

•
•
•

Structure of the talk

- What's Python?
- Utility of Python
- An Informal Introduction to Python
- More Control Flow Tools
- Data Structures
- Modules
- Input and Output
- How to learn more



python

• • • • • • • • • • •

-
-
-

What is Python?

python



•
•
•

Language features

- interpreted language

python

• • • • • • • • • • •

•
•
•

Language features

- interpreted language
- simple to use

python

• • • • • • • • • • •

•
•
•

Language features

- interpreted language
- simple to use
- very high-level language

python

• • • • • • • • • • •

•
•
•

Language features

- interpreted language
- simple to use
- very high-level language
- high-level data types built in

python

• • • • • • • • • • •

•
•
•

Utility of Python

- learn to program

python

• • • • • • • • • • •

•
•
•

Utility of Python

- learn to program
- teach to program

python

• • • • • • • • • • •

•
•
•

Utility of Python

- learn to program
- teach to program
- makes programming much easier

python

• • • • • • • • • • •

•
•
•

Utility of Python

- learn to program
- teach to program
- makes programming much easier
- rapid development

python

• • • • • • • • • • •

•
•
•

Utility of Python

- learn to program
- teach to program
- makes programming much easier
- rapid development
- prototyping

python

• • • • • • • • • • •

-
-
-

An Informal Introduction to Python

python

⋮

Using Python as a Calculator

```
>>> # This is a comment
... 7/3 # Integer division truncates
2
>>> x = y = z = 0 # Multiple assignment
>>> x + y + z
0
>>> (2 + 1j) * (1 - 3j) # Complex with j
(5-5j)
>>> _ ** 2 # Last expression in _
-50j
```



⋮

⋮

First Steps Towards Programming

```
>>> # Fibonacci series:
... # the sum of two elements
... # defines the next
... a, b = 0, 1
>>> while b < 500:
...     print b
...     a, b = b, a+b
1 1 2 3 5 8 13 21 34 55 89 144 233 377
```



⋮

⋮

More Control Flow Tools

```
>>> for i in range (2,10):  
...     if i % 2 == 0:  
...         print i,  
...     elif i % 3 == 0:  
...         continue  
...     else:  
...         break  
...  
2 4
```



⋮

⋮

Functions

```
>>> def fact (n):  
...     "Calculates n! recursively"  
...     if n:  
...         return n * fact (n-1)  
...     else:  
...         return 1  
...  
>>> map (fact, range (1, 8))  
[1, 2, 6, 24, 120, 720, 5040]
```



⋮

Data Structures: Lists

```
>>> a = ['spam', 'eggs', 100, 1234]
>>> a[0], a[3], a[-2]
('spam', 1234, 100)
>>> a[1:-1]
['eggs', 100]
>>> a[:2] + ['bacon', 2*2]
['spam', 'eggs', 'bacon', 4]
>>> a = a + ('bacon', 2*2)
>>> a
['spam', 'eggs', 100, 1234, 'bacon', 4]
```



⋮

Data Structures: Dictionaries

```
>>> tel = {'jack': 409, 'sape': 413}
>>> tel['guido'] = 412
>>> tel
{'sape': 413, 'guido': 412, 'jack': 409}
>>> tel['jack']
409
>>> tel.keys()
['guido', 'irv', 'jack']
```



•
•
•

Modules

- are files containing definitions

python

• • • • • • • • • • •

•
•
•

Modules

- are files containing definitions
- accessed with `import` statement

python

• • • • • • • • • • • •

•
•
•

Modules

- are files containing definitions
- accessed with `import` statement
- lots of standard modules

python

• • • • • • • • • • •

•
•
•

Modules

- are files containing definitions
- accessed with `import` statement
- lots of standard modules
- the `dir()` function

python

• • • • • • • • • • •

⋮

I/O: Fancier Output Formatting

```
>>> for x in range (1, 5):  
...     print string.rjust('x', 2),  
...     print string.rjust('x*x', 3),  
...     print string.rjust('x*x*x', 4)  
...  
1      1      1  
2      4      8  
3      9     27  
4     16     64
```



⋮

⋮

I/O: Reading and Writing Files

```
>>> f = open ('/tmp/workfile', 'w')
>>> print f
>>> f.readline()
'This is the first line of the file.\n'
>>> f.readline()
'Second line of the file\n'
>>> f.readline()
''
>>> f.write('This is a test\n')
>>> f.close()
```



⋮

-
-
-

How to learn more

python



⋮

Python documentation and people

- Python's standard documentation:

`http://www.python.org/doc`

- Tutorial
- Library Reference

- Mailing lists:

`http://mail.python.org/mailman/listinfo`



⋮

•
•
•

Interesting libraries

- wxPython: <http://www.wxpython.org>

python

• • • • • • • • • • •

•
•
•

Interesting libraries

- wxPython: <http://www.wxpython.org>
- Numeric Python:
<http://www.pfdubois.com/numpy>

python

• • • • • • • • • • •

•
•
•

Interesting libraries

- **wxPython:** <http://www.wxpython.org>
- **Numeric Python:**
<http://www.pfdubois.com/numpy>
- **Python Imaging Library:**
www.pythonware.com/products/pil



• • • • • • • • • • •

•
•
•

Interesting libraries

- **wxPython:** <http://www.wxpython.org>
- **Numeric Python:**
<http://www.pfdubois.com/numpy>
- **Python Imaging Library:**
www.pythonware.com/products/pil
- **Game development:** <http://www.pygame.org>



• • • • • • • • • • •

