

# Apex Plugins Cheatsheet

## apex\_plugin: functions

Returns the name attribute which has to be used for a HTML input element if you want that the value of the element is stored in session state when the page is submitted. If you have a HTML input element which returns multiple values (eg. select list with multiple="multiple") you have to set p\_is\_multi\_value. Note: This function has to be called before you write something to the HTTP buffer with HTP.P(RN)

```
function get_input_name_for_page_item (
    p_is_multi_value in boolean)
return varchar2;
```

## apex\_plugin: plugin

```
type t_plugin is record (
    name      varchar2(45),
    file_prefix varchar2(4000),
    attribute_01 varchar2(32767), ... );
```

## apex\_plugin: item

```
type t_page_item is record (
    id          number,
    name        varchar2(255),
    label       varchar2(4000),
    plain_label varchar2(4000),
    format_mask varchar2(255),
    is_required boolean,
    lov_definition varchar2(4000),
    lov_display_extra boolean,
    lov_display_null boolean,
    lov_null_text varchar2(255),
    lov_null_value varchar2(255),
    lov_cascade_parent_items varchar2(255),
    ajax_items_to_submit varchar2(255),
    ajax_optimize_refresh boolean,
    element_width   number,
    element_max_length number,
    element_height  number,
    element_attributes varchar2(2000),
    element_option_attributes varchar2(4000),
    escape_output    boolean,
    attribute_01     varchar2(32767), ... );
```

```
type t_page_item_render_result is record (
    is_navigable boolean default false,
    navigable_dom_id varchar2(255) );
```

```
type t_page_item_validation_result is record (
    message      varchar2(32767),
    display_location varchar2(40), /* if not set the app default is used */
    page_item_name varchar2(255) /* if not set the validated page item
                                name is used */
```

Render function

```
function <name of function> (
    p_item      in apex_plugin.t_page_item,
    p_plugin    in apex_plugin.t_plugin,
    p_value     in varchar2,
    p_is_READONLY in boolean,
    p_is_printer_friendly in boolean)
return apex_plugin.t_page_item_render_result;
```

Returns the AJAX identifier which has to be used for the on-demand call of a plug-in. Note: if the plug-in doesn't have an AJAX callback configured, null will be returned!

```
function get_ajax_identifier return varchar2;
```

## apex\_plugin: DA

```
type t_dynamic_action is record (
    id      number,
    action   varchar2(50),
    attribute_01 varchar2(32767), ... );
```

```
type t_dynamic_action_render_result is record (
    javascript_function varchar2(32767),
    ajax_identifier     varchar2(255),
    attribute_01        varchar2(32767), ... );
```

Render function

```
function <name of function> (
    p_dynamic_action in apex_plugin.t_dynamic_action,
    p_plugin         in apex_plugin.t_plugin)
return apex_plugin.t_dynamic_action_render_result;
```

## apex\_plugin: region

```
type t_region is record (
    id      number,
    static_id  varchar2(255),
    name     varchar2(255),
    type     varchar2(255),
    source    varchar2(32767),
    error_message varchar2(32767),
    attribute_01 varchar2(32767), ... );
```

Render function

```
function <name of function> (
    p_region     in apex_plugin.t_region,
    p_plugin     in apex_plugin.t_plugin,
    p_is_printer_friendly in boolean)
return apex_plugin.t_region_render_result;
```

## apex\_plugin: process

```
type t_process is record (
    id      number,
    name     varchar2(255),
    success_message varchar2(32767),
    attribute_01 varchar2(32767), ... );
```

```
type t_process_exec_result is record (
    success_message varchar2(32767));
```

Execute function

```
function <name of function> (
    p_process in apex_plugin.t_process,
    p_plugin  in apex_plugin.t_plugin)
return apex_plugin.t_process_exec_result;
```

```
function escape (
    p_value  in varchar2,
    p_escape in boolean) return varchar2;
```

Note: if both values are null, TRUE will be returned as well.

```
function is_equal (
    p_value1 in varchar2,
    p_value2 in varchar2) return boolean;
```

Returns the position in the list where p\_value is stored. If it's not found null will be returned.

```
function get_position_in_list (
    p_list  in www_flow_global_vc_arr2,
    p_value in varchar2) return number;
```

Outputs the value in an escaped form. It's also taking care of chunking big strings into smaller outputs.

```
procedure print_escaped_value (
    p_value in varchar2);
```

Outputs a hidden field to store the page item value if the page item is rendered readonly but not printer friendly

```
procedure print_hidden_if_READONLY (
    p_item_name   in varchar2,
    p_value       in varchar2,
    p_is_READONLY in boolean,
    p_is_printer_friendly in boolean,
    p_id_postfix  in varchar2 default null);
```

Outputs a hidden protected field to store the page item value.

```
procedure print_hidden_protected (
    p_item_name in varchar2,
    p_value     in varchar2,
    p_id_postfix in varchar2 default null);
```

Outputs a SPAN tag for a display only field.

```
procedure print_display_only (
    p_item_name   in varchar2,
    p_display_value in varchar2,
    p_show_line_breaks in boolean,
    p_escape      in boolean,
    p_attributes  in varchar2,
    p_id_postfix  in varchar2 default '_DISPLAY');
```

Outputs an OPTION tag. The values are always escaped.

```
procedure print_option (
    p_display_value in varchar2,
    p_return_value  in varchar2,
    p_is_SELECTED  in boolean,
    p_attributes   in varchar2,
    p_escape       in boolean default true);
```

Returns a Javascript array with all the page items which are really on the current and on page 0. p\_bind\_variable\_list are the bind variables returned by www\_flow\_utilities.get\_binds

```
function get_depending_fields (
    p_bind_variable_list in www_flow_global_vc_arr2)
return varchar2;
```

Outputs a JSON response based on the result of a two column LOV in the format:

```
[{"d": "display", "r": "return"}, {"d": "...", "r": "..."}, ...]
```

p\_component\_name is the name of the page item or report column, ... which is used in case an error has to be displayed.

Note: The HTTP header is initialized with mime type "application/json"

```
procedure print_lov_as_json (
    p_sql_statement in varchar2,
    p_component_name in varchar2,
    p_escape        in boolean,
    p_support_legacy_lov in boolean default false /*internal use only! */);
```

## apex\_plugin\_util

\* For a complete reference check the apex\_plugin and apex\_plugin\_util packages

## apex\_plugin\_util

Executes the query and returns the values for each column.  
Note: All column values are returned as a string  
p\_sql\_statement: SQL statement which should be executed.  
p\_min\_columns and p\_max\_columns: Used to check if the SQL statement complies to the number of required columns.  
p\_component\_name: The name of the page item or report column which is used in case an error has to be displayed.  
p\_search\_type: Use one of the c\_search\_\* constants.  
p\_search\_column\_no: Column which should be used to restrict the SQL statement. Has to be in range of p\_min\_column and p\_max\_column.  
p\_search\_string: Value which should be used to restrict the query.  
p\_first\_row: Skips all rows before the specified row. Can be used for pagination.  
p\_max\_rows: Limits the number of returned rows.

```
function get_data (
    p_sql_statement in varchar2,
    p_min_columns in number,
    p_max_columns in number,
    p_component_name in varchar2,
    p_search_type in varchar2 default null,
    p_search_column_no in varchar2 default 2,
    p_search_string in varchar2 default null,
    p_first_row in number default null,
    p_max_rows in number default null,
    p_support_legacy_lov in boolean default false /* internal use only! */)
return t_column_value_list;
```

Advanced version of get\_data, returns the column values with original data types.  
p\_data\_type\_list: If provided checks that the data type for each column matches to the specified data type in the array. Use the constants c\_data\_type\_\* for available data types.

```
function get_data2 (
    p_sql_statement in varchar2,
    p_min_columns in number,
    p_max_columns in number,
    p_data_type_list in www_flow_global_vc_arr2 default c_empty_data_type_list,
    p_component_name in varchar2,
    p_search_type in varchar2 default null,
    p_search_column_no in varchar2 default 2,
    p_search_string in varchar2 default null,
    p_first_row in number default null,
    p_max_rows in number default null)
return t_column_value_list2;
```

Gets the display value for the value specified in p\_search\_string.  
p\_component\_name: The name of the page item or report column, ... which is used in case an error has to be displayed.

p\_display\_column\_no: Column which should be returned from the SQL statement. Has to be in range of p\_min\_column and p\_max\_column.  
p\_search\_column\_no: Column which should be used to restrict the SQL statement. Has to be in range of p\_min\_column and p\_max\_column.  
p\_search\_string: Value which should be looked up.

p\_display\_extra: If set to TRUE and a value isn't found, the search value will be added to the result instead.

Note: In the case multiple rows are returned by the query, just the first one is returned.

```
function get_display_data (
    p_sql_statement in varchar2,
    p_min_columns in number,
    p_max_columns in number,
    p_component_name in varchar2,
    p_display_column_no in binary_integer default 1,
    p_search_column_no in binary_integer default 2,
    p_search_string in varchar2,
    p_display_extra in boolean default true,
    p_support_legacy_lov in boolean default false /* for internal use only! */)
return varchar2;
```

Gets the display values for the values specified in p\_search\_value\_list.

```
function get_display_data (
    p_sql_statement in varchar2,
    p_min_columns in number,
    p_max_columns in number,
    p_component_name in varchar2,
    p_display_column_no in binary_integer default 1,
    p_search_column_no in binary_integer default 2,
    p_search_value_list in www_flow_global_vc_arr2,
    p_display_extra in boolean default true,
    p_support_legacy_lov in boolean default false /* for internal use only! */)
return www_flow_global_vc_arr2;
```

Returns a jQuery selector based on a comma delimited string of page item names.

```
function page_item_names_to_jquery (
    p_page_item_names in varchar2)
return varchar2;
```

Returns a cleaned up version of a comma delimited string of page item names where all spaces are removed, colons are replaced with a comma and the page item names are converted to uppercase. This is important for JavaScript code, because page item names are case sensitive in the browser.

```
function cleanup_page_item_names (
    p_page_item_names in varchar2)
return varchar2;
```

Writes the data of the page item meta data into the HTTP buffer.

```
procedure debug_page_item (
    p_plugin in www_flow_plugin_t_plugin,
    p_page_item in www_flow_plugin_t_page_item);
procedure debug_page_item (
    p_plugin in www_flow_plugin_t_plugin,
    p_page_item in www_flow_plugin_t_page_item,
    p_value in varchar2,
    p_is_readonly in boolean,
    p_is_printer_friendly in boolean);
```

Writes the data of the region meta data into the HTTP buffer.

```
procedure debug_region (
    p_plugin in www_flow_plugin_t_plugin,
    p_region in www_flow_plugin_t_region);
procedure debug_region (
    p_plugin in www_flow_plugin_t_plugin,
    p_region in www_flow_plugin_t_region,
    p_is_printer_friendly in boolean);
```

Writes the data of the dynamic action meta data into the HTTP buffer.

```
procedure debug_dynamic_action (
    p_plugin in www_flow_plugin_t_plugin,
    p_dynamic_action in www_flow_plugin_t_dynamic_action);
```

Writes the data of the process meta data into the HTTP buffer.

```
procedure debug_process (
    p_plugin in www_flow_plugin_t_plugin,
    p_process in www_flow_plugin_t_process);
```

## apex\_javascript

Adds the script tag to load a javascript library.

```
p_name: has to be specified without .js
p_directory: has to have a trailing slash
p_version: version identifier which should be added to the library name
p_skip_extension: if true the extension .js is NOT added
procedure add_library(
    p_name in varchar2,
    p_directory in varchar2 default www_flow_g_image_prefix||'javascript/',
    p_version in varchar2 default c_apex_version,
    p_skip_extension in boolean default false);
```

Adds a javascript code snippets to the HTML output which is executed at the onload event. If an entry with the same key exists it will be ignored.

```
If p_key is null the snippet will always be added.
p_code: javascript code snippet. eg: $s('P1_TEST'; 'abc');
p_key: name of the key. eg: APEX_WIDGET
```

```
procedure add_onload_code(
    p_code in varchar2,
    p_key in varchar2 default null);
```

Adds a code snippet which is included inline into the HTML output eg. You can use this procedure to add new functions or global variable declarations.

Note: if you want to execute code, you should use add\_onload\_code  
If an entry with the same key exists it will be ignored.

```
If p_key is null the snippet will always be added.
p_code: javascript code snippet. eg: $s('P1_TEST'; 123);
```

p\_key: identifier for the code snippet. If specified and a code snippet with the same name has already been added the new code snippet will be ignored.

```
procedure add_inline_code(
    p_code in varchar2,
    p_key in varchar2 default null);
```

Escapes a text so that it can be used in JavaScript without problems.

Replaces \ with \\, / with \/, " with \u0022, ' with \u0027, tab with \t,chr(10) with \n.

```
function escape (
    p_text in varchar2) return varchar2;
```

## apex\_css

Adds the style tag to load a css library.

```
p_name: has to be specified without .js
p_directory: has to have a trailing slash
p_version: version identifier which should be added to the library name
p_skip_extension: if true the extension .css is NOT added
procedure add_file(
    p_name in varchar2,
    p_directory in varchar2 default www_flow_g_image_prefix||'css/',
    p_version in varchar2 default c_apex_version,
    p_skip_extension in boolean default false);
```

Adds a CSS style snippet which is included inline into the HTML output eg. You can use this procedure to add new css style declarations.

If an entry with the same key exists it will be ignored.  
If p\_key is null the snippet will always be added.

```
p_style: CSS style snippet. eg: #test { color:#fff }
```

p\_key: identifier for the style snippet. If specified and a style snippet with the same name has already been added the new style snippet will be ignored.

```
procedure add (
    p_css in varchar2,
    p_key in varchar2 default null);
```