## **Справочник phpDoc**

**@api** – помечает структурный элемент, используемый сторонними программами, и являющийся частью API.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | */\*\**  *\* Этот метод показывает версию релиза*  *\**  *\* @api*  *\**  *\* @return void*  *\*/*  function showVersion()  {  <...>  } |

**@author** – задает имя автора.

|  |  |
| --- | --- |
| 1  2  3  4 | */\*\**  *\* @author My Name*  *\* @author My Name <my.name@example.com>*  *\*/* |

**@copyright** – копирайтинг.

|  |  |
| --- | --- |
| 1  2  3 | */\*\**  *\* @copyright 2011-2012 The lifeexample.ru*  *\*/* |

**@deprecated** – тег для пометки элементов на удаление в будущих версиях.

|  |  |
| --- | --- |
| 1  2  3  4  5 | */\*\**  *\* @deprecated*  *\* @deprecated 1.0.0*  *\* @deprecated Больше не используется внутри кода, и не рекомендуется.*  *\*/* |

**@example** – пример использования метода или класса.

**@filesource** – сообщает в phpDocumentor, что нужно включить исходный код в текущий файл для разбора.

|  |  |
| --- | --- |
| 1  2  3 | */\*\**  *\* @filesource*  *\*/* |

**@global** – описывает глобальные переменные.

**@ignore** – сообщает в phpDocumentor, что следующий структурный элемент обрабатывать не надо.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | if ($ostest) {  */\*\**  *\* Константа определяет версию операционной системы*  *\*/*  define("OS","Unix");  } else {  */\*\**  *\* @ignore*  *\*/*  define("OS","Windows");  } |

**@internal** – говорит, что элемент является внутренним элементом этого приложения или библиотеки.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | */\*\**  *\* @internal*  *\**  *\* @return Возвращает целочисленное значение - количество чего либо.*  *\*/*  function count()  {  <...>  } |

**@license** – описывает соответствие лицензии.

|  |  |
| --- | --- |
| 1  2  3  4 | */\*\**  *\* @license GPL*  *\* @license http://opensource.org/licenses/gpl-license.php GNU Public License*  *\*/* |

**@link** – связывает структурный элемент с сайтом, откуда он взят.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | */\*\**  *\* @link http://lifeexample.ru/razrabotka-i-optimizacia-saita/phpdoc-i-phpdocumentor Документация к функции.*  *\**  *\* @return Возвращает целочисленное значение - количество чего либо.*  *\*/*  function count()  {  <...>  } |

**@method** – указывает какие магические (*не явные*) методы класса можно вызвать.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | class Parent  {  public function \_\_call()  {  <...>  }  }  */\*\**  *\* @method string getString()*  *\* @method void setInteger(integer $integer)*  *\* @method setString(integer $integer)*  *\*/*  class Child extends Parent  {  <...>  } |

**@package** – классификация структурных элементов в логических подразделениях.

**@param** – описывает тип и имя параметра передаваемого в функцию.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | */\*\**  *\* Counts the number of items in the provided array.*  *\**  *\* @param mixed[] $array Array массив значений.*  *\* @param int|string $value некоторое значение.*  *\* @return int возвращает номер элемента.*  *\*/*  function count(array $items, $value)  {  <...>  } |

**@property** – аналогичен тегу **@method**. Позволяет классу знать какие можно использовать магические свойства.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | class Parent  {  public function \_\_get()  {  <...>  }  }  */\*\**  *\* @property string $myProperty*  *\*/*  class Child extends Parent  {  <...>  } |

**@property-read** – указывает какие можно использовать магические свойства, только для чтения.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | class Parent  {  public function \_\_get()  {  <...>  }  }  */\*\**  *\* @property-read string $myProperty*  *\*/*  class Child extends Parent  {  <...>  } |

**@property-write** – указывает какие можно использовать магические свойства, только для записи.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | class Parent  {  public function \_\_set()  {  <...>  }  }  */\*\**  *\* @property-write string $myProperty*  *\*/*  class Child extends Parent  {  <...>  } |

**@return** – описывает возвращаемое значение функцией или методом.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | */\*\**  *\* @return string|null Метка для текста.*  *\*/*  function getLabel()  {  <...>  } |

**@see** – задает ссылку из структурного элемента, на веб-сайт или другой элемент.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | */\*\**  *\* @see http://lifeexample.ru/foo Документация функции.*  *\* @see MyClass::$items свойства элементов.*  *\* @see MyClass::setItems() установка элементов коллекции.*  *\**  *\* @return integer Показывает количество элементов в коллекции.*  *\*/*  function count()  {  <...>  } |

**@since** – говорит о том, что структурный элемент стал доступен с заданной версии пакета.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | */\*\**  *\* Файл проекта*  *\* @package test*  *\* @version 1.5.2*  *\*/*  */\*\**  *\* функция datafunction*  *\* @since Version 1.1.2*  *\*/*  function datafunction()  {  ...  } |

**@subpackage** – описывает под-пакет основного пакета, используется вместе с тегом **@package**.

**@todo** – документирует задачи, которые необходимо выполнить в ближайшем будущем.

**@uses** – показывает ссылку на документацию по этому элементу, и создает обратную ссылку в других документациях элементов на него.

**@var** – указывает описание для свойств класса.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | class class1{  */\*\**  *\* пример документирования переменной*  *\* @var string*  *\*/*  var $variable;  */\*\**  *\* пример документирования переменной с описанием*  *\* @var string содержит информацию о class1*  *\*/*  var $variable\_with\_desc;    } |

**@version** – версия документа.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | */\*\**  *\* Пример использования тега @version*  *\* @version v 1.2 дата релиза 2006-04-29;*  *\*/*  function datafunction() {  } |

Ознакомившись со всеми тегами из справочника по **phpDocumentor**, не рекомендую вам в изобилии снабжать ими **phpDoc** блоки. На мой взгляд, в большинстве случаев достаточно лишь нескольких из доступных таких как:

* @author;
* @deprecated;
* @var;
* @version;
* @todo;
* @return;
* @package;
* @param.

Но это не значит что остальные phpDoc параметры, являются бесполезными. Для каждой ситуации, нужно в индивидуальном порядке указывать тот или иной дескриптор.

***Полный список всех doc-комментариев расположен на официальном сайте phpdocumentor*** - <https://www.phpdoc.org/docs/latest/index.html>