
SmartRange

PythonCoderAS

Sep 24, 2023

CONTENTS

1 SmartRange Class	3
2 Exceptions	7
Python Module Index	9
Index	11

SmartRange is a simple module designed to make working with ranges not suck. It provides a single class that represents a list of `range` objects.

SMARTRANGE CLASS

```
class smart_range.SmartRange(range_str: str, *, min_val: Optional[int] = None, max_val: Optional[int] = None)
```

The class that represents a range of values. Input consists of a string containing the range.

A range string is a comma-separated list of range items. There are 5 types of range items supported by the class:

1. Classic range with a defined start and finish. Example: 1-5

Warning: Range items cannot have a negative difference between the start and finish values.

2. Range with a defined start and an undefined finish. This requires the `max_val` parameter to be supplied to the `__init__()` method. This type of range can only show up once at the end of the range string. Example: 1-
1-
3. Range with an undefined start and a defined finish. If this item is the first range in the range string, it requires the `min_val` parameter to be supplied to the `__init__()` method. Otherwise, it will assume that the starting value is one more than the previous range item.
 - Example at the start of the range string with a defined `min_val` parameter: -5 -> <min_val>-5
 - Example where the range string is 1-5, -8 -> 1-5, 6-8
4. A plus symbol and a number. This will indicate a range where the end of the previous range plus one is the starting number, and the end of the range is the starting number plus the provided number minus one. If specified at the beginning of the range string, it requires the `min_val` parameter to be supplied to the `__init__()` method.
 - Example at the start of the range string with a defined `min_val` parameter: +5 -> <min_val> - <min_val+5>
 - Example where the range string is 1-5, +8 -> 1-5, 6-13
5. A singular dash, indicating a range with an undefined start and an undefined finish. This requires the `min_val` and the `max_val` parameter to be supplied to the `__init__()` method. Additionally, this item can only show up once at the end of a range string. Example: -

```
__contains__(item: int) → bool
```

Returns `True` if the number is in the range, `False` otherwise.

```
__init__(range_str: str, *, min_val: Optional[int] = None, max_val: Optional[int] = None)
```

Create a new `SmartRange` object.

Parameters

- `range_str` (`str`) – The input range string.

- **min_val** (*Optional[int]*) – The minimum value of the range. If not supplied, range items #2 and #4 cannot be supplied at the beginning of the range string.
- **max_val** (*Optional[int]*) – The maximum value of the range. If not supplied, range item #3 cannot be supplied at the end of the range string.

Warning: If a `max_val` is supplied, no item in the range string can have an end value greater than the supplied maximum value.

Raises

- **NoMinimumValueError** – If a type 2 or 4 range item is used at the beginning of the range string and no `min_val` is supplied.
- **NoMaximumValueError** – If a type 3 range item is used at the end of the range string and no `max_val` is supplied.
- **NegativeRangeError** – If a range item has a negative difference between the start and finish values.
- **ExceedsMaximumValueError** – If `max_val` is provided and a range item has an end value greater than `max_val`.

__iter__() → *Generator[int, None, None]*

Returns a generator containing all the numbers in the smart range.

Note: In order to iterate over the list of range objects, use the *ranges* attribute, such as:

```
for range_obj in smart_range.ranges:
    print(range_obj.start, range_obj.stop)
```

__len__() → *int*

Returns the total number of values in all the range items combined.

__repr__() → *str*

Returns a string representation of the range.

__weakref__

list of weak references to the object (if defined)

classmethod from_range_list(*range_list: List[range]*) → *SmartRange*

Create a new *SmartRange* object from a list of *ranges*.

Note: *range* objects exclude the `stop` parameter, so `range(1, 2)` is not the same as the range string 1-2. The range object for 1-2 is `range(1, 3)`.

Warning: When using this method, none of the checks for `__init__()` are performed. In order to process these checks, this workaround can be used:

```
smart_range = SmartRange.from_range_list([range(1, 11), range(11, 16)])
validated_range = SmartRange(str(smart_range))
```

Parameters

range_list (*List*[*range*]) – The list of *ranges*.

ranges: *List*[*range*]

The list of *ranges*. Can be externally modified to add, change, or remove ranges.

Warning: Adding a range out of order may result in unordered output from <code>__iter__()</code> .

EXCEPTIONS

exception `smart_range.exceptions.ExceedsMaximumValueError`(*start: int, end: int, max_value: int*)

Raised when a value goes past the maximum.

end: `int`

The end of the range item.

max_value: `int`

The maximum value supplied to `SmartRange.__init__()`.

start: `int`

The start of the range item.

property string: `str`

A string representation of the range item.

exception `smart_range.exceptions.NegativeRangeError`(*start: int, end: int*)

Raised when a range is specified that is negative.

end: `int`

The end of the range item.

start: `int`

The start of the range item.

property string: `str`

A string representation of the range item.

exception `smart_range.exceptions.NoMaximumValueError`(*start: int*)

Raised when a range is specified that assumes a maximum value but none if specified.

start: `int`

The start of the range item.

property string: `str`

A string representation of the range item.

exception `smart_range.exceptions.NoMinimumValueError`(*end: int*)

Raised when a range is specified that assumes a minimum value but none if specified.

end: `int`

The end of the range item.

property string: `str`

A string representation of the range item.

exception `smart_range.exceptions.NoValueExpectedError`

Raised when a value is expected but none is provided.

exception `smart_range.exceptions.RangeError`

Raised when a value is invalid.

PYTHON MODULE INDEX

S

`smart_range.exceptions`, [7](#)

Symbols

`__contains__()` (*smart_range.SmartRange* method), 3
`__init__()` (*smart_range.SmartRange* method), 3
`__iter__()` (*smart_range.SmartRange* method), 4
`__len__()` (*smart_range.SmartRange* method), 4
`__repr__()` (*smart_range.SmartRange* method), 4
`__weakref__` (*smart_range.SmartRange* attribute), 4

E

`end` (*smart_range.exceptions.ExceedsMaximumValueError* attribute), 7
`end` (*smart_range.exceptions.NegativeRangeError* attribute), 7
`end` (*smart_range.exceptions.NoMinimumValueError* attribute), 7
`ExceedsMaximumValueError`, 7

F

`from_range_list()` (*smart_range.SmartRange* class method), 4

M

`max_value` (*smart_range.exceptions.ExceedsMaximumValueError* attribute), 7
`module`
`smart_range.exceptions`, 7

N

`NegativeRangeError`, 7
`NoMaximumValueError`, 7
`NoMinimumValueError`, 7
`NoValueExpectedError`, 7

R

`RangeError`, 8
`ranges` (*smart_range.SmartRange* attribute), 5

S

`smart_range.exceptions`
`module`, 7
`SmartRange` (class in *smart_range*), 3

`start` (*smart_range.exceptions.ExceedsMaximumValueError* attribute), 7
`start` (*smart_range.exceptions.NegativeRangeError* attribute), 7
`start` (*smart_range.exceptions.NoMaximumValueError* attribute), 7
`string` (*smart_range.exceptions.ExceedsMaximumValueError* property), 7
`string` (*smart_range.exceptions.NegativeRangeError* property), 7
`string` (*smart_range.exceptions.NoMaximumValueError* property), 7
`string` (*smart_range.exceptions.NoMinimumValueError* property), 7