

# Faker

[Packagist Downloads](https://packagist.org/packages/fakerphp/faker) [GitHub Workflow Status](https://github.com/FakerPHP/Faker/actions) [Type Coverage](https://shepherd.dev/github/FakerPHP/Faker) [Code Coverage](https://codecov.io/gh/FakerPHP/Faker)

Faker is a PHP library that generates fake data for you. Whether you need to bootstrap your database, create good-looking XML documents, fill-in your persistence to stress test it, or anonymize data taken from a production service, Faker is for you.

It's heavily inspired by Perl's [Data::Faker](https://metacpan.org/pod/Data::Faker), and by Ruby's [Faker](https://rubygems.org/gems/faker).

## Getting Started

### Installation

Faker requires PHP >= 7.4.

composer require fakerphp/faker

### Documentation

Full documentation can be found over on [fakerphp.github.io](https://fakerphp.github.io).

### Basic Usage

Use Faker\Factory::create() to create and initialize a Faker generator, which can generate data by accessing methods named after the type of data you want.

<?php

require\_once 'vendor/autoload.php';

// use the factory to create a Faker\Generator instance

$faker = Faker\Factory::create();

// generate data by calling methods

echo $faker->name();

// 'Vince Sporer'

echo $faker->email();

// 'walter.sophia@hotmail.com'

echo $faker->text();

// 'Numquam ut mollitia at consequuntur inventore dolorem.'

Each call to $faker->name() yields a different (random) result. This is because Faker uses \_\_call() magic, and forwards Faker\Generator->$method() calls to Faker\Generator->format($method, $attributes).

<?php

for ($i = 0; $i < 3; $i++) {

echo $faker->name() . "\n";

}

// 'Cyrus Boyle'

// 'Alena Cummerata'

// 'Orlo Bergstrom'

## Automated refactoring

If you already used this library with its properties, they are now deprecated and needs to be replaced by their equivalent methods.

You can use the provided [Rector](https://github.com/rectorphp/rector) config file to automate the work.

Run

composer require --dev rector/rector

to install rector/rector.

Run

vendor/bin/rector process src/ --config vendor/fakerphp/faker/rector-migrate.php

to run rector/rector.

*Note:* do not forget to replace src/ with the path to your source directory.

Alternatively, import the configuration in your rector.php file:

<?php

declare(strict\_types=1);

use Rector\Config;

return static function (Config\RectorConfig $rectorConfig): void {

$rectorConfig->import('vendor/fakerphp/faker/rector-migrate.php');

};

## License

Faker is released under the MIT License. See [LICENSE](http://license) for details.

## Backward compatibility promise

Faker is using [Semver](https://semver.org/). This means that versions are tagged with MAJOR.MINOR.PATCH. Only a new major version will be allowed to break backward compatibility (BC).

Classes marked as @experimental or @internal are not included in our backward compatibility promise. You are also not guaranteed that the value returned from a method is always the same. You are guaranteed that the data type will not change.

PHP 8 introduced [named arguments](https://wiki.php.net/rfc/named_params), which increased the cost and reduces flexibility for package maintainers. The names of the arguments for methods in Faker is not included in our BC promise.