

由于 `echo <<< EOF`

`xxx`

`EOF;`

和直接写 `<?php xxx ?>` 外写

`xxx`

的区别只在 `echo` 会解析 `xxx` 中的 php 变量，而后者会将变量名原样输出不解析，所以，当用例如 `if`、`while`、`for` 等语句时，若条件或循环执行的输出中没有需要 `echo` 解析的内容，可以直接用 `?>` 跳出解析，下一个 php

语句再用 `<?php` 跳回，这期间的内
容也归判断管，按其条件选择是否
输出。

如：

```
<?php
```

```
$name = 'xiaotao';
```

```
if (5 > 3) : ?>
```

名字是 `$name` → 不解析。

```
<?php endif ; ?>
```

相当于

```
<?php
```

```
$name = 'xiatao';
```

```
if (5 > 3):
```

```
echo ' $name ' ;
```

→ 被 ' ' 标记为字符串
3, 也不解析.

```
endif ; ? >
```

换种说法, 可以认为 .php 文件中所有
代码都被 php 解析, 只是会在 <?php

xxx ?> 外的内容中默认加上

```
<?php echo ' ' ?> 之后再析
```

解：这样的话，把文件的所有中间
的 `?>` 和 `<?php` 去除再看即能明白其
逻辑。也即所谓跳出，用 `?> xxx`
`<?php` 的所有操作相当于一个 `echo ' ';`

eg. `<?php` `?>`

我是谁

= `<?php`
`echo` `'我是谁'`;

`<?php`
`if (a=b) :` `?>`

我不知道

<?php ~~endif ; ?>~~

=
<?php
if (a=b) ;
echo ~~'我不知道'~~ ;
endif ;

<?php

for (\$i=0; \$i<5; \$i++) { ?>

all 6 times output

<?php } ?>

等效子 <?php

for (\$i=0; \$i<5; \$i++) {

echo 'all 6 times out put'

} ? >