Getting Started

Bok, Jong Soon
javaexpert@nate.com
https://github.com/swacademy/Python

What is Python?

- In Greek mythology, Python was the serpent, sometimes represented as a dragon, living at the center of the earth, believed by the ancient Greeks to be at Delphi.

Source from https://en.wikipedia.org/wiki/Python_(mythology)

What is Python? (Cont.)

- Is a general-purpose interpreted, interactive, object-oriented, and high-level programming language.
- Created by Guido van Rossum during 1985- 1990 at the National Research Institute for Mathematics and Computer Science in the Netherlands.
- First released in 1991.
- Is named after a TV Show called **'Monty Python's Flying Circus'** and not after Python-the snake.

What is Python? (Cont.)

- Is derived from many other languages
  - ABC → Statement nesting is indicated by indentation
  - Modula-2 : The module as a compilation unit for separate compilation
  - C, C++ : Basic syntax
  - ICON : Dictionary data structure, slice operator **`[n:m]`**
  - SETL : List and tuples data structure
  - SmallTalk : Object-Oriented concepts
  - Haskell, Lisp : Functional language concepts
  - Unix shell and other scripting languages.

What is Python? (Cont.)

- Have a large and comprehensive standard library.
- Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems.
- Have a community-based development model, as do nearly all of its variant implementations.
- Is managed by the non-profit Python Software Foundation.
- https://www.python.org/psf/

Python Language Features

- Multi-paradigm programming language
  - Functional, Object-Oriented → Common Lisp, Sather
  - Imperative, Object-Oriented → PHP, Simula
  - Concurrent, Functional → Erlang
  - Functional, Imperative, Object-Oriented → Java, Perl, Python
- Supports functional and structured programming methods as well as OOP.
- Can be used as a scripting language or can be compiled to byte-code for building large applications.

Python Language Features (Cont.)

- Dynamic Typing
  - Type constraints are not checked at compile time but at runtime.
  - Despite being dynamically typed, Python is strongly typed, forbidding operations that are not well-defined (for example, adding a number to a string).
- Provides very high-level dynamic data types.
- Supports dynamic type checking.

Python Language Features

- Is Interpreted
  - Is processed at runtime by the interpreter.
  - Do not need to compile your program before executing it.
- Is Interactive
  - Can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- Is Object-Oriented
  - Supports Object-Oriented style or technique of programming that encapsulates code within objects.

Python Language Features (Cont.)

- Is a Beginner's Language
  - Easy-to-learn.
  - Has few keywords, simple structure, and a clearly defined syntax.
  - Allows the student to pick up the language quickly.
- Portable
  - Can run on a wide variety of hardware platforms and has the same interface on all platforms.

Python Language Features (Cont.)

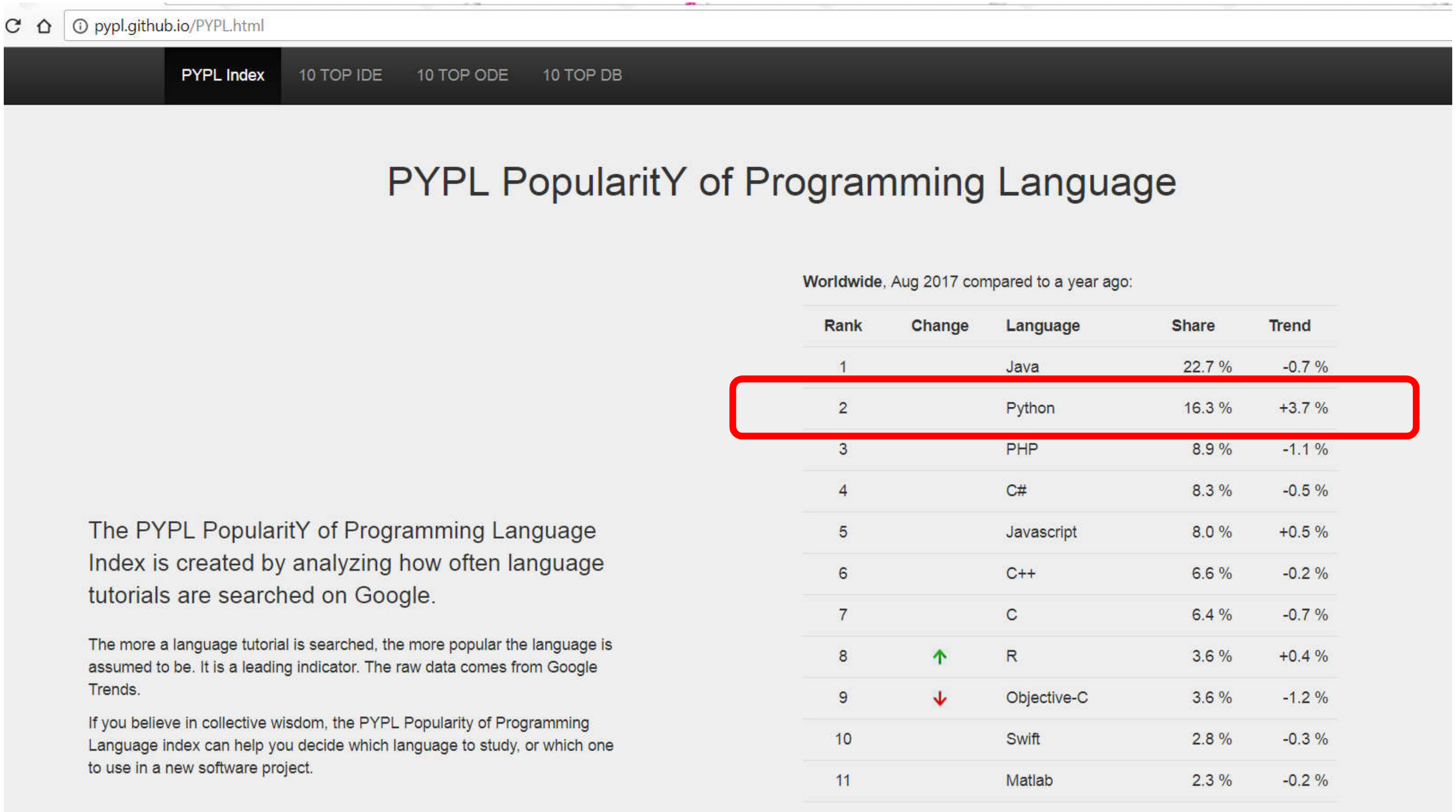
- Extendable
  - Can add low-level modules to the Python interpreter.
  - These modules enable programmers to add to or customize their tools to be more efficient.
- Databases
  - Provides interfaces to all major commercial databases.
- GUI Programming
  - Supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.

Python Language Features (Cont.)

- Scalable
  - Provides a better structure
  - Support for large programs than shell scripting.
- IT supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.
- Python source code is available under the GNU General Public License (GPL).

Python Popularity

- PYPL



PYPL PopularitY of Programming Language

Worldwide, Aug 2017 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Java	22.7 %	-0.7 %
2		Python	16.3 %	+3.7 %
3		PHP	8.9 %	-1.1 %
4		C#	8.3 %	-0.5 %
5		Javascript	8.0 %	+0.5 %
6		C++	6.6 %	-0.2 %
7		C	6.4 %	-0.7 %
8	↑	R	3.6 %	+0.4 %
9	↓	Objective-C	3.6 %	-1.2 %
10		Swift	2.8 %	-0.3 %
11		Matlab	2.3 %	-0.2 %

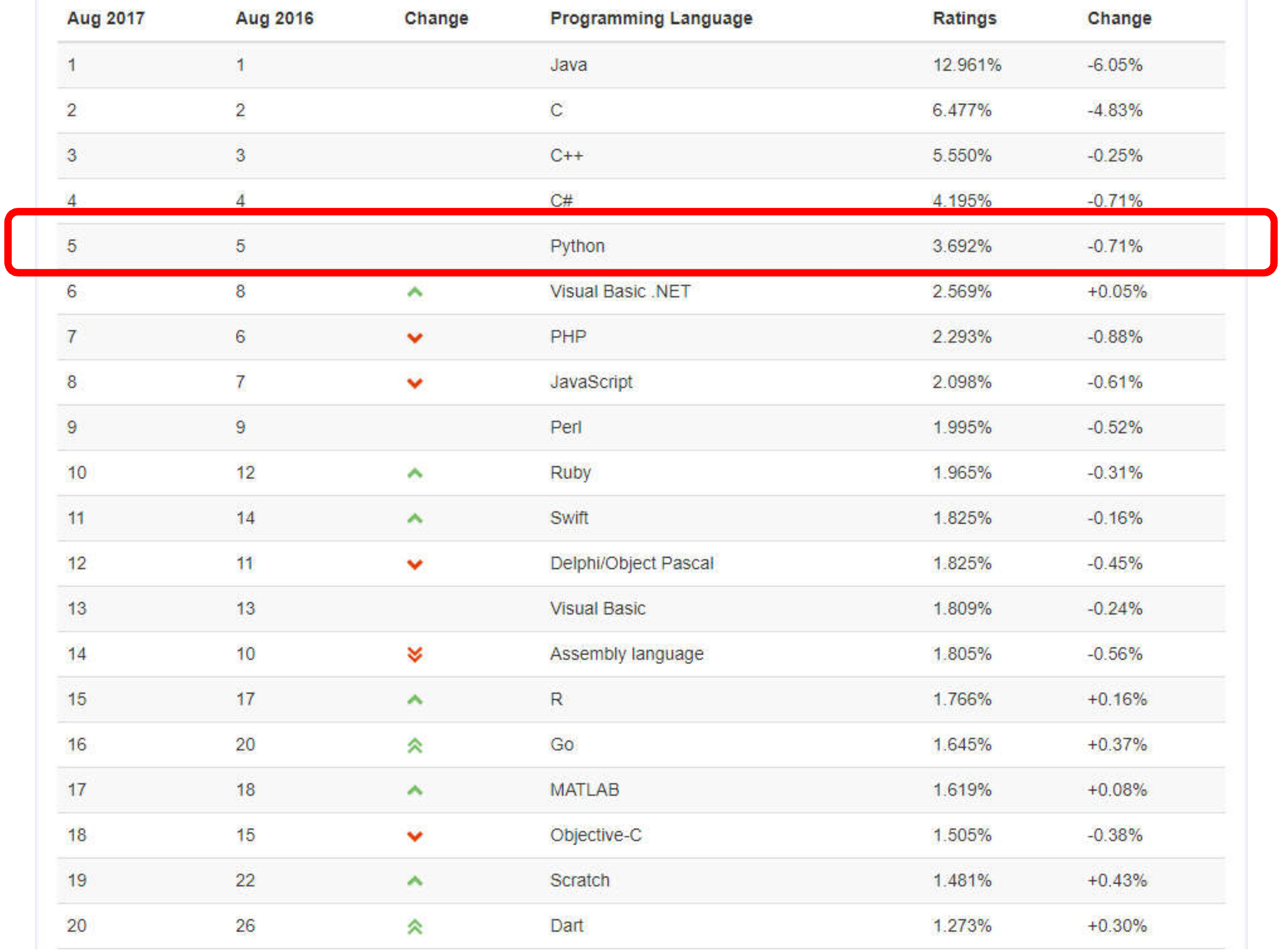
The PYPL PopularitY of Programming Language Index is created by analyzing how often language tutorials are searched on Google.

The more a language tutorial is searched, the more popular the language is assumed to be. It is a leading indicator. The raw data comes from Google Trends.

If you believe in collective wisdom, the PYPL Popularity of Programming Language index can help you decide which language to study, or which one to use in a new software project.

Python Popularity (Cont.)

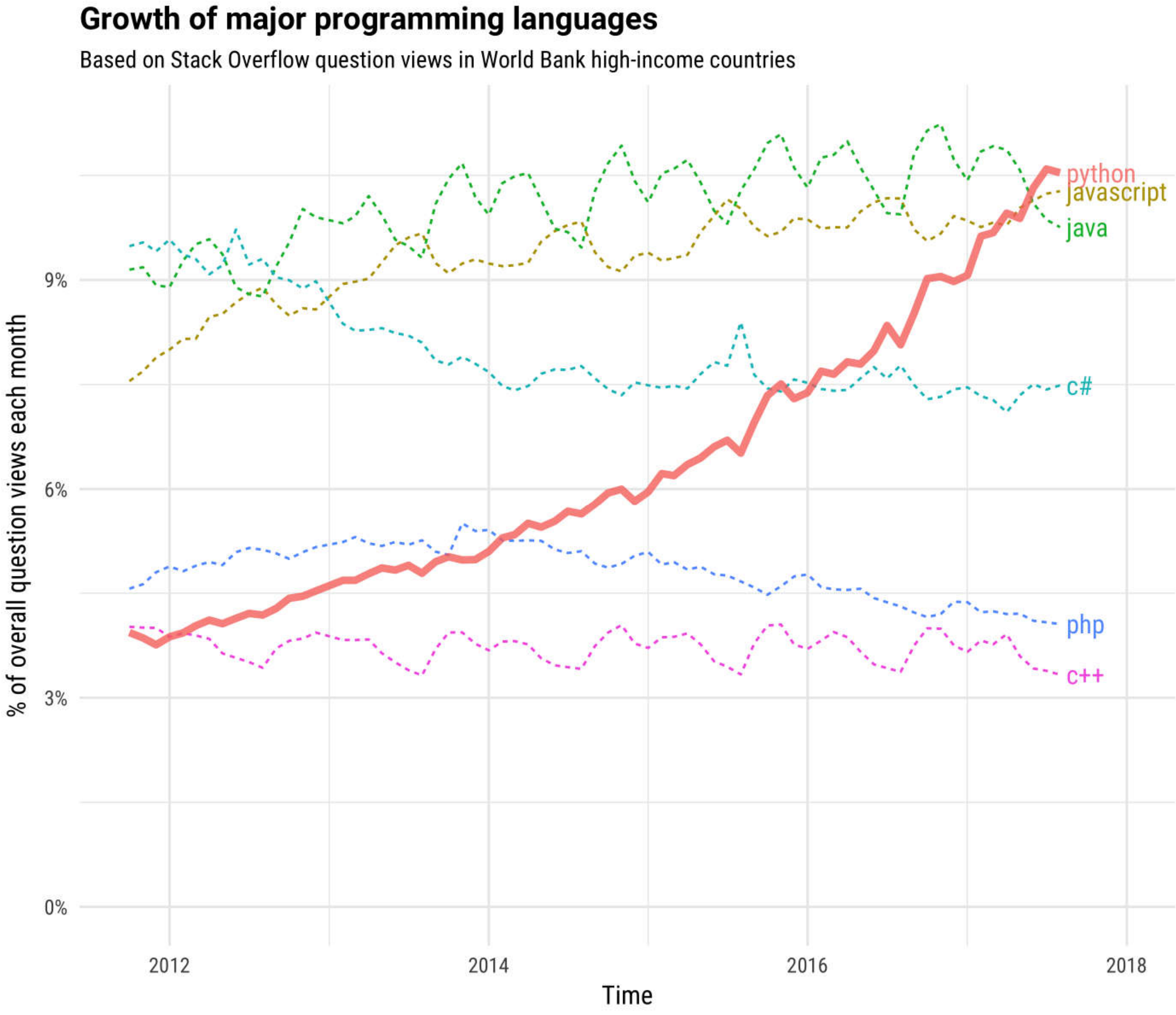
- TIOBE Index



Aug 2017	Aug 2016	Change	Programming Language	Ratings	Change
1	1		Java	12.961%	-6.05%
2	2		C	6.477%	-4.83%
3	3		C++	5.550%	-0.25%
4	4		C#	4.195%	-0.71%
5	5		Python	3.692%	-0.71%
6	8	↑	Visual Basic .NET	2.569%	+0.05%
7	6	↓	PHP	2.293%	-0.88%
8	7	↓	JavaScript	2.098%	-0.61%
9	9		Perl	1.995%	-0.52%
10	12	↑	Ruby	1.965%	-0.31%
11	14	↑	Swift	1.825%	-0.16%
12	11	↓	Delphi/Object Pascal	1.825%	-0.45%
13	13		Visual Basic	1.809%	-0.24%
14	10	⇊	Assembly language	1.805%	-0.56%
15	17	↑	R	1.766%	+0.16%
16	20	⇈	Go	1.645%	+0.37%
17	18	↑	MATLAB	1.619%	+0.08%
18	15	↓	Objective-C	1.505%	-0.38%
19	22	↑	Scratch	1.481%	+0.43%
20	26	⇈	Dart	1.273%	+0.30%

Python Popularity (Cont.)

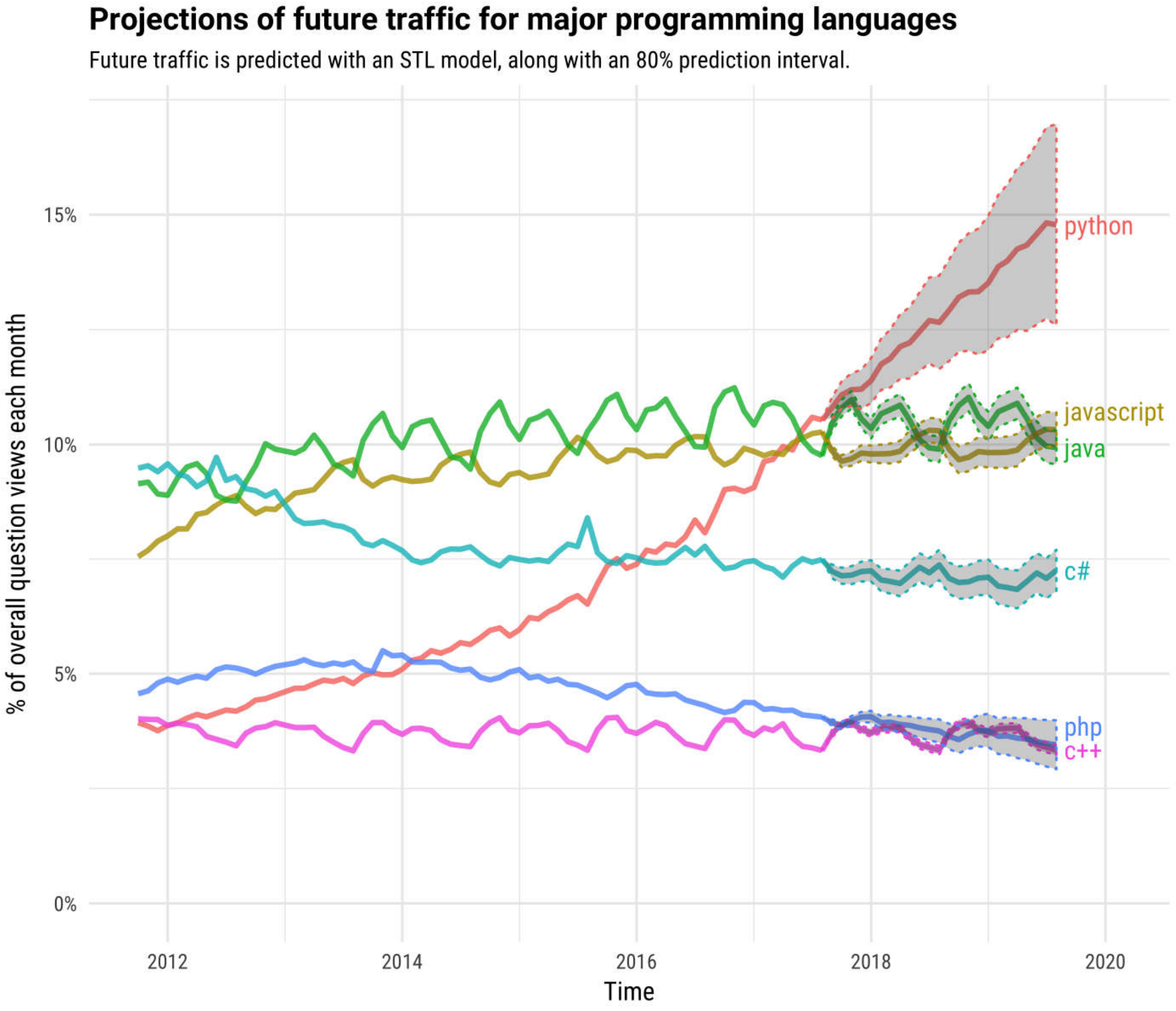
- stackoverflow



https://stackoverflow.blog/2017/09/06/incredible-growth-python/

Python Popularity (Cont.)

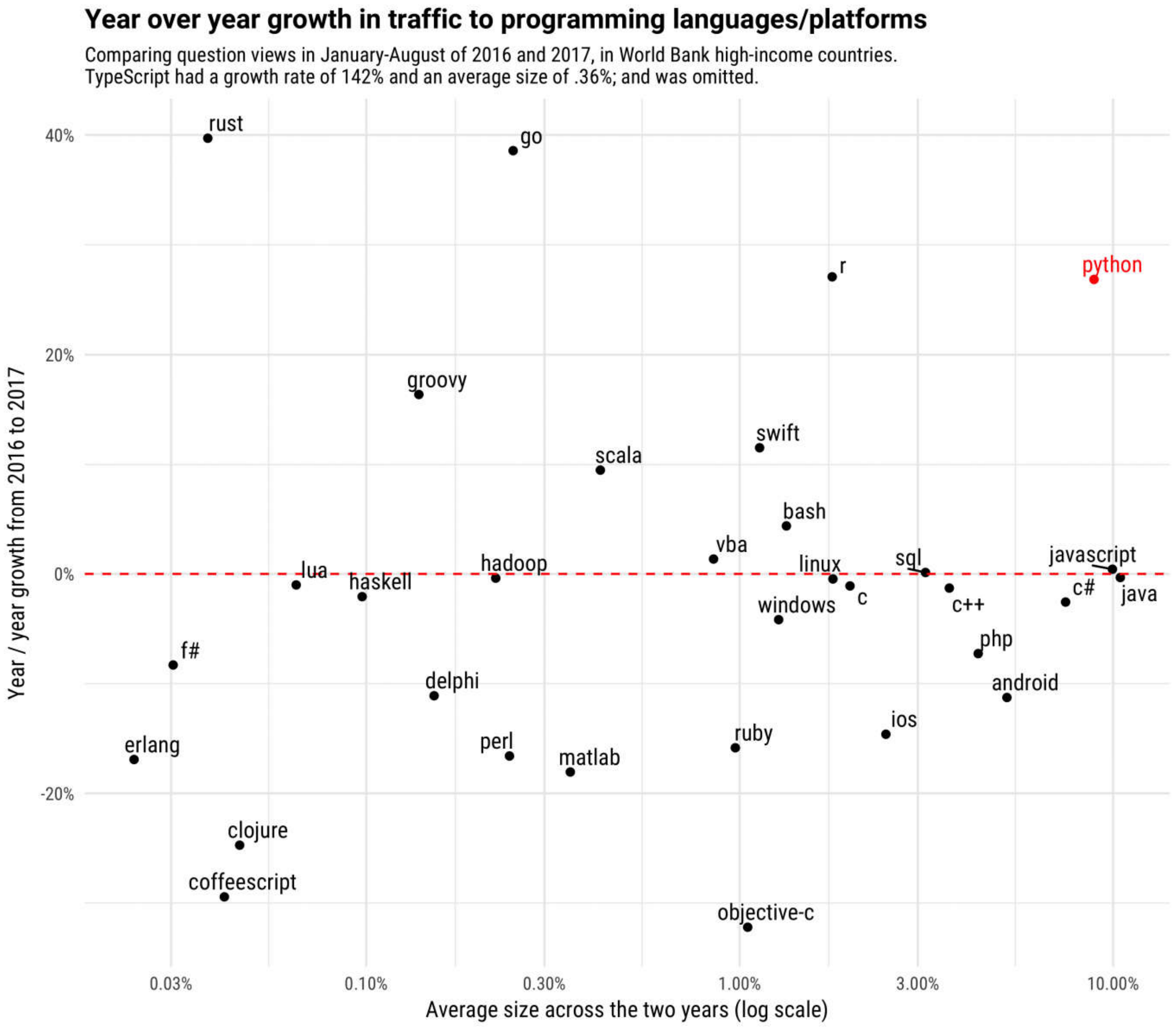
- stackoverflow



https://stackoverflow.blog/2017/09/06/incredible-growth-python/

Python Popularity (Cont.)

- stackoverflow



https://stackoverflow.blog/2017/09/06/incredible-growth-python/

Alternate Implementations

- CPython
  - Written in C, is the default and most widely used implementation of the Python language.
- Jython
  - Python implemented in Java.
  - Designed to run on the Java platform.
  - Can import and use any Java class.
  - A user interface in Jython could be written with Swing, AWT or SWT.
  - Compiles to Java bytecode.

Alternate Implementations (Cont.)

- IronPython
  - Is an implementation of the Python programming language targeting the .NET Framework and Mono.
  - Is written entirely in C#, although some of its code is automatically generated by a code generator written in Python.
- Pypy
  - A self-hosting interpreter for the Python programming language.

Cross-compilers to other languages

- **Jython**
  - Compiles into Java byte code, which can then be executed by every Java virtual machine implementation.
  - Enables the use of Java class library functions from the Python program.
- **IronPython**
  - Follows a similar approach in order to run Python programs on the .NET Common Language Runtime.

Cross-compilers to other languages (Cont.)

- **RPython**
  - Can be compiled to C, Java bytecode, or Common Intermediate Language, and is used to build the PyPy interpreter of Python.
- **Pyjs**
  - Compiles Python to JavaScript.
- **Cython**
  - Compiles Python to C and C++.

Cross-compilers to other languages (Cont.)

- **Pythran**
  - Compiles Python to C++.
- **Pyrex** (latest release in 2010) and **Shed Skin** (latest release in 2013)
  - Compile to C and C++ respectively.
- **Google's Grumpy**
  - Compiles Python to Go.
- **Nuitka**
  - Compiles Python into C++.

Who Uses Python Today?

- **Google** makes extensive use of Python in its web search systems.
- The popular **YouTube** video sharing service is largely written in Python.
- The **Dropbox** storage service codes both its server and desktop client software primarily in Python.
- The **Raspberry Pi** single-board computer promotes Python as its educational language.
- **EVE Online**, a massively multiplayer online game (MMOG) by CCP Games, uses Python broadly.

Who Uses Python Today? (Cont.)

- The widespread **BitTorrent** peer-to-peer file sharing system began its life as a Python program.
- **Industrial Light & Magic, Pixar**, and others use Python in the production of animated movies.
- **ESRI** uses Python as an end-user customization tool for its popular GIS mapping products.
- **Google's App Engine** web development framework uses Python as an application language.
- The **IronPort** email server product uses more than 1 million lines of Python code to do its job.

Who Uses Python Today? (Cont.)

- **Maya**, a powerful integrated 3D modeling and animation system, provides a Python scripting API.
- The **NSA** uses Python for cryptography and intelligence analysis.
- **iRobot** uses Python to develop commercial and military robotic devices.
- The **Civilization IV** game’s customizable scripted events are written entirely in Python.
- The One Laptop Per Child (**OLPC**) project built its user interface and activity model in Python.

Who Uses Python Today? (Cont.)

- **Netflix** and **Yelp** have both documented the role of Python in their software infrastructures.
- **Intel**, Cisco, **Hewlett-Packard**, **Seagate**, **Qualcomm**, and **IBM** use Python for hardware testing.
- **JPMorgan Chase**, **UBS**, **Getco**, and **Citadel** apply Python to financial market forecasting.
- **NASA**, **Los Alamos**, **Fermilab**, **JPL**, and others use Python for scientific programming tasks

Who Uses Python Today? (Cont.)

- Success stories
  - *http://www.python.org/about/success*
- Application domains
  - *http://www.python.org/about/apps*
- User quotes
  - *http://www.python.org/about/quotes*
- Wikipedia page
  - *http://en.wikipedia.org/wiki/List_of_Python_software*

What Can I Do with Python ?

- System Programming
- GUIs
- Internet Scripting
- Component Integration
- Database Programming
- Web Programming
- Rapid Prototyping
- Numeric and Scientific Programming

What Can I Do with Python? (Cont.)

- Game programming and multimedia with *pygame*, *cgkit*, *pyglet*, *PySoy*, *Panda3D*, and others.
- Serial port communication on Windows, Linux, and more with the *PySerial* extension
- Image processing with *PIL* and its newer *Pillow* fork, *PyOpenGL*, *Blender*, *Maya*, and more.
- Robot control programming with the *PyRo* toolkit.
- Natural language analysis with the *NLTK* package.
- Instrumentation on the *Raspberry Pi* and *Arduino* boards.

What Can I Do with Python? (Cont.)

- Mobile computing with ports of Python to the Google *Android* and Apple *iOS* platforms.
- Excel spreadsheet function and macro programming with the *PyXLL* or *DataNitro* add-ins.
- Media file content and metadata tag processing with *PyMedia*, *ID3*, *PIL/Pillow*, and more.
- Artificial intelligence with the *PyBrain* neural net library and the *Milk* machine learning toolkit.

What Can I Do with Python? (Cont.)

- Expert system programming with *PyCLIPS*, *Pyke*, *Pyrolog*, and *pyDatalog*.
- Network monitoring with *zenoss*, written in and customized with Python.
- Python-scripted design and modeling with *PythonCAD*, *PythonOCC*, *FreeCAD*, and others.
- Document processing and generation with *ReportLab*, *Sphinx*, *Cheetah*, *PyPDF*, and so on.

What Can I Do with Python? (Cont.)

- Data visualization with *Mayavi, matplotlib*, *VTK*, *VPython*, and more.
- XML parsing with the xml library package, the xmlrpclib module, and third-party extensions.
- JSON and CSV file processing with the json and csv modules.
- Data mining with the *Orange* framework, the *Pattern* bundle, *Scrapy*, and custom code.
- **Data Analysis, IoT**

Etc. Python Usage Cases

- Virtualization Solution Xen Managing Console
- Google Groups Mailing List for Service
- NC Soft Online Game Server-partly.
- Facebook Real-time Web-Server Tornado
- AWS Kinesis Real-time Stream Analysis Application
- Yogiyo
- GIMP, Maya, Paint Shop Pro

Python Possibilities and Limitations

- Possible
  - System Utilities
  - GUI Programming
  - Module Programming combined with a kind of languages.
  - Web Programming
  - Scientific Programming
  - Database Programming

- Limited
  - OS
  - Highly Iterative Operations
  - Compressed Application Algorithm Development
  - Mobile Programming

Python Version – 2.x vs 3.x

- Python 1.0 was released in November 1994.
- In 2000, Python 2.0 was released.
- Python 2.7.13 is the latest edition of Python 2.
- Python 3.0 was released in 2008.
- 3.3 in 2012, 3.4 in 2014, 3.5 in 2015, and 3.6 in 2016.

Python Version – 2.x vs 3.x (Cont.)

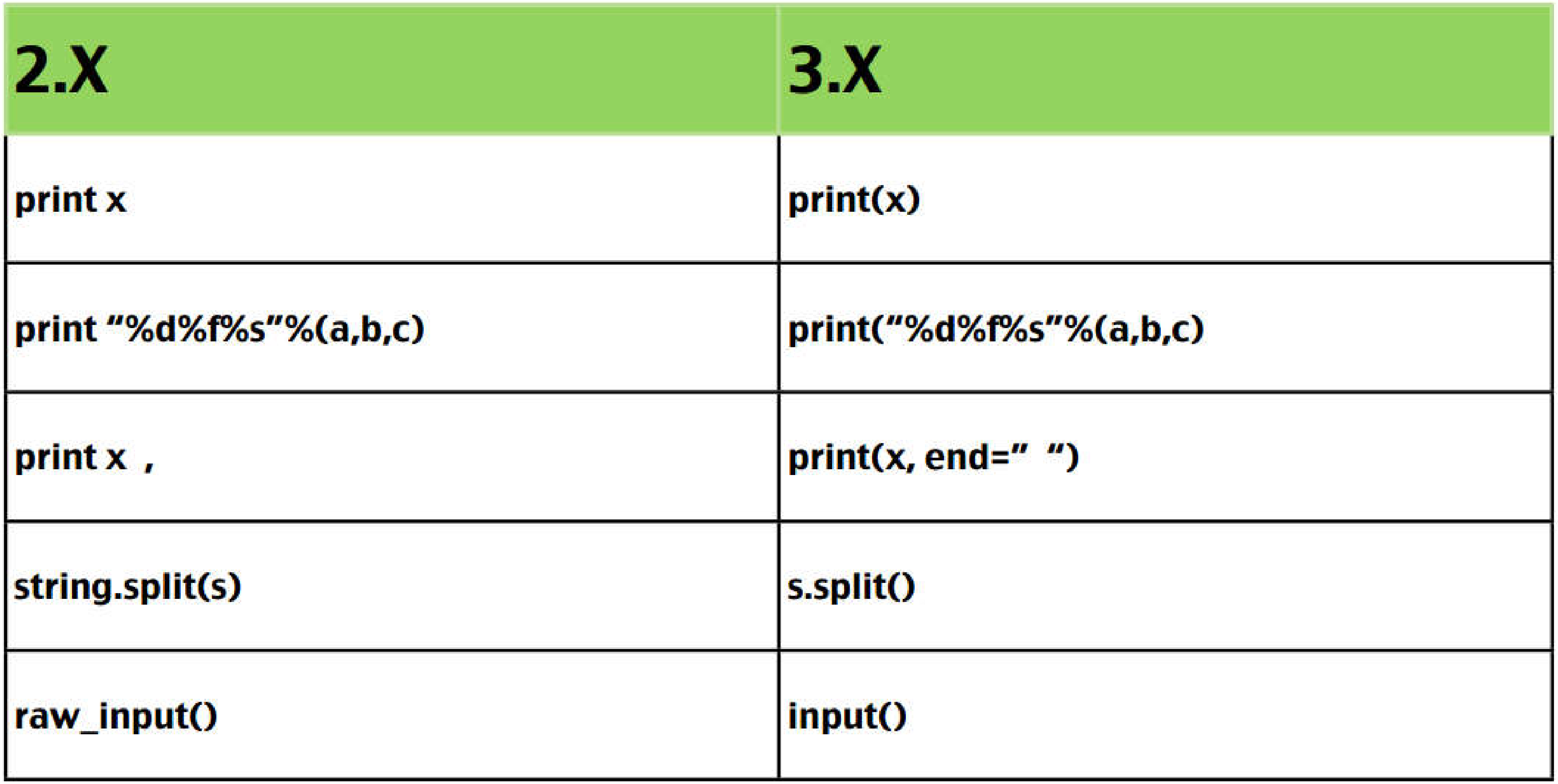
- Python 2.x is legacy, Python 3.x is the present and future of the language.
- Python 3 is not backward compatible with Python 2.
- All recent standard library improvements are only available by default in Python 3.x.
- More details refer to https://wiki.python.org/moin/Python2orPython3

https://docs.python.org/3.0/whatsnew/3.0.html

Python Version – 2.x vs 3.x (Cont.)

- A non-exhaustive list of features which are only available in 3.x releases and won't be backported to the 2.x series:
  - strings are Unicode by default
  - clean Unicode/bytes separation
  - exception chaining
  - function annotations
  - syntax for keyword-only arguments
  - extended tuple unpacking
  - non-local variable declarations

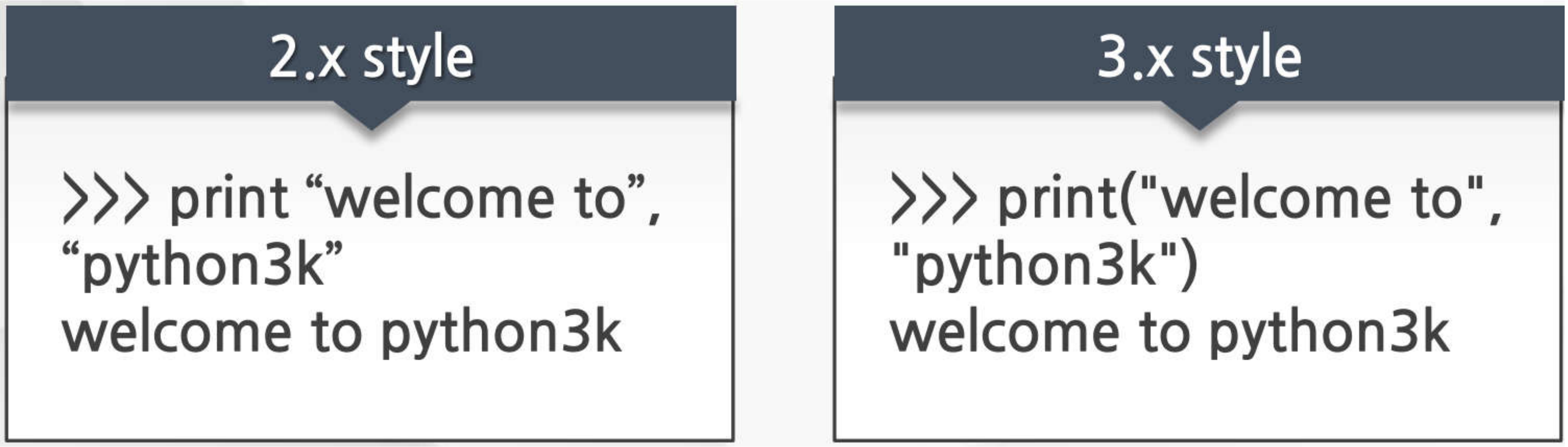
Python Version – 2.x vs 3.x (Cont.)



2.X	3.X
print x	print(x)
print “%d%f%s”%(a,b,c)	print(“%d%f%s”%(a,b,c)
print x ,	print(x, end=” “)
string.split(s)	s.split()
raw_input()	input()

Source from : https://docs.python.org/3.0/whatsnew/3.0.html

Python Version – 2.x vs 3.x (Cont.)



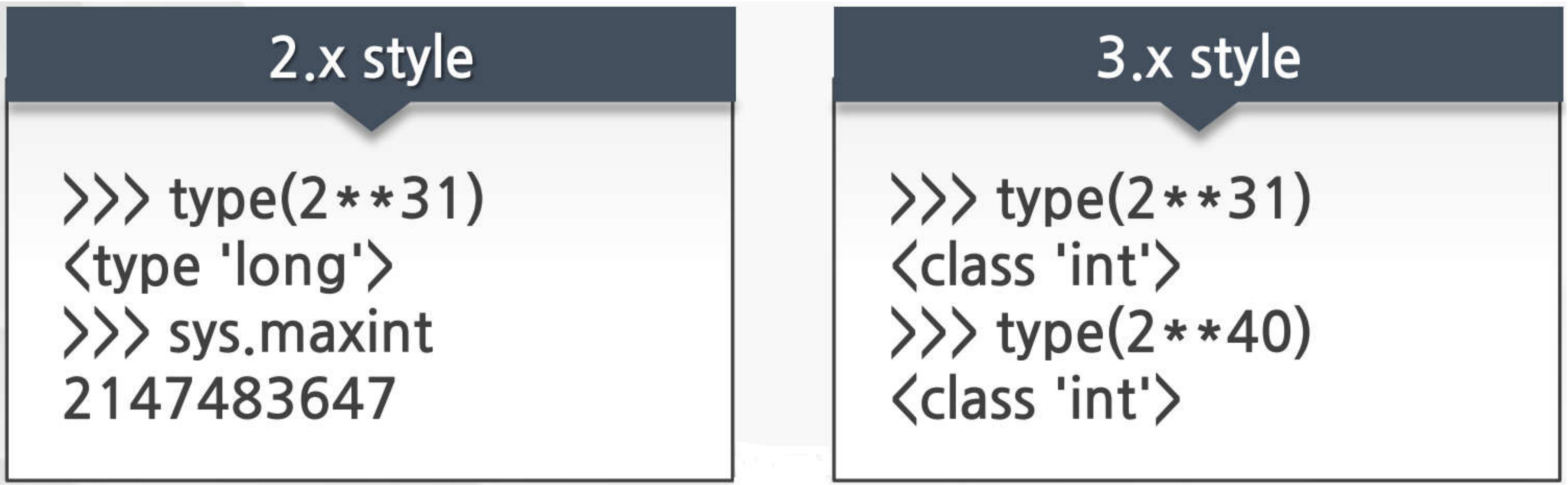
2.x style

```
>>> print “welcome to”,
“python3k”
welcome to python3k
```

3.x style

```
>>> print("welcome to",
"python3k")
welcome to python3k
```

Python Version – 2.x vs 3.x (Cont.)



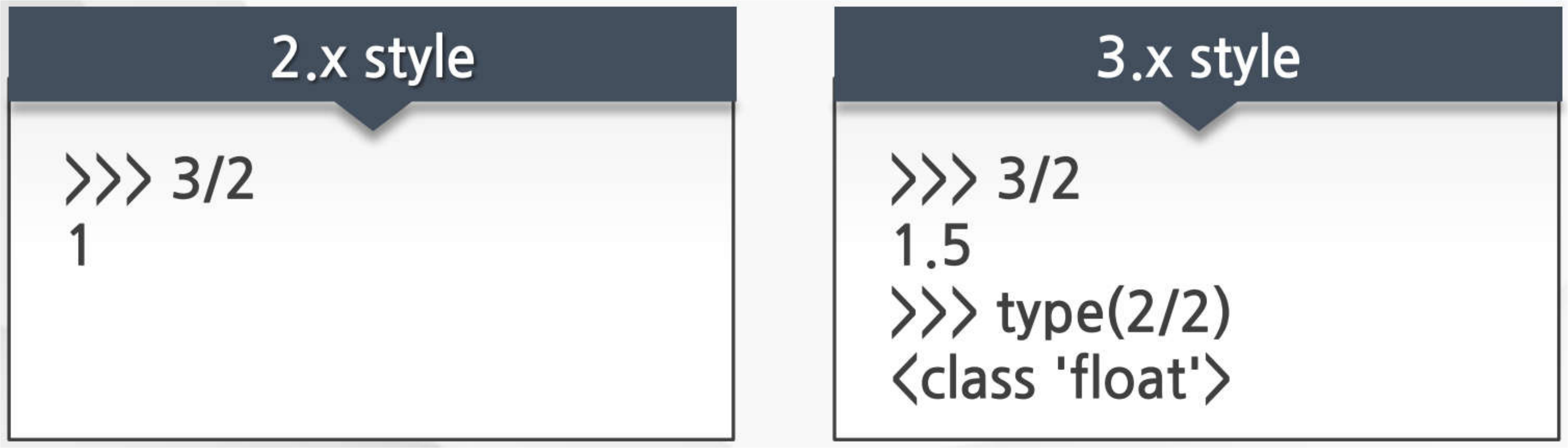
2.x style

```
>>> type(2**31)
<type 'long'>
>>> sys.maxint
2147483647
```

3.x style

```
>>> type(2**31)
<class 'int'>
>>> type(2**40)
<class 'int'>
```

Python Version – 2.x vs 3.x (Cont.)



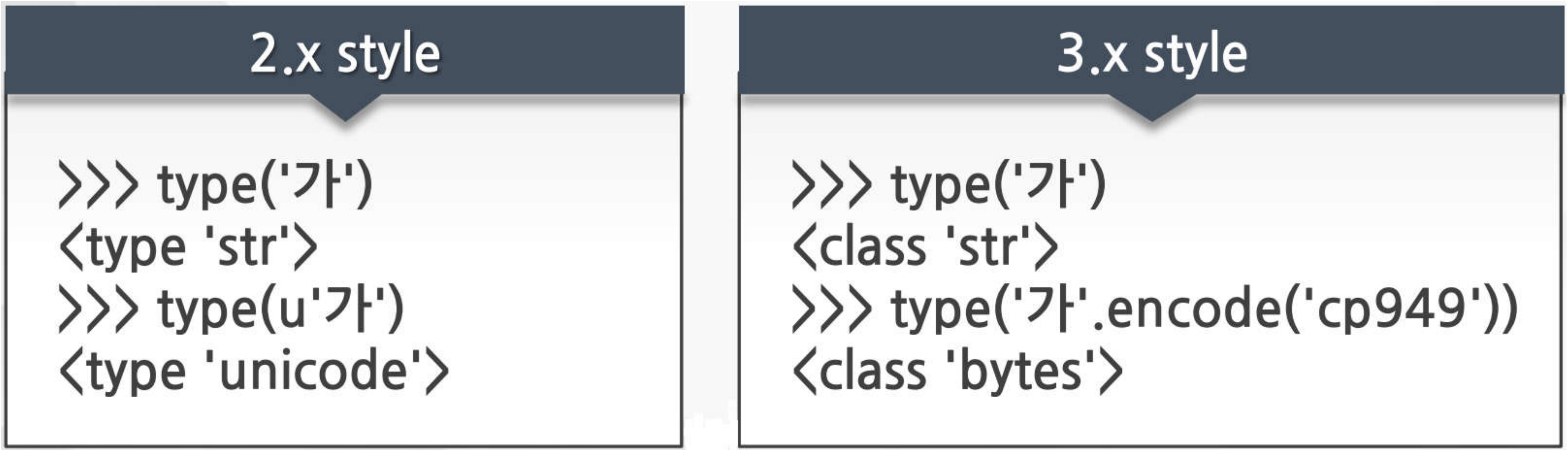
2.x style

```
>>> 3/2
1
```

3.x style

```
>>> 3/2
1.5
>>> type(2/2)
<class 'float'>
```

Python Version – 2.x vs 3.x (Cont.)



2.x style

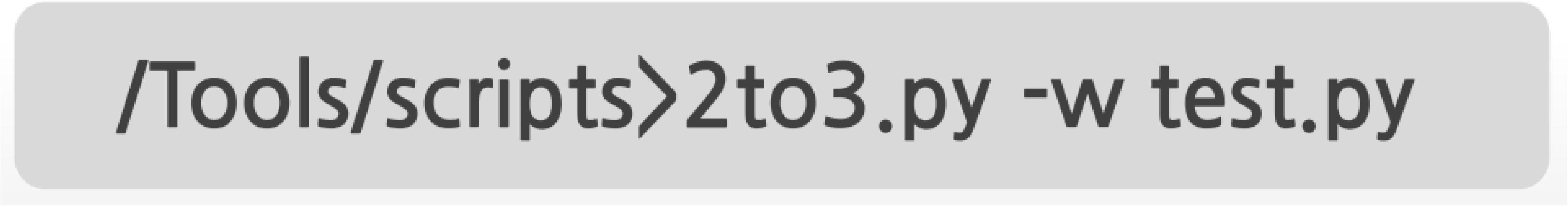
```
>>> type('가')
<type 'str'>
>>> type(u'가')
<type 'unicode'>
```

3.x style

```
>>> type('가')
<class 'str'>
>>> type('가'.encode('cp949'))
<class 'bytes'>
```

Python Version – 2.x vs 3.x (Cont.)

- Convert Python2 into Python3.



```
/Tools/scripts>2to3.py -w test.py
```