

# Tutoriel Argparse

**Argparse:** the recommended command-line parser in the standard Python library.

## First example:

```
import argparse
parser = argparse.ArgumentParser()
parser.parse_args()
```

## Running this script in the terminal:

```
$ python3 prog.py
$ python3 prog.py --help
usage: prog.py [-h]

$ python3 prog.py --verbose
usage: prog.py [-h]
prog.py: error: unrecognized arguments: --verbose
$ python3 prog.py foo
usage: prog.py [-h]
prog.py: error: unrecognized arguments: foo
```

→ Executing the script without any options results with nothing displayed on stdout. It is not very useful.

→ The `--help` option, which can also be shortened to `-h`, is the only option we have for free.

## Positional Arguments :

```
import argparse
parser = argparse.ArgumentParser()
parser.add_argument("echo")
args = parser.parse_args()
print(args.echo)
```

We execute the code :

```
$ python3 prog.py
usage: prog.py [-h] echo
```

```
prog.py: error: the following arguments are required: echo
$ python3 prog.py --help
usage: prog.py [-h] echo
```

positional arguments:

echo

optional arguments:

-h, --help show this help message and exit

- Adding the `add_argument()` method that is used to specify which command-line options the program can accept. In this case it's called `echo` for it to match its function.
- Using the program now requires that you specify an option. The `parse_args()` method actually returns some data from the specified options, in this case: `echo`.
- The variable is like a form of "magic" that `argparse` does for free (i.e. no need to specify in which variable the value is stored). The name is the same as the string argument given to the method: `echo`.

### Optional arguments:

```
import argparse
parser = argparse.ArgumentParser()
Et le résultat :parser.add_argument("--verbosity", help="increase output verbosity")
args = parser.parse_args()
if args.verbosity:
    print("verbosity turned on")
```

```
$ python3 prog.py --verbosity 1
verbosity turned on
$ python3 prog.py
$ python3 prog.py --help
usage: prog.py [-h] [--verbosity VERBOSITY]
```

optional arguments:

-h, --help show this help message and exit  
--verbosity VERBOSITY  
increase output verbosity

```
$ python3 prog.py --verbosity
usage: prog.py [-h] [--verbosity VERBOSITY]
prog.py: error: argument --verbosity: expected one argument
```

- The program is written so that it does not display anything unless the `--verbosity` option is set.
- To show that the option is optional, there will be no error if we run the program without it.