=1)

'region' => string '0' (length=1)

's' => string '1' (length=1)

'page' => string '1' (length=1)

\*/

function convertUrlQuery($query)

{

$queryParts = explode('&', $query);

$params = array();

foreach ($queryParts as $param) {

$item = explode('=', $param);

$params[$item[0]] = $item[1];

}

return $params;

}

/\*\*

\* 将**参数**变为字符串

\* @param $array\_query

\* @return string string 'm=content&c=index&a=lists&catid=6&area=0&author=0&h=0®ion=0&s=1&page=1' (length=73)

\*/

function getUrlQuery($array\_query)

{

$tmp = array();

foreach($array\_query as $k=>$param)

{

$tmp[] = $k.'='.$param;

}

$params = implode('&',$tmp);

return $params;

}

下面通过四种实例给大家介绍php url **参数**获取方式。

在已知URL**参数**的情况下，我们可以根据自身情况采用$\_GET来获取相应的**参数**信息（$\_GET['name']）;那，在未知情况下如何获取到URL上的**参数**信息呢？

**第一种、利用$\_SERVER内置数组变量**

相对较为原始的$\_SERVER['QUERY\_STRING']来获取，URL的**参数**，通常使用这个变量返回的会是类似这样的数据：name=tank&sex=1  
如果需要包含文件名的话可以使用$\_SERVER["REQUEST\_URI"](返回类似：/index.php?name=tank&sex=1)

**第二种、利用pathinfo内置函数**

 代码如下:

<?php

$test = pathinfo("http://localhost/index.php");

print\_r($test);

/\*

结果如下

Array

(

[dirname] => http://localhost //url的路径

[basename] => index.php //完整文件名

[extension] => php //文件名后缀

[filename] => index //文件名

)

\*/

?>

**第三种、利用parse\_url内置函数**

代码如下:

<?php

$test = parse\_url("http://localhost/index.php?name=tank&sex=1#top");

print\_r($test);

/\*

结果如下

Array

(

[scheme] => http //使用什么协议

[host] => localhost //主机名

[path] => /index.php //路径

[query] => name=tank&sex=1 // 所传的**参数**

[fragment] => top //后面根的锚点

)

\*/

?>

**第四种、利用basename内置函数**

代码如下:

<?php

$test = basename("http://localhost/index.php?name=tank&sex=1#top");

echo $test;

/\*

结果如下

index.php?name=tank&sex=1#top

\*/

?>

另外，还有就是自己通过正则匹配的处理方式来获取需要的值了。这种方式较为精确，效率暂不考虑。。。  
下面拓展实践下正则处理方式：

代码如下:

<?php

preg\_match\_all("/(\w+=\w+)(#\w+)?/i","http://localhost/index.php?name=tank&sex=1#top",$match);

print\_r($match);

/\*

结果如下

Array

(

[0] => Array

(

[0] => name=tank

[1] => sex=1#top

)

[1] => Array

(

[0] => name=tank

[1] => sex=1

)

[2] => Array

(

[0] =>

[1] => #top

)

)

\*/

?>

//------------------------

// http协议分析

$httptop;

if ($\_SERVER['HTTPS'] != "on") {

$httptop = 'http://';

}else{

$httptop = 'https://';

}

echo '<br /><br /><br />';

echo '获取域名或主机地址:<br />';

echo $\_SERVER['HTTP\_HOST']."<br /><br />";

echo '获取网页地址:<br />';

echo $\_SERVER['PHP\_SELF']."<br /><br />";

echo '获取网址参数:<br />';

echo $\_SERVER["QUERY\_STRING"]."<br /><br />";

echo '获取用户代理:<br />';

echo $\_SERVER['HTTP\_REFERER']."<br /><br />";

echo '获取完整的url:<br />';

echo 'http://'.$\_SERVER['HTTP\_HOST'].$\_SERVER['REQUEST\_URI']."<br />";

echo 'http://'.$\_SERVER['HTTP\_HOST'].$\_SERVER['PHP\_SELF'].'?'.$\_SERVER['QUERY\_STRING']."<br /><br />";

echo '包含端口号的完整url:<br />';

echo 'http://'.$\_SERVER['SERVER\_NAME'].':'.$\_SERVER["SERVER\_PORT"].$\_SERVER["REQUEST\_URI"]."<br /><br />";

echo '不包含端口号的完整url:<br />';

$url='http://'.$\_SERVER['SERVER\_NAME'].$\_SERVER["REQUEST\_URI"]."<br /><br />";

echo dirname($url);