Dokeos 1.8.5 Tools & Plugins Quick Reference Guide

Creating a new tool inside a Dokeos course

1. Add a toolname constant in main/inc/lib/main_api.lib.php
2. Create the tool directory in main/ (e.g. main/blog/)
3. Create the index.php file inside that directory

```
<?php
/* License terms */
/**
 * Description of the script
 */
$language_files = array('glossary');
require_once('../inc/global.inc.php');
$tool = TOOL_GLOSSARY;
$abc = 5 + 4; //code logic
Display::display_header($tool);
echo $abc; //display code
Display::display_footer();
```

4. Add the SQL queries for your tables in
   main/inc/lib/add_course.lib.inc.php
   (end of update_Db_course())
   and in
   main/install/migrate-db-x.y.z-x.y.z2-pre.sql
   (end of the xxCOURSExx section, generally)
   Also add the corresponding queries to insert into the
   dokeos_main.settings_current table
   (see course_create_active_tools),
   the course_module table and the course's tool table
5. Create the main/lang/[language]/newtool.inc.php language file for your tool, and add its name to the generic trad4all.inc.php file (then load it, see section 3)
6. Add an icon in main/img/ to represent your tool
7. Add some help to your tool, declaring the $help variable inside your index.php and defining it in main/lang/[language]/help.inc.php
8. Update the CSS styles in main/css/[styles]/default.css according to your needs, something like

```
.newtool-title { font-decoration: underline; }
```

Plugin development

1. A plugin is an external piece of code and can be integrated easily into Dokeos, without modifying its code
2. Prepare your SQL statements to create tables inside a separate SQL file (do not conditionally create tables in your code, checking the existence of a table kills efficiency). E.g. myplugin/install/myplugin.sql
3. Prepare your code, taking as an example the "date" plugin, shipped inside the default Dokeos (copy dir)

```
cp -r plugin/date/ plugin/myplugin/
cd plugin/myplugin/
```

4. Update plugin.php to describe your new plugin.
   $plugin_info['location'] should contain an array
   of the possible loccations of the plugin's appearance,
   which can be any of: loginpage_main, loginpage_menu,
   campushomepage_main, campushomepage_menu,
   mycourses_main, mycourses_menu, header and/or footer

5. Update index.act.php to set your code logic

```
<?php
require_once('../inc/global.inc.php');
$tool = 'My plugin';
$abc = 5 + 4; //code logic
```

6. Update index.dsp.php to set your code presentation

```
<?php
Display::display_header($tool);
echo $abc; //display code
Display::display_footer();
```

7. Try your plugin (not only on *your* portal)
8. Document your plugin (INSTALL.txt, README.txt, LICENSE.txt)
8. If you want it published by us, make it clear it is under GPL2 or + (requirement) inside the documentation and send it to extensions@dokeos.com