php

- Downloads
- Documentation
- Get Involved
- Help

Search

PHP 7.3.0.beta3 Released

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

Basics »
« Type Juggling

- PHP Manual
- Language Reference

Change language: English

Edit Report a Bug

Variables ¶

Table of Contents ¶

add a note

User Contributed Notes 13 notes

up
down
-13
jsb17 at cornell dot edu ¶
11 years ago

```
As an addendum to David's 10-Nov-2005 posting, remember that curly braces literally mean "evaluate
what's inside the curly braces" so, you can squeeze the variable variable creation into one line,
like this:

<?php
  ${"title_default_" . $title} = "selected";
?>

and then, for example:

<?php
  $title_select = <<<END
    <select name="title">
      <option>Select</option>
      <option $title_default_Mr  value="Mr">Mr</option>
      <option $title_default_Ms  value="Ms">Ms</option>
```

```
      <option $title_default_Mrs value="Mrs">Mrs</option>
      <option $title_default_Dr  value="Dr">Dr</option>
    </select>
END;
?>
```

up
down
-21
josh at PraxisStudios dot com ¶
13 years ago

As with echo, you can define a variable like this:

```
<?php

$text = <<<END

<table>
    <tr>
        <td>
             $outputdata
        </td>
     </tr>
</table>

END;

?>
```

The closing END; must be on a line by itself (no whitespace).

[EDIT by danbrown AT php DOT net: This note illustrates HEREDOC syntax. For more information on this and similar features, please read the "Strings" section of the manual here: http://www.php.net/manual/en/language.types.string.php]

up
down
-23
Mike at ImmortalSoFar dot com ¶
12 years ago

References and "return" can be flakey:

```
<?php
//  This only returns a copy, despite the dereferencing in the function definition
function &GetLogin ()
{
    return $_SESSION['Login'];
}

//  This gives a syntax error
function &GetLogin ()
{
    return &$_SESSION['Login'];
}

//  This works
```

```
function &GetLogin ()
{
    $ret = &$_SESSION['Login'];
    return $ret;
}
?>
```

up
down
-26
Anonymous ¶
10 years ago

[EDIT by danbrown AT php DOT net: The function provided by this author will give you all defined variables at runtime. It was originally written by (john DOT t DOT gold AT gmail DOT com), but contained some errors that were corrected in subsequent posts by (ned AT wgtech DOT com) and (taliesin AT gmail DOT com).]

```
<?php

echo '<table border=1><tr> <th>variable</th> <th>value</th> </tr>';
foreach( get_defined_vars() as $key => $value)
{
    if (is_array ($value) )
    {
        echo '<tr><td>$'.$key .'</td><td>';
        if ( sizeof($value)>0 )
        {
        echo '"<table border=1><tr> <th>key</th> <th>value</th> </tr>';
        foreach ($value as $skey => $svalue)
        {
            echo '<tr><td>[' . $skey .']</td><td>"'. $svalue .'"</td></tr>';
        }
        echo '</table>"';
        }
             else
        {
            echo 'EMPTY';
        }
        echo '</td></tr>';
    }
    else
    {
            echo '<tr><td>$' . $key .'</td><td>"'. $value .'"</td></tr>';
    }
}
echo '</table>';
?>
```

up
down
-29
Kubo2 ¶
3 years ago

Note that if you use runtime variable name recognition in your code, you are able to use any string as a variable name. Consider following code:

```
<?php

$varName = 'foo with bar';
${'foo with bar'} = 42;

// will output int(42)
var_dump($$varName);

?>
```

This can be useful for example when accessing a property of an object constructed from JSON:

```
<?php

$composerJson = json_decode(file_get_contents(__DIR__ . '/composer.json'));

// would output sth. similar to: object(stdClass)#...
var_dump(
    $composerJson->{'require-dev'}
);

?>
```

up
down
-29

Chris Hester ¶

13 years ago

Variables can also be assigned together.

```
<?php
$a = $b = $c = 1;
echo $a.$b.$c;
?>
```

This outputs 111.

up
down
-29

webmaster at daersys dot net ¶

14 years ago

You don't necessarily have to escape the dollar-sign before a variable if you want to output its name.

You can use single quotes instead of double quotes, too.

For instance:

```
<?php
$var = "test";

echo "$var"; // Will output the string "test"

echo "\$var"; // Will output the string "$var"
```

```
echo '$var'; // Will do the exact same thing as the previous line
?>

Why?
Well, the reason for this is that the PHP Parser will not attempt to parse strings encapsulated in
single quotes (as opposed to strings within double quotes) and therefore outputs exactly what it's
being fed with :)

To output the value of a variable within a single-quote-encapsulated string you'll have to use
something along the lines of the following code:

<?php
$var = 'test';
/*
Using single quotes here seeing as I don't need the parser to actually parse the content of this
variable but merely treat it as an ordinary string
*/

echo '$var = "' . $var . '"';
/*
Will output:
$var = "test"
*/
?>

HTH
- Daerion
```

up
down
-11
info at learnPHPonline dot in ¶
6 months ago

```
at php.net very difficult to learn PHP . I have better option for you

go w3schools.com for beginners

www.learnPHPonline.in for beginners to advance
```

up
down
-30
raja shahed at christine nothdurfter dot com ¶
14 years ago

```
<?php
error_reporting(E_ALL);

$name = "Christine_Nothdurfter";
// not Christine Nothdurfter
// you are not allowed to leave a space inside a variable name ;)
$$name = "'s students of Tyrolean language ";

print " $name{$$name}<br>";
print  "$name$Christine_Nothdurfter";
// same
?>
```

up
down
-30
dimitrov dot adrian at gmail dot com ¶
8 years ago

This is mine type casting lib, that is very useful for me.

```
<?php

function CAST_TO_INT($var, $min = FALSE, $max = FALSE)
{
    $var = is_int($var) ? $var : (int)(is_scalar($var) ? $var : 0);
    if ($min !== FALSE && $var < $min)
        return $min;
    elseif($max !== FALSE && $var > $max)
        return $max;
    return $var;

}

function CAST_TO_FLOAT($var, $min = FALSE, $max = FALSE)
{
    $var = is_float($var) ? $var : (float)(is_scalar($var) ? $var : 0);
    if ($min !== FALSE && $var < $min)
        return $min;
    elseif($max !== FALSE && $var > $max)
        return $max;
    return $var;
}

function CAST_TO_BOOL($var)
{
    return (bool)(is_bool($var) ? $var : is_scalar($var) ? $var : FALSE);
}

function CAST_TO_STRING($var, $length = FALSE)
{
    if ($length !== FALSE && is_int($length) && $length > 0)
        return substr(trim(is_string($var)
                    ? $var
                    : (is_scalar($var) ? $var : '')), 0, $length);

    return trim(
                is_string($var)
                ? $var
                : (is_scalar($var) ? $var : ''));
}

function CAST_TO_ARRAY($var)
{
    return is_array($var)
            ? $var
            : is_scalar($var) && $var
                ? array($var)
```

```
                : is_object($var) ? (array)$var : array();
}

function CAST_TO_OBJECT($var)
{
    return is_object($var)
            ? $var
            : is_scalar($var) && $var
                ? (object)$var
                : is_array($var) ? (object)$var : (object)NULL;
}

?>
```

up
down
-33
justgook at gmail dot com ¶
8 years ago

I found interstate solution to work with arrays

```
<?php
$vars['product']['price']=11;

$aa='product';
$bb='price';

echo $vars{$aa}{$bb};

//prints 11
?>
```

up
down
-31
Carel Solomon ¶
13 years ago

You can also construct a variable name by concatenating two different variables, such as:

```
<?php

$arg = "foo";
$val = "bar";

//${$arg$val} = "in valid";     // Invalid
${$arg . $val} = "working";

echo $foobar;     // "working";
//echo $arg$val;         // Invalid
//echo ${$arg$val};     // Invalid
echo ${$arg . $val};    // "working"

?>
```

Carel

up

down
-47
david at removethisbit dot futuresbright dot com ¶
12 years ago

```
When using variable variables this is invalid:

<?php
$my_variable_{$type}_name = true;
?>

to get around this do something like:

<?php
$n="my_variable_{$type}_name";
${$n} = true;
?>

(or $$n - I tend to use curly brackets out of habit as it helps t reduce bugs ...)
```

add a note