

---

# **Formatter2 Documentation**

*Release 1.3.0*

**Rick van Hattem**

**Apr 06, 2018**



---

# Contents

---


<b>1</b>	<b>Readme</b>	<b>3</b>
1.1	Introduction . . . . .	3
1.2	Install . . . . .	3
1.3	Usage . . . . .	3
<b>2</b>	<b>formatter Package</b>	<b>5</b>
2.1	formatter Package . . . . .	5
2.2	formatter Module . . . . .	7
2.3	main Module . . . . .	7
2.4	offsets Module . . . . .	7
2.5	tokens Module . . . . .	8
2.6	types Module . . . . .	9
<b>3</b>	<b>Indices and tables</b>	<b>11</b>
	<b>Python Module Index</b>	<b>13</b>



Contents:



## 1.1 Introduction

 *formatter2* is a Python formatter based on the *tokenize* library in Python. Due to a bug with line continuations we are currently running a fork of the *tokenize* library however.

During every format sequence the *compile* method is used to check if no code has been broken, but backing up is still a good idea.

The library is made to be very extendable and configurable but it's still in the beginning phase. Eventually this will become a fully featured formatting tool for Python.

## 1.2 Install

To install simply execute *python setup.py install* or *pip install formatter2*. If you want to run the tests first, run *nosetests*

## 1.3 Usage

To format all of your code recursively (MAKE BACKUPS!):

```
python-formatter -r DIRECTORY
```

Or:

```
format-python -r DIRECTORY
```

Formatting stdin is also possible:

```
cat some_python_file.py | format-python - > some_formatted_file.py
```





## 2.1 formatter Package

```
class formatter2.Formatter
    Bases: object
    format_directory (directory)
    format_file (name)
    format_path (path, recursive=False)
    classmethod format_string (string)
class formatter2.Token (offsets, tok_type, token, begin, end, line)
    Bases: object
    begin
        The begin coordinates.
        Settings this to a tuple will set the first to the row and the latter to the col.
        This returns a SmartList with the row and the column.
    begin_col
        The begin column.
    begin_row
        The begin row.
    col
        The column.
        Setting this to a tuple will set the first to the begin and the latter to the end.
        This returns a SmartList with the begin and end column.
    end
        The end.
```

Settings this to a tuple will set the first to the row and the latter to the col.

This returns a *SmartList* with the row and column.

**end\_col**

The end column.

**end\_row**

The end row.

**line**

The line.

Setting the line automatically updates the column as well.

Returns a `str()` object with the line.

**preprocess()**

`formatter2.types.TokenType.preprocess()`

**row**

The row.

Setting this to a tuple will set the first to the begin and the latter to the end.

This returns a *SmartList* with the begin and end row.

**type**

**class** `formatter2.Tokens` (*readline*)

Bases: `object`

**format()**

Format the string, expects the extra whitespace to be removed already

**classmethod** `from_fh` (*fh*)

**classmethod** `from_readline` (*readline*)

**generate\_tokens** (*readline*)

**strip()**

Strip all whitespace so we can begin formatting

**to\_str()**

Convert the tokens back to a string

**class** `formatter2.TokenOffsets` (*parent*, *default\_type=None*)

Bases: `dict`

**get** (*key*, *recurse=False*)

**get\_key** (*key*)

**update** (*other*)

**class** `formatter2.TokenOffset` (*parent*, *type\_*, *token*, *pre=0*, *post=0*, *children=None*, *end=None*,  
*pre\_collapse=True*, *post\_collapse=True*)

Bases: `object`

**copy** (*type\_=None*, *token=None*)

**end**

Set the end token in case of children.

Should either be a token or a tuple with `TokenType` and *token* which will be a string.

**parent**

The parent. `TokenOffsets`

**surround**

Surround the token with this amount of space. Setting will set the *pre* and *post* when given a tuple or `int()`.

**type**

The type. `TokenType`

**update** (*other*)

**class** `formatter2.DefaultTokenOffset` (*parent=None*)

Bases: `formatter2.offsets.TokenOffset`

**pprint** (*stream=<open file '<stderr>', mode 'w'>*)

## 2.2 formatter Module

**class** `formatter2.formatter.Formatter`

Bases: `object`

**format\_directory** (*directory*)

**format\_file** (*name*)

**format\_path** (*path, recursive=False*)

**classmethod format\_string** (*string*)

## 2.3 main Module

`formatter2.main.main` (*\*argv*)

## 2.4 offsets Module

**class** `formatter2.offsets.DefaultTokenOffset` (*parent=None*)

Bases: `formatter2.offsets.TokenOffset`

**pprint** (*stream=<open file '<stderr>', mode 'w'>*)

**class** `formatter2.offsets.TokenOffset` (*parent, type\_, token, pre=0, post=0, children=None, end=None, pre\_collapse=True, post\_collapse=True*)

Bases: `object`

**copy** (*type\_=None, token=None*)

**end**

Set the end token in case of children.

Should either be a token or a tuple with `TokenType` and *token* which will be a string.

**parent**

The parent. `TokenOffsets`

**surround**

Surround the token with this amount of space. Setting will set the *pre* and *post* when given a tuple or `int()`.

**type**

The type. `TokenType`

**update** (*other*)

**class** `formatter2.offsets.TokenOffsets` (*parent, default\_type=None*)

Bases: `dict`

**get** (*key, recurse=False*)

**get\_key** (*key*)

**update** (*other*)

`formatter2.offsets.get_token_offsets()`

## 2.5 tokens Module

**class** `formatter2.tokens.SmartList` (*\*items, \*\*kwargs*)

Bases: `list`

**copy** ()

**set** (*value*)

**class** `formatter2.tokens.Token` (*offsets, tok\_type, token, begin, end, line*)

Bases: `object`

**begin**

The begin coordinates.

Settings this to a tuple will set the first to the row and the latter to the col.

This returns a `SmartList` with the row and the column.

**begin\_col**

The begin column.

**begin\_row**

The begin row.

**col**

The column.

Setting this to a tuple will set the first to the begin and the latter to the end.

This returns a `SmartList` with the begin and end column.

**end**

The end.

Settings this to a tuple will set the first to the row and the latter to the col.

This returns a `SmartList` with the row and column.

**end\_col**

The end column.

**end\_row**

The end row.

**line**

The line.

Setting the line automatically updates the column as well.

Returns a `str()` object with the line.

**preprocess()**

`formatter2.types.TokenType.preprocess()`

**row**

The row.

Setting this to a tuple will set the first to the begin and the latter to the end.

This returns a `SmartList` with the begin and end row.

**type**

**class** `formatter2.tokens.Tokens(readline)`

Bases: `object`

**format()**

Format the string, expects the extra whitespace to be removed already

**classmethod from\_fh(fh)****classmethod from\_readline(readline)****generate\_tokens(readline)****strip()**

Strip all whitespace so we can begin formatting

**to\_str()**

Convert the tokens back to a string

## 2.6 types Module

**class** `formatter2.types.CommentTokenType(id_, name)`

Bases: `formatter2.types.TokenType`

**name** = 'COMMENT'

**preprocess(token)**

**class** `formatter2.types.DefaultTokenType`

Bases: `formatter2.types.TokenType`

**class** `formatter2.types.IndentTokenType(id_, name)`

Bases: `formatter2.types.TokenType`

**name** = 'INDENT'

**preprocess(token)**

Replace tabs with spaces as pep8 dictates

**class** `formatter2.types.StringTokenType(id_, name)`

Bases: `formatter2.types.TokenType`

**name** = 'STRING'

**preprocess** (*token*)

Preprocess the token

This automatically replaces strings with " to ' if possible

**class** `formatter2.types.TokenType` (*id\_*, *name*)

Bases: `object`

**get\_keys** ()

**preprocess** (*token*)

Preprocess the token, this can do pretty much anything including changing the column

**class** `formatter2.types.TokenTypes`

Bases: `dict`

**register** (*token\_type*)

`formatter2.types.get_token_types` ()

## CHAPTER 3

---

### Indices and tables

---

- `genindex`
- `search`





**f**

formatter2, 5  
formatter2.formatter, 7  
formatter2.main, 7  
formatter2.offsets, 7  
formatter2.tokens, 8  
formatter2.types, 9



**B**

begin (formatter2.Token attribute), 5  
 begin (formatter2.tokens.Token attribute), 8  
 begin\_col (formatter2.Token attribute), 5  
 begin\_col (formatter2.tokens.Token attribute), 8  
 begin\_row (formatter2.Token attribute), 5  
 begin\_row (formatter2.tokens.Token attribute), 8

**C**

col (formatter2.Token attribute), 5  
 col (formatter2.tokens.Token attribute), 8  
 CommentTokenType (class in formatter2.types), 9  
 copy() (formatter2.offsets.TokenOffset method), 7  
 copy() (formatter2.TokenOffset method), 6  
 copy() (formatter2.tokens.SmartList method), 8

**D**

DefaultTokenOffset (class in formatter2), 7  
 DefaultTokenOffset (class in formatter2.offsets), 7  
 DefaultTokenType (class in formatter2.types), 9

**E**

end (formatter2.offsets.TokenOffset attribute), 7  
 end (formatter2.Token attribute), 5  
 end (formatter2.TokenOffset attribute), 6  
 end (formatter2.tokens.Token attribute), 8  
 end\_col (formatter2.Token attribute), 6  
 end\_col (formatter2.tokens.Token attribute), 8  
 end\_row (formatter2.Token attribute), 6  
 end\_row (formatter2.tokens.Token attribute), 8

**F**

format() (formatter2.Tokens method), 6  
 format() (formatter2.tokens.Tokens method), 9  
 format\_directory() (formatter2.Formatter method), 5  
 format\_directory() (formatter2.formatter.Formatter method), 7  
 format\_file() (formatter2.Formatter method), 5  
 format\_file() (formatter2.formatter.Formatter method), 7

format\_path() (formatter2.Formatter method), 5  
 format\_path() (formatter2.formatter.Formatter method), 7  
 format\_string() (formatter2.Formatter class method), 5  
 format\_string() (formatter2.formatter.Formatter class method), 7  
 Formatter (class in formatter2), 5  
 Formatter (class in formatter2.formatter), 7  
 formatter2 (module), 5  
 formatter2.formatter (module), 7  
 formatter2.main (module), 7  
 formatter2.offsets (module), 7  
 formatter2.tokens (module), 8  
 formatter2.types (module), 9  
 from\_fh() (formatter2.Tokens class method), 6  
 from\_fh() (formatter2.tokens.Tokens class method), 9  
 from\_readline() (formatter2.Tokens class method), 6  
 from\_readline() (formatter2.tokens.Tokens class method), 9

**G**

generate\_tokens() (formatter2.Tokens method), 6  
 generate\_tokens() (formatter2.tokens.Tokens method), 9  
 get() (formatter2.offsets.TokenOffsets method), 8  
 get() (formatter2.TokenOffsets method), 6  
 get\_key() (formatter2.offsets.TokenOffsets method), 8  
 get\_key() (formatter2.TokenOffsets method), 6  
 get\_keys() (formatter2.types.TokenType method), 10  
 get\_token\_offsets() (in module formatter2.offsets), 8  
 get\_token\_types() (in module formatter2.types), 10

**I**

IndentTokenType (class in formatter2.types), 9

**L**

line (formatter2.Token attribute), 6  
 line (formatter2.tokens.Token attribute), 8

**M**

main() (in module formatter2.main), 7

**N**

name (formatter2.types.CommentTokenType attribute), 9  
name (formatter2.types.IndentTokenType attribute), 9  
name (formatter2.types.StringTokenType attribute), 9

**P**

parent (formatter2.offsets.TokenOffset attribute), 7  
parent (formatter2.TokenOffset attribute), 6  
pprint() (formatter2.DefaultTokenOffset method), 7  
pprint() (formatter2.offsets.DefaultTokenOffset method),  
7  
preprocess() (formatter2.Token method), 6  
preprocess() (formatter2.tokens.Token method), 9  
preprocess() (formatter2.types.CommentTokenType  
method), 9  
preprocess() (formatter2.types.IndentTokenType  
method), 9  
preprocess() (formatter2.types.StringTokenType method),  
9  
preprocess() (formatter2.types.TokenType method), 10

**R**

register() (formatter2.types.TokenTypes method), 10  
row (formatter2.Token attribute), 6  
row (formatter2.tokens.Token attribute), 9

**S**

set() (formatter2.tokens.SmartList method), 8  
SmartList (class in formatter2.tokens), 8  
StringTokenType (class in formatter2.types), 9  
strip() (formatter2.Tokens method), 6  
strip() (formatter2.tokens.Tokens method), 9  
surround (formatter2.offsets.TokenOffset attribute), 7  
surround (formatter2.TokenOffset attribute), 7

**T**

to\_str() (formatter2.Tokens method), 6  
to\_str() (formatter2.tokens.Tokens method), 9  
Token (class in formatter2), 5  
Token (class in formatter2.tokens), 8  
TokenOffset (class in formatter2), 6  
TokenOffset (class in formatter2.offsets), 7  
TokenOffsets (class in formatter2), 6  
TokenOffsets (class in formatter2.offsets), 8  
Tokens (class in formatter2), 6  
Tokens (class in formatter2.tokens), 9  
TokenType (class in formatter2.types), 10  
TokenTypes (class in formatter2.types), 10  
type (formatter2.offsets.TokenOffset attribute), 8  
type (formatter2.Token attribute), 6  
type (formatter2.TokenOffset attribute), 7  
type (formatter2.tokens.Token attribute), 9

**U**

update() (formatter2.offsets.TokenOffset method), 8  
update() (formatter2.offsets.TokenOffsets method), 8  
update() (formatter2.TokenOffset method), 7  
update() (formatter2.TokenOffsets method), 6