CCP 2006 Python

Python
A language for computational physics

Peter Borcherds

University of Birmingham
Downloading

Python may be downloaded from www.python.org

VPython may be downloaded from www.vpython.org

Installation is straightforward

VPython includes a folder of examples: stills from which are shown on the next two slides

Several of the slides in this presentation show stills from VPython programs: all of them are better seen by running the programs
Surreal Stonehenge

An example from the VPython examples folder
Color sliders

Another example from the VPython examples folder
What does Python look like?

The next slide shows what a Python program looks like. What the program does should be clear to anyone familiar with a high level language such as FORTRAN. The import statement loads the visual (VPython) module. Floor and ball are visual objects with properties such as ball.velocity. The first line of a block statement ends with a colon (:) and subsequent lines are indented. The end of the block is shown by the end of the indentation.
# bounce.py
from visual import *
floor = box(length=4)
bball = sphere(pos=(0,4,0), color=color.red)
bball.velocity = vector(0,-1,0)
dt = 0.01
while 1:
    rate(100)
bball.pos += bball.velocity*dt
if bball.y < 1:
bball.velocity.y *= (-1)
else:
bball.velocity.y -= 9.8*dt
Display of program

The next slide shows the display when the program is running. The scale of the display changes with the height of the ball (autoscale = True). This can be rectified by inserting autoscale = False, as shown in the following slide.
bounce.py autoscale=True
# bounce.py

```python
from visual import *

floor = box(length=4)
ball = sphere(pos=(0,4,0), color=color.red)
bball.velocity = vector(0,-1,0)
dt = 0.01
 autoscale = False  # new line inserted

while 1:
    rate(100)
    ball.pos += ball.velocity*dt
    if ball.y < 1:
        ball.velocity.y *= (-1)
    else:
        ball.velocity.y -= 9.8*dt
```
Helpful (free) documentation

Donaldson T,
www.cs.ubc.ca/wccce/Program03/papers/Toby.html,
Python as a First Programming Language for Everyone

Downey A, Elkner J and Meyers C,
www.greenteapress.com/thinkpython,
How to Think Like a Computer Scientist: Learning with Python
This textbook may be downloaded free
Finding your way: dir()

Finding your way in Python is easy
There are helpful functions such as dir() and help()
The next 2 slides show some examples of dir()

dir() what is currently available

dir(__builtins__) available at start

dir()
    when first bracket of a function is typed, python prompts with argument of the function
`dir()  dir(__builtins__)  & dir(

```python
>>> dir()
['__builtins__', '__doc__', '__name__']
>>> dir(__builtins__)
>>> dir([object]) -> list of strings
```

Peter Borcherds 12
dir('') properties of '' (string variables)
dir(math.sin)

```
>>> dir('')
[ '__add__', '__class__', '__contains__', '__delattr__', '__doc__', '__eq__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__getslice__', '__gt__', '__hash__', '__init__', '__le__', '__len__', '__lt__', '__mod__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__reversed__', '__rmul__', '__setattribute__', '__str__', 'capitalize', 'center', 'count', 'decode', 'encode', 'endswith', 'expandtabs', 'find', 'index', 'isalnum', 'isalpha', 'isdigit', 'islower', 'isspace', 'istitle', 'isupper', 'join', 'ljust', 'lower', 'lstrip', 'rjust', 'rindex', 'replace', 'rfind', 'rsplit', 'rstrip', 'split', 'splitlines', 'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
>>> import math
>>> dir(math.sin)
[ '__call__', '__class__', '__cmp__', '__delattr__', '__doc__', '__getattribute__', '__hash__', '__init__', '__module__', '__name__', '__new__', '__reduce__', '__reduce_ex__', '__repr__', '__self__', '__setattr__', '__str__']
```
The next two pages show examples of help()
help() by itself opens a help menu
help(something) gives help about something
help()

>>> help()

Welcome to Python 2.4! This is the online help utility.

If this is your first time using Python, you should definitely check out the tutorial on the Internet at http://www.python.org/doc/tut/.

Enter the name of any module, keyword, or topic to get help on writing Python programs and using Python modules. To quit this help utility and return to the interpreter, just type "quit".

To get a list of available modules, keywords, or topics, type "modules", "keywords", or "topics". Each module also comes with a one-line summary of what it does; to list the modules whose summaries contain a given word such as "spam", type "modules spam".

help> keywords

Here is a list of the Python keywords. Enter any keyword to get more help

<table>
<thead>
<tr>
<th>and</th>
<th>else</th>
<th>import</th>
<th>raise</th>
</tr>
</thead>
<tbody>
<tr>
<td>assert</td>
<td>except</td>
<td>in</td>
<td>return</td>
</tr>
<tr>
<td>break</td>
<td>exec</td>
<td>is</td>
<td>try</td>
</tr>
<tr>
<td>class</td>
<td>finally</td>
<td>lambda</td>
<td>while</td>
</tr>
<tr>
<td>continue</td>
<td>for</td>
<td>not</td>
<td>yield</td>
</tr>
<tr>
<td>def</td>
<td>from</td>
<td>or</td>
<td></td>
</tr>
<tr>
<td>del</td>
<td>global</td>
<td>pass</td>
<td></td>
</tr>
<tr>
<td>elif</td>
<td>if</td>
<td>print</td>
<td></td>
</tr>
</tbody>
</table>
help(sin)

>>> help(sin)

Traceback (most recent call last):
  File "<pyshell#8>", line 1, in -toplevel-
    help(sin)
NameError: name 'sin' is not defined

>>> help(math.sin)
Help on built-in function sin in module math:

math.sin(x)

    Return the sine of x (measured in radians).

>>>
Python documentation

The next 7 slides show examples of Python and VPython documentation and web pages
Python documentation

Release 2.4.3
29 March 2006

- Tutorial
  (start here)

- What's New in Python
  (changes since the last major release)

- Global Module Index
  (for quick access to all documentation)

- Library Reference
  (keep this under your pillow)

- Macintosh Module Reference
  (this too, if you use a Macintosh)

- Installing Python Modules
  (for administrators)

- Distributing Python Modules
  (for developers and packagers)

- Documentation Central
  (for everyone)

- Language Reference
  (for language lawyers)

- Extending and Embedding
  (tutorial for C/C++ programmers)

- Python/C API
  (reference for C/C++ programmers)

- Documenting Python
  (information for documentation authors)

- Python How-To Guides
  (special topics)

See About the Python Documentation for information on suggesting changes.
Python documentation 2

Python Documentation

Release 2.4.3
29 March 2006

- **Tutorial**
  (start here)

- **What's New in Python**
  (changes since the last major release)

- **Global Module Index**
  (for quick access to all documentation)

- **Library Reference**
  (keep this under your pillow)

- **Macintosh Module Reference**
  (this too, if you use a Macintosh)

- **Installing Python Modules**
  (for administrators)

- **Distributing Python Modules**
  (for developers and packagers)

- **Language Reference**
  (for language lawyers)

- **Extending and Embedding**
  (tutorial for C/C++ programmers)

- **Python/C API**
  (reference for C/C++ programmers)

- **Documenting Python**
  (information for documentation authors)

- **Documentation Central**
  (for everyone)

- **Python How-To Guides**
  (special topics)
The Python Programming Language

Python® is a dynamic object-oriented programming language that can be used for many kinds of software development. It offers strong support for integration with other languages and tools, comes with extensive standard libraries, and can be learned in a few days. Many Python programmers report substantial productivity gains and feel the language encourages the development of higher quality, more maintainable code.

Python runs on Windows, Linux/Unix, Mac OS X, OS/2, Amiga, Palm Handhelds, and Nokia mobile phones. Python has also been ported to the Java and .NET virtual machines.

Python is distributed under an OSI-approved open source license that makes it free to use, even for commercial products.

The Python Software Foundation (PSF) holds and protects the intellectual property rights behind Python, underwrites the PyCon conference, and funds grants and other projects in the Python community.

Read more - or - try Python now

» IronPython 1.0RC1 released
   IronPython (Python for the .NET virtual machine) version 1.0RC1 has been released
   Published: Wed, 26 Jul 2006 00:00:00 +0000

» Python 2.5 (beta 2)
   The second beta release of Python 2.5 is now available.
   Published: Tue, 11 July 2006 00:37 +1000

» SciPy 2006 - Conference for Scientific Computing
   The 5th annual SciPy Conference will be at CalTech. Some dates: abstracts
The Python Programming Language

Python® is a dynamic object-oriented programming language that can be used for many kinds of software development. It offers strong support for integration with other languages and tools, comes with extensive standard libraries, and can be learned in a few days. Many Python programmers report substantial productivity gains and feel the language encourages the development of higher quality, more maintainable code.

Python runs on Windows, Linux/Unix, Mac OS X, OS/2, Amiga, Palm Handhelds, and Nokia mobile phones. Python has also been ported to the Java and .NET virtual machines.

Python is distributed under an OSI-approved open source license that makes it free to use, even for commercial products.

The Python Software Foundation (PSF) holds and protects the intellectual property rights behind Python, underwrites the PyCon conference, and funds grants and other projects in the Python community.

Read more -or- try Python now

»IronPython 1.0RC1 released

IronPython (Python for the .NET virtual machine) version 1.0RC1 has been released

"Python has been an important part of Google since the beginning, and remains so as the system grows and evolves."

-- Peter Norvig, Google
VPython website www.vpython.org

3D Programming for Ordinary Mortals

VPython is a package that includes:

- the Python programming language
- the IDLE interactive development environment
- "Visual", a Python module that offers real-time 3D output, and is easily usable by novice programmers
- "Numeric", a Python module for fast processing of arrays

VPython is free and open-source.

Other links:

- The Python language [www.python.org](http://www.python.org)
- A module by Ruth Chabay to export a VPython scene to POV-Ray: [povexport-2005-12-06.zip](http://povexport-2005-12-06.zip)
- Contributed programs from users
- The cT archives

Peter Borcherds
Visual Module of Python

The Visual Module of VPython

VPython is the Python programming language plus a 3D graphics module called "Visual" developed by David Scherer.

To invoke the Visual module, place the following statement at the start of the file:

```python
from visual import *
```

**Introduction** for those new to Python and Visual

**Basic Display Objects**

cylinder  Start here: much of what is said here applies to other objects as well.

<table>
<thead>
<tr>
<th>Arrow</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cone</td>
<td>Frame: combining several objects into one</td>
</tr>
<tr>
<td>Pyramid</td>
<td>Faces: low-level object for special purposes</td>
</tr>
<tr>
<td>Sphere</td>
<td>Additional Attributes: visible, frame, display, class, members</td>
</tr>
<tr>
<td>Ring</td>
<td>Convenient Defaults</td>
</tr>
<tr>
<td>Box</td>
<td>Rotating an Object</td>
</tr>
<tr>
<td>Ellipsoid</td>
<td>Specifying Colors</td>
</tr>
<tr>
<td>Curve</td>
<td>Deleting an Object</td>
</tr>
<tr>
<td>Helix</td>
<td>Limiting the Animation Rate</td>
</tr>
<tr>
<td>Convex</td>
<td>Floating Division: 3/4 is 0, but 3/4.0 is 0.75 in Python</td>
</tr>
</tbody>
</table>

**Vector Computations**

* `vector`, including mag, mag2, norm, cross, dot, rotate, diff_angle
VPython Documentation

The Visual Module for 3D Graphics

- Reference manual: introduction, and all the details about using Visual
- VPython web site
- License information for the Visual module

Modules Automatically Imported by the Visual Module

- Math module: sin, cos, sqrt, etc.
- Numeric module: fast array processing (PDF); image mentioned in first section is not installed
  - Numeric web site
  Visual no longer imports the functions random, randint, and uniform from the random module

The Python Programming Language

- Python web site
Graphic displays

Graphic displays are important in computational physics. The remaining slides show some examples of VPython programs with graphic displays.
Kepler's law of equal areas

The next slide shows a program solving the equations of motion of a two body problem. It uses a modified version of the Euler method, suitable for display. Note the use of vectors simplifies the program. Features of the program are indicated by comments (#)

The following slide shows the display from a similar program, available on the VPython web site:

http://vpython.org/contributed/kepler.py
# Kepler's Laws with manual input.py  #  Peter Borcherds 060617

from visual import *  # import the vpython module

speed = input('speed of planet in range 0.7 to 1.2 ')

sun = sphere(radius = 0.1)  # all other properties have default values
planet = sphere(pos = (0,1), color = color.cyan, radius = 0.05)
exit = False  # erase the working program without deleting this file

dt = 0.025  # a convenient value for time step
month = int(1/dt)  # a convenient time interval

def trajectory(speed=1.2):  # defining the function 'trajectory', note the colon
    velocity = vector(speed, 0)  # indentation shows this is inside a function
    step = 0
    oldpos = vector(planet.pos)

    orbit = curve(pos = [oldpos, planet.pos], color = color.red, radius = 0.01)

    while not(oldpos.x < 0 and planet.pos.x>0):  # test for closure of orbit
        rate (40)  # sets rate for a comfortable display
        oldpos = vector(planet.pos)
        accel = - (planet.pos / (mag(planet.pos) ** 3))  # inverse square law
        velocity += accel * dt
        planet.pos += velocity * dt
        orbit.append(pos = planet.pos)  # update the orbit

        if step % month == 0:  # end of 'month'?
            curve(pos = [sun.pos, planet.pos], color = color.green)  # radius vector
            step += 1

    # end of indentations: end of function, loop and condition

trajectory(speed)  # call the function

''' When orbit has been plotted, you can plot another by typing
"trajectory(X)" where X is any value you want'''

Peter Borcherds 27
Kepler: law of equal areas

Period: 30.05 "months"
Initial speed: 1.2
Roots of unity

The next three slides show the basins of attraction of the roots of unity for the Newton Raphson method. The boundaries of the basins are complicated. As can be seen, to go from one basin to another, one has to pass through all the other basins. Without a graphical display, this would be difficult to visualise.
NR 3
Newton Raphson basins of attraction

fifth roots of unity
A pendulum

The next slide shows a VPython program with two graphic windows. The window on the left plots velocity against displacement (the phase plot). The window on the right shows the instantaneous position of the pendulum.
Pendulum phase plots and motion