

Python

Brandon Jeffcoat
Dashaun West



“Why settle for snake oil when you can have the whole snake?”
-- From Usenet posting by Mark Jackson, June 1998



History of Python

- Created in 1990 by Guido van Rossum
- Named after Monty Python
- First public release in 1991
- `comp.lang.python` founded in 1994
- Open source from the start

Python Overview



- Scripting Language
- Object-Oriented
- Portable
- Powerful
- Easy to learn and use
- Mixes good features from Java, Perl and Scheme



Major Uses of Python

- System Utilities
- GUIs (Tkinter, gtk, Qt, Windows)
- Internet Scripting
- Embedded Scripting
- Database Programming
- Artificial Intelligence
- Image Processing



Language Features

- Object-Oriented
- Interpreted
- Interactive
- Dynamic
- Functional
- Highly readable



Comparisons

Java

- Typically 3-5 times shorter than equivalent Java programs
- Run-time works harder than Java's
- Components can be developed in Java and combined to form applications in Python
- Python can be used to prototype components into Java implementation



Comparisons, cont'd

Perl

- Come from similar backgrounds
- Python is more applicable than Perl
- Perl emphasizes support for common application-oriented tasks
- Python emphasizes support for common programming methodologies



Comparisons, cont'd

C++

- Differences are similar to Java's
- Often 5-10 times shorter than equivalent C++ code
- Python shines as a glue language; used to combine components written in C++



Comparisons, cont'd

Common Lisp and Scheme

- Similar in dynamic semantics, different in approach to syntax
- Common Lisp is big and Scheme world is fragmented between many incompatible versions
- Python has a single, free compact implementation

Samples



- System Utility
- Functional Programming
- Object Oriented networking



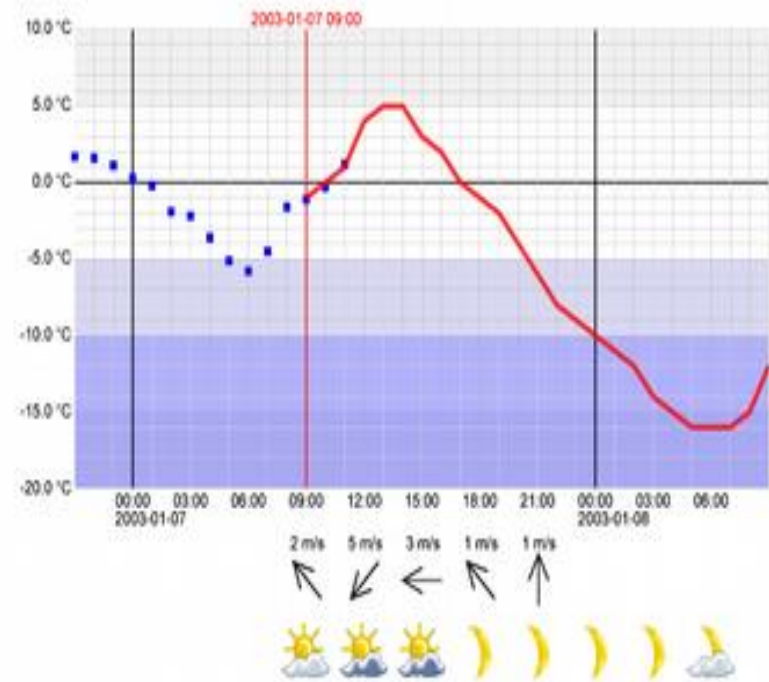
Samples

- The [Weather Server](#) product family consists of a number of products for production and display of forecast data, satellite images, and radar data.
- Secret Labs AB currently offers the following products:
- [Weather Production Server](#), for mission-critical processing and distribution of weather data. Servers based on this technology has been used in production since 1995.
- [Weather Media Server](#), for production and distribution of end-user products, for use via web browsers or custom weather workstations.

Samples



- [Weather Media Generator](#), for production of end-user products in existing distribution environments.
- [Weather Visualisation Client](#) is a toolbox for advanced visualisation of meteorological data.
- [Weather Workstation](#), for interactive display and analysis of satellite and radar images





Sample Code

- Here's a simple function written in Python, which inverts a table (represented as a Python dictionary):

```
def invert(table):  
    index = {}                # empty dictionary  
    for key in table.keys():  
        value = table[key]  
        if not index.has_key(value):  
            index[value] = [] # empty list  
            index[value].append(key)  
    return index
```

Comments are introduced by a **#** character



References

- Python homepage: <http://www.python.org/>
- Programming Python and Learning Python:
<http://python.oreilly.com/>
- Weather Server <http://www.pythonware.com/products/index.htm>