Python for Practice
NPRG067

http://d3s.mff.cuni.cz

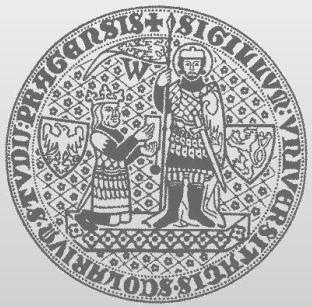
Tomas Bures

Petr Hnetynka

Ladislav Peška

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

peska@ksi.mff.cuni.cz



CHARLES UNIVERSITY IN PRAGUE

faculty of mathematics and physics

Course information

- https://d3s.mff.cuni.cz/teaching/nprg067/
- 0/2 Z
- “Zápočet“
  - Active participation + 1 homework

  or
  - 2 homeworks

Approx. time-line of the course

- GUI
- Web apps (REST)
- Machine learning
- Data analysis and processing

Python recap

About Python

- Dynamically-typed
  - *duck typing*
- Object-oriented language
  - there are classes but it is not a strictly class-based language
- Interpreted
  - no explicit compilation
  - “JIT” compilation to Python bytecode

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

- 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

Popularity

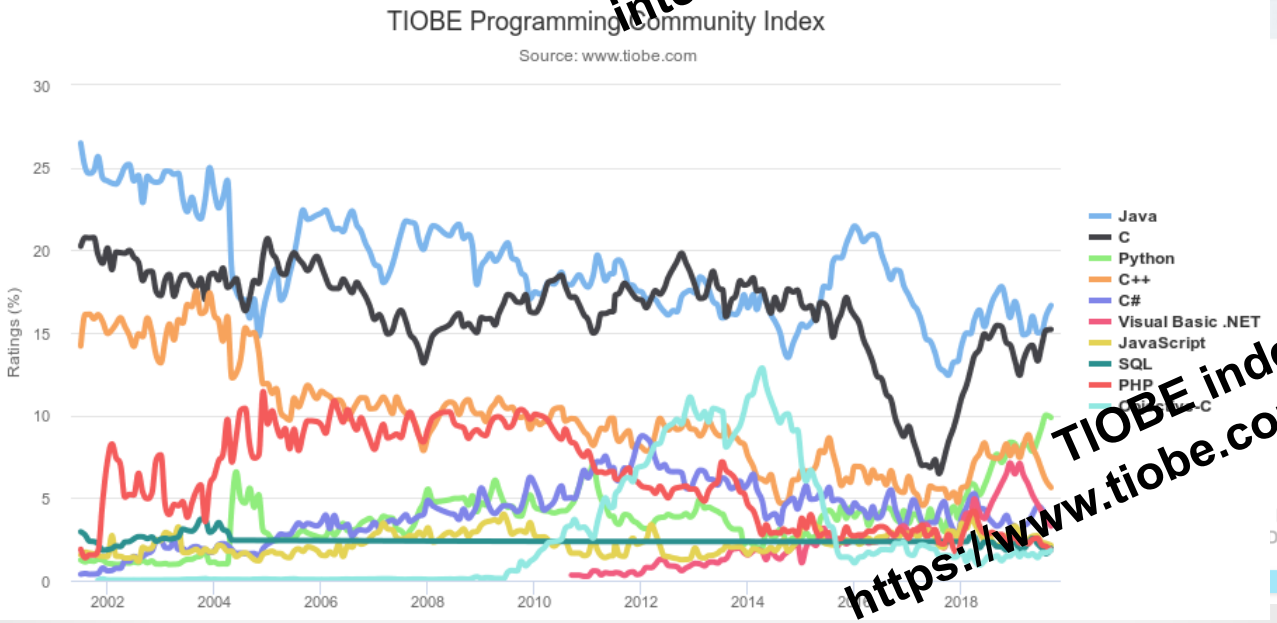
Worldwide, Sept 2019 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	29.21 %	+4.6 %
2		Java	19.9 %	-2.2 %
3		Javascript	8.39 %	+0.0 %
4		C#	7.23 %	-0.6 %
5		PHP	6.6%	0 %
6		C/C++	5.8 %	-0.4 %
7		R	3.91 %	-0.2 %
8		Objective-C	2.63 %	-0.7 %
9		Swift	2.46 %	-0.3 %
10		Matlab	1.82 %	-0.2 %

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

Rank	Language	Type	Score
1	Python		100.0
2	Java		96.3
3	C		94.4
4	C++		87.5
5	[illegible]		81.5
6	JavaScript		79.4
7	C#		74.5
8	Matlab		70.6
9	Swift		69.1
			68.0

IEEE Spectrum
https://spectrum.ieee.org/static/interactive-the-top-programming-languages-201



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

Department of Distributed and Dependable Systems

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

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
>>>
```

GUI in Python

GUI libraries for Python

- Many options

- Tkinter
- PyQt, PySide
- PyGTK
- wxPython
- Kiwi
- pygame
- …

Tkinter

- Available in the Python std library
- Python bindings to Tk toolkit
  - Tk – originally Tcl extensions
  - Now bindings to many languages
  - multiplatform

Hello world

```
from tkinter import *
from tkinter import ttk
root = Tk()
ttk.Button(root, text=
                "Hello World").grid()
root.mainloop()
```

See e01_hello_world.py

Events

```
from tkinter import *
from tkinter import ttk
root = Tk()

def exit_app():
    root.destroy()

ttk.Button(root, text="Exit",
           command=exit_app).grid()
root.mainloop()
```

See e02_hello_world_exit.py

Widgets

- Common set of widgets
  - 18
  - buttons, entries,…
- tkinter.ttk
  - themed tk
  - added to tk later
  - better support for styling

Widgets

- Frame
- Button
- Label
- Canvas
- Entry
- …

See
e03_widgets.py

- Styling widgets
  - ttk.Style()

See
e04_themes.py
05_widgets_themed.py
06_style.py

Layout managers

- grid
  - widgets in a table
    - widget can spread through several rows/columns

See e07_grid.py

- pack
  - arranging widgets side by side

See e08_pack.py

- place
  - absolute layout

Events and bindings

```
root = Tk()

def callback(event):
    print "clicked at", event.x, event.y

frame = Frame(root, width=100, height=100)
frame.bind("<Button-1>", callback)
frame.pack()

root.mainloop()
```

See e09_bind.py

- widget.bind(event, handler)
- event format – <modifier-type-detail>
  - <Double-Button-1>, <B1-Motion>, <Enter>, <Leave>, <Key>, <Return>, <Shift_L>, <Configure>, a, b, …

Menus

- Menu

  menu = Menu(root)

  root.config(menu=menu)

- Toolbar & status bar
  - no direct support
  - make own from the Frame

See e10_menu_and_bars.py

- Own component – class extending Frame
  - or different widget

See e11_statusbar_widget.py

Message boxes

```
from tkinter import messagebox

...

messagebox.showinfo("About", "Demo
                                application")
```

See e12_messagebox.py

- showinfo
- showwarning
- showerror
- askokcancel
- askquestion
- ...

More windows and dialogs

- Toplevel ~ window
- Modal dialogs
  - Toplevel + grab_set() + widget.wait_window(window)

See e13_dialog.py

- File choose dialog
  - from tkinter import filedialog
  - filename = filedialog.askopenfilename(initialdir = "/",title = "Select file",filetypes = (("jpeg files","*.jpg"),("all files","*.*")))

See e14_choose_file.py

Tk variables

- For obtaining values from widgets
- BooleanVar, DoubleVar, IntVar, StringVar

```
name = StringVar()
ttk.Entry(win, width=12, textvariable=name)
```

See e15_vars.py

More on events

- bind_class(className, event_descriptor, event_handler)
  - binding to all widgets of the particular type
- bind_all(event_descriptor, event_handler)
  - binding to all widgets
  - e.g. <F1> for help
- WARNING – tk is single threaded
  - (as most of the GUI frameworks in any language)
  - do not sleep (or any perform long actions) in handlers
- Long running actions -> own thread

See e16_sleep.py

See e17_text_tick.py

Images

- img = PhotoImage(file="…")
- Button(image=img,...

See
e18_images.py

Tasks

- Implement text editor (ala Windows notepad)
- Implement URL downloader (like a web browser but without rendering html)
  - Entry for URL
  - ScrolledText for result
- Implement Minesweeper

Help for tasks

- Manipulating text in ScrolledText
  - clean – text.delete(1.0, END)
  - get – text.get(1.0, END)
  - insert
    text.insert(INSERT, text)
    text.insert(END, text)