



## Overview

---

When processing a page, your code can send output to the debug console with commands such as `$debugger->msg("your message")`. This writes the output to the debugconsole which you can then view by displaying the console.

When executing debugging commands, the page is re-loaded with the output of the debugging commands written to the debugconsole. This may pose problems for pages which are processing user input, uploading files, or that crash and do not return.

The debug console is embedded within a div `class="debugconsole" id="debugconsole"` which is turned on by clicking on the link to toggle the console. Careful inspection will note that the footer.tpl includes `tiki-debug_console.php` which handles the processing after most of the rest of the page has already executed.

## General Notes

---

- Enable debugger console by selecting checkbox on Admin -> Features -> Administration Features.
- Debugger can be extended with commands and/or interface extensions (tabs).
- Every command in debugger is a PHP class.
- Sources `lib/debug`
- Search for `debug-command_*.php` it is filenames mask... for debugger commands
- one command == one class == one file

## Key Function and sub-features

---

# General info

```
*help    Display list of commands or help for specified command (help print for example)
*tikitable Show list of Tiki tables in DB schema
*features Show features on/off state
*perm    Show current permissions in a convenient way
*print   Print PHP variable. Indexes are OK.
*slist   Display list of Smarty variables. Better to specify partial name or
         very long list of vars will return.
*sprint  Print Smarty variable
*watch   Manage variables watch list
*sql     Exec SQL query on Tiki DB
```

# How to extend debugger

Look for example skeleton `lib/debug/debugger-command_test.php`

## Related Links

---

## Typical Uses

---

# Debug messages on console

It is possible to print debug info to special debug tab called 'Debug messages' during page generation. It is possible from PHP and Smarty as well.

```
Dump Smarty variable from template:  
{var_dump var=left_modules}
```

```
Dump Smarty variable from template using modifier:  
{$description|str_replace:"\n":"<br />"|dbg}
```

Of course modifier `dbg` can be used more than once:

```
{$description|dbg|str_replace:"\n":"<br />"|dbg}
```

... so you will get 2 messages before and after modification of `$description`.

```
// PHP code  
global $debugger;  
// Print message  
$debugger->msg('Value of $myvar = "'. $myvar. "'');  
// Print variable  
$debugger->var_dump('$myvar');
```

## Case Studies

---

### Bugs

---

Non command-line commands should not print help. — Fixed in CVS.

### Support Requests

---

We need more examples of the commands that are being passed here...

### TikiTeam

---

zaufi

### For more information

---

lookup sources lib/debug

## Related links

- [Debugging tools deployable within Tiki](#)

### alias

---

- [TikiDebuggerConsole](#)