

BEZPŁATNE WARSZTATY DLA POCZĄTKUJĄCYCH

# PYTHON

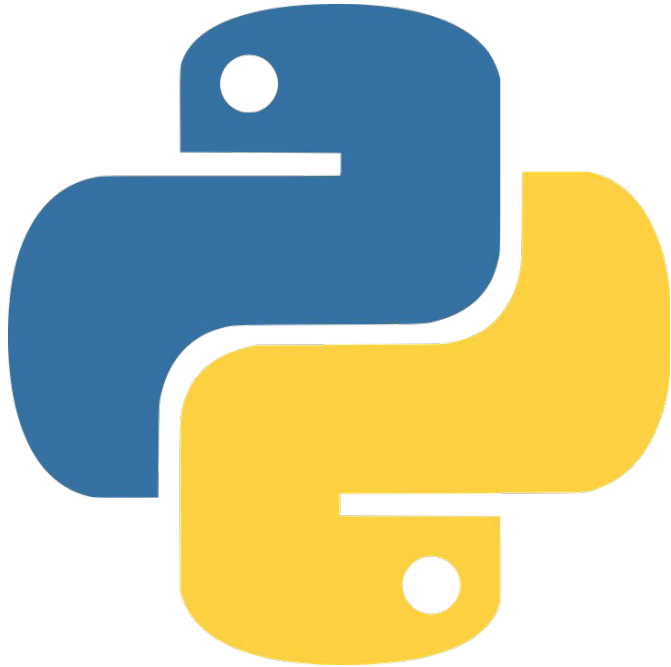


ŚRODA, 03.07.2019 | **GDAŃSK**

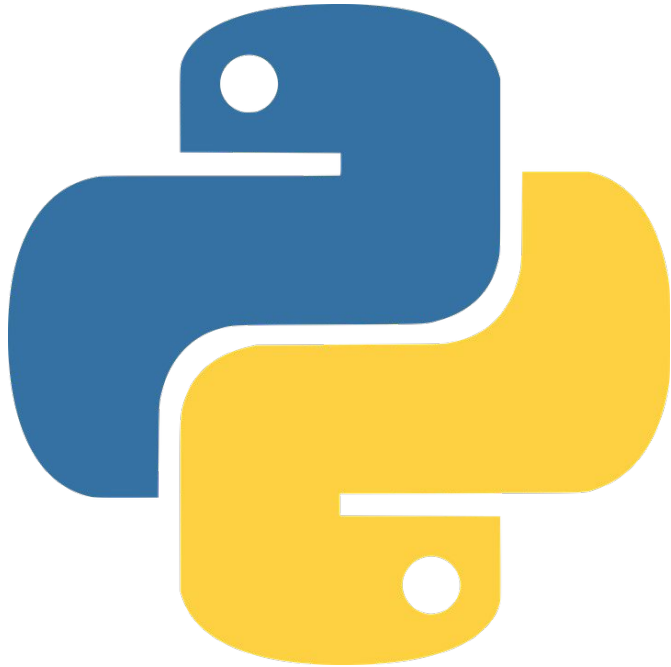
STWÓRZ APLIKACJĘ DO WIZUALIZACJI  
DANYCH O POŁOŻENIU ROWERÓW MIEJSKICH



# Dlaczego Python?



# Dlaczego Python?



# Narzędzia

Edytor kodu (edytor plików tekstowych, PyCharm)

Interpreter Python'a

Interaktywna konsola

# Czym jest kod?

# Czym jest kod?

Pokrój cebulę i dodaj ją do sosu



# Czym jest kod?

Pokrój cebulę i dodaj ją do sosu

# Czym jest kod?

Pokrój cebulę i dodaj ją do sosu

```
pokrojona_cebula = pokrój(cebula)  
sos_z_cebulą = połącz_składniki(sos, pokrojona_cebula)
```



# Czym jest kod?

Pokrój cebulę i dodaj ją do sosu

```
pokrojona_cebula = pokroj(cebula)  
sos_z_cebula = połącz_składniki(sos, pokrojona_cebula)
```

# Zmienna

Nadajemy czemuś nazwę (np. pokrojona\_cebula)

Możemy odwołać się do niej w różnych częściach programu

Reprezentuje dane, na których operujemy

# Operator

```
pokrojona_cebula = pokrój(cebula)  
sos_z_cebula = połącz_składniki(sos, pokrojona_cebula)
```

# Operator

```
pokrojona_cebula = pokrój(cebula)
```

```
sos_z_cebula = połącz_składniki(sos, pokrojona_cebula)
```

```
średnia = (ocena_z_egzaminu + ocena_z_odpowiedzi) / 2
```

# Operatory

Arytmetyczne (np. +, -, \*, /, %)

Przypisania (np. =, +=, -=)

Porównania (np. ==, !=, <, >)

Logiczne (and, or, not)

# Typ zmiennej

```
result = 3 + 4  
print(result)  
7
```

# Typ zmiennej

```
result = 3 + 4  
print(result)  
7
```

```
full_name = 'Mikołaj' + ' ' + 'Lewandowski'  
print(full_name)  
'Mikołaj Lewandowski'
```

# Typy zmiennych

**int** - liczby całkowite (np. **28**)

**float** - liczby zmiennoprzecinkowe (np. **47.15**)

**str** - tekst (np. **'Hello world!'**)

**bool** - wartość logiczna (Prawda albo Fałsz, **True/False**)



# Typy zmiennych

**list** - lista elementów (np. `[10, 27, 140]`)

**dict** - słownik, inaczej lista par klucz -> wartość (np. `{'name': 'Mikołaj'}`)

**NoneType** - ten typ posiada specjalna zmienna **None**, oznacza 'nic'

**Możemy tworzyć własne typy (KLASY)**

# Instrukcja warunkowa

Jeżeli kupujesz więcej niż 5 produktów otrzymasz 20% rabatu

# Instrukcja warunkowa

Jeżeli kupujesz więcej niż 5 produktów otrzymasz 20% rabatu

```
if number of products > 5:  
    total_price = total_price * 0.8
```

# Instrukcja warunkowa

Jeżeli kupujesz więcej niż 5 produktów otrzymasz 20% rabatu, a jeżeli przynajmniej 3 to 10% rabatu

```
if number of products > 5:  
    total_price = total_price * 0.8  
elif number of products >= 3:  
    total_price = total_price * 0.9
```

# Instrukcja warunkowa

Jeżeli kupujesz więcej niż 5 produktów otrzymasz 20% rabatu, a jeżeli przynajmniej 3 to 10% rabatu. Jeżeli nie dostałaś/eś rabatu otrzymasz niespodziankę.

```
if number of products > 5:  
    total price = total price * 0.8  
elif number of products >= 3:  
    total_price = total_price * 0.9  
else:  
    gift = get_random_gift()
```

# Pętle for

```
student_grades = [5, 4, 6, 3]  
sum_of_grades = 0
```

```
for grade in student_grades:  
    sum_of_grades += grade
```

# Funkcje - definicja

```
def get_full_name(first_name, last_name):  
    full_name = first_name + ' ' + last_name  
    return full_name
```

# Funkcje - wywołanie

```
def get_full_name(first_name, last_name):  
    full_name = first_name + ' ' + last_name  
    return full_name
```

```
my_name = get_full_name('Mikołaj', 'Lewandowski')  
your_name = get_full_name('Ola', 'Kowalska')
```



# Metoda - funkcja na zmiennej (obiekcie)

```
words = 'I really like Python'.split(' ')
print(words)
['I', 'really', 'like', 'Python']
```

# String - przydatne operacje

```
some_text = 'abc'
```

Długość:

```
len(some_text)
```

```
>> 3
```

Wartość pod indeksem:

```
some_text[0]
```

```
>> a
```

```
some_text[1]
```

```
>> b
```

# String - przydatne operacje

Podział po separatorze:

```
some_text.split('b')
```

```
>> ['a', 'c']
```

Łączenie (konkatenacja):

```
'Mikołaj' + ' ' + 'Lewandowski'
```

```
>> 'Mikołaj Lewandowski'
```

# Lista - przydatne operacje

```
student_grades = [5, 4, 6, 3]
```

Długość (liczba elementów):

```
len(student_grades)
```

```
>> 4
```

Wartość pod indeksem:

```
student_grades[0]
```

```
>> 5
```

```
student_grades[2]
```

```
>> 6
```

# Słownik - przydatne operacje

```
student_grades = {'Math': 5, 'Biology': 4}
```

Wartość pod kluczem:

```
student_grades['Math']
```

```
>> 5
```

```
student_grades['Biology']
```

```
>> 4
```

# Operacje na plikach

```
with open('C:\Documents\file.txt', 'r') as bikes_file:  
    bikes_data = bikes_file.readlines()
```

# Wykorzystanie bibliotek, import

```
import csv
```

```
from csv import DictReader
```