# Installing new.py Into Your PATH

I hate writing code from scratch! This week you learned about using a program called `new.py` that will create a program for you to start from. Now, we need to add this program to our PATH, so we can just use it without having to figure out where it is!

In the *bin* directory of your repo, you should find a program called `new.py` that will help you make a new Python program. From this directory, you can provide the full path using `..` to indicate the parent directory:

```
cd ./assignments/01_salutations
../../bin/new.py -h
usage: new.py [-h] [-n NAME] [-e EMAIL] [-p PURPOSE] [-f] program

Create Python argparse program

positional arguments:
  program               Program name

optional arguments:
  -h, --help            show this help message and exit
  -n NAME, --name NAME  Name for docstring (default: Ken Youens-Clark)
  -e EMAIL, --email EMAIL
                        Email for docstring (default: kyclark@gmail.com)
  -p PURPOSE, --purpose PURPOSE
                        Purpose for docstring (default: Rock the Casbah)
  -f, --force           Overwrite existing (default: False)
```

It will be unpleasant to always indicate the full path to `new.py` as you will use it often. I suggest you create a directory in your `$HOME` (which is often written using the tilde `~` AKA "twiddle") to put useful programs you'll write. It's common to create a *~/local* or *~/.local* (so it's hidden) to install software, and inside of that a *bin* directory:

```
mkdir ~/.local
mkdir ~/.local/bin
```

You will need to ensure that this directory is included in your `$PATH`. First check what Unix shell you are using:

```
echo $SHELL
```

If are using the `bash` shell, you can edit *~/.bashrc*. If are using the `zsh` shell, you can edit *~/.zshrc*. For instance, you can use `nano`:

```
nano ~/.bashrc
```

Add this line to the end:

```
export PATH=~/.local/bin:$PATH
```

Then use the `source` command to read this file and alter your `$PATH`:

```
source ~/.bashrc
```

You can view your `$PATH` to ensure this directory is included:

```
echo $PATH
```

Then you can copy the `new.py` program to that location:

```
cp ../../bin/new.py ~/.local/bin
```

Verify that the program can be found using `which`:

```
$ which new.py
```

My new.py file is located here: /Users/bhurwitz/.local/bin/new.py

## Getting Started with new.py for the first homework assignment

Here is how you can create the `salutations.py` using `new.py`:

```
$ new.py -p 'Print greeting' salutations.py
Done, see new script "salutations.py."
```