

## Character type checking

# Introduction

The functions provided by this extension check whether a character or string falls into a certain character class according to the current locale (see also [setlocale\(\)](#) ).

When called with an integer argument these functions behave exactly like their C counterparts from *ctype.h*. It means that if you pass an integer smaller than 256 it will use the ASCII value of it to see if it fits in the specified range (digits are in 0x30-0x39). If the number is between -128 and -1 inclusive then 256 will be added and the check will be done on that.

When called with a string argument they will check every character in the string and will only return **TRUE** if every character in the string matches the requested criteria. When called with an empty string the result will always be **TRUE** in PHP < 5.1 and **FALSE** since 5.1.

Passing anything else but a string or integer will return **FALSE** immediately.

It should be noted that ctype functions are always preferred over regular expressions, and even to some equivalent `str_*` and `is_*` functions. This is because of the fact that ctype uses a native C library and thus processes significantly faster.

# Installing/Configuring

## Requirements

None besides functions from the standard C library which are always available.

## Installation

Beginning with PHP 4.2.0 these functions are enabled by default. For older versions you have to configure and compile PHP with `--enable-ctype`. You can disable ctype support with `--disable-ctype`.

The Windows version of PHP has built-in support for this extension. You do not need to load any additional extensions in order to use these functions.

<b>Note</b>
Builtin support for ctype is available with PHP 4.3.0.

## Runtime Configuration

This extension has no configuration directives defined in *php.ini*.

## Resource Types

This extension has no resource types defined.

# Predefined Constants

This extension has no constants defined.

# Ctype Functions

# ctype\_alnum

ctype\_alnum -- Check for alphanumeric character(s)

## Description

bool **ctype\_alnum** ( string *\$text* )

Checks if all of the characters in the provided string, *text*, are alphanumeric. In the standard C locale letters are just *[A-Za-z]*.

## Parameters

*text*  
The tested string.

## Return Values

Returns **TRUE** if every character in *text* is either a letter or a digit, **FALSE** otherwise.

## Examples

### Example #1 - A [ctype\\_alnum\(\)](#) example (using the default locale)

```
<?php
$strings = array('AbCd1zyZ9', 'foo!#$bar');
foreach ($strings as $testcase) {
    if (ctype_alnum($testcase)) {
        echo "The string $testcase consists of all letters or digits.\n";
    } else {
        echo "The string $testcase does not consist of all letters or
digits.\n";
    }
}
?>
```

The above example will output:

```
The string AbCd1zyZ9 consists of all letters or digits.
The string foo!#$bar does not consist of all letters or digits.
```

## See Also

- `ctype_alpha()`
- `ctype_digit()`
- `setlocale()`

# ctype\_alpha

ctype\_alpha -- Check for alphabetic character(s)

## Description

bool **ctype\_alpha** ( string \$text )

Checks if all of the characters in the provided string, *text*, are alphabetic. In the standard C locale letters are just *[A-Za-z]* and [ctype\\_alpha\(\)](#) is equivalent to *(ctype\_upper(\$text) || ctype\_lower(\$text))* if *\$text* is just a single character, but other languages have letters that are considered neither upper nor lower case.

## Parameters

*text*

The tested string.

## Return Values

Returns **TRUE** if every character in *text* is a letter from the current locale, **FALSE** otherwise.

## Examples

### Example #2 - A [ctype\\_alpha\(\)](#) example (using the default locale)

```
<?php
$strings = array('KjgWZC', 'arf12');
foreach ($strings as $testcase) {
    if (ctype_alpha($testcase)) {
        echo "The string $testcase consists of all letters.\n";
    } else {
        echo "The string $testcase does not consist of all letters.\n";
    }
}
?>
```

The above example will output:

```
The string KjgWZC consists of all letters.
The string arf12 does not consist of all letters.
```

## See Also



- `ctype_upper()`
- `ctype_lower()`
- `setlocale()`

# ctype\_cntrl

ctype\_cntrl -- Check for control character(s)

## Description

bool **ctype\_cntrl** ( string *\$text* )

Checks if all of the characters in the provided string, *text*, are control characters. Control characters are e.g. line feed, tab, escape.

## Parameters

*text*  
The tested string.

## Return Values

Returns **TRUE** if every character in *text* is a control character from the current locale, **FALSE** otherwise.

## Examples

### Example #3 - A [ctype\\_cntrl\(\)](#) example

```
<?php
$strings = array('string1' => "\n\r\t", 'string2' => 'arfl2');
foreach ($strings as $name => $testcase) {
    if (ctype_cntrl($testcase)) {
        echo "The string '$name' consists of all control characters.\n";
    } else {
        echo "The string '$name' does not consist of all control
characters.\n";
    }
}
?>
```

The above example will output:

```
The string 'string1' consists of all control characters.
The string 'string2' does not consist of all control characters.
```

## See Also

- `ctype_print()`

# ctype\_digit

ctype\_digit -- Check for numeric character(s)

## Description

bool **ctype\_digit** ( string *\$text* )

Checks if all of the characters in the provided string, *text*, are numerical.

## Parameters

*text*

The tested string.

## Return Values

Returns **TRUE** if every character in *text* is a decimal digit, **FALSE** otherwise.

## Examples

### Example #4 - A [ctype\\_digit\(\)](#) example

```
<?php
$strings = array('1820.20', '10002', 'wsl!12');
foreach ($strings as $testcase) {
    if (ctype_digit($testcase)) {
        echo "The string $testcase consists of all digits.\n";
    } else {
        echo "The string $testcase does not consist of all digits.\n";
    }
}
?>
```

The above example will output:

```
The string 1820.20 does not consist of all digits.
The string 10002 consists of all digits.
The string wsl!12 does not consist of all digits.
```

## See Also

- [ctype\\_alnum\(\)](#)
- [ctype\\_xdigit\(\)](#)

# ctype\_graph

ctype\_graph -- Check for any printable character(s) except space

## Description

bool **ctype\_graph** ( string \$text )

Checks if all of the characters in the provided string, *text*, creates visible output.

## Parameters

*text*

The tested string.

## Return Values

Returns **TRUE** if every character in *text* is printable and actually creates visible output (no white space), **FALSE** otherwise.

## Examples

### Example #5 - A [ctype\\_graph\(\)](#) example

```
<?php
$strings = array('string1' => "asdf\n\r\t", 'string2' => 'arfl2', 'string3'
=> 'LKA#@%.54');
foreach ($strings as $name => $testcase) {
    if (ctype_graph($testcase)) {
        echo "The string '$name' consists of all (visibly) printable
characters.\n";
    } else {
        echo "The string '$name' does not consist of all (visibly) printable
characters.\n";
    }
}
?>
```

The above example will output:

```
The string 'string1' does not consist of all (visibly) printable characters.
The string 'string2' consists of all (visibly) printable characters.
The string 'string3' consists of all (visibly) printable characters.
```

## See Also

- [`ctype\_alnum\(\)`](#)
- [`ctype\_print\(\)`](#)
- [`ctype\_punct\(\)`](#)

# ctype\_lower

ctype\_lower -- Check for lowercase character(s)

## Description

bool **ctype\_lower** ( string *\$text* )

Checks if all of the characters in the provided string, *text*, are lowercase letters.

## Parameters

*text*

The tested string.

## Return Values

Returns **TRUE** if every character in *text* is a lowercase letter in the current locale.

## Examples

### Example #6 - A [ctype\\_lower\(\)](#) example (using the default locale)

```
<?php
$strings = array('aac123', 'qiutoas', 'QASsdks');
foreach ($strings as $testcase) {
    if (ctype_lower($testcase)) {
        echo "The string $testcase consists of all lowercase letters.\n";
    } else {
        echo "The string $testcase does not consist of all lowercase
letters.\n";
    }
}
?>
```

The above example will output:

```
The string aac123 does not consist of all lowercase letters.
The string qiutoas consists of all lowercase letters.
The string QASsdks does not consist of all lowercase letters.
```

## See Also



- `ctype_alpha()`
- `ctype_upper()`
- `setlocale()`

# ctype\_print

ctype\_print -- Check for printable character(s)

## Description

bool **ctype\_print** ( string *\$text* )

Checks if all of the characters in the provided string, *text*, are printable.

## Parameters

*text*  
The tested string.

## Return Values

Returns **TRUE** if every character in *text* will actually create output (including blanks).  
Returns **FALSE** if *text* contains control characters or characters that do not have any output or control function at all.

## Examples

### Example #7 - A [ctype\\_print\(\)](#) example

```
<?php
$strings = array('string1' => "asdf\n\r\t", 'string2' => 'arf12', 'string3'
=> 'LKA#@%.54');
foreach ($strings as $name => $testcase) {
    if (ctype_print($testcase)) {
        echo "The string '$name' consists of all printable characters.\n";
    } else {
        echo "The string '$name' does not consist of all printable
characters.\n";
    }
}
?>
```

The above example will output:

```
The string 'string1' does not consist of all printable characters.
The string 'string2' consists of all printable characters.
The string 'string3' consists of all printable characters.
```

## See Also

- [ctype\\_cntrl\(\)](#)
- [ctype\\_graph\(\)](#)
- [ctype\\_punct\(\)](#)

# ctype\_punct

ctype\_punct -- Check for any printable character which is not whitespace or an alphanumeric character

## Description

bool **ctype\_punct** ( string *\$text* )

Checks if all of the characters in the provided string, *text*, are punctuation character.

## Parameters

*text*  
The tested string.

## Return Values

Returns **TRUE** if every character in *text* is printable, but neither letter, digit or blank, **FALSE** otherwise.

## Examples

### Example #8 - A [ctype\\_punct\(\)](#) example

```
<?php
$strings = array('ABasdk!@$#', '!@ # $', '*&$()');
foreach ($strings as $testcase) {
    if (ctype_punct($testcase)) {
        echo "The string $testcase consists of all punctuation.\n";
    } else {
        echo "The string $testcase does not consist of all punctuation.\n";
    }
}
?>
```

The above example will output:

```
The string ABasdk!@$# does not consist of all punctuation.
The string !@ # $ does not consist of all punctuation.
The string *&$() consists of all punctuation.
```

## See Also

- [ctype\\_cntrl\(\)](#)
- [ctype\\_graph\(\)](#)

# ctype\_space

ctype\_space -- Check for whitespace character(s)

## Description

bool **ctype\_space** ( string \$text )

Checks if all of the characters in the provided string, *text*, creates whitespace.

## Parameters

*text*

The tested string.

## Return Values

Returns **TRUE** if every character in *text* creates some sort of white space, **FALSE** otherwise. Besides the blank character this also includes tab, vertical tab, line feed, carriage return and form feed characters.

## Examples

### Example #9 - A [ctype\\_space\(\)](#) example

```
<?php
$strings = array('string1' => "\n\r\t", 'string2' => "\narfl2", 'string3' =>
'\n\r\t');
foreach ($strings as $name => $testcase) {
    if (ctype_space($testcase)) {
        echo "The string '$name' consists of all whitespace characters.\n";
    } else {
        echo "The string '$name' does not consist of all whitespace
characters.\n";
    }
}
?>
```

The above example will output:

```
The string 'string1' consists of all whitespace characters.
The string 'string2' does not consist of all whitespace characters.
The string 'string3' does not consist of all whitespace characters.
```

## See Also

- [ctype\\_cntrl\(\)](#)
- [ctype\\_graph\(\)](#)
- [ctype\\_punct\(\)](#)

# ctype\_upper

ctype\_upper -- Check for uppercase character(s)

## Description

bool **ctype\_upper** ( string *\$text* )

Checks if all of the characters in the provided string, *text*, are uppercase characters.

## Parameters

*text*

The tested string.

## Return Values

Returns **TRUE** if every character in *text* is an uppercase letter in the current locale.

## Examples

### Example #10 - A [ctype\\_upper\(\)](#) example (using the default locale)

```
<?php
$strings = array('AKLWC139', 'LMNSDO', 'akwSKWsm');
foreach ($strings as $testcase) {
    if (ctype_upper($testcase)) {
        echo "The string $testcase consists of all uppercase letters.\n";
    } else {
        echo "The string $testcase does not consist of all uppercase
letters.\n";
    }
}
?>
```

The above example will output:

```
The string AKLWC139 does not consist of all uppercase letters.
The string LMNSDO consists of all uppercase letters.
The string akwSKWsm does not consist of all uppercase letters.
```

## See Also



- `ctype_alpha()`
- `ctype_lower()`
- `setlocale()`

# ctype\_xdigit

ctype\_xdigit -- Check for character(s) representing a hexadecimal digit

## Description

bool **ctype\_xdigit** ( string *\$text* )

Checks if all of the characters in the provided string, *text*, are hexadecimal 'digits'.

## Parameters

*text*

The tested string.

## Return Values

Returns **TRUE** if every character in *text* is a hexadecimal 'digit', that is a decimal digit or a character from *[A-Fa-f]*, **FALSE** otherwise.

## Examples

### Example #11 - A [ctype\\_xdigit\(\)](#) example

```
<?php
$strings = array('AB10BC99', 'AR1012', 'abl2bc99');
foreach ($strings as $testcase) {
    if (ctype_xdigit($testcase)) {
        echo "The string $testcase consists of all hexadecimal digits.\n";
    } else {
        echo "The string $testcase does not consist of all hexadecimal
digits.\n";
    }
}
?>
```

The above example will output:

```
The string AB10BC99 consists of all hexadecimal digits.
The string AR1012 does not consist of all hexadecimal digits.
The string abl2bc99 consists of all hexadecimal digits.
```

## See Also

- `ctype_digit()`