

The `dotenv` Package

Prerequisites: [Environment Variables](#), [The `os` Module](#)

The `dotenv` package allows a program to reference environment variables from a project-specific `.env` file. This makes environment variables much easier to manage, especially for Windows users.

Reference: <https://github.com/theskumar/python-dotenv>.

Installation

First install the package, if necessary:

```
pip install python-dotenv # note: NOT just "dotenv"
```

Usage

To setup this example, create a new directory on your Desktop named "my-secure-project". Then navigate there from the command-line:

```
cd Desktop/my-secure-project/
```

Create two files in the "my-secure-project" directory named `.env` and `my_script.py`, respectively, and place inside the following contents:

```
# my-secure-project/.env

SECRET_MESSAGE="Hello World"
```

```
# my-secure-project/my_script.py

from dotenv import load_dotenv
import os

print(os.environ.get("SECRET_MESSAGE")) #> None

load_dotenv() #> loads contents of the .env file into the script's
environment
```

```
print(os.environ.get("SECRET_MESSAGE")) #> "Hello World"
```

And run the script to see the output:

```
python my_script.py
```

The lesson is that the `load_dotenv()` function will load environment variables from the ".env" file into the Python script's environment so they can be accessed via the `os` module.

Ignoring ".env" Files from Version Control

SECURITY NOTE: Because these ".env" files often contain sensitive information like secret passwords and API Keys, we should absolutely avoid checking them into version control! To do this, we'll use a special ".gitignore" file.

Create another file in the "my-secure-project" directory named ".gitignore", and place inside the following contents:

```
# my-secure-project/.gitignore

# ignore the ".env" file:
.env
```

Great! Now all subsequent commits will ignore the ".env" file from your project's version history, so you can push your code to GitHub without divulging your secret credentials.

NOTE: if your repository already contained a ".env" file before you added the corresponding entry to the ".gitignore" file, you'll need to commit the ".gitignore" file then delete / move the ".env" file and make another commit, and then afterwards you can feel free to restore your ".env" file and it will be ignored