

---

# **astrolabe Documentation**

***Release 0.3.0***

**Timothee Peignier**

June 13, 2016



<b>1</b>	<b>Table Of Contents</b>	<b>3</b>
1.1	Interval . . . . .	3
<b>2</b>	<b>Indices and tables</b>	<b>5</b>
	<b>Python Module Index</b>	<b>7</b>



Fast, high resolution timer library for recording performance metrics.



---

## Table Of Contents

---

### 1.1 Interval

**class** `astrolabe.interval.Interval`

This is the lowest level timing mechanism available.

It allows for easy measuring based upon a block:

```
interval = Interval()
with interval:
    ...
```

Or measuring something specifically:

```
interval = Interval()
interval.start()
duration = interval.stop()
```

Allocating and starting an interval can be done in one method call with:

```
interval = Interval.now()
```

**duration**

Returns the integer value of the interval, the value is in milliseconds.

If the interval has not had stop called yet, it will report the number of milliseconds in the interval up to the current point in time.

**duration\_so\_far**

Return how the duration so far.

**Returns** the duration from the time the Interval was started if the interval is running, otherwise False.

**classmethod** `now()`

Create an interval that has already started

**running**

Returns whether or not the interval is running or not.

This means that it has started, but not stopped.

**split()**

Immediately stop the current interval and start a new interval that has a `start_instant` equivalent to the `stop_interval` of self

**start ()**

Mark the start of the interval.

Calling start on an already started interval has no effect. An interval can only be started once.

**Returns** `True` if the interval is truly started `True` otherwise `False`.

**start\_instant**

The integer representing the start instant of the Interval.

This value is not useful on its own. It is a platform dependent value.

**started**

Returns whether or not the interval has been started.

**stop ()**

Mark the stop of the interval.

Calling stop on an already stopped interval has no effect. An interval can only be stopped once.

**Returns** the duration if the interval is truly stopped otherwise `False`.

**stop\_instant**

The integer representing the stop instant of the Interval.

This value is not useful on its own. It is a platform dependent value.

**stopped**

Returns whether or not the interval has been stopped.



---

## Indices and tables

---

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



**a**

`astrolabe.interval`, 3



## A

`astrolabe.interval` (module), 3

## D

`duration` (`astrolabe.interval.Interval` attribute), 3

`duration_so_far` (`astrolabe.interval.Interval` attribute), 3

## I

`Interval` (class in `astrolabe.interval`), 3

## N

`now()` (`astrolabe.interval.Interval` class method), 3

## R

`running` (`astrolabe.interval.Interval` attribute), 3

## S

`split()` (`astrolabe.interval.Interval` method), 3

`start()` (`astrolabe.interval.Interval` method), 3

`start_instant` (`astrolabe.interval.Interval` attribute), 4

`started` (`astrolabe.interval.Interval` attribute), 4

`stop()` (`astrolabe.interval.Interval` method), 4

`stop_instant` (`astrolabe.interval.Interval` attribute), 4

`stopped` (`astrolabe.interval.Interval` attribute), 4