TWINCLING™
freedom of innovation

TWINCLING.org

Python Programming Language

11th February, 2006
Hyderabad, AP India

TWINCLING™
freedom of innovation

What is Python

TWINCLING.org

- Object-oriented
- Dynamic
- Easy to learn syntax
- High-level data types
- Scripting language
- Embeddable in C/C++

Guido Von Rossum

TWINCLING™
freedom of innovation

Python vs Java (or C++)

- No compilation
  - Fast edit-build-debug cycle
- Dynamic Typing
  - No need to declare variables for use
- Easy Syntax
  - No curly braces, no semicolons, no `new` ...

TWINCLING™
freedom of innovation

Python vs Java (or C++) ...

- Embeddable
  - Scripting support for your applications
- Interactive
  - Create, view, change objects at runtime
- 50% less code
  - Compact and natural syntax
- 300% more productive
  - Closer to the way you think

TWINCLING.org

TWINCLING™
freedom of innovation

Hello World

Java

```
public class HelloWorld
{
  public static void main( String args[] )
  {
     System.out.println( "Hello World" );
  }
}
```

Python

```
print 'Servus Austria'
```

TWINCLING.org

TWINCLING™
freedom of innovation

Java + Swing

TWINCLING.org

Java

```
public class HelloSwing extends JFrame
{
   public HelloSwing( String title )
   {
      super( title );

      addWindowListener( new WindowAdapter() {
         public void windowClosing( WindowEvent ev )
         {
            exit();
         }
      } );

      JButton button = new JButton( "Servus Austria" );
      button.setPreferredSize( new Dimension( 200, 50 ) );
      button.addActionListener( new ActionListener() {
```

TWINCLING™
freedom of innovation

Java + Swing ...

Java

```
            public void actionPerformed( ActionEvent ev )
            {
               exit();
            }
         } );

         getContentPane().add( button );
      }

      public void exit(){
         System.exit( 0 );
      }

      public static void main( String args[] ){
         HelloSwing frame = new HelloSwing( "Python" );
         frame.pack();
         frame.show();
      }
   }
```


TWINCLING™
freedom of innovation

Python + XUL

Python

```
import java.lang as lang

def exit( event ):
  lang.System.exit(0)
```

```
<window title="Python" onclosing="exit()">
  <button label="Hello World"
          style="width: 200; height: 50;"
          onclick="exit()" />
</window>
```

TWINCLING.org

TWINCLING™
freedom of innovation

Why Python ?

- One language can not do it all
  - Single purpose scripting languages on the rise
    - XHTML
    - XUL (menus, toolbars, forms, grids, trees)
    - SVG (charts, maps, logos ...)
    - CSS (visual styling)
    - Xpath (xml tree node addressing)
    - SQL (data queries)
    - XSL-T (templates)
    - **Python/Scripting**

TWINCLING™
freedom of innovation

Why Python ? ...

- **Object-oriented**
  - Ideal for scripting
  - Ideal for creating gluing together components written in Java, C# or C++
  - Easy reuse through classes, polymorphism, operator overloading, multiple inheritance
- **Portable**
  - Written in ANSI C
  - Runs anywhere
  - Python scripts run on any Python runtime

TWINCLING™
freedom of innovation

Why Python ? ...

- **Mixable**
  - **Extend** Python with Components written in C++, Java, C
  - **Embed** python into your app and call it from C, C++
  - Python on windows supports COM
- **Powerful**
  - Ease of use of scripting language
  - Built in object types
  - Extensive libraries
  - Automatic memory management
  - Modules, Classes and Exceptions

TWINCLING™
freedom of innovation

Why Python ? ...

- **Robust**
  - Exception Handling
  - Automatic memory management (Garbage Collection)
  - Dynamic Type Checking
  - No unsafe pointers
- **Dynamic**
  - Late binding language
  - Add methods at runtime
  - Call methods using reflection

Python features not there in Java

- syntactic sugar for lists
- syntactic sugar for maps/dictionaries
- raw strings
- for loop shortcuts (=foreach)
- named method parameters
- string formatting shortcuts

TWINCLING™
freedom of innovation

List

Java

```
List list = new LinkedList();
list.add( new Integer( 1 ) );
list.add( new Integer( 2 ) );
list.add( new Integer( 3 ) );
```

Python

```
list = [1, 2]
list.append( 3 )
```


TWINCLING™
freedom of innovation

Maps

Java

```
Map map = new HashMap();
map.put( "one", new Integer( 1 ) );
map.put( "two", new Integer( 2 ) );
map.put( "three", new Integer( 3 ) );

System.out.println( map.get( "one" ) );
```

Python

```
map = { "one" : 1, "two" : 2, "three" : 3 }
print map[ "one" ]
```


TWINCLING™
freedom of innovation

Loops shortcut

Java

```
double sum = 0.0;
for( Iterator it=nums.iterator(); it.hasNext() )
{
  sum += ((Double)it.next()).doubleValue();
}
```

Python

```
sum = 0.0
for x in nums:
  sum = sum + x
```


TWINCLING™
freedom of innovation

Named Method Parameters

Java

```
JFrame frame = new JFrame( "Servus" );
frame.setSize( new Dimension( 200, 200 ) );
frame.setVisible( true );
```

Python

```
frame = JFrame( "Servus", visible=1, size=(200,200) )
```


TWINCLING™
freedom of innovation

String Formatting Shortcuts

TWINCLING.org

Java

```
double x = 10000.0 / 3.0;
NumberFormat nf = NumberFormat.getNumberInstance();
nf.setMinimumFractionDigits( 2 );
nf.setMaximumFractionDigits( 2 );
String s = nf.format( x );
for( int i = s.length(); i < 10; i++ )
    System.out.print( ' ' );
System.out.print( s );
```

Python

```
x = 10000.0 / 3.0
print "%10.2f" % x
```

TWINCLING™
freedom of innovation

Raw strings

Java	Python
"\\$\\d+,\\d+\\."	r'\$\d+,\d+\.'
"\\s((::)(\\w+))\\b"	r'\s((::)(\w+))\b'
"c:\\sandbox\\doc\\talk"	r'\s((::)(\w+))\b'
"Christina says, \"Python”“	'Christina says, "Python”'

TWINCLING.org

In Python you can use triple-quotes (""") strings
for multiline text snippets without escaping newlines or single or double quotes

Raw String Especially Useful for Regular Expressions (Regex)

TWINCLING™
freedom of innovation

Data types

TWINCLING.org

Java

- Boolean
- Char
- Byte
- Short
- Int
- Long
- Float
- double

Python

- String
- Int
- Long
- Float
- complex

TWINCLING™
freedom of innovation

Exception Handling

Java

```
FileInputStream in =
  new FileInputStream( new File(
    name ) );
try
{
  process_file( in );
}
finally
{
  in.close();
}
```

Python

```
file = open( name )
try:
  process_file( file )
finally:
  file.close()
```


TWINCLING™
freedom of innovation

Python class

Python

Dynamic Properties/Getters

```
lass DynProps:

   def __init__( self, **args ):
       self.props = args

   def __getattr__( self, attribute ):
       return self.props[ attribute ]
```

```
capital = DynProps( austria="vienna", canada="ottawa" )

print capital.austria

capital.props[ "peru" ] = "lima"

print capital.peru
```

TWINCLING.org

TWINCLING™
freedom of innovation

Embed Python in Java

TWINCLING.org

```
import org.python.util.PythonInterpreter;
import org.python.core.*;

public class SimpleEmbedded
{
  public static void main( String args[] ) throws PyException
  {
    // create a phyton interpreter
    PythonInterpreter interp = new PythonInterpreter();

    // execute a statement
    interp.exec( "import sys" );
    interp.exec( "print sys" );
```

TWINCLING™
freedom of innovation

Embed Python in Java ...

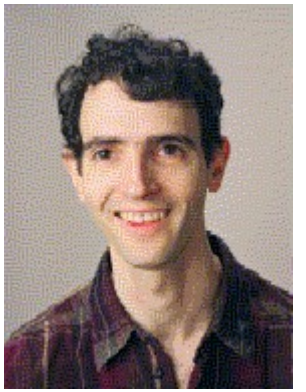
```
// create a int variable n with the value 7
    interp.set( "n", new PyInteger( 7 ));

    // print value of n
    interp.exec( "print n" );

    // assign value to new variable x
    interp.exec( "x = 2+2" );

    PyObject x= interp.get( "x" );

    // print value of x
    System.out.println( "x: " + x );
  }
}
```

TWINCLING.org

TWINCLING™
freedom of innovation

Python – C++ integration

- Boost.Python Library



Dave Abrahams

- Key goals of the library
  - Reference / Pointer Support
  - Globally Registered Type Coercions
  - Full Cross module Support
  - Improve Overloading support
  - C++ to Python Exception Translation
  - Default Argument Support
  - Generic C++ to Python Object Interface

TWINCLING™
freedom of innovation

Python – C++ integration …

- Key goals of the library
  - Standard C++ algos to Python objects
  - Python LONG support
  - Improved built-in Numeric Type Coercion
  - Python Iterator support
  - Automatic C++ object initialization
  - DocString support
  - C++ long long support
  - Code Footprint Reduction
  - Data memory Footprint Reduction

TWINCLING™
freedom of innovation

What is Python used for ?

- System Utilities
  - system admin tools, portable shell scripts
- Internet Scripting
  - CGI scripts, parse HTML, process XML, email tools
- User Interfaces (Uis) - rapid prototyping
- Component Glue
  - scripting for apps, COM scripting
- Distributed Programming
  - COM, CORBA, XML-RPC

TWINCLING™
freedom of innovation

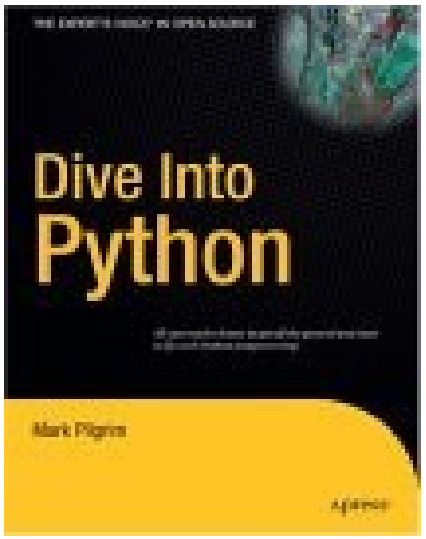
What is Python used for ? ...

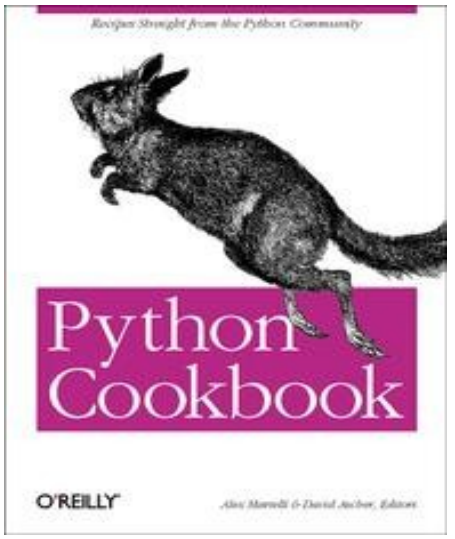
- Database Programming
  - `http://www.python.org/peps/pep-0249.html`
- Image Processing
  - `Python Image Library`
    `http://www.pythonware.com/products/pil/`
- OpenGL Programming, Writing Games
  - `PyOpenGL, py3d`
- Artifical Intelligence
  - `http://www.strout.net/python/ai/`

TWINCLING™
freedom of innovation

Python Resources

- http://www.python.org/
- http://wiki.python.org/moin/PythonBooks
- Boost Python – C++ tutorial
  `http://www.boost.org/libs/python/doc/tutorial/doc/html/index.html`





TWINCLING.org

TWINCLING™
freedom of innovation

Special thanks

TWINCLING.org

AppLabs Technologies Pvt. Ltd.

Coordinating for TWINCLING Saturday Meet (TSM)
Providing LCD projector and Meeting space.

http://www.applabs.com/

TWINCLING™
freedom of innovation

About Us

TWINCLING.org

TWINCLING™ Society
freedom of innovation

India's first, independent, "not-for-profit",
OpenSource software development
& promotion society.

www.twincling.org

groups.yahoo.com/group/twincling