

# python?

BDFL



The Netherlands



python SOFTWARE FOUNDATION

Guido van Rossum







AUSTRALIA



python  
for web  
development

django

Flask  
web development,  
one drop at a time



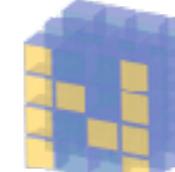
Pyramid™



# python for science

 astropy

 biopython

 NumPy

 SciPy



python  
for cloud  
configuration



ANSIBLE

Boto3 for  
 **amazon**  
web services™



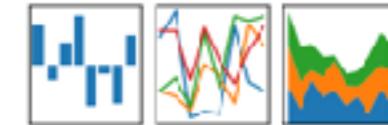
Microsoft Azure  
SDK for Python



python  
for  
data analytics

pandas

$$y_t = \beta' x_t + \mu_t + \epsilon_{it}$$



matplotlib



Bokeh



TensorFlow





# Who is using Python?



Instagram



Quora

# Python Implementations

Python implementation

written in

runs on

CPython



native



JVM

IronPython

C#



pypy

RPython

native & others

# Python Release Timeline

2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014

Python 2

2.4

2.5

straightforward

upgrade paths

2.6

2.7

Python 3

3.0

3.1

3.2

3.3

3.4



# Python 2 vs Python 3

Python 2	Python 3
Maintained, but no new features	New features being added
End of Life (EOL) in 2020	Unicode support by default
Still default on many systems	Cleared some Python 2 confusion
print "Hello World"	print("Hello World")
	Minor differences from Python 2



# Installing Python

---





# Platform Specific Installation



Windows

macOS

ubuntu The Ubuntu logo, which is a red circle containing a white dash-dot pattern.



**Read  
Evaluate  
Print  
Loop**

>>> REPL



# Significant Whitespace Rules

1. Prefer **four spaces**
2. Never mix spaces and tabs
3. Be **consistent** on consecutive lines
4. Only deviate to **improve** readability

Programming as Guido ~~intended it~~  
*indented*

Moment of Zen

# Readability Counts

Clarity Matters  
So readability makes  
For valuable code





python

# Scalar types and values



# python **Scalar types and values**

int  
42

arbitrary precision integer

float  
4.2

64-bit floating point numbers

NoneType  
None

the null object

bool  
True

bool  
False

boolean logical values



**int**

unlimited precision signed integer



# float

IEEE-754 double precision (64-bit)

53 bits of binary precision

15 to 16 bits of decimal precision



# None

The sole value of `NoneType`.

Often used to represent the absence of a value.

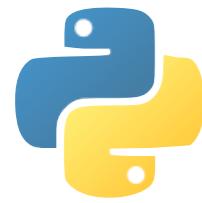
Not displayed by the REPL.



bool

Boolean logical value.

Either True or False.



# python Relational Operators

`==` value equality / equivalence

`!=` value inequality / inequivalence

`<` less-than

`>` greater-than

`<=` less-than or equal to

`>=` greater-than or equal to

**Equal objects are interchangeable**



== ==  
== ==



**True**



# python Conditional Statements

```
if expr:  
    print("expr is True")
```

expr is converted to bool as if by  
the bool() constructor



# python Conditional Statements

```
if expr:  
    print("expr is True")
```



# python Conditional Statements

```
if expr:  
    print("expr is True")  
else:  
    print("expr is False")
```

```
if h > 50:  
    print("Greater than 50")  
else:  
    if h < 20:  
        print("Less than 20")  
    else:  
        print("Between 20 and 50")
```

Python provides the `elif` keyword to eliminate the need for nested `if ... else` structures in many cases.



Flat is better  
than nested

```
if h > 50:  
    print("Greater than 50")  
elif h < 20:  
    print("Less than 20")  
else:  
    print("Between 20 and 50")
```

Python provides the elif keyword to  
eliminate the need for nested if ... else  
structures in many cases.



# python while loops

```
while expr:  
    print("loop while expr is True")
```

expr is converted to bool as if by  
the bool() constructor



# python breaking out

```
while True:  
    if expr:  
        break  
    print("Go here on break")
```

The `break` keyword terminates the innermost loop, transferring execution to the first statement after the loop



# python Getting Started – Summary

- **Obtaining and installing Python 3**
  - Windows
  - Ubuntu Linux
  - Mac OS X
- **Read-Eval-Print-Loop or REPL**
- **Simple arithmetic with + - \* / % and //**
- **Assigning objects to named variables with the = operator**
- **print()**
- **Exiting the REPL**
  - Ctrl-Z on Windows
  - Ctrl-D on Linux and Mac.
- **Significant indentation - usually four spaces**
- **Python Enhancement Proposals**
  - PEP 8 – Python Style Guide
  - PEP 20 – The Zen of Python



# python Getting Started – Summary

- **Importing Python Standard Library modules:**
  - import module
  - from module import function
  - from module import function as alias
- **Finding and browsing help()**
- **Scalar built-in types**
  - int float None bool
  - conversions between types
- **Relational operators == != < > <= >= for equivalence and ordering**
- **Conditional statements with if ... elif ... else**
- **while loops**
- **Interrupting execution with Ctrl-C to create a KeyboardInterrupt exception**
- **Breaking out of loops with break**
- **Augmented assignment operators for modifying objects in-place**
- **Requesting text from the user with input()**