

---

# **wttr.py**

***Release 1.0.0***

**Jul 16, 2020**



---

Getting started:

---

<b>1</b>	<b>0. Getting started with wttr.py</b>	<b>1</b>
<b>2</b>	<b>wttr.py API documentation</b>	<b>3</b>
	<b>Index</b>	<b>5</b>



## 0. Getting started with wttr.py

**Warning:** This tutorial assumes that you have installed at least Python 3.6 with pip.

You can install wttrpy package like that:

```
$ python -m pip install wttrpy --user
```

Now you can test it out:

```
$ python
>>> import wttrpy
>>> wttr = wttrpy.getWttr("Amsterdam", "en")
>>> print(wttr)
```

The output should be something like this:

```

Weather report: Amsterdam

\ /      Partly cloudy
- /" ".-. 17 °C
\_( ) .   17 km/h
/(__(_)   10 km
          0.2 mm

Tue 14 Jul
Morning      Noon      Evening
Night
Light rain sho... Patchy rain po... \ /
Partly cloudy | \ / Partly cloudy | - /" ".-. 15..17
, \_( ) .   17 °C | , \_( ) .   16 °C | - /" ".-.
°C | - /" ".-. 14 °C |

```

(continues on next page)

(continued from previous page)

[illegible]

---

### wtr.py API documentation

---

**getWtr** (*[where=None[, loc="en"]]*)

A function that returns a string with wtr.in's response.

*where* is None or a string that contains city name, (ex. `paris`) any location, (+ for spaces, ex. `~Eiffel+tower`) Unicode name of any location in any language, (ex. ) airport code, (3 letters, ex. `muc`) domain name, (ex. `@stackoverflow.com`) area codes (ex. `94107`) or GPS coordinates (ex. `-78.46, 106.79`)

If *where* is not given or None, then location will be auto-detected by wtr.in depending on your IP.

*loc* is the language code (ex. `en`, `fr`. Look for the full list of supported languages [here](#)) that you want wtr.in respond in.

wtr.py is a Python package that shows you forecast using wtr.in.

CLI is coming soon.





## G

`getWttr()` (*built-in function*), 3