PODSTAWY PROGRAMOWANIA W PYTHON

Dzień 1

AGENDA

Część pierwsza – podstawy programowania – spotkania 1-8

- Podstawowe pojęcia, pamięć,
- Instrukcje warunkowe, pętle
- Kolekcje

Część druga – wstęp do obiektowości – spotkania 9-15

- obiektowość
- Klasy, hierarchia klas, dziedziczenie
- Praktyczne wykorzystanie

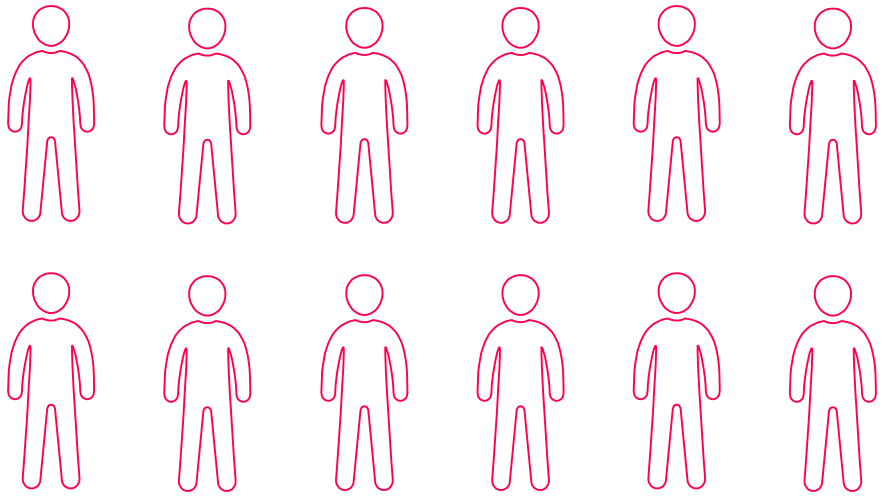
Przemysław Lalak

lalak@protonmail.com

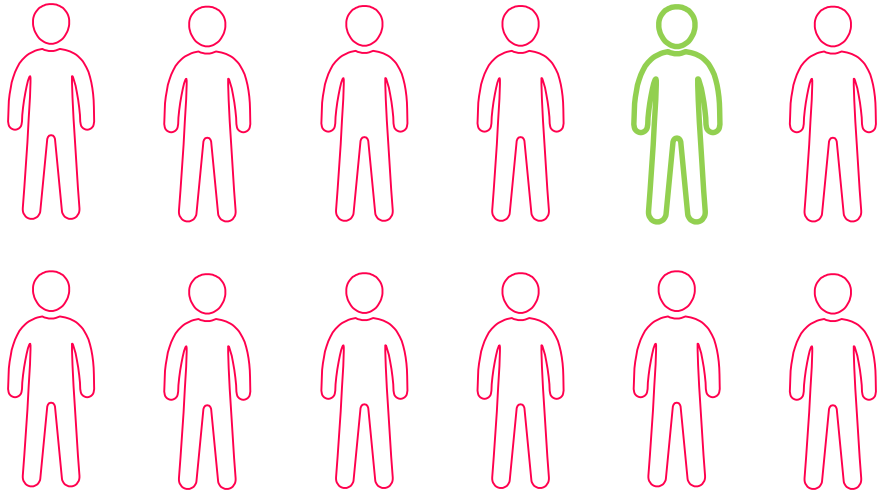
github.com/stemaan

Slajdy: Arek Gutkowski

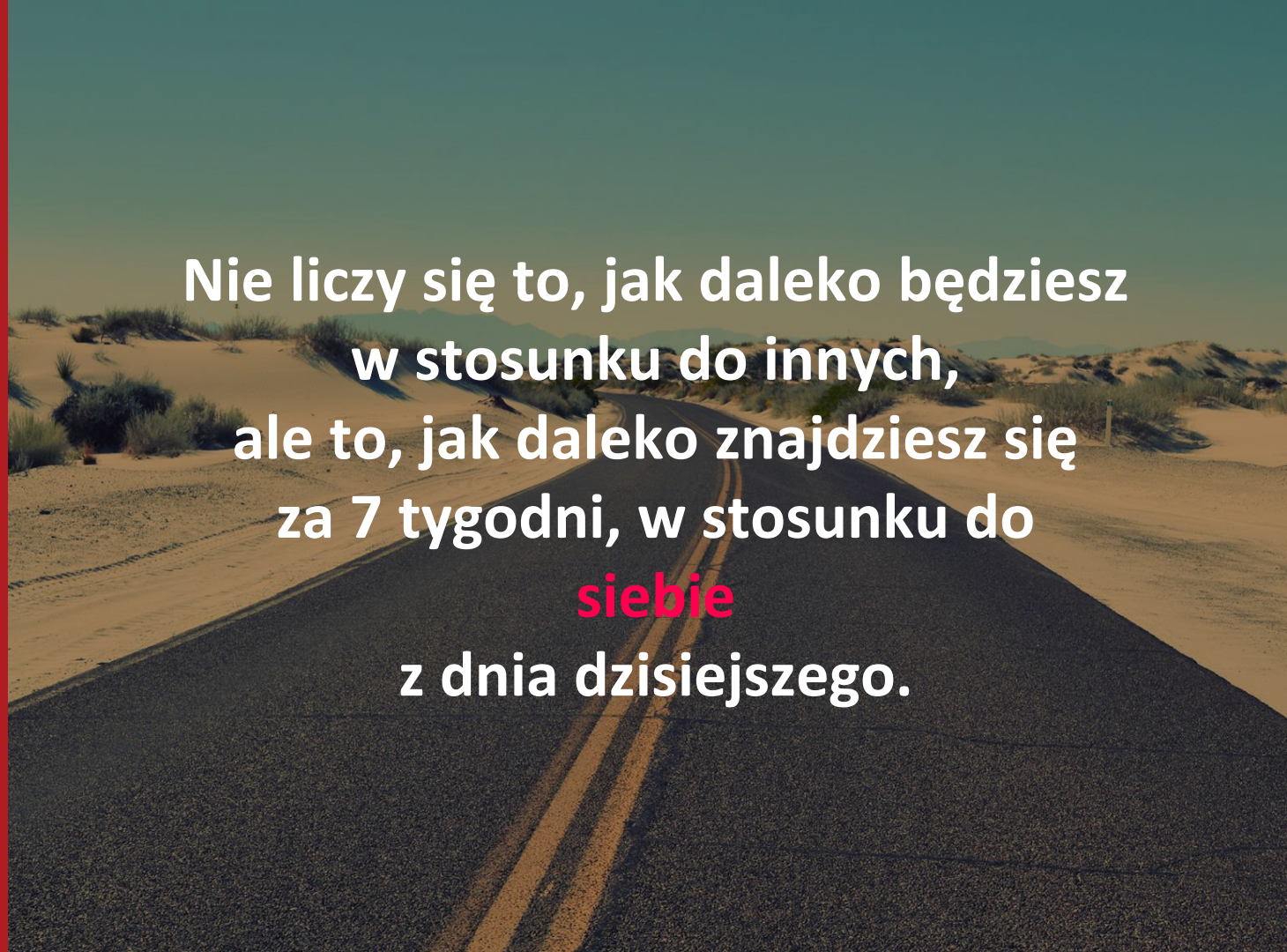
POZNAJMY SIĘ



POZNAJMY SIĘ



TWOJA DROGA



Nie liczy się to, jak daleko będziesz
w stosunku do innych,
ale to, jak daleko znajdziesz się
za 7 tygodni, w stosunku do
siebie
z dnia dzisiejszego.

CEL KURSU

- Fundamentalne pojęcia
- Analiza problemów
- Dobre praktyki
- Składnia języka

MINDSET

ZASOBY

- **Google**
- Dokumentacja Python: docs.python.org/3/
- StackOverflow
- 4programmers: 4programmers.net/Forum
- Slack: **isapythonkrk3.slack.com**
- GitHub: https://github.com/stemaan/isapythonkrk3

AGENDA
DAY 1

- Myślenie algorytmiczne
- Input, algorytm, output
- Pseudokod
- Języki kompilowane vs interpretowane
- Python
- Terminal / wiersz poleceń
- Git / GitHub

1.
Myślenie algorytmiczne

input -> algorytm -> output

SYSTEM BINARNY

0, 1

SYSTEM DZIESIĘTNY

0, 1, 2, 3, 4, 5, 6, 7, 8 , 9

1 2 3

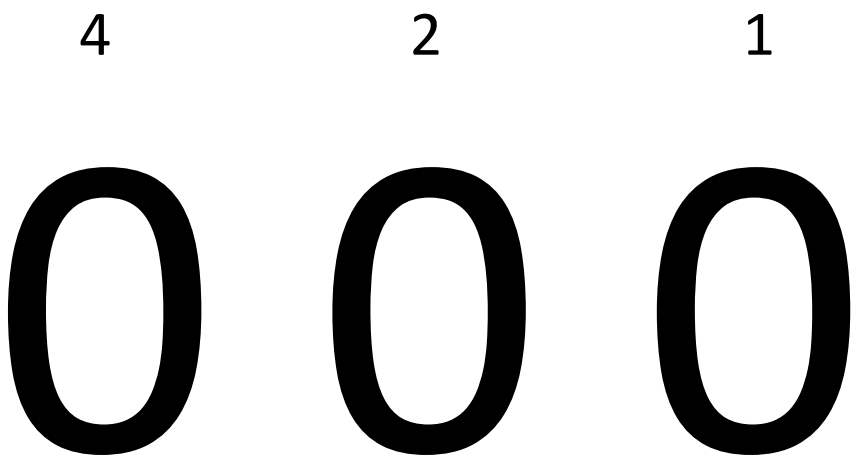
100	10	1
1	2	3

100	10	1
1	2	3

100 x 1 + 10 x 2 + 1 x 3

100	10	1
1	2	3

100 + 20 + 3


4
2
1
0
0
0

4 2 1

0 0 1

4	2	1
0	1	0

4	2	1
0	1	1

ASCII

A	B	C	D	E	F	G	H	I	J	K	L	M
65	66	67	68	69	70	71	72	73	74	75	76	77

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
78	79	80	81	82	83	84	85	86	87	88	89	90

i	S	A
105	83	65

105 83 65

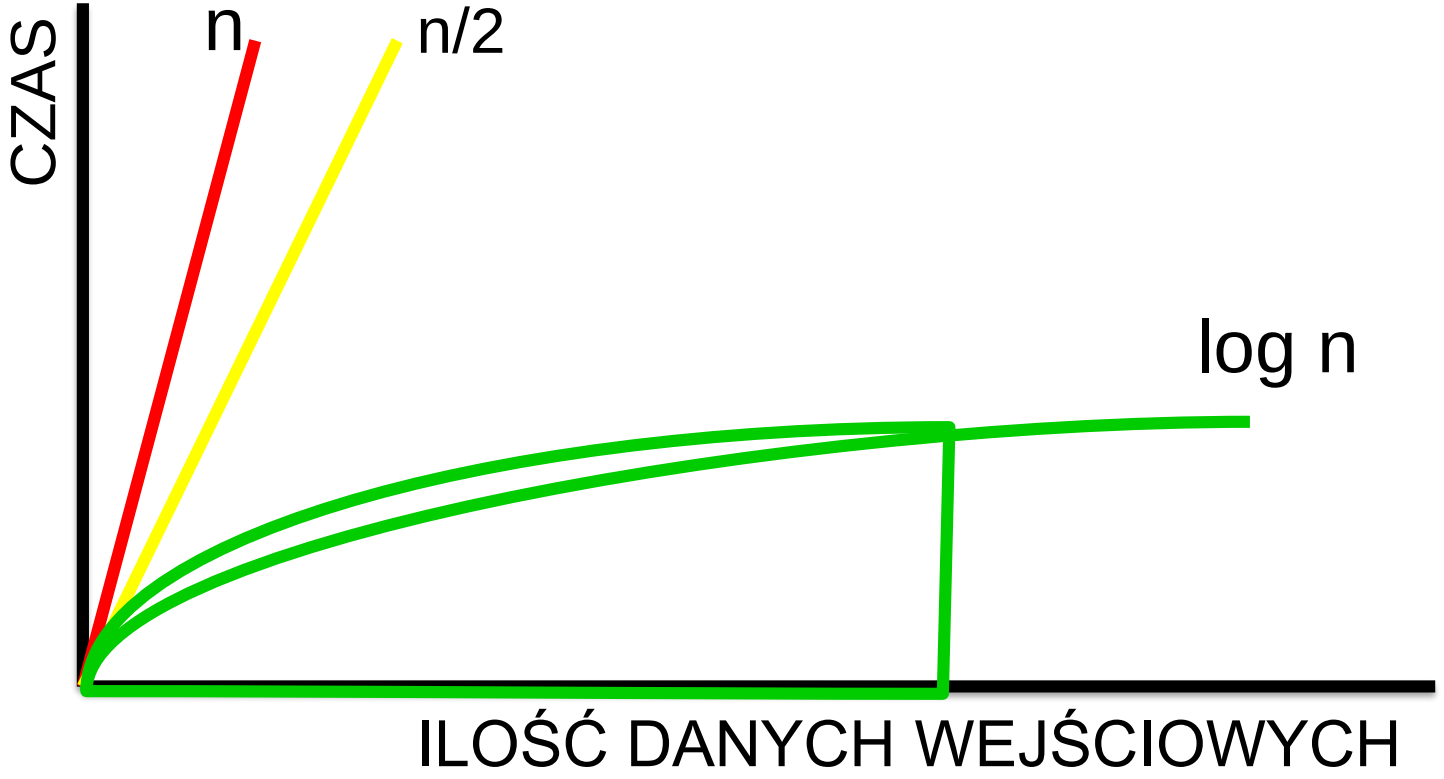
R G B

105	83	65
R	G	B

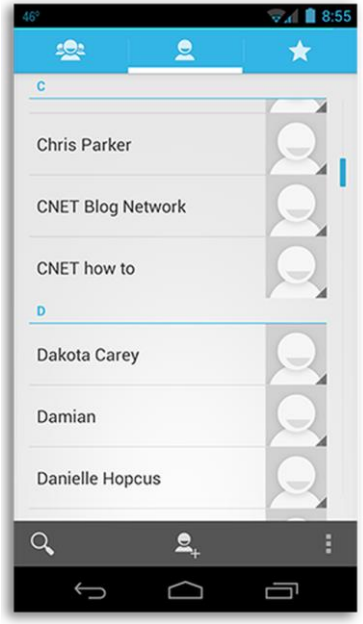
ALGORYTM

> *Skończony zestaw instrukcji potrzebnych do wykonania zadania.*

ZŁOŻONOŚĆ ALGORYTMICZNA



ALGORYTM


8:55
C
Chris Parker
CNET Blog Network
CNET how to
D
Dakota Carey
Damian
Danielle Hopcus


Adresser
eniro
Din vejviser på nettet - www.eniro.dk
Adresser
Find it easy - www.eniro.dk
eniro

PSEUDOKOD

PSEUDOKOD

1. weź książkę telefoniczną
2. otwórz książkę na środku
3. zobacz nazwiska
4. jeśli ”Wojtkowiak” jest wśród osób
5. zadzwoń do niego
6. w przeciwnym razie jeśli ” Wojtkowiak” jest wcześniej w książce
7. otwórz lewą połowę po środku
8. Idź do kroku 3
9. w przeciwnym razie jeśli ” Wojtkowiak” jest później w książce
10. otwórz prawą połowę po środku
11. idź do kroku 3
12. w przeciwnym razie
13. poddaj się

Polecenia

1. **weź książkę telefoniczną**
2. **otwórz książkę na środku**
3. **zobacz nazwiska**
4. jeśli ”Wojtkowiak” jest wśród osób
5. **zadzwoń do niego**
6. w przeciwnym razie jeśli ” Wojtkowiak” jest wcześniej w książce
7. **otwórz lewą połowę po środku**
8. Idź do kroku 3
9. w przeciwnym razie jeśli ” Wojtkowiak” jest później w książce
10. **otwórz prawą połowę po środku**
11. idź do kroku 3
12. w przeciwnym razie
13. **poddaj się**

Instrukcje warunkowe

1. weź książkę telefoniczną
2. otwórz książkę na środku
3. zobacz nazwiska
4. **jeśli ”Wojtkowiak” jest wśród osób**
5. zadzwoń do niego
6. **w przeciwnym razie jeśli ” Wojtkowiak” jest wcześniej w książce**
7. otwórz lewą połowę po środku
8. Idź do kroku 3
9. **w przeciwnym razie jeśli ” Wojtkowiak” jest później w książce**
10. otwórz prawą połowę po środku
11. idź do kroku 3
12. **w przeciwnym razie**
13. poddaj się

Pętle

1. weź książkę telefoniczną
2. otwórz książkę na środku
3. zobacz nazwiska
4. jeśli ”Wojtkowiak” jest wśród osób
5. zadzwoń do niego
6. w przeciwnym razie jeśli ” Wojtkowiak” jest wcześniej w książce
7. otwórz lewą połowę po środku
8. **Idź do kroku 3**
9. w przeciwnym razie jeśli ” Wojtkowiak” jest później w książce
10. otwórz prawą połowę po środku
11. **idź do kroku 3**
12. w przeciwnym razie
13. poddaj się

infoShare
<academy/>

PYTHON

DLACZEGO PYTHON

- Prosta składnia (syntax)
- Kompaktowy kod
- Kod niezależny od systemu
- Wszechstronny
- Popularność

Języki kompilowane vs interpretowane

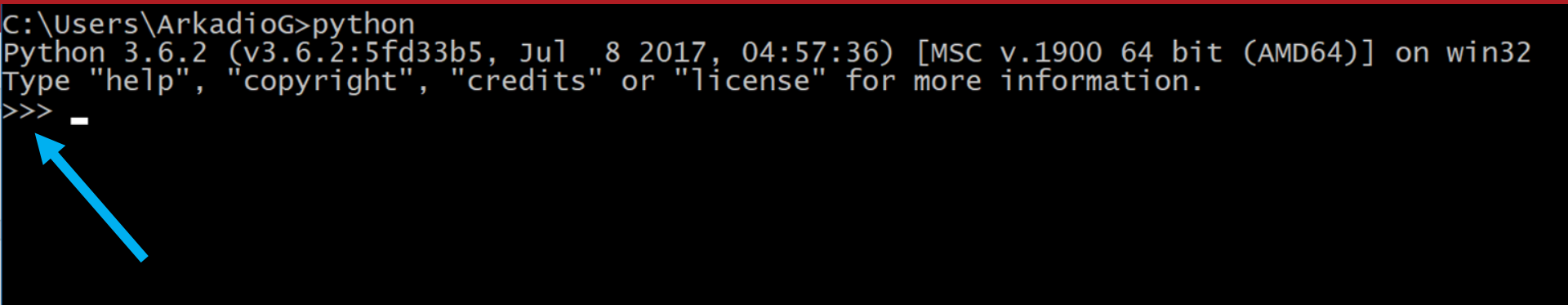
KOMPILOWANE

- Cały program jest kompilowany
- Z reguły szybszy
- Poprawka błędu wymaga ponownej kompilacji
- Niektóre błędy możliwe do wykrycia już na etapie kompilacji

INTERPRETOWANE

- Interpretowana jest linijka po linijce
- Z reguły wolniejszy
- Łatwiejszy w obsłudze
- Prostsza składnia
- Błędy dopiero podczas uruchomienia kodu

INTERPRETER PYTHON



INTERPRETER PYTHON

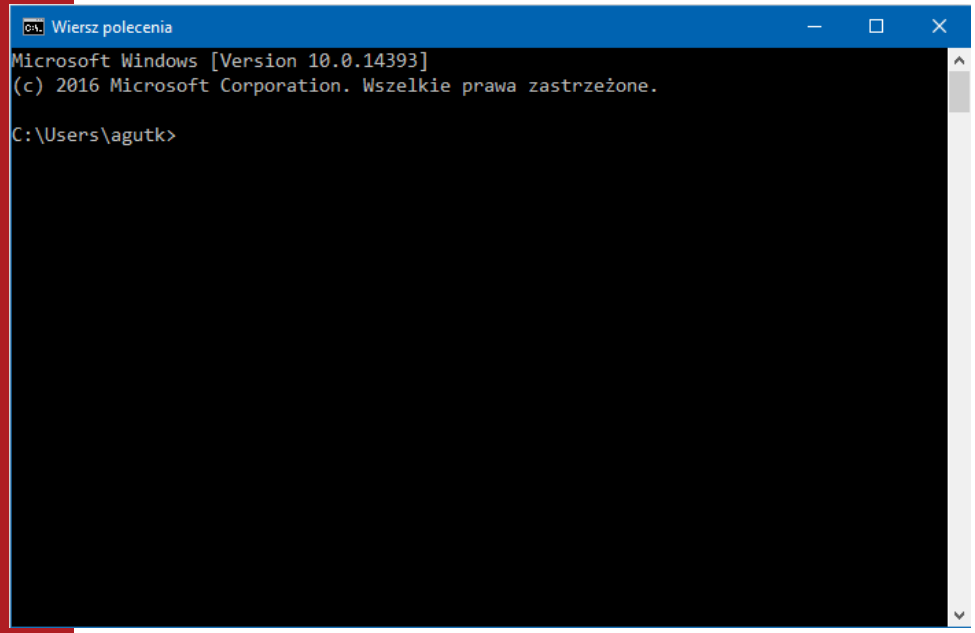
- uruchamianie

w wierszu poleceń wpisujemy `python` lub `python3.6`
i naciskamy enter

- wychodzenie

wpisujemy `exit()` i naciskamy enter

Wiersz poleceń / terminal



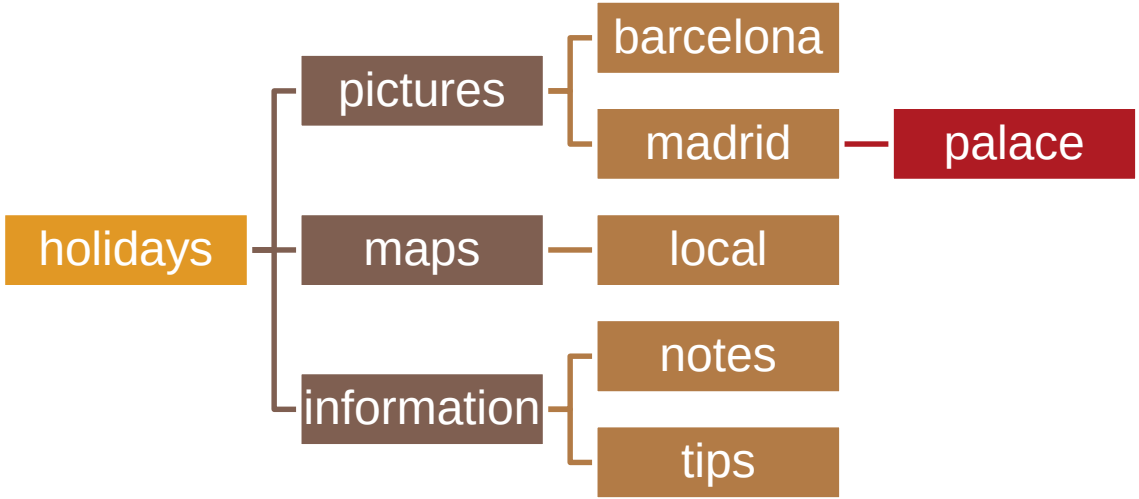
Cheatsheets

- http://bit.ly/mac-ter
- http://bit.ly/wincommands

Wiersz poleceń

/? –help -- help --h - pomoc
mkdir / md – tworzenie folderu
rmdir / rd – usuwanie folderu
move – przenoszenie zmiana nazwy
dir / ls – wyświetlenie zawartości folderu
cd – przejście do innej lokalizacji
pwd – obecny folder (linux, osx, powershell)
type / cat – wyświetlenie zawartości pliku
touch – utworzenie pliku
echo Lalalalla **>** plik.txt

WIERSZ POLECEŃ



1. Utwórz strukturę folderów

1. Zmień nazwę folderu barcelona na valencia
2. Usuń folder information
3. Będąc w folderze pictures dodaj folder barcelona do folderu maps
4. Będąc w folderze maps wyświetl zawartość folderu madrid

Git & GitHub

Git

Autor: Linus Torvalds

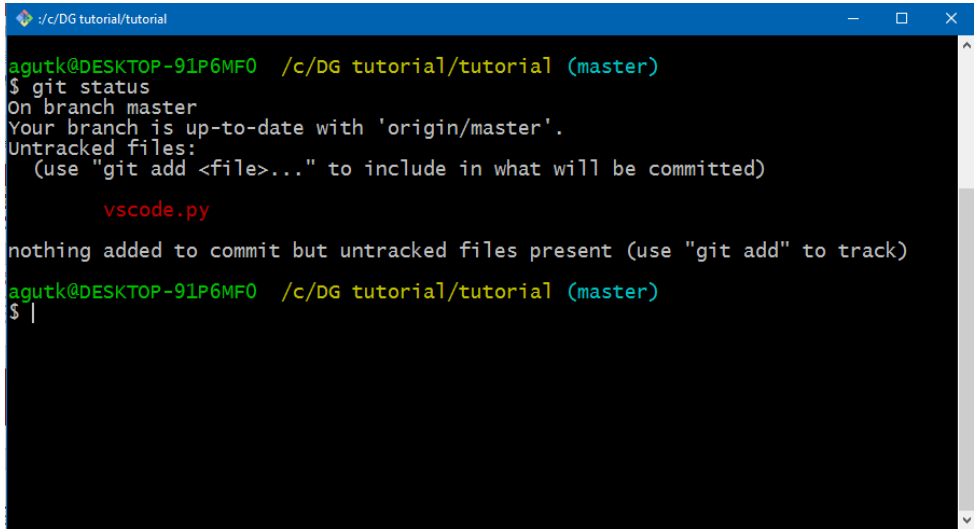
Rozproszony system wersjonowania plików

- każdy developer może pracować nad częścią kodu
- dev może mieć kilka wersji kodu
- umożliwia cofanie zmian, łączenie gałęzi (branch)

https://git-scm.com/

Git - konsola

wydajemy polecenia konsolą (wymagana instalacja git na Windows)



ściąga: https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf

PODSTAWOWE KOMENDY GIT

- git clone *https://...* - *skopiowanie(klonowanie) zdalnego repo na komputer*
- **git pull** - **pobranie zmian ze zdalnego repo**

- git init – zainicjowanie nowego lokalnego repozytorium
- git add – dodanie plików do wersjonowania
- git commit – zapamiętanie lokalnego stanu
- git push – wysłanie commita na zdalne repo

- git status

Git

Dokumentacja, video, książka Git Pro
https://git-scm.com/doc

Książka w wersji polskiej (1 edycja):
https://git-scm.com/book/pl/v1

Online tutorial z komend:
https://try.github.io/levels/1/challenges/1

GitHub

https://github.com

- repozytoria kodu w chmurze
- bezpłatne publiczne repozytoria
- najpopularniejsze miejsce z projektami opensource
- must-have dla programisty

Konkurencja – gitlab, bitbucket – dają bezpłatne prywatne repozytoria

GitHub

- zakładamy konto
- tworzymy repozytorium
- klonujemy na swój komputer
- zmieniamy kod
- commitujemy zmiany (zapisujemy do lokalnego repo)
- synchronizujemy z github

Polecenia w wierszu komend

Pliki ignorowane:
https://help.github.com/articles/ignoring-files/

GitHub

Kurs: https://services.github.com/on-demand/

Cheatsheet: https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf

Pomoc: https://help.github.com/

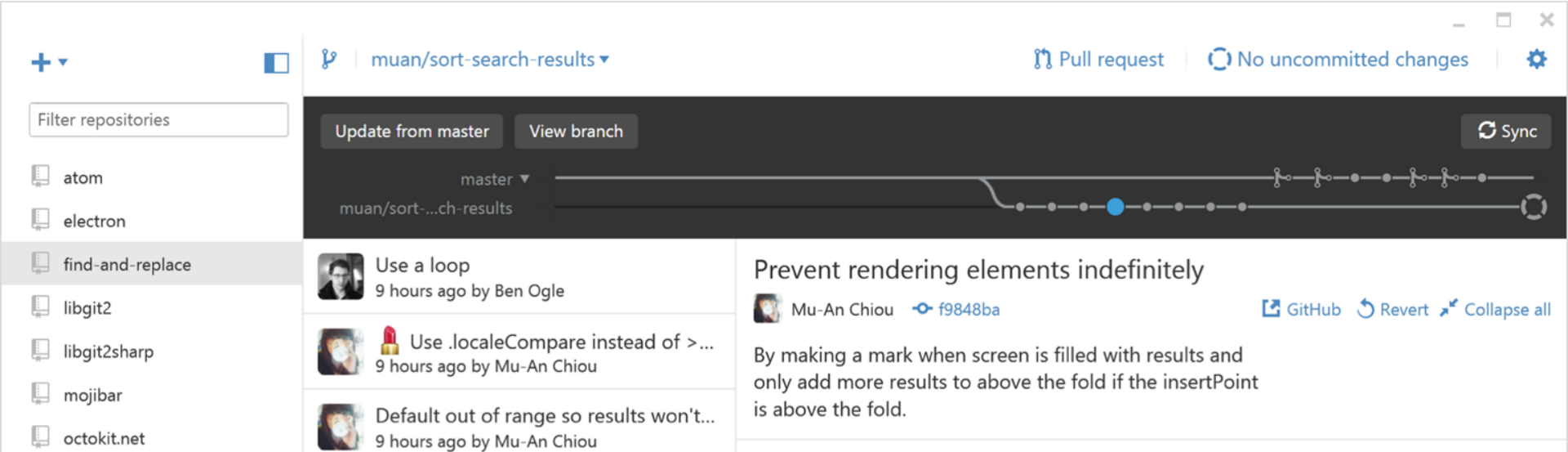
Video: https://www.youtube.com/githubguides , https://youtu.be/HVsySz-h9r4

https://services.github.com/resources/

GitHub Desktop

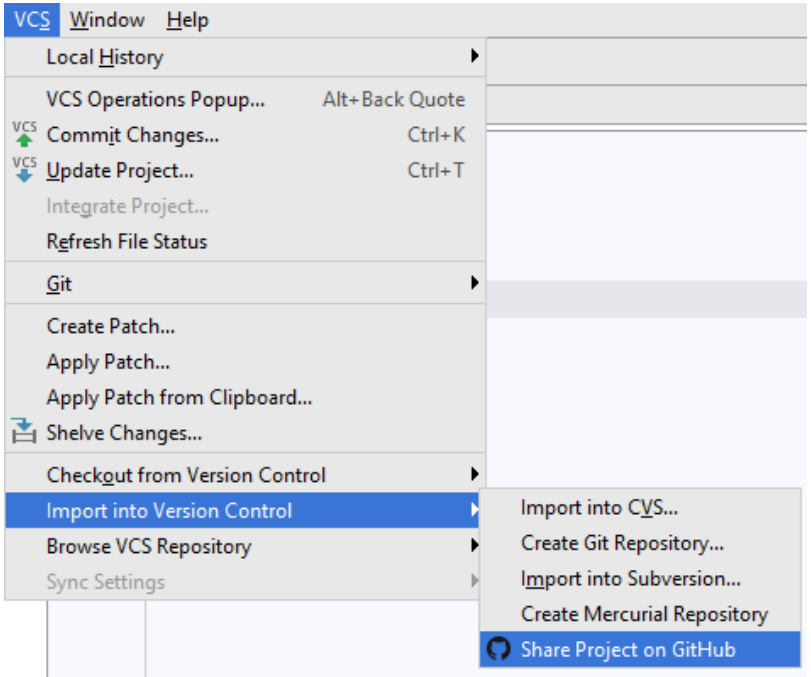
https://desktop.github.com/

alternatywne programy okienkowe:
https://git-scm.com/downloads/guis



GitHub & PyCharm

- tworzymy projekt
- w ustawieniach łączymy się z github (login, hasło)
- menu VCS:



PODSUMOWANIE

- Myślenie algorytmiczne
- Input, algorytm, output
- Pseudokod
- Języki kompilowane vs interpretowane
- Python
- Terminal / wiersz poleceń
- Git/GitHub

Thanks!!