GreyOS

"Coding Style"

Ver. 1.0

All the developers will have to follow the coding style guidelines:

- Always use a space after a comma.
- Use only one line with no braces around when you use a one-line "if", "for", "do" or "while" statement.
- Always use a 4 spaces (1 tab) indentation.
- Put function and class braces always under the declaration.
- Declare variables with comma separation when possible (design pattern).
- Use multiline comments with "/* … */" and one-line comments with "// …".
- Always leave a blank row after { and before }.
- Do not use "camel case" mode. Instead, use all capitals for constants and every first letter capital for function names or classes or variables with more than one word. For JS you may use non capital letters for the full function names or variables.
- For every naming, divide each word with an underscore ("_").
- Always leave one row between two commands that are not closely related.
- All functions should return a result. If no result is needed then they should return a status: "true" or "false".

Coding style examples:

```
/*

   GreyOS Inc. - spl@sh

   Version: 3.2

   File name: splash.php
   Description: This file contains the SPLASH wrapper class.

   Coded by George Delaportas (G0D)

   GreyOS Inc.
   Copyright © 2013

*/
```

Example 1. Header comments

```
// Load AJAX support
ALPHA_CMS::Load_Extension('bull', 'ajax');

// Include HELPERS class
require_once('helpers/helpers.php');

// Include EVENTS wrapper
require_once('events/events.php');

...
```

Example 2. Loading dependencies

```
// Wrapper class: [SPLASH]
class SPLASH
{

   // [ACTION]
   private $__link;
   private $__button;

   // [GROUP]
   private $__div;
   private $__table;
   private $__fieldset;

   ...
```

Example 3. Declaring classes and private variables

```
public function Test()
{

   /* ----------------------- [CONTROL] ----------------------- */

   // [ACTION]
   $this->__link = new LINK();
   $this->__button = new BUTTON();
```

Example 4. Declaring functions and initializing objects

```
public function Link($mode, $content, $attributes = null, $events = null)
{

   if (HELPERS::Is_Valid_Mode($mode))
   {

      if ($mode == 1)
         $result = $this->__link->Show($content, $attributes, $events);

      else
         $result = $this->__link->Fetch($content, $attributes, $events);

      return $result;

   }

   return false;

}
```

Example 5. Full PHP function implementation

```
// Initialize an AJAX object
function init()
{

   var counter = 0;

   // Initialize global XML HTTP objects
   If (utils.is_undefined(ajax_data_num) &&
      utils.is_undefined(ajax_response_num))
   {

      for (counter = 0; counter < (__default_ajax_data_num +__default_ajax_response_num);
           counter++)
         __global_xml_http[counter] = new utils.ajax_object();

   }

   else
   {

      if (!utils.is_integer(ajax_data_num) || !utils.is_integer(ajax_response_num))
         return false;

      for (counter = 0; counter < (ajax_data_num + ajax_response_num); counter++)
         __global_xml_http[counter] = new utils.ajax_object();

      __ajax_data_num = ajax_data_num;
      __ajax_response_num = ajax_response_num;
```

```
        __init_mode = 1;

    }

    return true;

}
```

Example 6. Full JS function implementation

```
// Pythia
function pythia()
{

    var self = this;

    var __results = [],
        __this_result = null;

    this.generate = function(length)
    {

        var __res_len = __results.length;

        __this_result = Math.floor((Math.random() * length) + 1);

        for (var i = 0; i < __res_len; i++)
        {

            if (__this_result === __results[i])
                return self.generate(length);

        }

        __results.push(__this_result);

        return __this_result;

    };

}
```

Example 7. Full declaration of an object in JS

Best regards,
George A. Delaportas
GreyOS Inc.