Стиль написания кода

На самом деле, нет никаких строгих правил форматирования программного кода, но всё же следует придерживаться некоторых принципов при добавлении кода в «Докувики».

Скобки и отступы

Для выделения логических блоков вы должны использовать отступы либо в два, либо четыре **пробела**. Пожалуйста, **не используйте** символы табуляции. Одно и то же количество пробелов должно использоваться а протяжении всего файла, поэтому проверьте сколько пробелов используется в существующем коде и используйте его же.

Открывающие скобки должны начинаться в той же строке что и ключевое слово, закрывающая скобка должна быть расположена под первым символом начального ключевого слова. Т. е.:

```
if ($foo == "bar"){
  call_bar();
}elseif($foo == "baz"){
  call_baz();
}else{
  call_other();
}
```

Завершение строки

Строка должна заканчиваться единственным символом перевода строки LF (так называемый, стиль UNIX). Пожалуйста, избегайте пробелов в конце строки. Заглянув в мой файл vimrc вы сможете понять, как легко в редакторе VIM придерживаться этого автоматически.

Комментирование

Каждой функции и классу следует иметь комментарии в стиле PHPDocumentor, содержащие по крайней мере предназначение функции и автора. Описания параметров и возвращаемого значения приветствуются, но обязательно только если их назначение не очевидно. Если вы улучшаете существующую функцию, просто добавьте еще одну строку с автором.

Пример:

```
/**
 * Check for foo in bar
 *
 * Checks if there is a foo in bar
 *
 * @author   Joe Schmoe <joe@example.com>
```

```
 * @param    string $in your input
 * @returns  boolean    true if foo in bar
 *
 */
function is_foo($in){
...
}
```

Эти комментарии используются для автоматической генерации документации по API.

Закрывающие тэги PHP

Вам следует опускать закрывающие тэги PHP (?>) в конце всех php-файлах, чтобы избежать нежелательного вывода. Это может показаться странным, но, на самом деле, это отмечается в руководстве по PHP:

Замечание: Закрывающие теги блока PHP в конце файла не обязательно, и в некоторых случаях их отсутствие даже полезно, когда используются функции `include()` или `require()`, таким образом, нежелательные пробелы не возникнут в конце файла, и у вас все еще будет возможность добавлять заголовки (с помощью функции `header()`). Кроме того, это удобно, если вы используете буферизацию вывода, и не хотите добавлять нежелательные пробелы в конце части, сгенерированной включаемым файлом.

From:
https://wwoss.ru/ - **worldwide open-source software**

Permanent link:
https://wwoss.ru/doku.php?id=wiki:coding_style

Last update: **2023/08/18 15:26**