Intro to Python

Maverick (@mavjs)
CSFC Core Member a.k.a 0xtr0ll
Fedora Ambassador
0xn00b Python ninja

Agenda

- Warm Up!
- Get Python
- Good Practices
- Modules
- Reserved Words
- Read Input
- Operators
- If-Else, While, For
- Practical (Fibonacci Series)
- List, Tuple, Dict, Set

Warm Up!

In your favourite programming language code a simple get input and print input in <5mins :)

```
>> input_word = raw_input('Please type a word: ')
>> print(input_word)
```

Get Python!

Debian/Ubuntu:
apt-get install python -y

Red Hat/Fedora/Centos:
yum install python -y

Most Linux Distributions come with python installed.

```
python --version
Python 2.7.5

~$ python
Python 2.7.5 (default, May 16 2013, 13:44:12)
[GCC 4.8.0 20130412 (Red Hat 4.8.0-2)] on linux2
Type "help", "copyright", "credits" or "license" for
more information.
>>>
```

Designed by – Guido van Rossum
Python – Monty Python
IDLE – Eric Idle

IDLE == 'Integrated DeveLopment Environment'

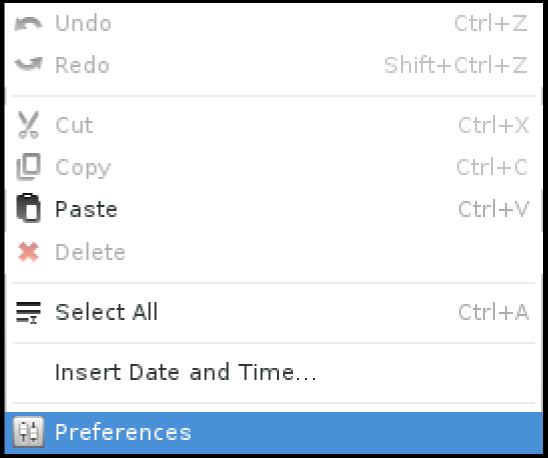
Get iPython

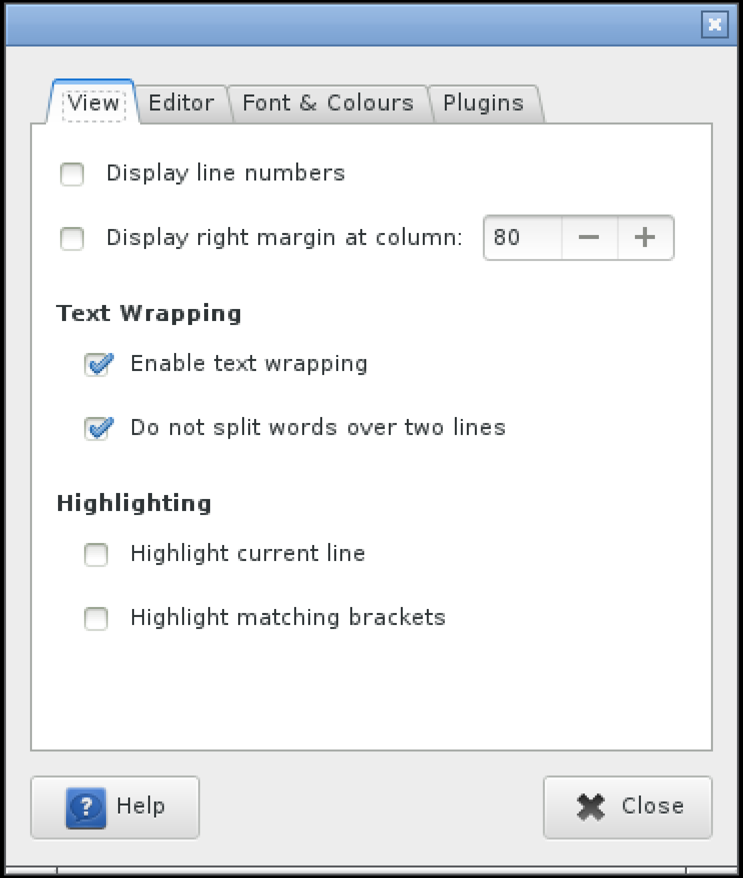
```
apt-get install ipython -y

yum install ipython -y
```

Good Practices

- Use 4 spaces for indentation.
- Never mix tab and spaces.
- One blank line between functions.
- Two blank lines between classes.
- Add a space after "," in dicts, lists, tuples, and argument lists and after ":" in dicts.
- Spaces around assignments and comparisons (except in argument list)
- No spaces just inside parentheses.

4 spaces and some





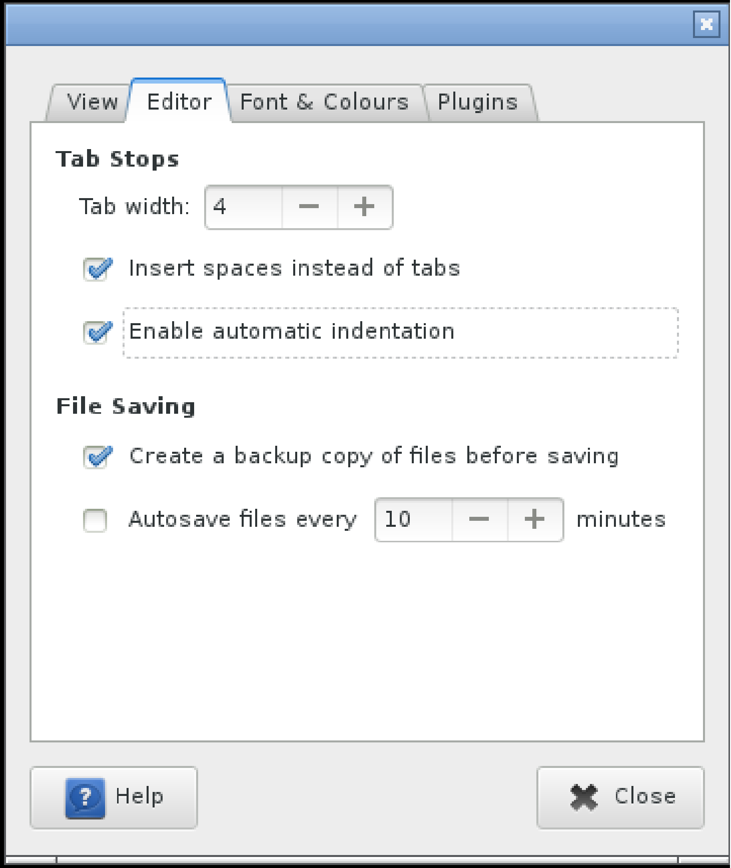
- [ ] Display line numbers
- [ ] Display right margin at column: 80

Text Wrapping

- [✓] Enable text wrapping
- [✓] Do not split words over two lines

Highlighting

- [ ] Highlight current line
- [ ] Highlight matching brackets



Tab Stops

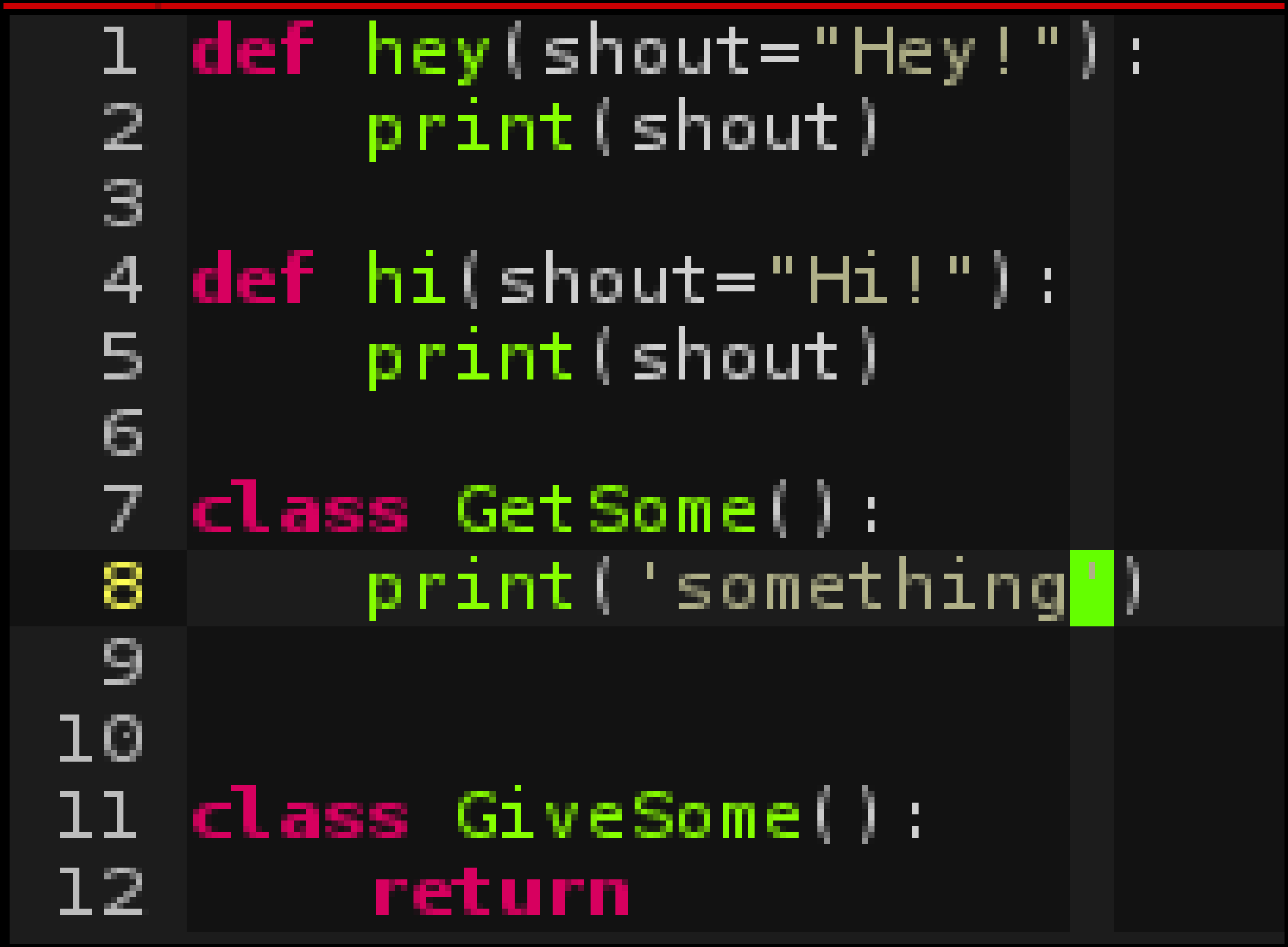
Tab width: 4

- [✓] Insert spaces instead of tabs
- [✓] Enable automatic indentation

File Saving

- [✓] Create a backup copy of files before saving
- [ ] Autosave files every 10 minutes

1BlankFunc 2BlankClass



```
def hey(shout="Hey!"):
    print(shout)

def hi(shout="Hi!"):
    print(shout)

class GetSome():
    print('something')


class GiveSome():
    return
```

Spaces after ',', comparison, assign

```
Dict = {'Name' : 'CSFC', 'Location' : 'TPM'}

List = ['a', 'b', 'c', 1, 2, 3]

Tuple = ('a', 'b', 'c', 1, 2, 3)

word = 'a'

word == 'a'
```

Modules

```
>>> import math
>>> print math.e
2.71828182846
```

Reserved Words

and	del	from	not	while
as	elif	global	or	with
assert	else	if	pass	yield
break	except	import	print	
class	exec	in	raise	
continue	finally	is	return	
def	for	lambda	try	

Operators

/ * + -

<

<=

>

>=

==

!=

and

or

not

If-Else

```
>>> if True:
...     print('hi!')
... else:
...     print('bye!')
...
hi!
>>>
```

If-elif-else

```
>>> now = 12
>>> if now < 12:
...     print('goedemorgen!')
... elif now == 12 and now < 18:
...     print('goedemiddag!')

... else:
...     print('goedenavond!')
...
goedemiddag!
>>>
```

While

```
>>> n = 0
>>> while n < 5:
...     print n
...     n += 1
...
0
1
2
3
4
```

For

```
 >>> a = ['Python', 'is', 'OHSM!']
>>> for x in a:
...     print x,
...
Python is OHSM!
```

Apply your knowledge (1) ;)

Write^h^h^hCode a fibonacci series for 20 digits

```
>>> a, b = 0, 1
>>> while b < 20:
...     print b
...     a, b = b, a + b
...
1
1
2
3
5
8
13

>>>
```

List

word_list = ['a', 'b', 'c', 'd', 'e']

Like a zero-based array, non-empty array

word_list[0] = 'a'
word_list[-1] = 'e'
word_list[0:3] = ['a', 'b', 'c']
word_list[:3] = ['a', 'b', 'c']

```
word_list[0] = 'a'
word_list[-1] = 'e'
word_list[0:3] = ['a', 'b', 'c']
word_list[:3] = ['a', 'b', 'c']

word_list.append('f')
['a', 'b', 'c', 'd', 'e', 'f']

word_list2 = ['g', 'h', 'i']
word_list.extend(word_list2)
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i']
```

Tuple

word_tuple = ('a', 'b', 'c', 'd')

A tuple is an immutable list. A tuple can not be changed in any way once it is created.

Zero-based indices.

word_tuple[0] = 'a'
word_tuple[0:1] = 'a'
'e' in word_tuple

Dictionaries

```
word_dict = {'Name' : 'Maverick', 'Location' : 'KUL'}
```

Dictionaries have keys and values. Like a Hashtable Class in Java.

```
word_dict['Name'] = 'Maverick'
```

Apply your knowledge (2) ;)

Questions?

End of Intro

Outro:
1) Functions
2) Classes
3) $$$$$

Likes what I do?
tip me @ https://www.gittip.com/mavjs/