Programming in Python NPRG065

http://d3s.mff.cuni.cz

Department of Distributed and Dependable Systems

Tomas Bures

Petr Hnetynka

{bures,hnetynka}@d3s.mff.cuni.cz

CHARLES UNIVERSITY IN PRAGUE

faculty of mathematics and physics

Course information

- https://d3s.mff.cuni.cz/teaching/nprg065/

- 2/2 Exam + “Zápočet“
- Exam
  - practical in lab
    - implement a simple assignment
- “Zápočet“
  - homework
    - via ReCodEx
    - https://recodex.ms.mff.cuni.cz

Department of Distributed and Dependable Systems

Approx. time-line of the course

- Introduction
- Core types
- Control structures
- Data structures
- Classes and objects
- Core parts of the std. library

About Python

- Dynamically-typed
  - *duck typing*

"If it walks like a duck and it quacks like a duck, then it must be a duck."

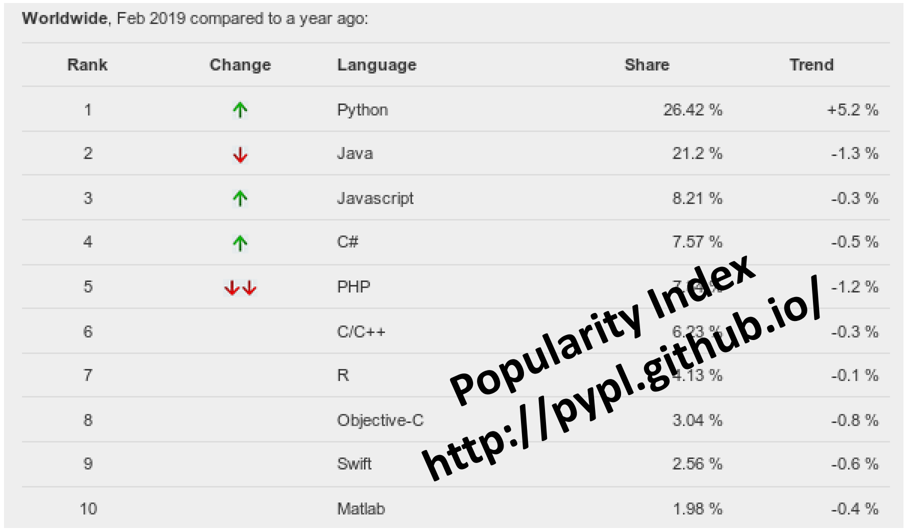
- Object-oriented language
  - there are classes but it is not a strictly class-based language
- Interpreted
  - no explicit compilation
  - "JIT" compilation to Python bytecode

- Started around 1990 by Guido Van Rossum
- Now in version 3.7
  - 2.7 – the last version of Python 2 still supported too
    - but only till January 1, 2020

- One of the most popular languages today
  - mainly for data analysis and machine learning

Department of Distributed and Dependable Systems

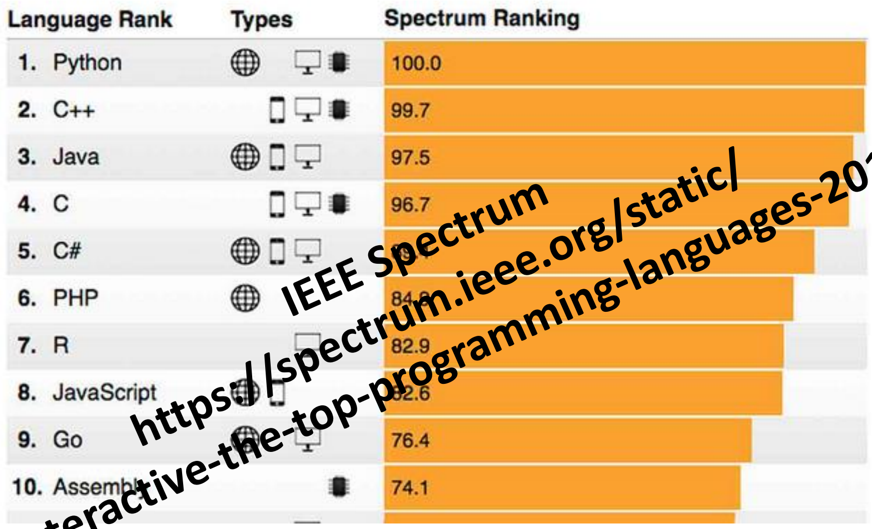
Popularity



Worldwide, Feb 2019 compared to a year ago:

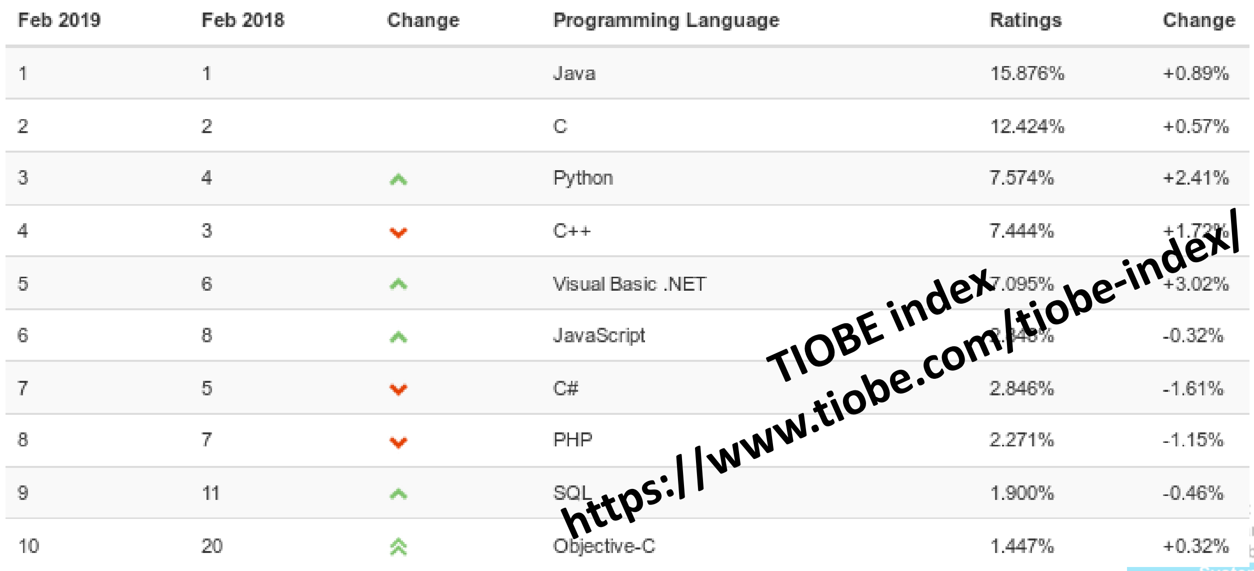
Rank	Change	Language	Share	Trend
1	↑	Python	26.42 %	+5.2 %
2	↓	Java	21.2 %	-1.3 %
3	↑	Javascript	8.21 %	-0.3 %
4	↑	C#	7.57 %	-0.5 %
5	↓↓	PHP	[illegible]	-1.2 %
6		C/C++	6.23 %	-0.3 %
7		R	4.13 %	-0.1 %
8		Objective-C	3.04 %	-0.8 %
9		Swift	2.56 %	-0.6 %
10		Matlab	1.98 %	-0.4 %

Popularity Index
http://pypl.github.io/



Language Rank	Types	Spectrum Ranking
1. Python	🌐 🖥 ▪	100.0
2. C++	📱 🖥 ▪	99.7
3. Java	🌐 📱 🖥	97.5
4. C	📱 🖥 ▪	96.7
5. C#	🌐 📱 🖥	[illegible]
6. PHP	🌐	[illegible]
7. R	🖥	82.9
8. JavaScript	🌐 📱	82.6
9. Go	🌐 🖥	76.4
10. Assembly	▪	74.1

IEEE Spectrum
https://spectrum.ieee.org/static/interactive-the-top-programming-languages-2018



Feb 2019	Feb 2018	Change	Programming Language	Ratings	Change
1	1		Java	15.876%	+0.89%
2	2		C	12.424%	+0.57%
3	4	▲	Python	7.574%	+2.41%
4	3	▼	C++	7.444%	+1.72%
5	6	▲	Visual Basic .NET	7.095%	+3.02%
6	8	▲	JavaScript	[illegible]	-0.32%
7	5	▼	C#	2.846%	-1.61%
8	7	▼	PHP	2.271%	-1.15%
9	11	▲	SQL	1.900%	-0.46%
10	20	▲▲	Objective-C	1.447%	+0.32%

TIOBE index
https://www.tiobe.com/tiobe-index/

About Python

- Name – why Python
  - Monty Python's Flying Circus ;-)
- Portable
  - Windows, Linux, *BSD,…, anywhere
- Installation https://www.python.org/downloads/
  - on Windows – download installer
  - on Linux – use a package manager
- License
  - Python Software Foundation license
    - BSD style license, can be used for anything
- PyPI – https://pypi.python.org/
  - Python Package Index
  - the repository of python packages

Department of Distributed and Dependable Systems

IDE

- PyCharm
  - https://www.jetbrains.com/pycharm/
  - Community edition – free
  - Professional edition – free for students/teachers
    - register via your university email
- Other IDEs

Sources

- Scripts
  - **`my_script.py`**
  - no explicit main – just start code
  - executable programs
    - **`python my_script.py`**

      or
    - **`my_script.py`**
      - on unix systems
      - shebang line: **`#!/usr/bin/env python3`**

Shell

- Interactive shell
  - immediate evaluation
  - history (like in bash)
  - …
  - run just `python`

```
>>> 1 + 2
3
>>>
```

Department of Distributed and Dependable Systems

Multiple Python implementations

- **CPython**
  - “the” Python
- MicroPython
  - a variant of CPython
  - runs on microcontrollers (pyboard, ESP32,…)
- PyPy
  - implementation in Python
  - JIT
- Jython
  - in Java, Python2 only
  - can be embedded in Java
- IronPython
  - in .NET
- …

Python introduction…

- …via comparison with Pascal

Department of Distributed and Dependable Systems

Hello world

Pascal

```
program Hello;
begin
  writeln('Hello, world.');
end.
```

No begin, no main method,…

No semicolons

Python

```
print('Hello, world.')
```

Department of Distributed and Dependable Systems

Case sensitivity

```
program Hello;
var
  a: integer;
begin
  a := 1;
  A := 2;
  Writeln(a);
  writeln(A);
end.
```

Single variable

writeln / Writeln
does not matter

```
a = 1
A = 2
print(a)
print(A)
```

Two variables

Fibonacci numbers

```
function fib( i: integer ): integer;
begin
  if i<=1 then fib := 1
         else fib := fib( i-1 ) + fib( i-2 )
end;

begin
 writeln( fib(10) )
end.
```

No return type
No difference between functions/procedures

```
def fib(a):
    if a <= 1:
        return 1
    else:
        return fib(a - 1) + fib(a - 2)

print(fib(10))
```

No begin/end, no { }
Blocks by indentation

Multiplication table

```
procedure printMutliTable(number: integer );
var i: integer;
begin
 writeln( 'Multiplication table of ', number )
  for i:=1 to 10 do
      write( i * number );
end;
```

No variable declaration

```
def multi(number):
    print('Multiplication table of ', number)
    for i in range(11):
        print(i * number)
```

No “classical” **for** cycle

Fibonacci numbers v. 2

```
function Fib( k: integer ): integer;
var  prev, prevprev, tmp: integer;
begin
  prev := 1;
  prevprev := 1;
  while k>0 do
  begin
    tmp := prev + prevprev;
    prevprev := prev;
    prev := tmp;
    Dec( k )
  end;
  Fib := min
end;
```

```
def Fib(k):
    prev = 1
    prevprev = 1
    while k > 0:
        tmp = prev + prevprev
        prevprev = prev
        prev = tmp
        k -= 1
    return prev
```

Department of Distributed and Dependable Systems

Command line arguments

```
Program Cmdline;
Var
  i : Longint;
begin
  Writeln ('Num. of args', ParamCount);
  For i:=0 to ParamCount do
    Writeln (ParamStr (i));
end.
```

```
import sys

print('Num. of args', len(sys.argv))
for arg in sys.argv:
  print(arg)
```

Max value in “array”

```
Program MaxValue;
var
  max:integer = 0;
  i:integer;
  arr: array [0..9] of integer = (0, 9, 1, 8, 2, 7, 3, 6, 4, 5);
begin
  for i := 1 to 10 do
  begin
    if arr[i] > max then
      max := arr[i];
  end;
  writeln(max);
end.
```

```
arr = [0, 9, 1, 8, 2, 7, 3, 6, 4, 5]
max = 0
i = 0
while i < len(arr):
    if arr[i] > max:
        max = arr[i]
    i += 1
print(max)
```