Contributing to phpDocumentor

=============================

Introduction

------------

phpDocumentor aims to be a high quality Documentation Generation Application (DGA) but at the same time wants to give

contributors freedom when submitting fixes or improvements.

As such we want to \*encourage\* but not obligate you, the contributor, to follow these guidelines. The only exception to

this are the guidelines regarding \*GitHub Usage & Branching\* to prevent `merge-hell`.

Having said that: we really appreciate it when you apply the guidelines in part or wholly as that will save us time

which, in turn, we can spend on bugfixes and new features.

GitHub Usage & Branching

------------------------

Once you decide you want to contribute to phpDocumentor (which we really appreciate!) you can fork the project from

https://github.com/phpDocumentor/phpDocumentor.

Currently phpDocumentor version 3.0 is developed in our `master` branch. Large parts of the code in there will be

rewritten or removed. If you want to contribute to phpDocumentor v3.0, create your feature branch from `master`. If you

want to fix a bug in the current released version, base your branch on `2.9`. Please \*always\* create a new branch for

each feature/bugfix you want to contribute.

> If you create your branch from the wrong base branch we won't be able to merge your feature in to the right version.

> Which means that either your feature will only be released with v3.0 or bugfix will never be in a new 2.\* release.

To get more information about the usage of Git, please refer to the [Pro Git book][PROGIT] written

by Scott Chacon and/or [this help page of GitHub][GITHUB\_HELP\_PR].

Setting Up Your Development Environment

---------------------------------------

You need the following:

- Git

- Composer (https://getcomposer.org)

- Phive (https://phar.io)

- Docker (https://docker.com), including Docker Compose

Once you cloned the repository you should be able to run the following commands to get started

$ composer install

$ phive install

$ docker-compose run phpdoc

To run the tests you can use the following command:

$ make test

Before issuing a pull request it is also recommended to run the following commands:

$ make phpcs

$ make phpstan

These command will check the quality of your code; this is also done by Travis during the pull request process but

performing these checks yourself will help getting your pull request merged.

Coding Standards

----------------

phpDocumentor uses the [PSR-2 Coding Standard][PSR2] as defined by the

[PHP Framework Interoperability Group (PHP-FIG)][PHP\_FIG].

It is recommended to check your code using phpCodeSniffer using the \*PSR2\* standard using the following command:

$ make phpcs

Unit testing

------------

phpDocumentor aims to be have at least 90% Code Coverage using unit tests using PHPUnit. It is appreciated to include

unit tests in your pull requests as they also help understand what the contributed code exactly does.

In order to run the unit tests you can execute the following command from your phpDocumentor folder:

$ make test

### Running tests without Docker

To make sure that the application is tested in a controlled way we use Docker. However, sometimes it is not possible to

use Docker on a development machine. In these cases it is recommended to run the following commands:

$ make install-phive

$ ./tools/phive.phar install --copy --trust-gpg-keys 4AA394086372C20A,D2CCAC42F6295E7D,E82B2FB314E9906E,8E730BA25823D8B5,D0254321FB74703A --force-accept-unsigned

And after this you can run phpunit using `./tools/phpunit`.

[PROGIT]: https://git-scm.com/book

[GITHUB\_HELP\_PR]: https://help.github.com/articles/using-pull-requests

[PSR2]: https://www.php-fig.org/psr/psr-2/

[PHP\_FIG]: https://www.php-fig.org/