

Python, zajęcia 09.05.20

# Plan zajęć

- Algorytmy – pętla w pętli
- Algorytmy - sortowanie
- Tkinter – modyfikowanie plików

# Pętle – przypomnienie

Pętle już znacie. Wykorzystywaliśmy takie pętle:

```
lista = [1, 2, 3, 4, 5]
```

```
for liczba in lista:  
    print(liczba)
```

← Ta pętla przechodzi (**iteruje**)  
po elementach listy.

```
licznik = 1  
while licznik<=5:  
    print(licznik)  
    licznik += 1
```

← Ta pętla działa, dopóki jest  
spełniony warunek.

# Pętle – pętla for

Teraz poznacie trzecią, wykorzystywaną pętlę. To też jest pętla **for**, ale z użyciem funkcji **range()**:

```
for i in range(5):  
    print(i)
```

← Ta pętla przejdzie po wszystkich wartościach **i**, od 0 do 4 (**uwaga: nie będzie 5!**)

```
=====
```

```
0  
1  
2  
3  
4
```

← Wynik wywołania

# Pętle – pętla for

Wykorzystuje ona funkcję **range**. Funkcja **range** przyjmuje jeden albo dwa argumenty:

- **range(b)** – zwróci wartości od **0** do **b - 1**, czyli `range(10)` zwróci wartości od 0 do 9.
- **range(a, b)** – zwróci wartości od **a** do **b - 1**, czyli `range(1, 5)` zwróci wartości od 1 do 4.

Jakie wartości zwróci:

`range(100)` ?

`range(10, 15)` ?

# Pętle – pętla for

Przejdźmy teraz za pomocą tej pętli przez listę. Przypominam o numerowaniu listy (indeksach):

```
lista = [9, 3, 4, 1, 8]
```

Będzie miała takie indeksy (u góry):

0	1	2	3	4
9	3	4	1	8

Jakie wartości będą miały te elementy?

```
lista[0]  
lista[3]  
lista[6]
```

# Pętle – pętla for

Więc tą listę możemy przejść taką pętlą:

```
lista = [9, 3, 4, 1, 8]
for i in range(5):
    print(lista[i])
```

Jaką wadę ma to rozwiązanie? Co dałoby się zautomatyzować?

# Pętle – pętla for

Odpowiedź: **długość listy**.

Możemy ją pobrać używając funkcji **len(lista)**:

```
len(lista)
```

Czyli pętli użyjemy w ten sposób:

```
for i in range(len(lista)):  
    print(lista[i])
```

Teraz **i** będzie miało wartości od **0** do **długość listy – 1**.  
Idealnie!



# Zadania part 1

Zadania z pętli:

Plik *petle\_zadania.py* na githubie:

[github.com/jerzyklos/zajecia](https://github.com/jerzyklos/zajecia) folder *zajecia\_09\_05*

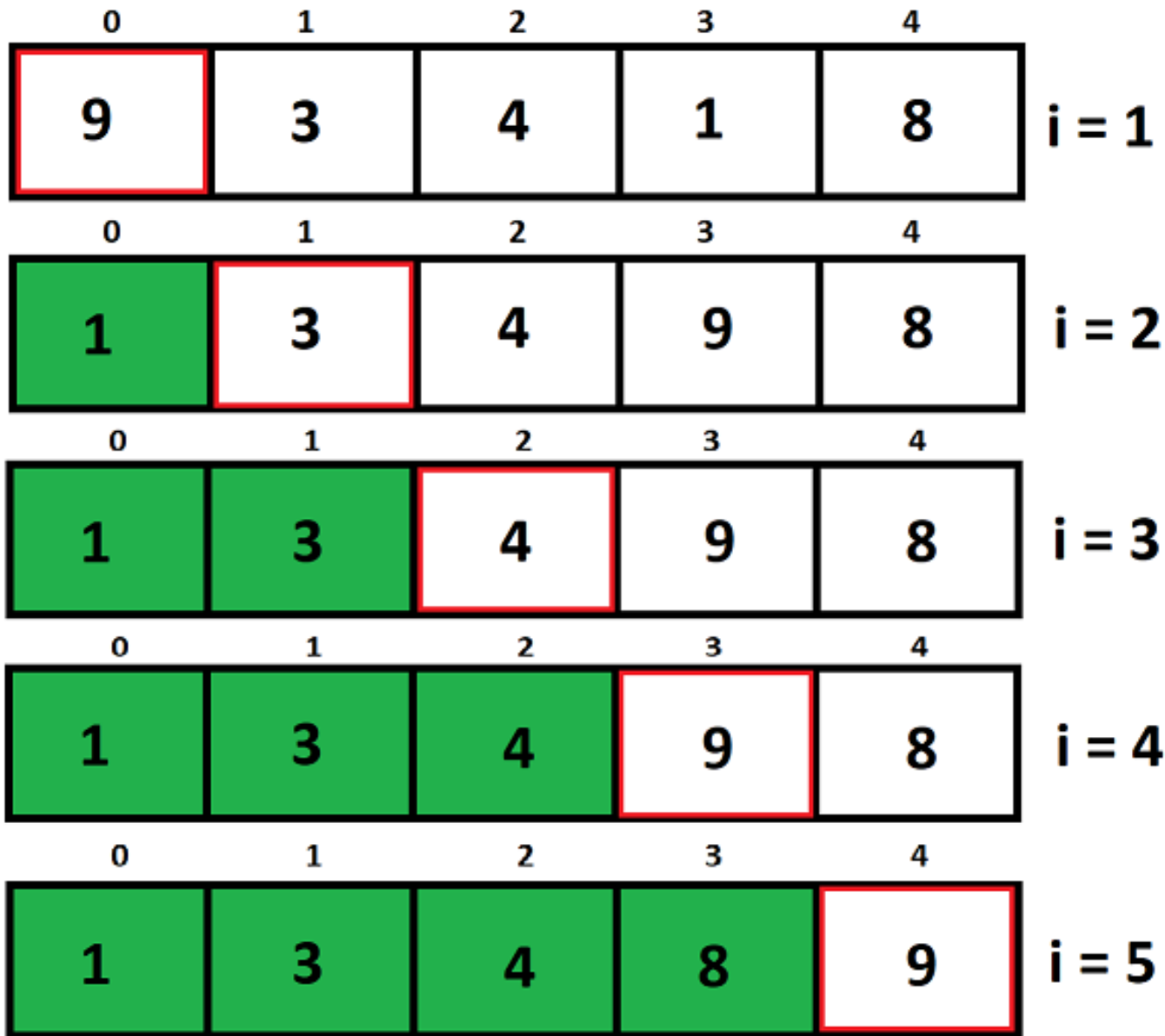
Discord: <https://discord.gg/svQQE3> - jeśli link nie będzie działał, nowy będzie na czacie

# Sortowanie przez wybieranie

Napiszemy funkcję, która będzie sortować listę. Najpierw musimy zrozumieć algorytm. W skrócie jest taki:

- Znajdź w liście najmniejszy element
- Daj go na pierwsze miejsce w liście
- Znajdź w liście (już bez pierwszego elementu) najmniejszy element
- Daj go na drugie miejsce w liście
- Znajdź w liście (już bez pierwszego i drugiego elementu) najmniejszy element
- Daj go na trzecie miejsce w liście...
- Powtarzamy, aż cała lista będzie posortowana (czyli tyle razy, ile jest elementów w liście)

# Sortowanie przez wybieranie



# Sortowanie przez wybieranie

0	1	2	3	4	
9	3	4	1	8	$i = 1$

j zmienia się od 0 do 4

0	1	2	3	4	
1	3	4	9	8	$i = 2$

j zmienia się od 1 do 4

0	1	2	3	4	
1	3	4	9	8	$i = 3$

j zmienia się od 2 do 4

0	1	2	3	4	
1	3	4	9	8	$i = 4$

j zmienia się od 3 do 4

0	1	2	3	4	
1	3	4	8	9	$i = 5$

j zmienia się od 4 do 4

# Zadania part 2

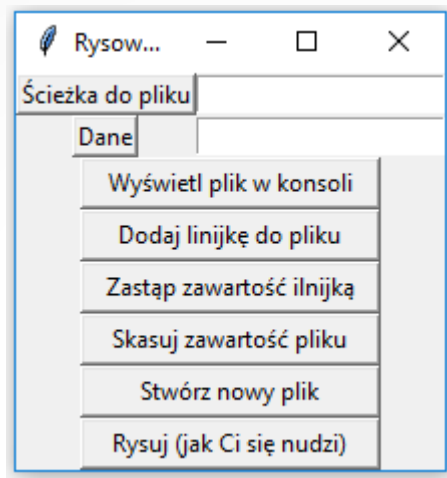
Teraz zrobimy razem przykłady a potem spróbujecie napisać sami sortowanie.

Plik *sortowanie\_zadanie.py* na githubie:  
[github.com/jerzyklos/zajecia](https://github.com/jerzyklos/zajecia) folder *zajecia\_09\_05*

Discord: <https://discord.gg/svQQE3> - jeśli link nie będzie działał, nowy będzie na czacie

# Tkinter – apka do modyfikacji plików

Chcemy zrobić taką aplikację:



A więc podpiąć obsługę plików pod odpowiednie guziki.

# Zadania part 3

Wyświetlanie listy elementów z pliku:

Plik *guziki\_zadanie.py* na githubie:

[github.com/jerzyklos/zajecia](https://github.com/jerzyklos/zajecia) folder *zajecia\_09\_04*

Discord: <https://discord.gg/svQQE3> - jeśli link nie będzie działał, nowy będzie na czacie