

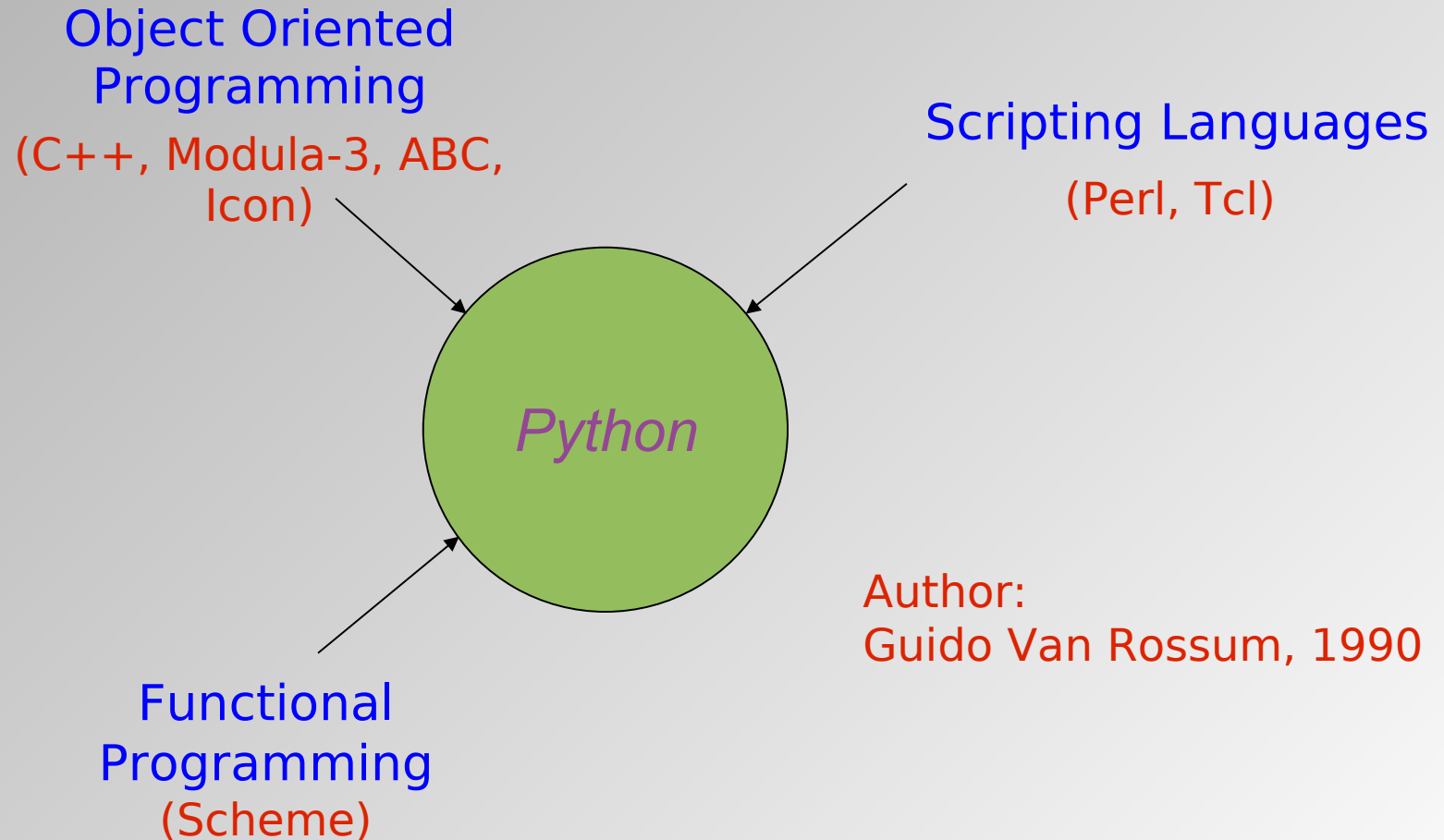


OOP and Scripting in Python

Introduction

Giuliano Armano - DMI Univ. di Cagliari

Python Roots ...





Timeline

- 1989/1990 First code
- 1991 First Release
- 1992 Mailing list
- 1992 Python 0.9.6 for MS-DOS.
- 1993 comp.lang.python Python newsgroup
- 1995 Website



Timeline

Python 2.y.x

- 1998 Z Object Publishing Env., aka Zope
- 2000 Python 2.0
- 2001 Official GPL-compatibility
- 2003 Release 2.3.x
- 2005 Release 2.4.x
- 2006 Release 2.5.x
- 2008 Releases 2.6.x
- 2010 Release 2.7.x



Timeline

Python 3000

- 2008 Release 3.0
- 2009 Release 3.1.x
- 2011 Release 3.2.x
- 2012 Release 3.3.x
- 2014 Release 3.4.x
- 2015 Release 3.5.x
- 2016 Release 3.6.x
- ... etc. ...



Ongoing Work

- **Python 2.7.18** is the latest advancement in the Python 2 series
- **Python 3.9.7** is the latest of the Python 3000 series (backward-incompatible)

Notes

- Python 2.7 will be the last backward-compatible transitional release
- As such, many features are being backported from Python 3.x to 2.7
- Thus, it makes sense to release both versions at the same time



<http://www.python.org>

- Python is an easy to learn, powerful programming language
- It has efficient high-level data structures and a simple but effective approach to object-oriented programming
- Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms



Bibliography

<http://www.python.org>

- Python in a Nutshell: A Desktop Quick Reference (3rd ed.), A. Martelli, A. Ravenscroft, S. Holden, O'Reilly, 2017.
- Pensare in Python, A. Downey, O'Reilly, 2018
- Programming Python (4th ed.), M. Lutz, O'Reilly, 2011.
- Programmare con Python - Guida Completa: M. Buttu, LSWR Edizioni, 2014.
- Learning Python: M. Lutz, D. Ascher, O'Reilly, 5th ed. 2013.
- Head First Python: A Brain-Friendly Guide (2nd ed.), P. Barry, O'Reilly, 2016.



Bibliography (free download books)

<http://www.python.org>

- Learn Python 3 the Hard Way, Z. A. Shaw
- Think Python: How to Think Like a Computer Scientist, A. B. Downey
- What You Need to Know about Python, P.Riti
- ... etc ...



Some Links

- **Python.org** <http://www.python.org/links/>
- **Python docs** <http://docs.python.org/>
- **Cetus Links** http://www.objenv.com/cetus/oo_python.html
- **Pypi** <https://pypi.org/>
- **Python LF** <http://www.awaretek.com/plf.html>
- **Python course** <https://www.python-course.eu/>



The Zen of Python

- Beautiful is better than ugly
- Explicit is better than implicit
- Simple is better than complex
- Complex is better than complicated
- Flat is better than nested
- Sparse is better than dense
- Readability counts
- Special cases aren't special enough to break the rules (although practicality beats purity)
- ...



The Zen of Python

- Errors should never pass silently. Unless explicitly silenced
- In the face of ambiguity, refuse the temptation to guess
- There should be one –and preferably only one– obvious way to do it (although that way may not be obvious at first unless you're Dutch)
- ...



The Zen of Python

- Now is better than never. Although never is often better than **right** now
- If the implementation is hard to explain, it's a bad idea
- If the implementation is easy to explain, it may be a good idea
- Namespaces are one honking great idea -- let's do more of those !