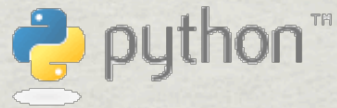


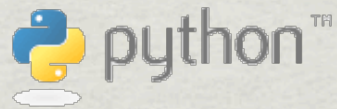
Python for Astronomers

Lauren Palladino
February 15, 2011

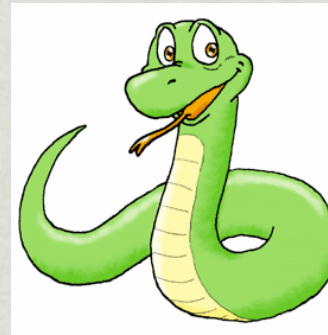


What is Python?

- ✱ **Interpreted:** Do not need to compile your program before executing it.
- ✱ **Interactive:** Can sit at a prompt and interact with the interpreter directly.
- ✱ **Object-Oriented** Style of programming that encapsulates code within objects.



About Python



CSI.UPENN.EDU

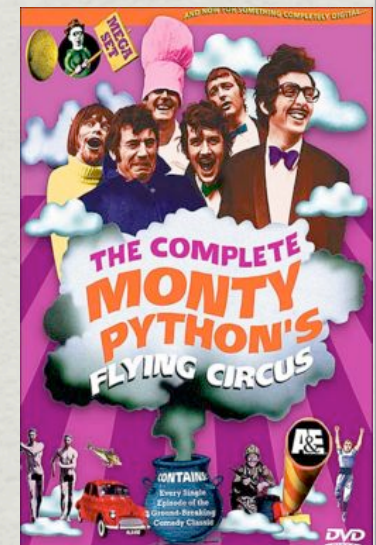
- * Developed by Guido van Rossum in 1989
- * It is **FREE!!**

<http://www.python.org/download/>

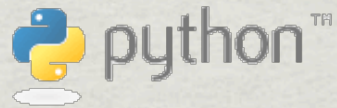
- * Designed to be fun to use and easy to learn.
- * Named after Monty Python's Flying Circus



WWW.PYTHON.ORG

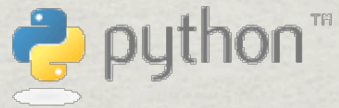


WWW.SEARCH.COM



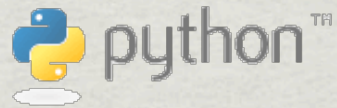
Today's Topics

- * how to run Python interactively
- * Assignment
- * Input/Output
- * Mathematical operators
- * Data Types and Conversions
- * Conditional Statements



Reserved Words

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

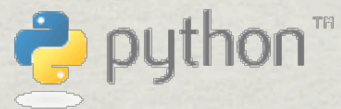


Interactive Python

- ✱ To begin an interactive Python session, type python on the command line.

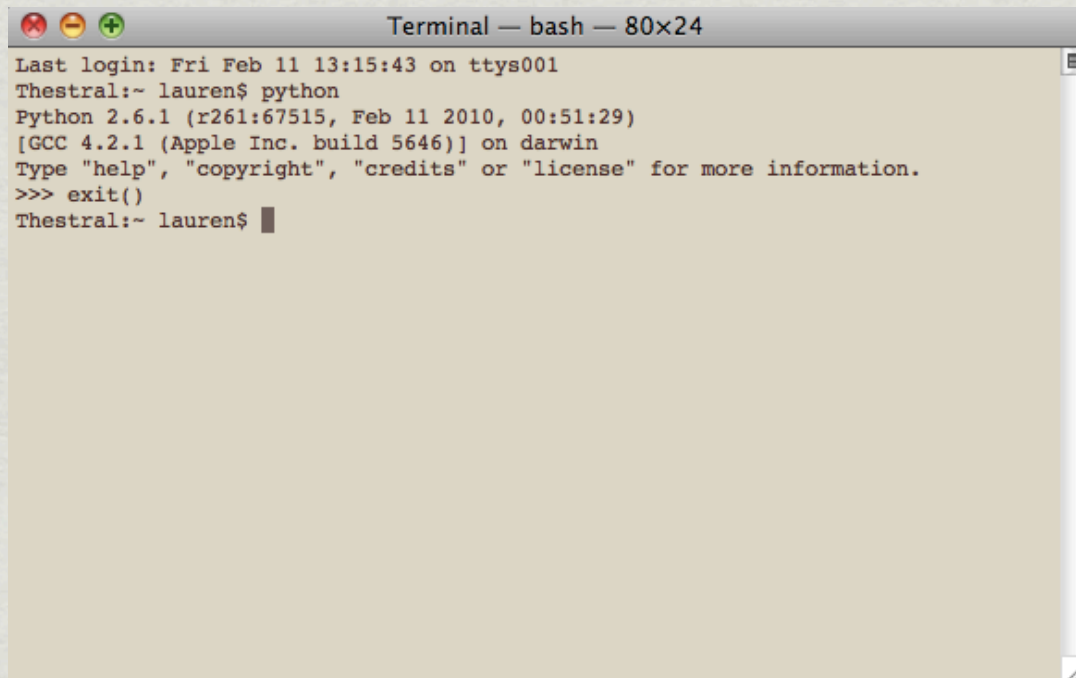
A screenshot of a terminal window titled "Terminal — bash — 80x24". The window shows the following text:

```
Last login: Fri Feb 11 13:15:33 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

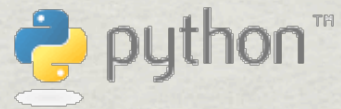


Interactive Python

- * To exit an interactive Python session: `ctrl-d`, `exit()`, or `quit()`

A screenshot of a terminal window titled "Terminal — bash — 80x24". The window shows a shell prompt "Thestral:~ lauren\$" where the user has entered "python". The terminal displays the Python version "Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)" and the compiler "[GCC 4.2.1 (Apple Inc. build 5646)] on darwin". It also shows the instruction "Type 'help', 'copyright', 'credits' or 'license' for more information." followed by the user entering ">>> exit()". The terminal then returns to the shell prompt "Thestral:~ lauren\$".

```
Terminal — bash — 80x24
Last login: Fri Feb 11 13:15:43 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
Thestral:~ lauren$
```



Assignment

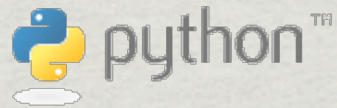
* `<variable> = <expression>`

* for example:

```
>>> number = 5
```

```
>>> number
```

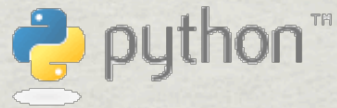
```
5
```

Assignment

- * Variables can be reassigned. Python will retain the most recent assignment.
- * for example: swap values of x and y

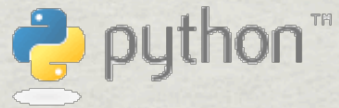
```
Terminal — bash — 80x24
Last login: Fri Feb 11 13:39:27 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> x=5
>>> x
5
>>> y=12
>>> y
12
>>> x=y
>>> y=x
>>> x,y
(12, 12)
>>> █
```



Assignment

- * Variables can be reassigned. Python will retain the most recent assignment.
- * for example: swap values of x and y

```
Terminal — bash — 80x24
Last login: Fri Feb 11 14:14:26 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> x=5
>>> x
5
>>> y=12
>>> y
12
>>> temp=x
>>> x=y
>>> y=temp
>>> x,y
(12, 5)
>>> █
```



Multiple Assignment

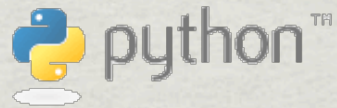
* `<var1>, <var2>, ... = <expr1>, <expr2>, ...`

* for example:

```
>>> x, y = 5, 12
```

```
>>> y
```

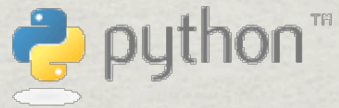
```
12
```



Multiple Assignment

- * Variables can be reassigned. Python will retain the most recent assignment.
- * for example: swap values of x and y

```
Terminal — bash — 80x24
Last login: Mon Feb 14 11:41:07 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> x, y = 5, 12
>>> x, y = y, x
>>> x, y
(12, 5)
>>> █
```



Input & Output

* `<variable> = input(<prompt>)`

```
>>> name = input("What is your name?: ")
```

What is your name?: "Lauren"

```
>>> print "Hello,", name
```

Hello, Lauren

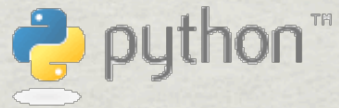
* `<variable> = raw_input(<prompt>)`

```
>>> name = raw_input("What is your name?: ")
```

What is your name?: Lauren

```
>>> print "Hello,", name
```

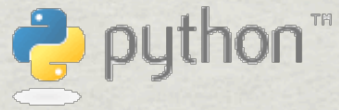
Hello, Lauren



Operators

MATHEMATICAL

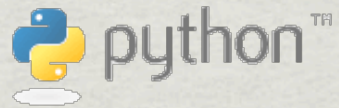
- + Addition
- Subtraction
- * Multiplication
- / Division
- ** Exponents
- % Remainder
- abs() Absolute Value



Operators

RELATIONAL

==	Equal to	and	Logical and
!=	Not equal to	or	Logical or
>	Greater than		
<	Less than		
>=	Greater than or equal to		
<=	Less than or equal to		



Operators

EXAMPLES

* $2 + 2 * 3$

* $(2 + 2) * 3$

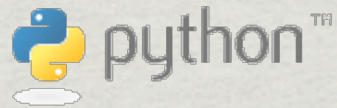
* $3.1415 * 2$

* `print 3.1415 * 2`

* $5/2$

* $5.0/2$

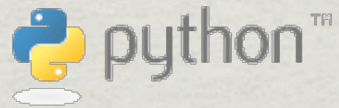
* 2^{100}



Data Types

NUMBERS

- * integer: int, 1 2 3 4
- * floating-point: float, 1. 2.0 3.1415
- * unlimited precision: long, 2^{100}



Type Conversion

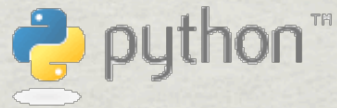
* to float: float()

```
>>> x=float(5)
```

```
>>> x/2
```

* to integer: int()

* to string: str()

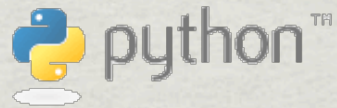


Data Types

STRINGS

- * Immutable sequence of characters
- * examples: “Spam”, “eggs”, ‘spam and eggs’

```
Terminal — bash — 80x24
Last login: Fri Feb 11 14:32:31 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> s= "Spam"
>>> █
```

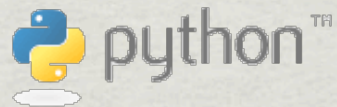


Data Types

STRINGS

- * To get the length of a string: `len(stringname)`

```
Terminal — bash — 80x24
Last login: Fri Feb 11 14:32:31 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> s= "Spam"
>>> len(s)
4
>>> █
```

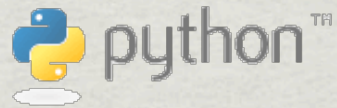


Data Types

STRINGS

- * Indexing: items are ordered left-to-right, beginning with 0
- * `stringname[index]`

```
Terminal — bash — 80x24
Last login: Fri Feb 11 14:32:31 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> s= "Spam"
>>> len(s)
4
>>> s[0]
's'
>>> s[2]
'a'
>>> s[-1]
'm'
>>> █
```

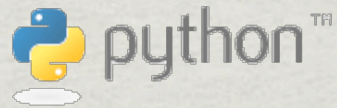


Data Types

STRINGS

- * Slicing: extracts whole sections
- * `stringname[begin:end]` Try: `S[1:3]`

```
Terminal — bash — 80x24
Last login: Fri Feb 11 14:32:31 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> S= "Spam"
>>> len(S)
4
>>> S[0]
'S'
>>> S[2]
'a'
>>> S[-1]
'm'
>>> S[1:3]
'pa'
>>> █
```

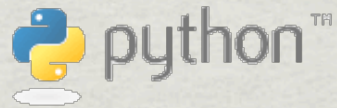


Data Types

STRINGS

- * Concatenation: join two string with +
- * Repitition: repeat a string with *

```
Terminal — bash — 80x24
Last login: Fri Feb 11 14:32:31 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> s= "Spam"
>>> len(s)
4
>>> s[0]
's'
>>> s[2]
'a'
>>> s[-1]
'm'
>>> s[1:3]
'pa'
>>> s + 'Eggs'
'SpamEggs'
>>> s * 4
'SpamSpamSpamSpam'
>>> █
```

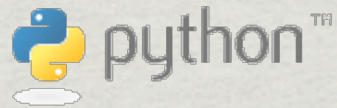


Data Types

LISTS

- * Mutable collection of objects [`<element1>`,`<element2>`,...]
- * example: `[1, 'spam', 4.0, 'U']`

```
Terminal — bash — 80x24
Last login: Mon Feb 14 10:48:36 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> L = [1, 'spam', 4.0, 'U']
>>> █
```

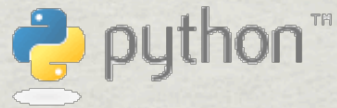



Data Types

LISTS

- * Indexing and slicing just like strings
- * Try: `L[1]` and `L[0:-1]`

```
Terminal — bash — 80x24
Last login: Mon Feb 14 10:48:36 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> L = [1, 'spam', 4.0, 'U']
>>> L[1]
'spam'
>>> L[0:-1]
[1, 'spam', 4.0]
>>> █
```

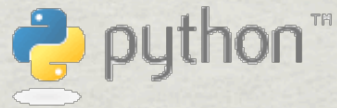


Data Types

LISTS

- * Can grow and shrink on command
- * Try: `L.append('eggs')` and `del L[3]`

```
Terminal — bash — 80x24
Last login: Mon Feb 14 10:48:36 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> L = [1, 'spam', 4.0, 'U']
>>> L[1]
'spam'
>>> L[0:-1]
[1, 'spam', 4.0]
>>> L.append('eggs')
>>> L
[1, 'spam', 4.0, 'U', 'eggs']
>>> del L[3]
>>> L
[1, 'spam', 4.0, 'eggs']
>>> █
```

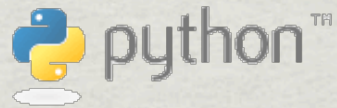


Data Types

LISTS

- * Can create multidimensional lists.
- * Start with an empty list

```
Terminal — bash — 80x24
Last login: Tue Feb 15 10:19:18 on ttys000
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> M = []
>>> █
```

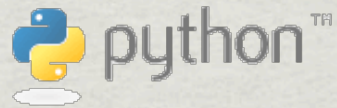


Data Types

LISTS

- * Can create multidimensional lists.
- * Add a new list, [1, 2, 3]

```
Terminal — bash — 80x24
Last login: Tue Feb 15 10:19:18 on ttys000
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> M = []
>>> M.append([1, 2, 3])
>>> M
[[1, 2, 3]]
>>> █
```

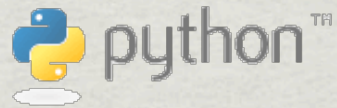


Data Types

LISTS

- * Can create multidimensional lists.
- * Add another list, [4, 5, 6]

```
Terminal — bash — 80x24
Last login: Tue Feb 15 10:19:18 on ttys000
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> M = []
>>> M.append([1, 2, 3])
>>> M
[[1, 2, 3]]
>>> M.append([4, 5, 6])
>>> M
[[1, 2, 3], [4, 5, 6]]
>>> █
```

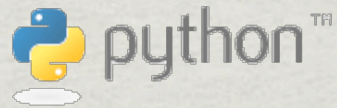


Data Types

LISTS

- * Can create multidimensional lists.
- * Index the list, Try extracting the 3...

```
Terminal — bash — 80x24
Last login: Tue Feb 15 10:19:18 on ttys000
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> M = []
>>> M.append([1, 2, 3])
>>> M
[[1, 2, 3]]
>>> M.append([4, 5, 6])
>>> M
[[1, 2, 3], [4, 5, 6]]
>>> M[0][2]
3
>>> █
```

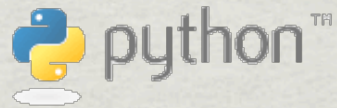


Data Types

LISTS

- * Lists are dynamic and heterogeneous.
- * Don't need to allocate space, and can mix types.

```
>>> A = []
>>> A.append([1,2,3])
>>> A
[[1, 2, 3]]
>>> A.append(['spam','eggs'])
>>> A
[[1, 2, 3], ['spam', 'eggs']]
>>> A.append([4.0, 'spam', [1,2,'string'], 5])
>>> A
[[1, 2, 3], ['spam', 'eggs'], [4.0, 'spam', [1, 2, 'string'], 5]]
>>> █
```

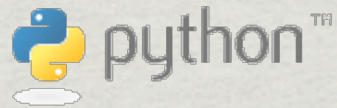


Data Types

TUPLES

- * Like lists, but immutable (<element1>,<element2>,...)
- * example: (1, 'spam', 4.0, 'U')

```
Terminal — bash — 80x24
Last login: Mon Feb 14 16:25:27 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> T = (1, 'spam', 4.0, 'U')
>>> █
```

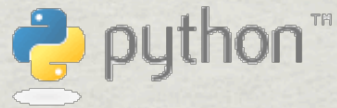



Data Types

TUPLES

- * Length, Concatenation, Indexing
- * Try: `len(T)` `T+(5,6)` `T[1]`

```
Terminal — bash — 80x24
Last login: Mon Feb 14 11:18:03 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> T = (1, 'spam', 4.0, 'U')
>>> len(T)
4
>>> T + (5, 6)
(1, 'spam', 4.0, 'U', 5, 6)
>>> T[1]
'spam'
>>> █
```

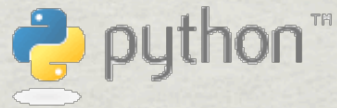


Data Types

TUPLES

- * Change or remove elements
- * Try: `del T[2]`

```
Terminal — bash — 80x24
Last login: Mon Feb 14 16:25:27 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> T = (1, 'spam', 4.0, 'U')
>>> len(T)
4
>>> T + (5, 6)
(1, 'spam', 4.0, 'U', 5, 6)
>>> T[1]
'spam'
>>> del T[2]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: 'tuple' object doesn't support item deletion
>>> █
```

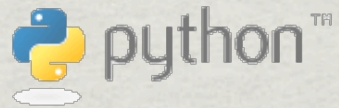


Data Types

DICTIONARIES

- * Mutable key-value pairs {<key>:<value>,<key>:<value>,...}
- * example: {'food': 'spam', 'color': 'pink', 'quantity': 4}

```
Terminal — bash — 80x24
Last login: Mon Feb 14 11:32:59 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> D = {'food': 'spam', 'color': 'pink', 'quantity': 4}
>>> █
```

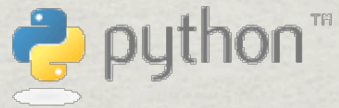


Data Types

DICTIONARIES

- * Index with keyword
- * Try: D['food']

```
Terminal — bash — 80x24
Last login: Mon Feb 14 11:32:59 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> D = {'food': 'spam', 'color': 'pink', 'quantity': 4}
>>> D['food']
'spam'
>>> █
```

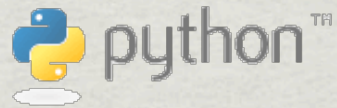


Data Types

DICTIONARIES

- * Change values
- * Try: `D['quantity'] += 1`

```
Terminal — bash — 80x24
Last login: Mon Feb 14 11:32:59 on ttys001
Thestral:~ lauren$ python
Python 2.6.1 (r261:67515, Feb 11 2010, 00:51:29)
[GCC 4.2.1 (Apple Inc. build 5646)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> D = {'food': 'spam', 'color': 'pink', 'quantity': 4}
>>> D['food']
'spam'
>>> D['quantity'] += 1
>>> D
{'food': 'spam', 'color': 'pink', 'quantity': 5}
>>> █
```



Conditional Statements

if <condition>:

<statements>

```
>>> if 2 > 1:
```

elif <condition2>:

```
    print "Hello, World"
```

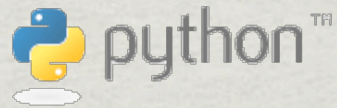
<statements>

```
>>> else:
```

else <condition3>:

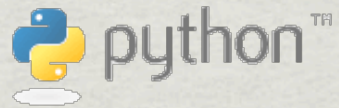
```
    print "Something went wrong."
```

<statements>



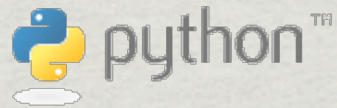
Example

```
>>> spam = raw_input("2 letter word for (and rhymes with) you & me: ")
>>> guido = [spam, 'eggs']
>>> if guido[1] == 'eggs':
>>>     guido[1] = 'are the'
>>> x = guido
>>> sausage = raw_input("Like a time of day, but begins with 'k': ")
>>> y = sausage + 's'
>>> bacon = raw_input("Fill in the blank in this children's game: Simon ____: ")
>>> z = ['who', bacon[0:-1]]
>>> print x, y, z, "Ni!"
```



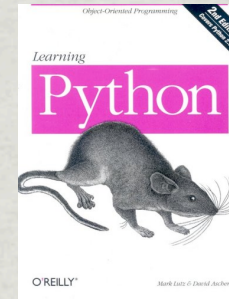
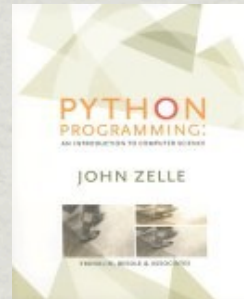
Next Week's Topics

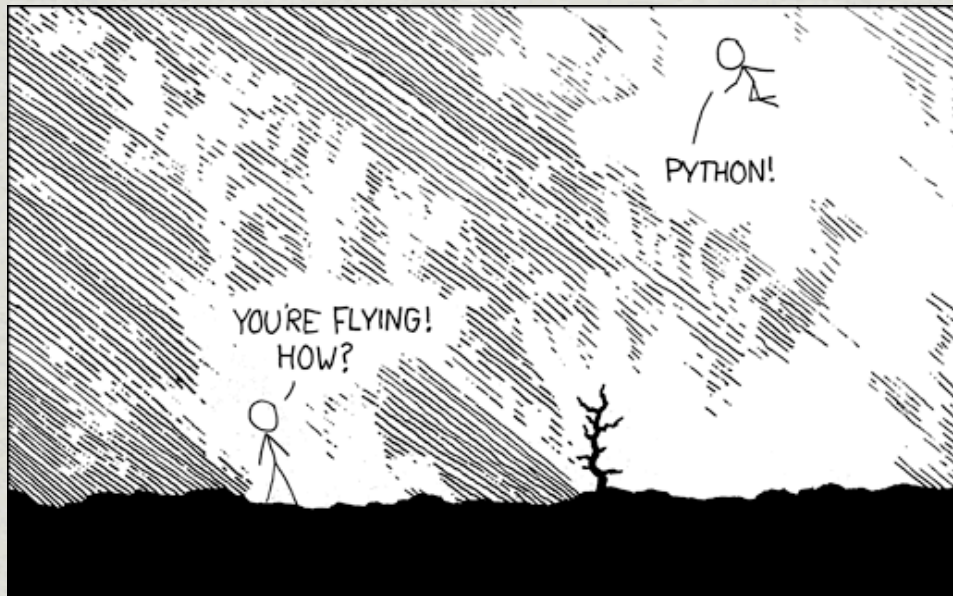
- * Scripting
- * Functions
- * Loops
- * Reading & Writing Files
- * Libraries
 - * math, string, os, PyFits



Useful References

- * Python Programming: An Introduction to Computer Science, by John Zelle
- * Learning Python, by Mark Lutz
- * <http://www.python.org/>
- * [http://en.wikipedia.org/wiki/Python_\(programming_language\)#Standard_library](http://en.wikipedia.org/wiki/Python_(programming_language)#Standard_library)
- * <http://www.tutorialspoint.com/python/index.htm>
- * http://www.astro.umd.edu/~chapman/python_tutorial.html
- * <http://www.sthurlow.com/python/>
- * <http://pytut.infogami.com/node4.html>





I LEARNED IT LAST NIGHT! EVERYTHING IS SO SIMPLE!
HELLO WORLD IS JUST
`print "Hello, world!"`

A stick figure is flying in the sky.

I DUNNO...
DYNAMIC TYPING?
WHITESPACE?

COME JOIN US!
PROGRAMMING IS FUN AGAIN!
IT'S A WHOLE NEW WORLD UP HERE!

BUT HOW ARE YOU FLYING?

A stick figure is on the ground.

I JUST TYPED
`import antigravity`

THAT'S IT?

... I ALSO SAMPLED EVERYTHING IN THE MEDICINE CABINET FOR COMPARISON.

BUT I THINK THIS IS THE PYTHON.

A stick figure is on the ground.