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
Course information

- [https://d3s.mff.cuni.cz/teaching/nprg067/](https://d3s.mff.cuni.cz/teaching/nprg067/)

- 0/2 Z

- “Zápočet“
  - 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

- Started around 1990 by Guido Van Rossum
- Now in version 3.7
  - 2.7 – the last version of Python 2 deprecated and unsupported
    - since January 1, 2020

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

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

Worldwide. Sept 2020 compared to a year ago:

<table>
<thead>
<tr>
<th>Rank</th>
<th>Change</th>
<th>Language</th>
<th>Share</th>
<th>Trend</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td>Python</td>
<td>31.56%</td>
<td>+2.9%</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>Java</td>
<td>16.4%</td>
<td>-3.1%</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>JavaScript</td>
<td>8.38%</td>
<td>+0.3%</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>C#</td>
<td>6.5%</td>
<td>-0.8%</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td>PHP</td>
<td>4.5%</td>
<td>-0.5%</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>C/C++</td>
<td>2.1%</td>
<td>+0.0%</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>R</td>
<td>4.08%</td>
<td>+0.3%</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>Objective-C</td>
<td>2.79%</td>
<td>+0.2%</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td>Swift</td>
<td>2.35%</td>
<td>-0.1%</td>
</tr>
<tr>
<td>10</td>
<td></td>
<td>TypeScript</td>
<td>1.92%</td>
<td>+0.1%</td>
</tr>
</tbody>
</table>

TIOBE Programming Community Index
Source: www.tiobe.com

IEEE Spectrum
Interactive-the-top-programming-languages-20/
http://pypl.github.io/
About Python

- Name – why Python
  - Monty Python’s Flying Circus ;-)  
- Portable
  - Windows, Linux, *BSD,…, anywhere
- Installation [https://www.python.org/downloads/](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/](https://pypi.python.org/)
  - Python Package Index
  - the repository of python packages
• PyCharm
  - [https://www.jetbrains.com/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

```python
>>> 1 + 2
3
```
GUI in Python
GUI libraries for Python

- Many options
- Tkinter
- PyQt, PySide
- PyGTK
- wxPython
- Kiwi
- pygame
- …
• Available in the Python std library
• Python bindings to Tk toolkit
  ▪ Tk – originally Tcl extensions
  ▪ Now bindings to many languages
  ▪ multiplatform
from tkinter import *
from tkinter import ttk
root = Tk()
ttk.Button(root, text= "Hello World").grid()
root.mainloop()
See e01_hello_world.py
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
• ...

• Styling widgets
  - ttk.Style()

See e03_widgets.py
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
  
- pack
  - arranging widgets side by side

- place
  - absolute layout

See e07_grid.py

See e08_pack.py
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()

- 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**
  
  ```python
  menu = Menu(root)
  root.config(menu=menu)
  ```

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

- **Own component – class extending Frame**
  - or different widget

See e10_menu_and_bars.py

See e11_statusbar_widget.py
Message boxes

from tkinter import messagebox

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

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

See e12_messagebox.py
More windows and dialogs

- Toplevel ~ window

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

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

See e13_dialog.py
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 perform long actions) in handlers

- Long running actions -> own thread
Images

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