

---

# **pyinter Documentation**

***Release 0.1***

**Inti Ocean**

August 01, 2015



|          |  |          |
|----------|--|----------|
| <b>1</b> | <b><i>Interval</i> Class</b>                   | <b>3</b> |
| 1.1      | <i>Interval</i> Construction helpers . . . . . | 4        |
| <b>2</b> | <b><i>IntervalSet</i> Class</b>                | <b>5</b> |
| <b>3</b> | <b>Indices and tables</b>                      | <b>7</b> |
|          | <b>Python Module Index</b>                     | <b>9</b> |



Pyinter is a python interval library which deals with interval arithmetic and sets of intervals (discontinuous ranges).



---

## Interval Class

---

**class** `pyinter.Interval`

Instances of *Interval* provide the following operations:

**Standard comparison operators:** `<=`, `<`, `==`, `!=`, `>`, `>=`

Note: comparison is performed solely on the lower value of the *Interval*.

**`x in i`**

Tests `x` to see if it is in the range specified by the *Interval* `i`.

**`interval | other`**

Performs interval union just as `union()`

`Interval.intersection(other)`

Returns a new *Interval* representing the intersection of this *Interval* with the other *Interval*

**`interval & other`**

Performs interval intersection just as `intersect()`

**`interval + other`**

Performs interval union just as `union()`

`Interval.union(other)`

Returns a new *Interval* or an *IntervalSet* representing the union of this *Interval* with the other *Interval*.

If the two intervals are overlapping then this will return an *Interval*, otherwise this returns an *IntervalSet*.

**`interval - other`**

Performs interval difference just as `difference()`

`Interval.difference(other)`

Returns a new *Interval* or an *IntervalSet* representing the subtraction of this *Interval* with the other *Interval*.

The result will contain everything that is contained by the left interval but not contained by the second interval.

If the *other* interval is enclosed in this one then this will return a *IntervalSet*, otherwise this returns a *Interval*.

`Interval.empty()`

## 1.1 *Interval* Construction helpers

`pyinter.interval.closed(lower_value, upper_value)`

Helper function to construct an interval object with closed lower and upper.

For example:

```
>>> closed(100.2, 800.9)
[100.2, 800.9]
```

`pyinter.interval.closedopen(lower_value, upper_value)`

Helper function to construct an interval object with a closed lower and open upper.

For example:

```
>>> closedopen(100.2, 800.9)
[100.2, 800.9)
```

`pyinter.interval.open(lower_value, upper_value)`

Helper function to construct an interval object with open lower and upper.

For example:

```
>>> open(100.2, 800.9)
(100.2, 800.9)
```

`pyinter.interval.openclosed(lower_value, upper_value)`

Helper function to construct an interval object with a open lower and closed upper.

For example:

```
>>> openclosed(100.2, 800.9)
(100.2, 800.9]
```



---

## *IntervalSet* Class

---

**class** `pyinter.IntervalSet` (*iterable=None*)

A class to hold collections of intervals, otherwise known as discontinuous ranges

**add** (*other*)

Add an Interval to the IntervalSet by taking the union of the given Interval object with the existing Interval objects in self.

This has no effect if the Interval is already represented. :param other: an Interval to add to this IntervalSet.

**difference** (*other*)

Subtract an Interval or IntervalSet from the intervals in the set.

**intersection** (*other*)

Returns a new IntervalSet which represents the intersection of each of the intervals in this IntervalSet with each of the intervals in the other IntervalSet. :param other: An IntervalSet to intersect with this one.

**union** (*other*)

Returns a new IntervalSet which represents the union of each of the intervals in this IntervalSet with each of the intervals in the other IntervalSet :param other: An IntervalSet to union with this one.



---

## Indices and tables

---

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



**p**

`pyinter.__init__`, 3  
`pyinter.interval`, 4



## A

`add()` (`pyinter.IntervalSet` method), 5

## C

`closed()` (in module `pyinter.interval`), 4

`closedopen()` (in module `pyinter.interval`), 4

## D

`difference()` (`pyinter.IntervalSet` method), 5

`difference()` (`pyinter.pyinter.Interval.Interval` method), 3

## E

`empty()` (`pyinter.pyinter.Interval.Interval` method), 3

## I

`intersect()` (`pyinter.pyinter.Interval.Interval` method), 3

`intersection()` (`pyinter.IntervalSet` method), 5

`IntervalSet` (class in `pyinter`), 5

## O

`open()` (in module `pyinter.interval`), 4

`openclosed()` (in module `pyinter.interval`), 4

## P

`pyinter.__init__` (module), 1

`pyinter.Interval` (class in `pyinter.__init__`), 3

`pyinter.interval` (module), 4

## U

`union()` (`pyinter.IntervalSet` method), 5

`union()` (`pyinter.pyinter.Interval.Interval` method), 3