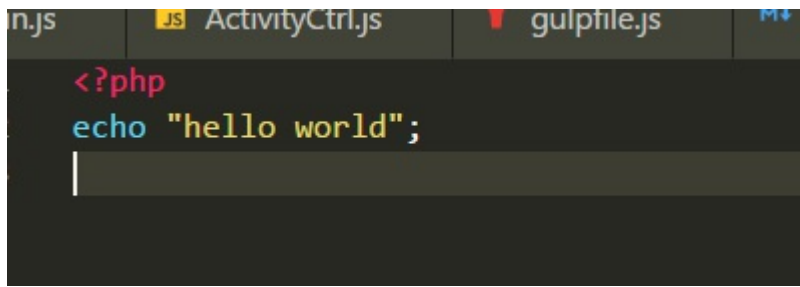


# hello world

A screenshot of a code editor with a dark theme. The top of the editor shows several tabs: 'n.js', 'ActivityCtrl.js', 'gulpfile.js', and a partially visible 'M'. The active tab is 'ActivityCtrl.js'. The code in the editor is: 

```
<?php
echo "hello world";
```

## 语法

### 基本语法

php 文件，以 .php 后缀结尾

以 <?php 开始

以 ?> 结尾（如果末尾 没有其他语言，就是都是php的语法，就可以把 ?> 省略，如果当前文件都是php语法，建议把 ?> 省略）

```
<?php
echo "hello world";
```

### echo

只能输出 字符串 如果是数组的话，在页面上会给你一个 notice

```
Notice: Array to string conversion in E:\FE11\php\demo\05.array.php on line 5
Array
```

只会显示 Array echo 把字符串输出到页面 可以用来调试，必须输出到页面的时候

### 注释

以 // 注释单行 以 /\* \*/ 注释多行

注意：

在php中每行必须以分号";"，结尾，不然就报错;

### 拼接字符串

使用 点 "." 拼接字符串

```
$name = 'stark';  
$headname = 'wang';  
echo $name.$headname; //stark.wang
```

## 变量

以\$符号开头

PHP 中的变量用一个美元符号后面跟变量名来表示。变量名是区分大小写的。变量名与 PHP 中其它的标签一样遵循相同的规则。一个有效的变量名由字母或者下划线开头，后面跟上任意数量的字母，数字，或者下划线。按照正常的正则表达式，它将被表述为：'[a-zA-Z\_\x7f-\xff][a-zA-Z0-9\_\x7f-\xff]\*'。

0. 以\$开头
1. 可以使用字母和下划线开头
2. 区分大小写
3. \$this 是一个特殊的变量，它不能被赋值。

## 常量

语法

可以用 `define()` 函数来定义常量，在 PHP 5.3.0 以后，可以使用 `const` 关键字在类定义之外定义常量。一个常量一旦被定义，就不能再改变或者取消定义。

常量只能包含标量数据（boolean，integer，float 和 string）。可以定义 resource 常量，但应尽量避免，因为会造成不可预料的结果。

可以简单的通过指定其名字来取得常量的值，与变量不同，不应该在常量前面加上 \$ 符号。如果常量名是动态的，也可以用函数 `constant()` 来获取常量的值。用 `get_defined_constants()` 可以获得所有已定义的常量列表。

- 常量和变量有如下不同：
- 常量前面没有美元符号（\$）；
  - 常量只能用 `define()` 函数定义，而不能通过赋值语句；
  - 常量可以不用理会变量的作用域而在任何地方定义和访问；
  - 常量一旦定义就不能被重新定义或者取消定义；
  - 常量的值只能是标量。

## 字符串

单引号

定义一个字符串的最简单的方法是用单引号把它包围起来（字符 '）。

要表达一个单引号自身，需在它的前面加个反斜线（\）来转义。要表达一个反斜线自身，则用两个反斜线（\\）。其它任何方式的反斜线都会被当成反斜线本身：也就是说如果想使用其它转义序列例如 \r 或者 \n，并不代表任何特殊含义，就单纯是这两个字符本身。

双引号

如果字符串是包围在双引号（"）中， PHP 将对一些特殊的字符进行解析：

定界符

以 <<< 开头，后面跟任意大写字母 结束 以后面跟的大写字母结束 必须靠墙边写 贴着边

demo：

```
$fuzu = <<<STARK

.....我佛慈悲.....
*
*          _oo0oo_
*          o8888888o
*          88" . "88
*          (| -_- |)
*          0\  =  /0
*          ___/\___
*          .'\\|      |//'.
*          / \\\||| : |||// \
*          / _||| - 卍 - ||| - \
*          | | \\\ - /// | |
*          | \_| ' '\---/'' |_/ |
*          \ .-\_   '- '   _/-. /
*          ___.'.' /--.--\   .'. ___
*          ."" ' <  \_<|>/_>' ""'.
*          | | : ' - \.;\ _/;.\ / - ' : | |
*          \ \ \_.' \_ \_/_/_/ .-\' / /
*          =====\_._____._____\_____/_____.-\'=====
*          \_.'_____._____._____._____.-\'_
*          \_.'_____._____._____._____.-\'_
*          .....佛祖开光 ,永无BUG.....
*

STARK;

echo $fuzu;
```

数组

Array 数组

PHP 中的数组实际上是一个有序映射。映射是一种把 values 关联到 keys 的类型。此类型在很多方面做了优化，因此可以把它当成真正的数组，或列表（向量），散列表（是映射的一种实现），字典，集合，栈，队列以及更多可能性。由于数组元素的值也可以是另一个数组，树形结构和多维数组也是允许的。

定义数组

定义数组 array()

可以用 `array()` 语言结构来新建一个数组。它接受任意数量用逗号分隔的 键 (key) => 值 (value) 对。 `array(key => value, ...)` // 键 (key) 可是是一个整数 `integer` 或字符串 `string` // 值 (value) 可以是任意类型的值

最后一个数组单元之后的逗号可以省略。通常用于单行数组定义中，例如常用 `array(1, 2)` 而不是 `array(1, 2, )`。对多行数组定义通常保留最后一个逗号，这样要添加一个新单元时更方便。

自 5.4 起可以使用短数组定义语法，用 `[]` 替代 `array()`。

```
$hero = array('法师' => '安琪拉', '刺客' => '李白', '坦克' => '程咬金');

$hero = ['法师'=>'王昭君', '刺客'=>'兰陵王', '坦克' => '张飞'];
```

## 打印 调试 输出 数组

```
加上格式化 输出比较好看
echo '<pre>';
print_r($hero);
var_dump($hero);
```

## 多维数组

```
$hero = [
    '法师' => ['安琪拉', '王昭君', '女娲', '妲己', '貂蝉', ],
    '刺客' => ['兰陵王', '李白', '阿珂', '猴子', '韩信'],
    '坦克' => ['程咬金', '曹操', '梦琪', '吕布', '苏烈']
];
```

## 注意

如果不设置key，则自动转义 数字，如果设置了key，读取的时候，直接写key值 `$hero['法师'];`

## 遍历

```
$hero = ['法师'=>'王昭君', '刺客'=>'兰陵王', '坦克' => '张飞',];
$heros = ['王昭君', '兰陵王', '张飞',];

for ($i=0; $i < count($heros); $i++) {
    # code...
    echo "<li>". $heros[$i]. "</li>";
}
```

```
foreach ($hero as $key => $value) {  
    # code...  
    echo "英雄: ".$value. "---".$key ."<br>";  
}
```

```
$hero = [  
    '法师' => ['安琪拉','王昭君','女娲','妲己','貂蝉'],  
    '刺客' => ['兰陵王','李白','阿珂','猴子','韩信'],  
    '坦克' => ['程咬金','曹操','梦琪','吕布','苏烈']  
];  
  
foreach ($hero as $key => $value) {  
    # code...  
    echo $key ."<br>";  
    foreach ($value as $heroname) {  
        echo "英雄: ".$heroname. "---".$key ."<br>";  
        # code...  
    }  
}
```