Composer Cheat Sheet

Installation

Download Composer from https://getcomposer.org/download/ and follow the instructions.

Enable bash autocompletion:

```
composer completion bash | sudo tee /etc/bash_completion.d/composer
```

Installing dependencies

Command	Description
`composer install`	Install dependencies from `composer.lock`
`composer install --no-dev`	Install without development dependencies

Managing dependencies

Command	Description
`composer require <package>`	Add a new dependency to `composer.json` and install it
`composer require <package>:<version>`	Require a specific version of a dependency
`composer require --dev <package>`	Add a new development dependency to `composer.json` and install it
`composer update`	Update all dependencies to their latest installable versions
`composer update <package>`	Update a dependency to its latest installable version
`composer update <vendor>/*`	Update all dependencies of a vendor to their latest versions
`composer remove <package>`	Remove a dependency from `composer.json` and uninstall

Global dependencies

Command	Description
`composer global require <package>`	Add a new global dependency to `composer.json` and install it
`composer global update`	Update all global dependencies

Configuration

Command	Description
`composer config sort-packages true`	Sort packages in `composer.json` alphabetically

Authentication

Command	Description
`composer config --global github-oauth.github.com <token>`	Authenticate with GitHub using a personal access token
`composer config --global http-basic.repo.packagist.com <username> <password>`	Authenticate with Private Packagist using a username and password

Dependency analysis

Command	Description
`composer show`	List all installed dependencies
`composer show <package>`	Display detailed information about a package
`composer outdated`	List dependencies that are outdated
`composer audit`	Check for known security vulnerabilities in dependencies
`composer licenses`	List all licenses of installed dependencies
`composer status`	Show a list of packages with local modifications

Creating new Composer projects

Command	Description
`composer init`	Create a `composer.json` file in the current directory
`composer create-project <package>`	Create a new project from a package

Other commands

Command	Description
`composer --version`	Show the current Composer version
`composer self-update`	Update Composer to the latest version

Version constraints

Constraint	Description
`1.0.2`	Exactly version `1.0.2`
`>=1.0.2`	Version `1.0.2` or higher
`>=1.0.2 <1.1.0`	Version `1.0.2` or higher but lower than `1.1.0`
`>=1.0.2 <1.1.0 || >=1.2.0`	Version `1.0.2` or higher but lower than `1.1.0` or version `1.2.0` or higher
`1.0 - 2.0`	Version `1.0` to `2.0` (including `1.0` and `2.0`)
`1.0.*`	Version `1.0` with any patch version
`~1.2`	Version `1.2` or higher but lower than `2.0`
`~1.2.3`	Version `1.2.3` or higher but lower than `1.3`
`^1.2`	Version `1.2` or higher but lower than `2.0`
`^1.2.3`	Version `1.2.3` or higher but lower than `2.0`

Repositories

Command	Description
`composer config repositories.foo composer <url>`	Add a new repository named `foo` with the given URL as a Composer repository
`composer config repositories.foo vcs <url>`	Add a new repository named `foo` with the given URL as a VCS repository
`composer config repositories.foo vcs git@github.com:foo/bar.git`	Add a new GitHub repository named `foo/bar` as a VCS repository
`composer config repositories.foo path <path>`	Add a new repository named `foo` with the given path as a path repository