Search

Keyboard Shortcuts

?
: This help

j
: Next menu item

k
: Previous menu item

g p
: Previous man page

g n
: Next man page

G
: Scroll to bottom

g g
: Scroll to top

g h
: Goto homepage

g s
: Goto search
(current page)

/
: Focus search box

Разделение инструкций »
« Теги PHP

- Руководство по PHP
- Справочник языка
- Основы синтаксиса

Change language: Russian

Изолирование от HTML

Все, что находится вне пары открывающегося и закрывающегося тегов, игнорируется интерпретатором PHP, у которого есть возможность обрабатывать файлы со смешанным содержимым. Это позволяет PHP-коду быть встроенным в документы HTML, к примеру, для создания шаблонов.

```
<p>Это будет проигнорировано PHP и отображено браузером.</p>
<?php echo 'А это будет обработано.'; ?>
<p>Это тоже будет проигнорировано PHP и отображено браузером.</p>
```

Это работает так, как и ожидается, потому что когда интерпретатор PHP встречает закрывающие теги ?>, он просто начинает выводить все что найдёт (за исключением сразу следующего символа перевода строки - смотрите раздел разделение инструкций) пока не встретит другой открывающий тег за исключением случая с содержащимся внутри кода условным оператором, в котором интерпретатор определяет результат условия перед принятием решения что пропустить. Ознакомьтесь со следующим примером.

Использование структур с условиями

Пример #1 Продвинутое изолирование с использованием условий

```
<?php if ($expression == true): ?>
Это будет отображено, если выражение истинно.
<?php else: ?>
В ином случае будет отображено это.
<?php endif; ?>
```

В этом примере PHP пропускает блоки, где условие не соблюдается. Даже несмотря на то, что они находятся вне пары открывающих/закрывающих тегов, PHP пропустит их в соответствии с условием, так как интерпретатор PHP будет перепрыгивать через блоки, содержащиеся внутри условия, которое не соблюдается.

При выводе больших блоков текста выход из режима синтаксического разбора PHP обычно более эффективен, чем отправка текста с помощью функций echo или print.

> **Замечание**:
>
> Кроме того, если вы намереваетесь вставлять PHP-код в XML или XHTML, чтобы соответствовать XML стандартам, вам следует использовать форму <?php ?>.

User Contributed Notes 4 notes

385
quickfur at quickfur dot ath dot cx ¶
13 years ago

When the documentation says that the PHP parser ignores everything outside the <?php ... ?> tags, it means literally EVERYTHING. Including things you normally wouldn't consider "valid", such as the following:

```
<html><body>
<p<?php if ($highlight): ?> class="highlight"<?php endif;?>>This is a paragraph.</p>
</body></html>
```

Notice how the PHP code is embedded in the middle of an HTML opening tag. The PHP parser doesn't care that it's in the middle of an opening tag, and doesn't require that it be closed. It also doesn't care that after the closing ?> tag is the end of the HTML opening tag. So, if $highlight is true, then the output will be:

```
<html><body>
<p class="highlight">This is a paragraph.</p>
</body></html>
```

Otherwise, it will be:

```
<html><body>
<p>This is a paragraph.</p>
</body></html>

Using this method, you can have HTML tags with optional attributes, depending on some PHP condition. Extremely flexible and useful!
```

up
down
77

ravenswd at gmail dot com ¶

14 years ago

```
One aspect of PHP that you need to be careful of, is that ?> will drop you out of PHP code and into HTML even if it appears inside a // comment. (This does not apply to /* */ comments.) This can lead to unexpected results. For example, take this line:

<?php
$file_contents = '<?php die(); ?>' . "\n";
?>

If you try to remove it by turning it into a comment, you get this:

<?php
// $file_contents = '<?php die(); ?>' . "\n";
?>

Which results in ' . "\n"; (and whatever is in the lines following it) to be output to your HTML page.

The cure is to either comment it out using /* */ tags, or re-write the line as:

<?php
$file_contents = '<' . '?php die(); ?' . '>' . "\n";
?>
```

up
down
13

sgurukrupa at gmail dot com ¶

9 years ago

```
Although not specifically pointed out in the main text, escaping from HTML also applies to other control statements:

<?php for ($i = 0; $i < 5; ++$i): ?>
Hello, there!
<?php endfor; ?>

When the above code snippet is executed we get the following output:

Hello, there!
Hello, there!
Hello, there!
Hello, there!
```

up
down
0

gazianis005 at gmail dot com ¶

10 months ago

```
There are two types of escaping from HTML.
1.Normal escaping of using outside of a pair of opening and closing tag
2.Advance escaping of using conditions.

Example of normal escaping

<p>This is going to be ignored by the php parser and displayed by the browser</p>
<?php echo 'This is going to be parsed';?>
```

```
<>This will also ignored by the php parser and displayed by the browser</>

Example of advanced escaping

<?php if($expression == true): ?>
This will show if expression is true
<?php else : ?>
Otherwise this will show
<?php endif ;?>
```