

Podstawy programowania w języku Python

Dzień 1

Mateusz Cebula

- cebula.mat@gmail.com
- LinkedIn

Doświadczenie

- Apisystems Sp. z O. O. (C# developer, automation engineer)
- Apisystems Sp. z O. O. (Tech Leader of developers team)
- Staples Solutions (Full Stack Developer)
- SDA (Trener)
- InfoShare Academy (Trener)

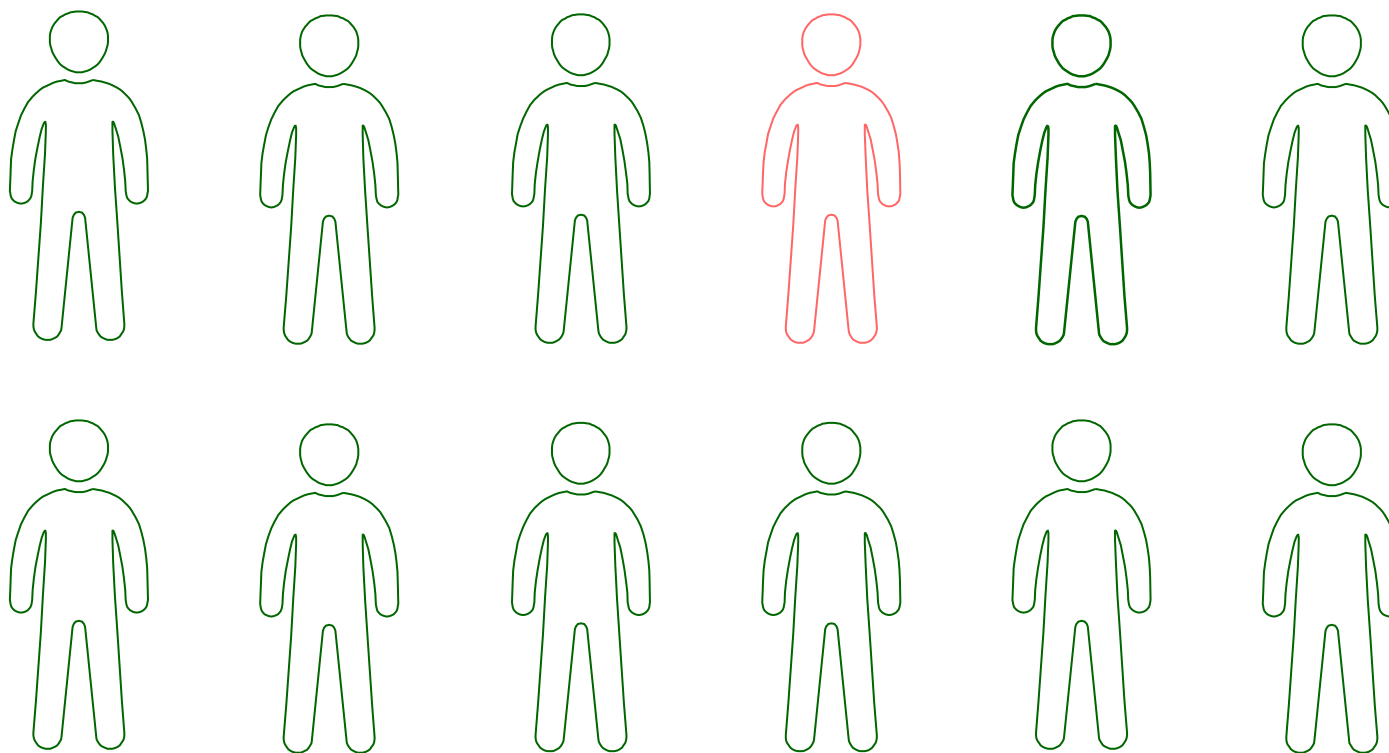
Kontakt

- cebula.mat@gmail.com
- <https://isapy12.slack.com>

Materiały i pomoc

- <https://docs.python.org>
- Trener :)
- StackOverflow
- <https://github.com/infoshareacademy/isa-python12>

Poznajmy się



Część pierwsza – podstawy programowania – spotkania 1-8

- podstawowe pojęcia, pamięć, typy danych
- instrukcje warunkowe, pętle
- kolekcje

Część druga – wstęp do obiektowości – spotkania 9-15

- obiektowość
- klasy, hierarchia klas, dziedziczenie
- praktyczne wykorzystanie (API, www, okienka)

Cel kursu

- fundamentalne pojęcia
- analiza problemów
- dobre praktyki
- składnia języka
- szukanie rozwiązań

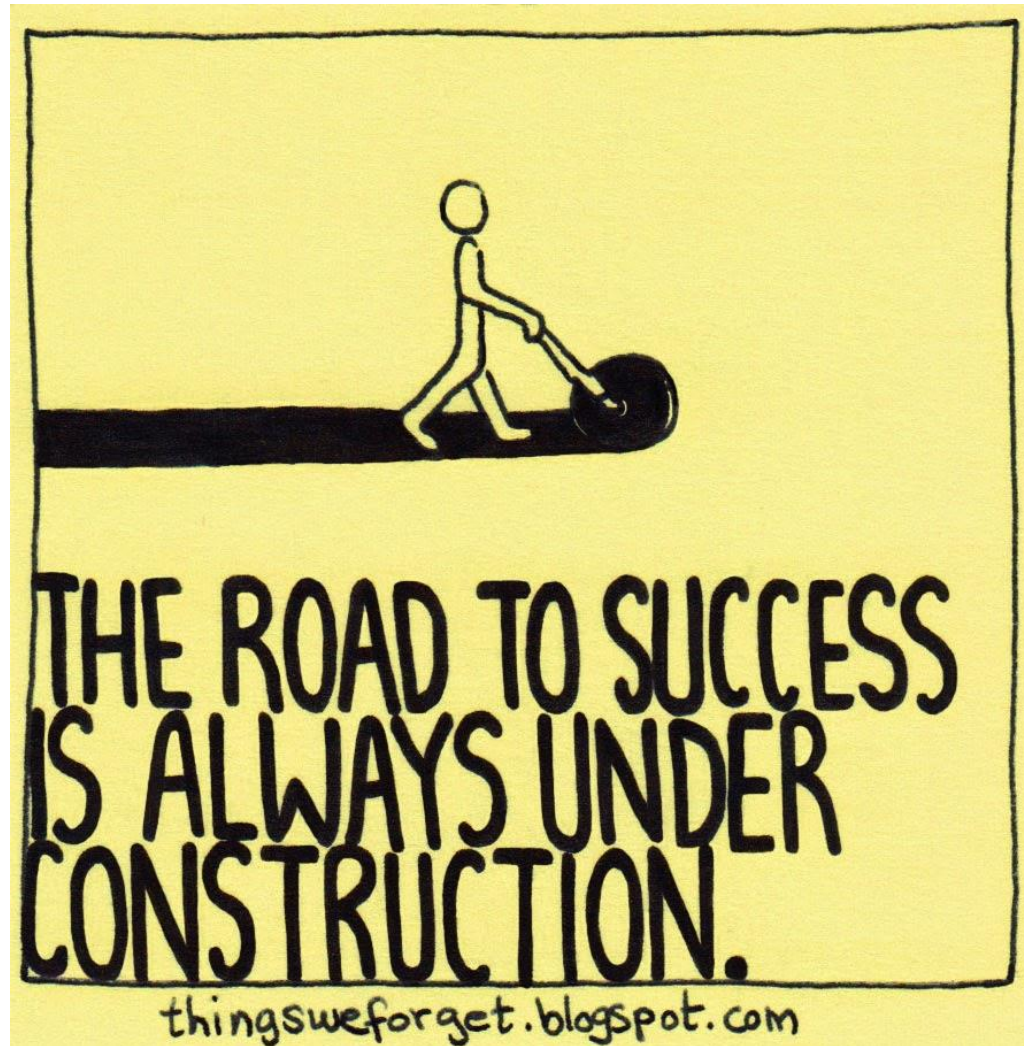


**I have not failed.
I've just found 10000
ways that won't work.**

Thomas A.
Edison



Nie liczy się to, jak daleko będziesz w
stosunku do kolegów, ale to, jak daleko
znajdziesz się po kursie, w stosunku
do **siebie** z dnia dzisiejszego



Agenda

Dzień 1

- jak to działa?
- myślenie algorytmiczne
- input, algorytm, output
- pseudokod
- Python i języki programowania
- terminal
- Git



Jak to działa?

System dziesiętny

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



Jak to działa?

System dziesiętny

1 2 3



Jak to działa?

System dziesiętny

100

10

1

1 2 3

Jak to działa?

System dziesiętny

100		10		1
1	2	3		
1 x 100	+	2 x 10	+	3 x 1

Jak to działa?

System dziesiętny

100		10		1
1	2	3		
100	+	20	+	3

Jak to działa?

System binarny

0, 1

Jak to działa?

System binarny

4	2	1
0	1	1
0 x 4	+ 1 x 2	+ 1 x 1

Jak to działa?

System binarny

4		2		1
0	1	1		
0	+	2	+	1

Jak to działa?

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

Jak to działa?

ASCII

i
105

S
83

A
65

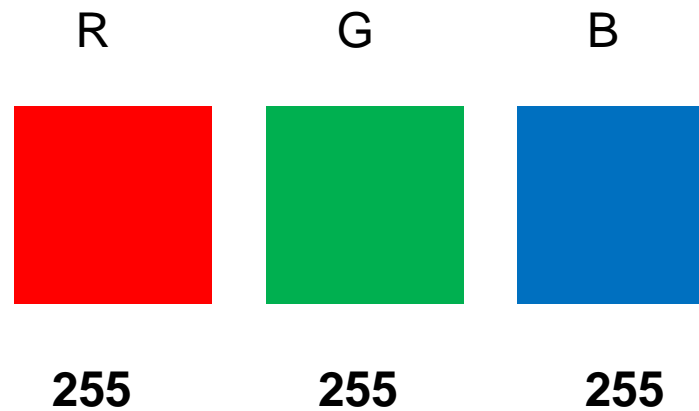
Jak to działa?

ASCII

Dec	Bin	Hex	Char	Dec	Bin	Hex	Char	Dec	Bin	Hex	Char	Dec	Bin	Hex	Char
0	0000 0000	00	[NUL]	32	0010 0000	20	space	64	0100 0000	40	@	96	0110 0000	60	`
1	0000 0001	01	[SOH]	33	0010 0001	21	!	65	0100 0001	41	A	97	0110 0001	61	a
2	0000 0010	02	[STX]	34	0010 0010	22	"	66	0100 0010	42	B	98	0110 0010	62	b
3	0000 0011	03	[ETX]	35	0010 0011	23	#	67	0100 0011	43	C	99	0110 0011	63	c
4	0000 0100	04	[EOT]	36	0010 0100	24	\$	68	0100 0100	44	D	100	0110 0100	64	d
5	0000 0101	05	[ENQ]	37	0010 0101	25	%	69	0100 0101	45	E	101	0110 0101	65	e
6	0000 0110	06	[ACK]	38	0010 0110	26	&	70	0100 0110	46	F	102	0110 0110	66	f
7	0000 0111	07	[BEL]	39	0010 0111	27	'	71	0100 0111	47	G	103	0110 0111	67	g
8	0000 1000	08	[BS]	40	0010 1000	28	(72	0100 1000	48	H	104	0110 1000	68	h
9	0000 1001	09	[TAB]	41	0010 1001	29)	73	0100 1001	49	I	105	0110 1001	69	i
10	0000 1010	0A	[LF]	42	0010 1010	2A	*	74	0100 1010	4A	J	106	0110 1010	6A	j
11	0000 1011	0B	[VT]	43	0010 1011	2B	+	75	0100 1011	4B	K	107	0110 1011	6B	k
12	0000 1100	0C	[FF]	44	0010 1100	2C	,	76	0100 1100	4C	L	108	0110 1100	6C	l
13	0000 1101	0D	[CR]	45	0010 1101	2D	-	77	0100 1101	4D	M	109	0110 1101	6D	m
14	0000 1110	0E	[SO]	46	0010 1110	2E	.	78	0100 1110	4E	N	110	0110 1110	6E	n
15	0000 1111	0F	[SI]	47	0010 1111	2F	/	79	0100 1111	4F	O	111	0110 1111	6F	o
16	0001 0000	10	[DLE]	48	0011 0000	30	0	80	0101 0000	50	P	112	0111 0000	70	p
17	0001 0001	11	[DC1]	49	0011 0001	31	1	81	0101 0001	51	Q	113	0111 0001	71	q
18	0001 0010	12	[DC2]	50	0011 0010	32	2	82	0101 0010	52	R	114	0111 0010	72	r
19	0001 0011	13	[DC3]	51	0011 0011	33	3	83	0101 0011	53	S	115	0111 0011	73	s
20	0001 0100	14	[DC4]	52	0011 0100	34	4	84	0101 0100	54	T	116	0111 0100	74	t
21	0001 0101	15	[NAK]	53	0011 0101	35	5	85	0101 0101	55	U	117	0111 0101	75	u
22	0001 0110	16	[SYN]	54	0011 0110	36	6	86	0101 0110	56	V	118	0111 0110	76	v
23	0001 0111	17	[ETB]	55	0011 0111	37	7	87	0101 0111	57	W	119	0111 0111	77	w
24	0001 1000	18	[CAN]	56	0011 1000	38	8	88	0101 1000	58	X	120	0111 1000	78	x
25	0001 1001	19	[EM]	57	0011 1001	39	9	89	0101 1001	59	Y	121	0111 1001	79	y
26	0001 1010	1A	[SUB]	58	0011 1010	3A	:	90	0101 1010	5A	Z	122	0111 1010	7A	z
27	0001 1011	1B	[ESC]	59	0011 1011	3B	;	91	0101 1011	5B	[123	0111 1011	7B	{
28	0001 1100	1C	[FS]	60	0011 1100	3C	<	92	0101 1100	5C	\	124	0111 1100	7C	
29	0001 1101	1D	[GS]	61	0011 1101	3D	=	93	0101 1101	5D]	125	0111 1101	7D	}
30	0001 1110	1E	[RS]	62	0011 1110	3E	>	94	0101 1110	5E	^	126	0111 1110	7E	~
31	0001 1111	1F	[US]	63	0011 1111	3F	?	95	0101 1111	5F	_	127	0111 1111	7F	[DEL]

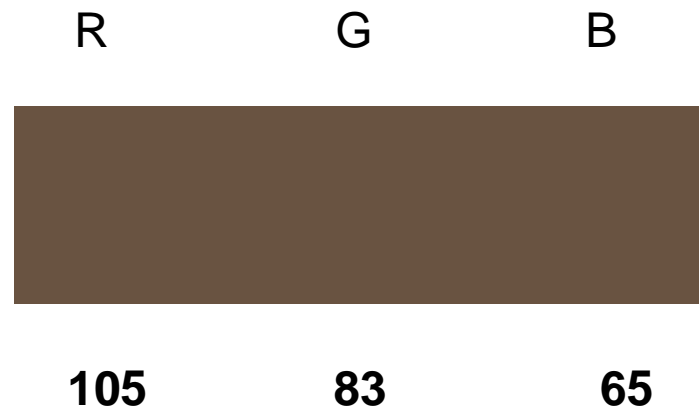
Jak to działa?

RGB



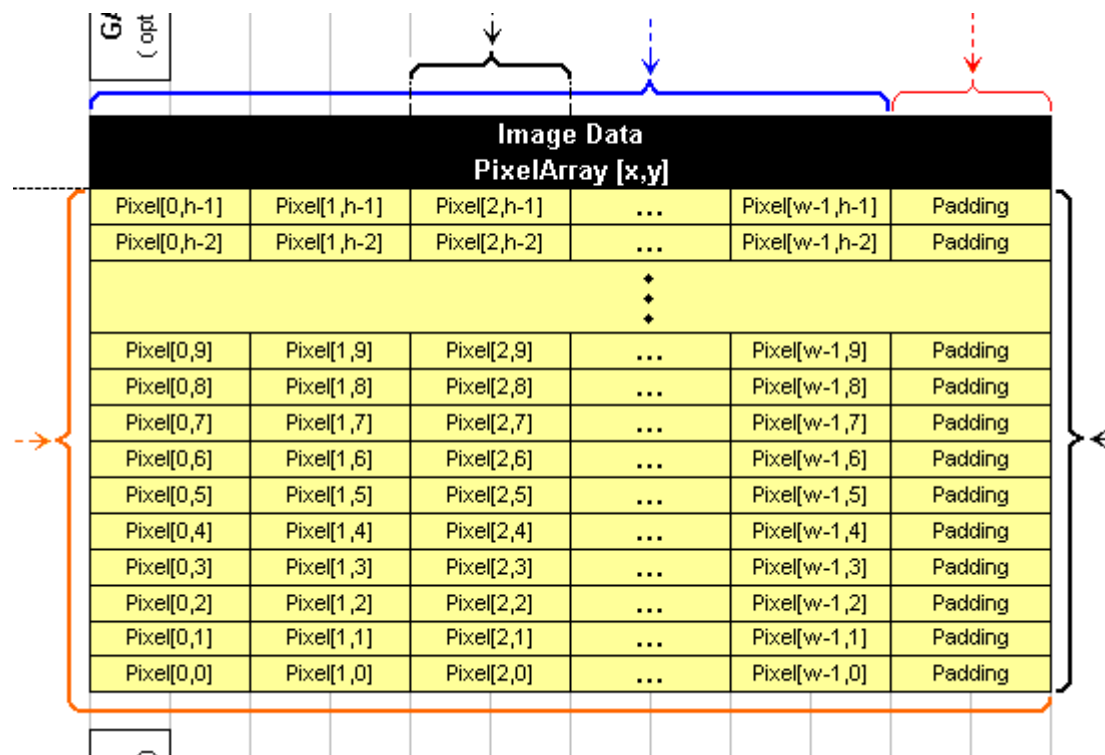
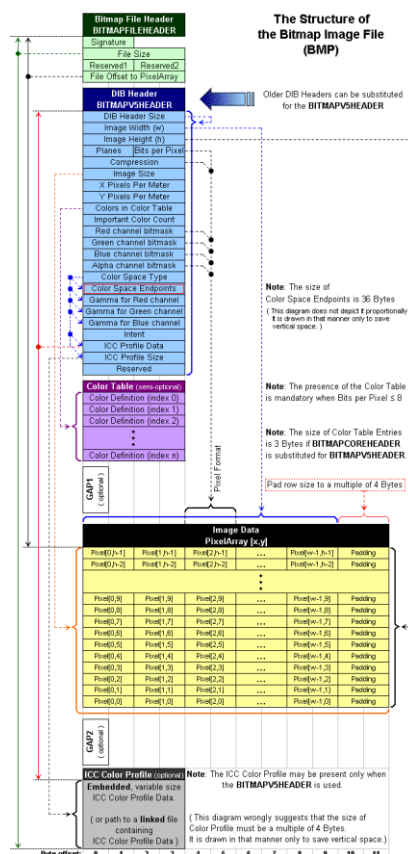
Jak to działa?

RGB



Jak to działa?

Obrazy (BMP)

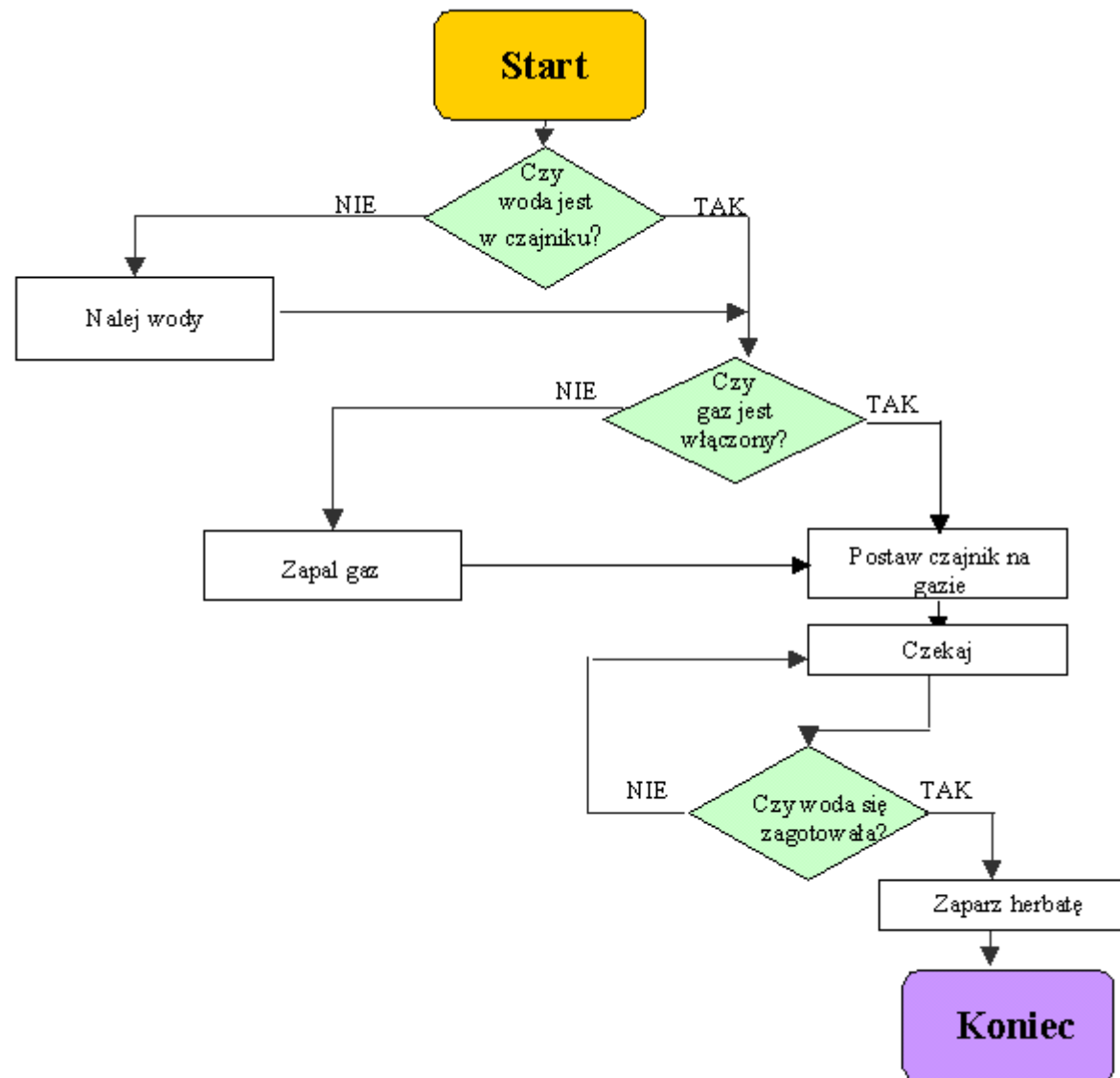


**Skończony zestaw instrukcji potrzebny
do wykonania zadania.**

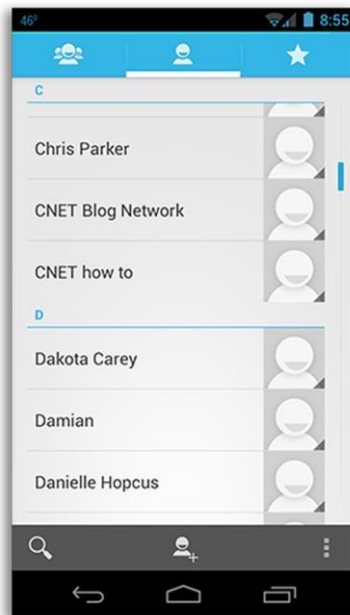
Przykłady z życia

- pieczenie ciasta
- mycie zębów
- ubieranie się
- jedzenie
- odrabianie lekcji
- sądownictwo
- w matematyce

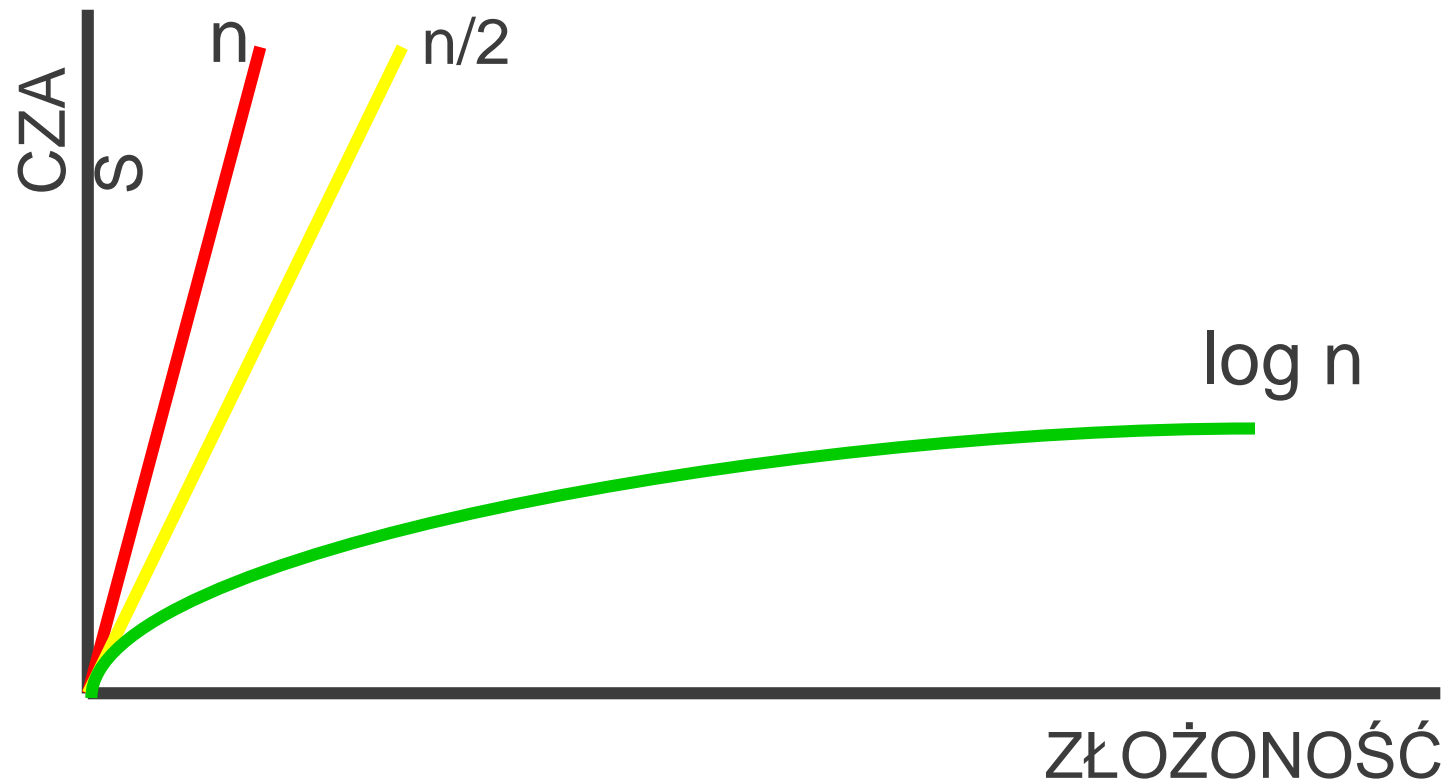
Schemat blokowy



Algorytm



Złożoność obliczeniowa (algorytmiczna)



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

Wiersz poleceń

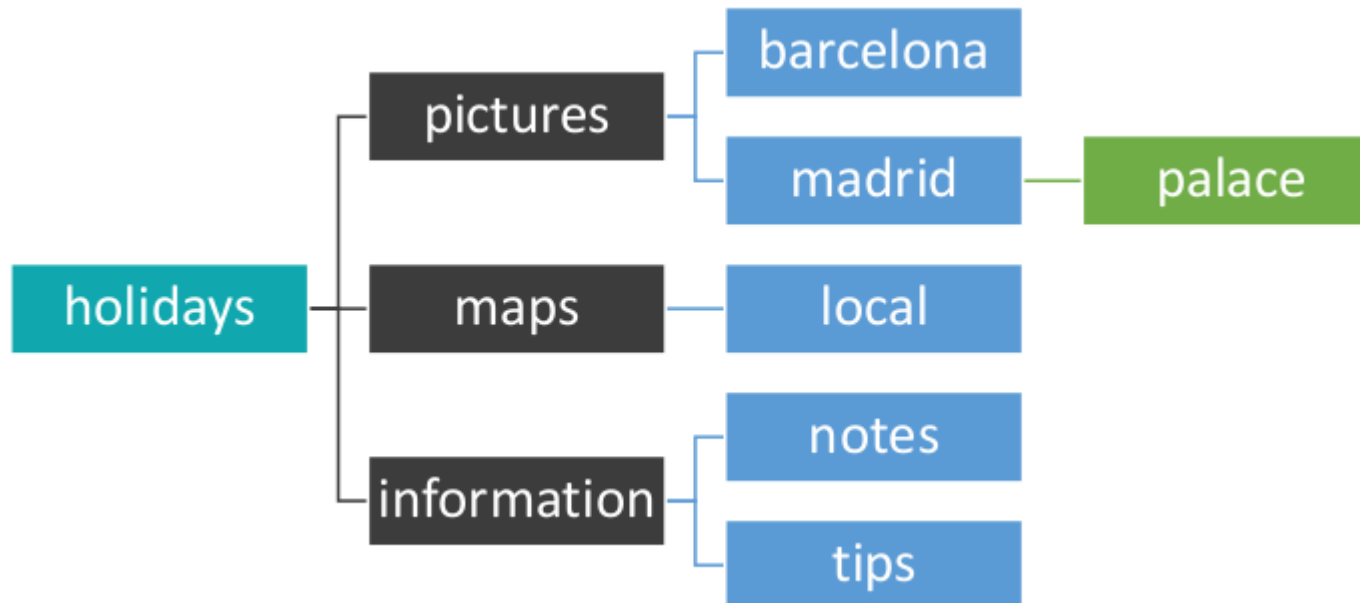
Terminale

- OSX: <https://bit.ly/mac-ter>
- Windows: <https://bit.ly/wincommands>
- Linux: <https://bit.ly/1Yxy9Ws>
 - **/?** **--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)
 - **type** / **cat** – wyświetlenie zawartości pliku
 - **touch** – utworzenie pliku
 - **echo** Lalalalla > plik.txt – zapis do pliku

Wiersz poleceń

Zadania

- utwórz strukturę katalogów
- zmień nazwę folderu barcelona na valencia
- usuń folder information
- będąc w folderze pictures dodaj folder barcelona do folderu maps
- będąc w folderze maps wyświetl zawartość folderu madrid



Języki programowania

Kompilowane

- cały program jest kompilowany
- z reguły szybszy
- poprawka błędy wymaga ponownej kompilacji
- dużo klamer {} i średników ;

Interpretowane

- interpretowana jest linijka po linijce
- z reguły wolniejszy
- łatwiejszy w obsłudze i utrzymaniu
- prostsza składania

Dlaczego

- prosta składnia
- kompaktowy kod
- kod niezależny od systemu
- wszechstronny
- popularny



Windows

- python

Linux / OSX

- python3 lub python3.6

- aby wyjść wpisujemy **exit()**

```
weblink@SonyVaio:~$ python3
Python 3.5.2 (default, Nov 23 2017, 16:37:01)
[GCC 5.4.0 20160609] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

```
C:\Users\ArkadioG>python
Python 3.6.2 (v3.6.2:5fd33b5, Jul 8 2017, 04:57:36) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```

Git:

- Linus Torvalds
- rozproszony system wersjonowania plików
- każdy developer może pracować nad częścią kodu
- można mieć wiele wersji kodu
- możliwość cofania zmian
- możliwość łączenia gałęzi (branch)



```
:/c/DG tutorial/tutorial

agutk@DESKTOP-91P6MF0 /c/DG tutorial/tutorial (master)
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    vscode.py

nothing added to commit but untracked files present (use "git add" to track)
agutk@DESKTOP-91P6MF0 /c/DG tutorial/tutorial (master)
$ |
```


Dokumentacja:

- <https://git-scm.com/doc>

Książka wersja polska:

- <https://git-scm.com/book/pl/v1>

Interaktywny tutorial z komend:

- <https://learngitbranching.js.org/>

Cheat sheet:

- <https://services.github.com/on-demand/downloads/github-git-cheat-sheet.pdf>
- Google Drive

GitHub

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

Alternatywy

- GitLab
- Bitbucket



Pomoce

- Pomoc: <https://help.github.com>
- Kurs: <https://lab.github.com>
- Video: <https://youtube.com/githubguides> / <https://youtu.be/HVsySz-h9r4>

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

<https://github.com>

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

Pliki ignorowane

- <https://help.github.com/articles/ignoring-files/>

Polecenia

`git clone https://...` - skopiowanie 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` – sprawdzenie stanu lokalnego repozytorium

IDE

- zintegrowane środowisko programistyczne
- pełne wsparcie dla Pythona
- uzupełnianie składni
- sprawdzanie składni
- integracja z git



Dzień 1

- jak to działa?
- myślenie algorytmiczne
- input, algorytm, output
- pseudokod
- Python i języki programowania
- terminal
- Git



Koniec

DZIĘKUJĘ