

---

# PySlot Documentation

*Release 1.0.0*

**Julien Kauffmann**

February 17, 2016



<b>1</b>	<b>Installation</b>	<b>3</b>
<b>2</b>	<b>Basic usage</b>	<b>5</b>
<b>3</b>	<b>Table of contents</b>	<b>7</b>
3.1	API . . . . .	7
<b>4</b>	<b>Indices and tables</b>	<b>9</b>



*PySlot* a dead-simple signal/slot library for Python.



---

## Installation

---

You may install it by using *pip*:

```
pip install pyslot
```



---

## Basic usage

---

*PySlot* provides two signal classes:

- `Signal` for basic signals in mono-threaded code.
- `ThreadSafeSignal` for cross-thread signal instance usage.

Both have the same interface and can be used like so:

```
from pyslot import Signal

def greet(name, msg):
    print("{name} says: {msg}".format(name=name, msg=msg))

signal = Signal()
signal.connect(greet)

signal.emit("alice", msg="Hi Bob !")
signal.emit("bob", msg="Hi Alice !")
```

The `connect` function takes a weak-reference to any callable that will in turn be called whenever the `emit` method gets called.

A signal can be connected to several callables, which will all be called *in their registration order*.

It is also possible to disconnect a callable from the signal by calling the `disconnect` method with that callable as the single argument. Since only a weak-reference to the callable is kept, destroying the callable will implicitly disconnect it from the signal as well.

---

**Note:** No matter what variant of the *Signal* class you use, it is **always** safe for a callable to call `disconnect`, be it for itself or for another callable.

---



---

**Table of contents**

---

## **3.1 API**

*PySlot* comes with two different signal classes:



---

## Indices and tables

---

- `genindex`
- `modindex`
- `search`