

# PyStart #7 Kolekcje - wprowadzenie

## Co to jest typ złożony?

12

True

3.49

alamakota

("Kacper", "Sieradziński")

[1, 20, 15, 30]

{ "Polska": "Warszawa", "Francja": "Paryż" }

**Kolekcje**

# PyStart #7 Kolekcje - wprowadzenie

## Jakie poznamy kolekcje?



Tuple / Krotki

Listy

Słowniki

Zbiory

Stringi / napisy

# PyStart #7 Kolekcje wprowadzenie

## Jakie poznamy kolekcje?



A diagram illustrating the classification of Python collections. It features two rounded rectangular boxes: a dark purple one on the left labeled 'Mutowalne' and a bright pink one on the right labeled 'Niemutowalne'. A vertical wavy line separates the two boxes. Above the pink box, a horizontal wavy line extends to the right edge of the frame.

**Mutowalne**

**Niemutowalne**

# PyStart #8 Range - zakres liczb

## Range - Jak wygląda range?



Wartości od zera do limitu

```
range(limit)
```

Wartości od start do limitu

```
range(start, limit)
```

Wartości od start do limitu z krokiem

```
range(start, limit, step)
```

- Limit nie zalicza się do range'a
- Wartość domyślna dla start to 0
- Wartość domyślna dla step to 1

# PyStart #8 Range - zakres liczb

## Range - Jak go wyświetlić?



By zobaczyć wartości z których jest stworzony range można zamienić na tuple.

```
Można to wykonać za pomocą funkcji tuple()  
tuple(range(10))
```

```
Zobaczysz wynik  
(0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
```

# PyStart #8 Range - zakres liczb

## Range - zakres liczb



```
>>> range(9)
range(0, 9)
>>>

>>> tuple(range(9))
(0, 1, 2, 3, 4, 5, 6, 7, 8)

>>> tuple(range(1,9))
(1, 2, 3, 4, 5, 6, 7, 8)

>>> tuple(range(1,27,3))
(1, 4, 7, 10, 13, 16, 19, 22, 25)
```

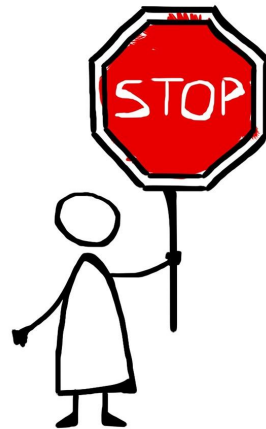
- Zakres od 0 do 9
- Wyświetlanie zakresu
- Określenie od-do
- Modyfikacja kroku w zakresie

# PyStart #8 Range - zadania

## Zadania dla nabrania wprawy

Korzystając z `tuple(range(...))` utwórz:

- Zakres liczb od 99 do 110
- Zakres liczb parzystych od 100 do 140
- Wszystkie liczby podzielne przez 3 od 3 do 33.



# PyStart #9 Tupla

## Cechy dobrej Tupli



→ Do deklaracji można użyć nawiasu, ale można też go pominąć

```
personal_information = 'Kacper', 'Sieradziński'  
children = (1, 2, 3)  
cities = ('Gdynia', )  
  
print(cities)  
print(personal_information)
```



# PyStart #9 Tupla

## UWAGA NA RÓŻNICE!



→ Uważaj na ten błąd!

```
test1 = (1024)
print(test1)

test2 = (1024,)
print(test2)
```

# PyStart #9 Tupla

## Cechy dobrej Tupli



→ Indeksowana -> element wskazywany liczbą

```
>>> cities = "Gdynia", "Sochaczew"  
>>> cities[0]  
'Gdynia'  
>>> cities[1]  
'Sochaczew'
```

# PyStart #9 Tupla

## Cechy dobrej Tupli



→ Jest niemutowalna

```
>>> cities = "Gdynia", "Sochaczew"  
>>> cities[0] = "Gdańsk"  
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
TypeError: 'tuple' object does not support item assignment
```

# PyStart #9 Tupla

## Cechy dobrej Tupli



- Do deklaracji można użyć nawiasu
- Do rzutowania używamy funkcji `tuple()`
- Indeksowana -> element wskazywany liczbą
- Jest niemutowalna
- Kolejność jest zachowana

# PyStart #9 Tuple - zadania

## Zadania dla nabrania wprawy

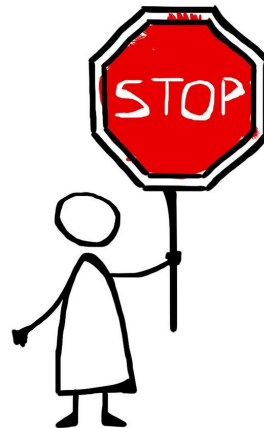
Sprawdź ostrzeżenie, czym różni się wyrażenie

`print(('owca'))` od `print(('owca',))`

możesz też sprawdzić typy za pomocą funkcji `type()`

-----

Utwórz tuplę zawierającą nazwy 5 owoców,  
a następnie wyświetl nazwy pierwszego, trzeciego  
i piątego elementu, korzystając z jednej instrukcji `print`



# PyStart #9 Operacje na tuplach

## Co możemy z nimi zrobić?



```
>>> (1,2,3) * 3  
(1, 2, 3, 1, 2, 3, 1, 2, 3)
```

```
>>> a = (1, 2, 3)  
>>> b = (4, 5, 6)  
>>> a += b  
>>> a  
(1, 2, 3, 4, 5, 6)
```

# PyStart #9 Operacje na tuplach

## Wypakowywanie



```
>>> first_name, _, last_name = 'Cyprian', 'Kamil', 'Norwid'
>>> first_name
'Cyprian'
>>> last_name
'Norwid'
>>> _
'Kamil'
```

# PyStart #9 Operacje na tuplach

## Operator in



```
>>> animals = 'Frog', 'Pig', 'Cow'
>>> 'Pig' in animals
True
>>> 'Python' in animals
False
```



# PyStart #9 Tuple - zadania

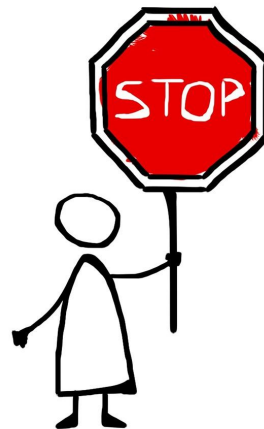
## Zadania dla nabrania wprawy

Przygotuj tuple zawierającą listę imion. Za każdym uruchomieniem programu pytaj użytkownika o imię. Jeśli znajduje się ono na liście wyświetl komunikat pomyślny, a jeśli nie to powiadom go o błędzie.

```
Podaj imię?
```

```
Alina
```

```
To imię znajduje się na liście!
```



# PyStart #10 Slicing

## Operator : do wycinania



Wywołanie wygląda tak:

**nazwa[początek:koniec:krok]**

- **początek** - wskazanie indeksu od którego zaczynamy. Z wartością dodatnią od początku, jako wartość ujemna od końca.
- **koniec** - analogicznie jak początek, jeśli nie będzie takich elementów zostanie zwrócony pusty zakres
- **krok** - co która wartość ma być zwracana, od początku lub od końca (ujemna)

# PyStart #10 Slicing

## Operator : do wycinania



```
>> fruits = 'mango', 'peach', 'lemon', 'grapefruit', 'apple', 'banana', 'cherry'
>> fruits[0]
mango

>> fruits[0:3] # to samo co fruits[:3]
('mango', 'peach', 'lemon')

>> fruits[1:4]
('peach', 'lemon', 'grapefruit')

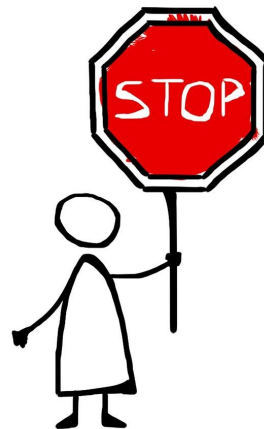
>> fruits[::2]
('mango', 'lemon', 'apple', 'cherry')

>> fruits[::-1]
('cherry', 'banana', 'apple', 'grapefruit', 'lemon', 'peach', 'mango')
```

# PyStart #10 Slicing

## Operator : do wycinania

- Za pomocą range i tuple stwórz listę liczb od 1 do 100
- Wyciągnij tylko pierwsze 10 liczb.
- Wyciągnij tylko ostatnie 10 liczb. (minus od końca)
- Wyciągnij z tej listy tylko wartości nieparzyste
- Wyciągnij z listy tylko wartości parzyste w kolejności od największej do najmniejszej.



# PyStart #10 Przydatne funkcje

## len, max, min, sum



```
>>> my_fav = (1, 9, 11, 18, 6, 3)
>>> len(my_fav)
6
```

```
>>> max(my_fav)
18
>>> min(my_fav)
1
>>> _
```

```
>>> my_fav = [1, 9, 11, 18, 6, 3]
>>> sum(my_fav)
48
```

# PyStart #10 Przydatne funkcje

## Sortowanie



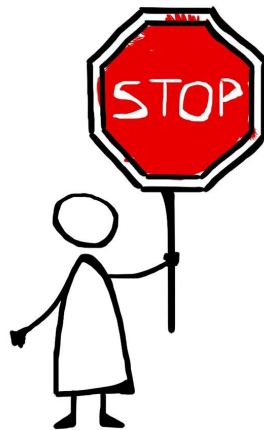
```
numbers = 1, 10, 9, 11, 18, 6  
sorted(numbers)
```

```
numbers = 1, 10, 9, 11, 18, 6  
sorted(numbers, reverse=True)
```

## #10 Przydatne funkcje `len`, `max`, `min`, `sum`

- Stwórz w IDE program, w którym zadeklarujesz tuplę odpowiadającą ocenom, jakie uzyskał uczeń.
- Wyświetl posortowane oceny od najniższej do najwyższej.
- Policz średnią jego ocen oraz odpowiedz czy zostanie świadectwo z paskiem (użyj `if'a`)

Aby otrzymać pasek, uczeń musi uzyskać  
średnią ocen  $\geq 4.75$



# PyStart #10 For... jak przejść po liście

## Informacje wstępne



- Do czego służy pętla?
- Czym jest indeks?
- Kiedy pętla się kończy, czyli czym jest **break** ?
- Idziemy dalej czyli **continue**



# PyStart #10 For... jak przejść po liście

## Budowa pętli



```
>>>  
>>> for fruit in fruits:  
...     print(fruit)  
...  
apple  
mango  
lemon  
>>>
```

# PyStart #10 For... jak przejść po liście

## Kilka przydatnych funkcji



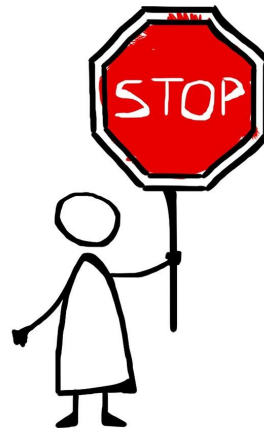
- Dostęp zarówno do indeksu i elementu  
for **indeks**, **element** in **enumerate**(**tupla**):
- Iteracja, przechodzenie od końca  
for **element** in **reversed**(**tupla**):
- Posortowane elementy rosnąco  
for **element** in **sorted**(**tupla**)
- Jak to połączyć? :)  
for **element** in **reversed**(**sorted**(**tupla**))  
for **element** in **sorted**(**tupla**, reversed=True)

# PyStart #10 For... jak przejść po liście

## Zadania dla nabrania wprawy

1. Z podanej tupli liczb wyświetl te, które są podzielne przez 4 i te które są podzielne przez 5.
2. Z podanej listy imion o różnej długości wyświetl tylko te które są dłuższe niż 5 znaków. Długość imienia sprawdź funkcją `len()`
3. Umieszczając pętle for w drugiej pętli for wyświetl trójkąt jak w przykładzie:

```
1 2 3 4
1 2 3
1 2
1
```



# PyStart #11 Stringi - łańcuchy znaków

## Charakterystyka ogólna



- Do deklaracji używamy apostrofów'
- Do rzutowania używamy funkcji `str()`
- Indeksowana -> element wskazywany liczbą
- Stringi są niemutowalne
- Kolejność elementów jest zachowana

# PyStart #11 Stringi, czyli praca z tekstem

## Przydatne funkcje



```
>>> name = 'PyStart'
>>> name.upper()
'PYSTART'
>>> name.lower()
'pystart'
>>> name.replace('start', 'www')
'Pywww'
>>>
>>> name.find('Py')
0
>>> name.find('start')
2
```

# PyStart #11 Stringi, czyli praca z tekstem

## Przydatne funkcje



```
>>> fruits = 'lemon,apple,peach,melon'
>>> fruits.split(',')
['lemon', 'apple', 'peach', 'melon']
```

```
>>> person = ['Zofia', 'Nałkowska']
>>> ' '.join(person)
'Zofia Nałkowska'
```

# PyStart #11 Stringi, czyli praca z tekstem

## Przydatne funkcje



```
>>> string = 'Programuj w Pythonie, Python to dobry wybór'
>>> string.count('Python')
2
```

```
>>> string.startswith('Programuj')
True
```

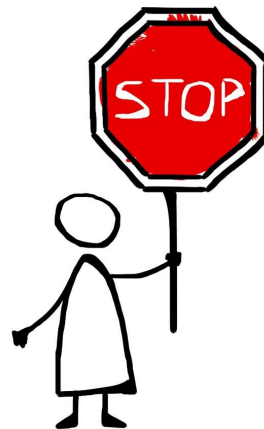
Inne przydatne funkcje: **isnumeric**, **isalpha**,

# PyStart #11 Stringi, czyli praca z tekstem

## Zadania dla nabrania wprawy

1. Poproś użytkownika o podanie imienia. W zależności od jego ostatniej litery wyświetl “Witam Panią” lub “Witam Pana”.
2. Napisz program generujące bardzo “silne” hasła. Zamień każdą literę “a” na @ oraz każdą literę s na \$.
3. Policz ile razy w zdaniu “Pies to najlepszy przyjaciel człowieka, lecz nie każdy pies o tym wie”. Wystąpiło słowo “pies”
4. Poszukaj innych funkcji niż upper i lower i spróbuj ze zdania:  
“12 kilogramów jabłek, 10 kilogramów gruszek, 20 kilogramów śliwek”  
policzyć ile łącznie razem ważą zakupione produktu. (spacje przed kilogramami muszą być :)

11.1





# DRUGA PRACA DOMOWA

## PĘTLA FOR

1. Napisz program, który odbierze od użytkownika liczbę całkowitą, a następnie wyświetli na ekranie wszystkie liczby parzyste od 2 do tej liczby. Wyświetl wszystkie te liczby oddzielone przecinkami.

2. Poniżej wyświetl sumę, liczbę, średnią oraz iloczyn tych liczb.

3. Odbierz od użytkownika liczbę, a następnie wyświetl tę liczbę podniesioną do potęgi od 2, 3, 4 oraz 5.

**Użyj pętli for!**

4. Napisz kod, który przejdzie po wszystkich znakach łańcucha znaków i wyświetli osobno każdy znak oraz odpowiednik z tablicy ASCII za pomocą funkcji `ord()`
5. Zapytaj użytkownika o liczbę, w wyniku wyświetl informację czy jest ona liczbą pierwszą.



# DRUGA PRACA DOMOWA

Stwórz tuple zawierającą liczby podzielne przez 6 z zakresu od 12 do 1024 i odpowiedz na poniższe pytania:

- Ile jest tych liczb?
- Wyświetl trzy pierwsze liczby
- Przedostatni element
- Co szósty element licząc od czwartego
- Co trzeci element licząc od końca
- Ile wynosi średnia 10 ostatnich wartości?



# DRUGA PRACA DOMOWA

Przygotuj tuplę przechowującą imię, nazwisko oraz wiek osoby. Następnie wyświetl te dane w postaci:

Imię: John

Nazwisko: Smith

Wiek: 40

---

Zadeklaruj tuplę w postaci  $a=((1,2,3), (4,5,6), (7,8,9))$ , a następnie przechodząc po nich za pomocą pętli for wyświetl:

$$1 + 2 + 3 = 6$$

$$4 + 5 + 6 = 15$$

$$7 + 8 + 9 = 24$$



# DRUGA PRACA DOMOWA

## STRINGI - łańcuchy znaków

1. Napisz program, który pobierze od użytkownika dowolny ciąg znaków i wypisze go na ekranie w odwrotnej kolejności.
2. Napisz program, który pobierze od użytkownika ciąg znaków i zwróci ten sam tekst bez spacji oraz znaków interpunkcyjnych.
3. Odbierz od użytkownika zdanie, a następnie wypisz je w taki sposób by poszczególne wyrazy były zapisane raz MAŁYMI raz WIELKIMI literami.
4. Poniżej wypisz małymi literami tylko te słowa z podanego zdania, które zaczynają się i kończą na tą samą literę.
5. Poniżej wyświetl wszystkie słowa ze zdania z usuniętymi samogłoskami.

