# PHPDebugr

This is a debugger for inspecting values of variables.

## usage

Debugr::eDbg(mixed *$var* [, string *$description* [, string $writeOption]])

Debugr::eDbgScreen(mixed *$var* [, string *$description* [, string $writeOption]])

Debugr::eDbgLog(mixed *$var* [, string *$description* [, string $writeOption]])

Debugr::eDbgConsole(mixed *$var* [, string *$description* [, string $writeOption]])

Debugr::eDbg - writes the value of $var and the description text (optional) to the default output defined in config.php using the $writeOption.  
Possible options for the default output are: Screen, Log, Console, and None.   
(The predefined value is Screen)

Debugr::eDbgScreen - writes the value of $var to the screen regardless of the default config.php file entry

Debugr::eDbgLog - writes the value of $var to the log file defined in config.php

Debugr::eDbgConsole writes the value of $var to the browsers console/Firebug (careful with that on live servers)

**Var**

The variable to inspect

Description

(optional)

Text to be displayed before the variable value e.g. ´The value of $thisVar is:

writeOption

(optional)

you can chose the way the output is written/formatted ??

options:

'e ' or 'echos '

'v ' or 'varDump '

'r ' or 'printR '

'c ' or 'custom ' – not implemented (you can write you own)

If you omit this the defaults are used. For scalar types (integer, double, string) the default is echos and for composite types (array, object, resource, boolean, null, unknown type) the default is varDump.

(I know boolean is technically scalar and Null is… well, Null but they are fitting better in the composite group)

## Notes

If 'None' is used as the default output, Debugr::eDbg will not produce any output. This is not true for eDbgScreen, eDbgLog, eDbgConsole.

You can disable all by settting: ´disable:true´ in config.php. This is some sort of kill switch.

## how to use

Use eDbg to output the variable values to the default output. On a developer server you can choose screen, and then change it to log for the live server. All Debugr::eDbg calls will now write to the log file instead of the screen.

On some occasions you want a different output then the default e.g., when outputting a specific variable on screen breaks the site layout. In this case you can use eDbgConsole to use the browser console regardless of the default output. In the same way you can use eDbgScreen, eDbgLog, according to the situation and your needs.

The value is formatted according to the variables type or the writeOption given.

## examples

## Licence

## Requirements