Smarty Cheat Sheet for Template Designers

Initializing Smarty

include("smarty/Smarty.class.php")
\$smarty = new smarty();

Assigning Variables



General Format
\$smarty->assign("var","value");
Example
\$smarty->assign("who", "world");

Sample Template



Filename : templates\greet.tpl
Hello {\$who}

Display Output (?



\$smarty->display("greet.tpl");

Debugging templates [

Add this line at the top {debug}

Capturing output in php variable 📳



Use Fetch() function
\$output =
\$smarty->fetch("greet.tpl");
echo \$output;

Processing indexed array



{section name=id loop=\$array}
Current Item : {\$array[id]}
{/section}

Processing associated array



{student.name}
{student.roll}
{\$student.age}

Passing associated array



\$student=array("name"=>"shumi");
\$smarty->assign("student",\$std);



{section name=id loop=\$variable}
element:{\$variable[id]}
{/section}

{foreach item=curltem from=\$items}
element : {\$curltem}
{/foreach}

Logic

{if \$var == condition}
 something to do
{elseif \$var == condition}
 something to do
{else}
 something else to do
{/if}

\$smarty.version

\$smarty.ldelim

\$smarty.rdelim

Common Modifiers

capitalize, count_characters, cat, count_paragraphs, count_sentences, count_words, date_format, default, escape, indent, lower, nl2br, regex_replace, replace, spacify, string_format, strip, strip_tags, truncate, upper, wordwrap



Builtin Functions

capture, config_load, foreach, foreachelse, include, include_php, insert if, elseif, else, Idelim, rdelim, literal, php, section, sectionelse, strip



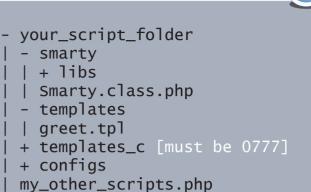
Custom Functions

assign, counter, cycle, debug, eval, fetch, html_checkboxes, html_image html_options, html_radios, html_select_date, html_select_time html_table, math, mailto, popup_init, popup, textformat



Smarty reserved variables

\$smarty.get \$smarty.now \$smarty.post \$smarty.const \$smarty.session \$smarty.capture \$smarty.session \$smarty.config \$smarty.env \$smarty.section \$smarty.request \$smarty.foreach \$smarty.env \$smarty.template **Directory Structure**



Processing objects



```
Access Object Methods {object->method
```

blah_blah_script.php

{object->method
 param1="val"
 param2="val"}

Accesing Object Properties {object->property}

Assign method output to variable {object->method param1="val" param2="val"} assign="storage_var"}

Output: {\$storage_var}

Passing objects to template



Using modifiers



General Format

{\$variable | modifier:parameter}
Example
{\$name|count_characters}

Execute PHP inside template



