isset

WMPRO, WMMINI FW >= 1.0 WMMEGA FW >= 2.0

Check if a variable exists

Description

int isset (mixed $variable)

Determine if a variable is set

Parameter

$variable: Variable to check for existence[1)]

Return Values

Integer: 1 (true) if variable exists, 0 (false) if variable does not exist

Examples

```
<?
  print(isset($x)); // outputs 0, the variable doesn't exist yet
  $x=0;
  print(isset($x)); // outputs 1

  $y='';
  print(isset($y)); // outputs 1, $y exists (although it is an empty string)

  $z=chr(0);
  print(isset($z)); // outputs 1, $z exists with a single character (ASCII
code 0 or NUL)

  $array=array();
  print(isset($array)); // outputs 1, $array exists (an empty array)

  $array[1]="test";
  print(isset($array[0])); // outputs 0, the first element in the array does
not exist
  print(isset($array[1])); // outputs 1, the second element in the array
exists
```

```
?>
```

Notes

There is no way to unset a variable once it has been set (except to exit the local scope or script where it was defined). Unlike mainline PHP, uPHP does not have a construct `unset()` or the constant `null`. Unlike many other programming languages, you cannot simply remove a variable by setting it equal to nothing, such as by trying the statement `$x=;`[2)].

Also note that the uPHP isset() does not always generate a parsing error if the parameter is not a variable[3)]. All of these examples will return 0:

```
<?
  print(isset(1));
  print(isset("123"));
  print(isset(array(1,2,3)));
  print(isset(2+3));
?>
```

Although maybe useful in some situations, it is better not to use isset() for expressions or values instead of variables, because in some cases it *will* generate a parsing error. Example:

```
<?
  print(isset(z));
?>
```

For the above example the parser answers:

```
Error on line 1: Undefined constant z
Error on line 1: Invalid function
```

Also note that in many situations where using if to test for the existence of a variable it is possible to simplify the conditional test by using `if($x)` instead of `if(isset($x))` but only if you are certain that `$x` does not contain the value 0 (false). Unlike mainline PHP, uPHP does not generate an error for using `if($x)` on an undefined variable. Nevertheless, this is not a good practice even though this hack *is* commonly used *somewhat successfully* by Wattmon uPHP programmers. It is much better to use isset() if you are not certain that the variable has been initialized, and the code will be easier to understand later on.

See Also

is_int() - Check if a variable is an **integer**

is_float() - Check if a variable is a **float**

is_numeric() - Check if a value is numeric (**int**, **float** or numeric **string**)

is_array() - Check if a variable is an **array**

is_string() - Check if a variable is a **string**

uPHP Variable Types and Limits

1)
Multiple parameters are not supported as in mainline PHP
2)
As tested on a WattmonPRO with firmware 1.1051 the statement `$x=;` causes a Watchdog Timeout error and a reboot of the Wattmon
3)
See http://php.net/manual/en/function.isset.php where it says “Warning: isset() only works with variables as passing anything else will result in a parse error.”

From:
http://wattmon.com/dokuwiki/ - **Wattmon Documentation Wiki**

Permanent link:
http://wattmon.com/dokuwiki/uphp/functions/isset

Last update: **2021/09/13 05:57**