



akademia kodu

Python od podstaw

2020

Zasady

- ✓ Szkolenie dla osób początkujących.
- ✓ Pytania, komentarze merytoryczne.
- ✓ 10:00 do 18:00, może trochę krócej.
- ✓ Przerwy 11:30-11:50.
- ✓ 13:00-13:15.
- ✓ 14:00-15:00.

I te cele udało się osiągnąć.



Dla tych co będą do końca.

- ✓ Możliwość otrzymania certyfikatu ze szkolenia.
- ✓ Certyfikat będzie wysłany drogą mailową do 28 kwietnia.

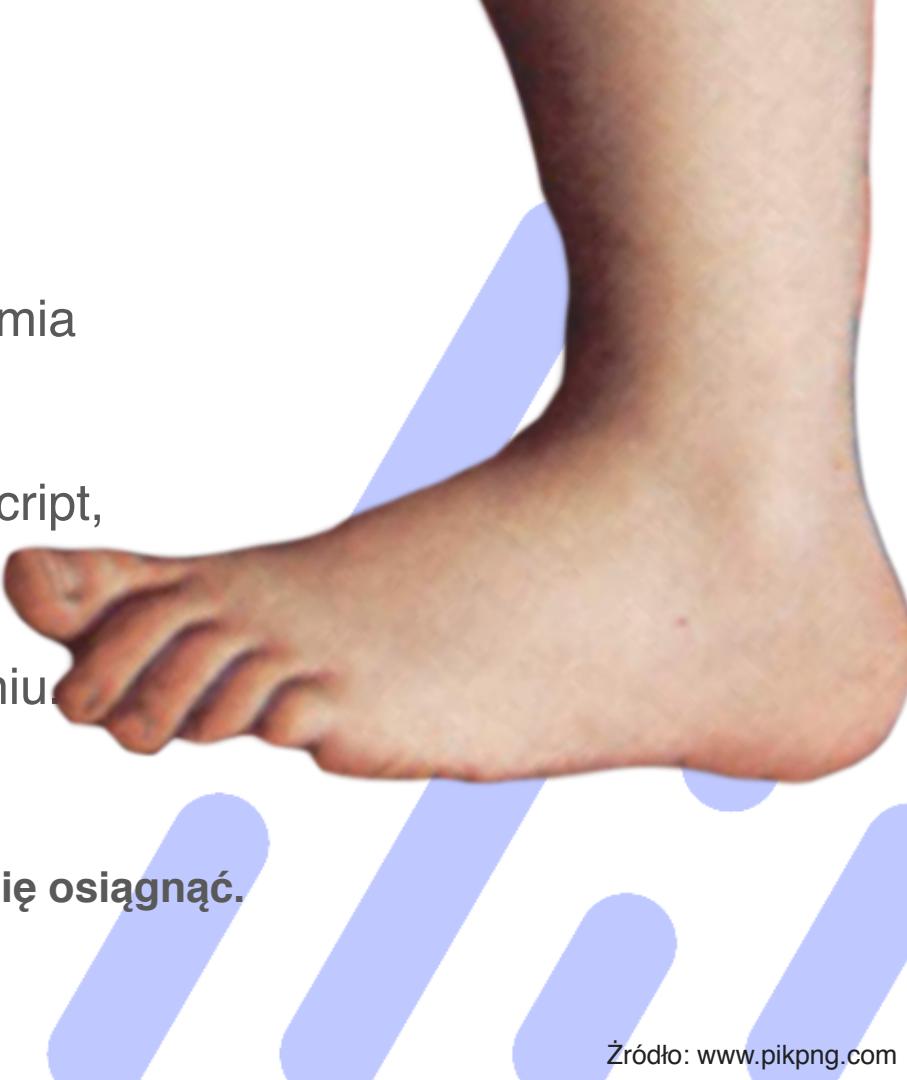
I te cele udało się osiągnąć.



Michał Makaruk

- ✓ Założyciel szkoły programowania Akademia Kodu.
- ✓ Programista Ruby on Rails, Java, Javascript, Python.
- ✓ 10 letnie doświadczenie w programowaniu.
- ✓ Aktywny programista.

I te cele udało się osiągnąć.



Plan

- ✓ Teoria.
- ✓ Ćwiczenia.
- ✓ Pierwsze programy.
- ✓ Instrukcje warunkowe.
- ✓ Pętle.
- ✓ Instalacja zewnętrznych bibliotek.
- ✓ Rysowanie wykresów, aplikacja okienkowa.
- ✓ Test.

I te cele udało się osiągnąć.



Czy warto się uczyć programować?



Nowe zawody będą pojawiać się w obszarze nowych technologii.

Już teraz znajomość programowania można wykorzystać też w innych zawodach, m.in. Tester automatyzujący, rekruter HR, menedżer IT, project manager IT, specjalista digital marketingu.

Według World Economic Forum aż 65% dzisiejszych uczniów szkoły podstawowej będzie pracowało w zawodzie, który jeszcze nie istnieje.

133 mln,

tyle miejsc pracy ma pojawić się w ciągu najbliższych dwóch lat w zawadach związanych z nowymi technologiami .

Szacunki WEF

Czy będzie praca dla programistów?

W 2018 r. amerykańskie Bureau of Labor Statistics (główny urząd statystyczny w USA) opublikowało, że w ciągu dekady popyt na programistów wzrośnie o **21 proc.**

Komisja Europejska też jeszcze nie zmieniła szacunków deficytu programistycznego dla Polski – zakłada on, że brakuje **50 tys. programistów.**

Autorzy unijnego raportu przewidywali, że w **Polsce**, Hiszpanii i Portugalii do 2021 r. powstanie najwięcej stanowisk IT.

A jak to wygląda obecnie?

Najmniejszy stopień zagrożenia zwolnieniami notujemy w branżach:



Z kolei najbardziej zagrożone redukcją zatrudnienia są branże:

łotnictwo (80 proc. pracodawców przewiduje konieczność zwolnień), **produkcja** (80 proc.), **handel i gastronomia** (59 proc.)

Zwolnienia dotknęły również branżę IT, ale w mniejszym stopniu niż pozostałe sektory.

W marcu 2020 r. IDC obniżyła prognozę wzrostu światowego rynku IT w tym roku z ponad 5,3 proc. na 1,3 proc. w pesymistycznym scenariuszu.

Co mówią przedstawiciele branży IT?

Mikołaj Gwóźdż, dyrektor finansowy gliwickiej firmy Future Processing, w rozmowie z Tomaszem Czoikiem, „Branża IT w dobie pandemii. "Co nam grozi? Że naszym klientom zabraknie pieniędzy„, Gazeta Wyborcza, 17.04.2020.

Czego zatem najbardziej obawiacie się obecnie w kontekście biznesowym?

- Największą obawą jest ciągłość przychodów naszych klientów. To może z kolei wpływać na problemy z zapłatą za powierzone nam zlecenia. Od paru klientów otrzymaliśmy już informacje, że muszą ograniczyć współpracę, a jeden z naszych partnerów był nawet zmuszony wycofać się z powodu cięć przeprowadzonych w jego firmie.

Na szczęście nawet w tym trudnym czasie cały czas jesteśmy w stanie pozyskiwać nowych klientów. Zauważamy jednak, że oczekują teraz od nas większej elastyczności, przerzucają też część ryzyka na nas, np. prosząc o krótszy termin wypowiedzenia danego projektu.

My - w przeciwieństwie np. do firm produkcyjnych - nie obawiamy się, czy będziemy mogli kontynuować pracę. Nowoczesne firmy z branży IT ze względu na specyfikę pracy mają pod tym względem znacznie łatwiej. Potrafimy sobie radzić zdalnie, nasi programiści mogą przecież pracować z domu.

Branża IT upatruje nawet pewnych szans w obecnej sytuacji. Ludzie we współpracy z nami widzą możliwości szybszego wychodzenia z trudnych momentów, wzrasta zaufanie do nowoczesnych firm. Wiele instytucji właśnie teraz dochodzi do wniosku, że e-komunikacja i e-administracja pozwalają na uniknięcie wielu zbędnych czynności i uproszczenie procesów. A jeszcze większy rozwój tych dziedzin umożliwią właśnie firmy technologiczne.

Wypowiedzi z artykułu Katarzyny Kucharczyk, „Niejednoznaczny wpływ pandemii na branżę IT” Parkiet, 20.03.2020r.

– Obecnie nie widzimy realnego zagrożenia dla ciągłości funkcjonowania spółki – informuje zarząd Asseco Poland. Podobny wydźwięk mają wieś napływające z Comarchu. – Część pracowników z polskich biur oraz z oddziałów zagranicznych, m.in. w Belgii, we Francji, w Hiszpanii i we Włoszech, pracuje w formie zdalnej – dodaje Marek Wiśniewski, rzecznik Comarchu.

– Nie widzimy jednoznacznych, istotnych sygnałów dotyczących wpływu pandemii na nasz biznes. Jego struktura w dużej części opiera się na długoterminowych kontraktach z klientami i dotyczy strategicznych obszarów państwa (administracja, energetyka, finanse państwa) – mówi z kolei Bogdan Zborowski, prezes Sygnity.

Już wkrótce okaże się, jak zawirowania z wirusem wpłyną na kwestię podziału ubiegłorocznego zysku w branży IT.

Szanse i zagrożenia

– Paradoksalnie właśnie lekcje „z wirusa” zdynamizują podejście do pracy zdalnej, wykorzystywania odpowiedniego oprogramowania oraz chmury – uważa Konrad Świński, prezes Transition Technologies. Wtóruję mu Roman Szwed, prezes Atende.

– Widzimy, że oferowane przez nas rozwiązania IT związane np. z zapewnieniem pracy zdalnej, wykorzystaniem chmury czy systemów wideokonferencyjnych, choć w części, dały wielu naszym klientom szansę na zachowanie ciągłości prowadzonych operacji. Dziś te narzędzia przechodzą poważny test bojowy i naszym zdaniem zdają go pozytywnie – mówi. Dodaje, że jeśli chodzi o wypłaty dywidend, to spółka zamierza kontynuować dotychczasową politykę, jeśli sytuacja na to pozwoli. Niedawno Atende opublikowało strategię na lata 2020–2021. – Mimo nietypowej sytuacji pozostaje ona aktualna. Pomaga nam fakt, że Atende nie jest zadłużone, a ponadto mamy znaczną rezerwę finansową w wyniku niedawnej sprzedaży spółki z grupy – mówi prezes.

Źródło:

<https://www.parkiet.com/Technologie/303209982-Niejednoznaczny-wplyw-pandemii-na-branze-IT.html>

O negatywnym wpływie pandemii na swój biznes poinformowało natomiast Wasko. Podało, że istnieje ryzyko opóźnienia przy dużym kontrakcie dla PKP-PLK .

Rynek IT w trakcie pandemii

No Fluffjobs ogłosiło, że choć w marcu 2020 r. kryzys dotknął też branżę IT, to wciąż prowadzą rekrutację firmy z sektorów:

- ✓ Firmy wytwarzające i utrzymujące narzędzia online (21 proc.), w tym narzędzia do komunikacji i analityczne.
- ✓ Software house'y i firmy fintech tworzące oraz utrzymujące różnego rodzaju oprogramowanie online, w tym oprogramowanie e-commerce (łącznie 43 proc.).
- ✓ Bankowość i finanse (8 proc.)
- ✓ konsulting (5 proc.).

Zarobki junior developerów

Średnia z median wynagrodzeń
oferowanych juniorom



Umowa cywilnoprawna jest proponowana 5 razy częściej juniorom niż midom i seniorom.

37% udział umów B2B w 2019 r.
↑ 7,27% w porównaniu z 2018 r.

Na stanowiskach juniorskich zarobki w 2019 r. wahały się od 2 tys. (staże) do ok. 8 tys. zł.

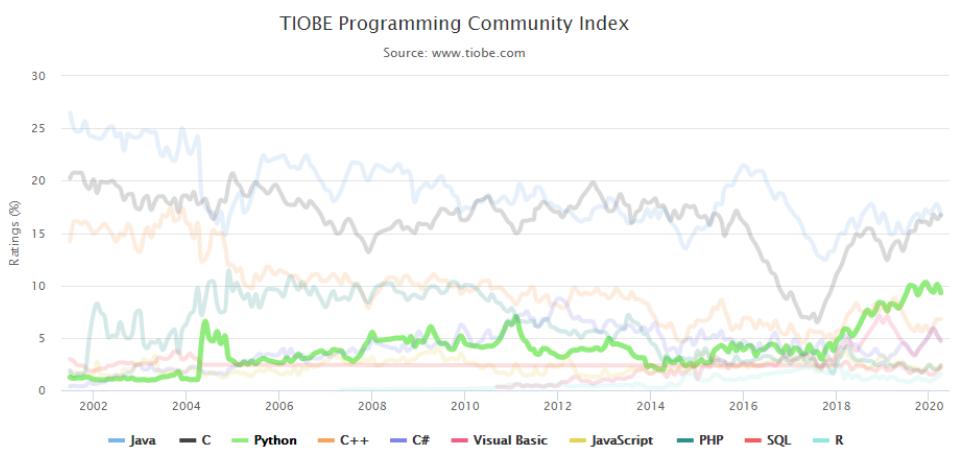
Jak podaje No Fluff jobs w **marcu 2020 r.**, czyli już po wybuchu pandemii juniorzy mogli liczyć o wynagrodzenie o **1 proc.** wyższe niż w styczniu.

Na razie wynagrodzenia nie są obniżane.



Raport „Rynek pracy IT w Polsce w 2019 r.”
No Fluff Jobs, 2020

A jaką pozycję ma Python?



| Apr 2020 | Apr 2019 | Change | Programming Language | Ratings | Change |
|----------|----------|--------|----------------------|---------|--------|
| 1 | 1 | | Java | 16.73% | +1.69% |
| 2 | 2 | | C | 16.72% | +2.64% |
| 3 | 4 | ▲ | Python | 9.31% | +1.15% |
| 4 | 3 | ▼ | C++ | 6.76% | -2.06% |
| 5 | 6 | ▲ | C# | 4.74% | +1.23% |
| 6 | 5 | ▼ | Visual Basic | 4.72% | -1.07% |
| 7 | 7 | | JavaScript | 2.38% | -0.12% |
| 8 | 9 | ▲ | PHP | 2.37% | +0.13% |
| 9 | 8 | ▼ | SQL | 2.17% | -0.10% |
| 10 | 16 | ▲ | R | 1.54% | +0.35% |

Według **TIOBE Programming Community Index** popularność Pythona od początku 2018 r. systematycznie rośnie.

Obecnie ten język programowania uplasował się na **trzecim miejscu**, wyprzedzając C++ (w stosunku do analogicznego okresu ubiegłego roku).

Python, czyli co?



To wysokopoziomowy opensourcowy język programowania stworzony przez **Guido van Rossum**. na początku lat dziewięćdziesiątych w Centrum Matematyki i Informatyki w Amsterdamie.

Opensourcowy, czyli darmowy (zarządzany przez fundację Python Software Fundation)

Wysokopoziomowy, czyli mający maksymalnie ułatwić zrozumienie kodu przez człowieka.

Skąd nazwa?

Po prostu, Guido von Rossum był wielkim fanem...
Latającego Cyrku Monty Pythona.

I dlatego właśnie
Python to Python.



A do tego jest:



Wieloplatformowy



działa na Macu, Windowsie i Linuxie.



Wielofunkcyjny



za jego pomocą można tworzyć aplikacje desktopowe i webowe, serwisy internetowe, czy gry.



Wydajny



generatory dostępne są jako wyrażenia i funkcje. Pozwalają na iteracje przetwarzanych danych po jednym elemencie, bez konieczności zajmującej dużo pamięci listy.

I jest prosty!

Widać to na przykładzie „Hello world”

W Javie:

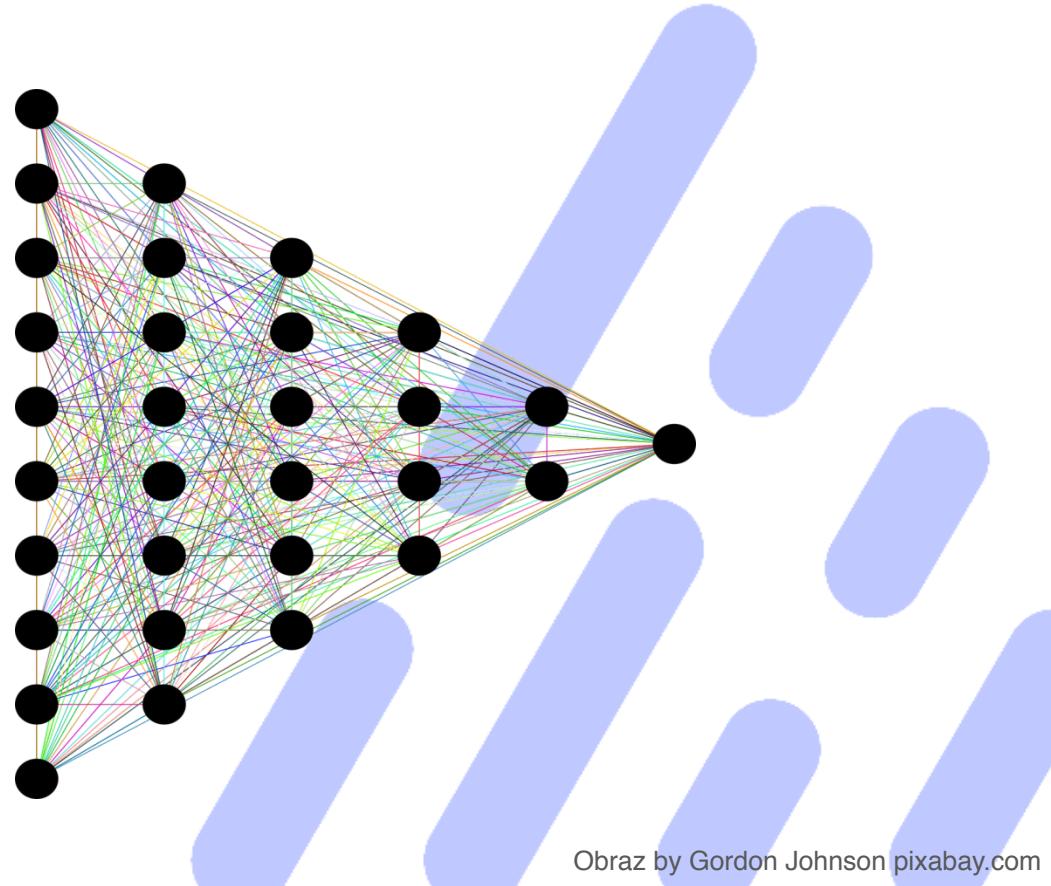
```
public class Main {  
    public static void main(String[] args) {  
        System.out.println("hello world");  
    }  
}
```

I w Pythonie:

```
print("hello world")
```

Gdzie stosuje się Pythona?

- ✓ Strony internetowe w Django
- ✓ Analiza danych
- ✓ Uczenie maszynowe
- ✓ Marketing
- ✓ Internet od Thinks
- ✓ Testowanie oprogramowania
- ✓ Arduino



To zaczynamy! Hello World, czyli witaj świectie

```
#!/usr/bin/env python
print("Witaj")
print("Jeszcze raz")
print('I po raz trzeci')
a = 'Witaj w swiecie
Pythona'
b = 'w 2020 roku'
print(a + b)
```



Zmienne

- ✓ Zmienna to inaczej wydzielone miejsce w pamięci komputera.
- ✓ W Pythonie w zmiennej nie trzeba deklarować typu danych.
Zmienne są uniwersalne.
- ✓ Przy jej tworzeniu wystarczy podać nazwę zmiennej i przypisać jej wartość:
- ✓ Każda zmienne jest obiektem.

```
Zmienna = 2
```

Nazwy zmiennych nie mogą zawierać:

polskich znaków

cyfry na pierwszym miejscu

spacji

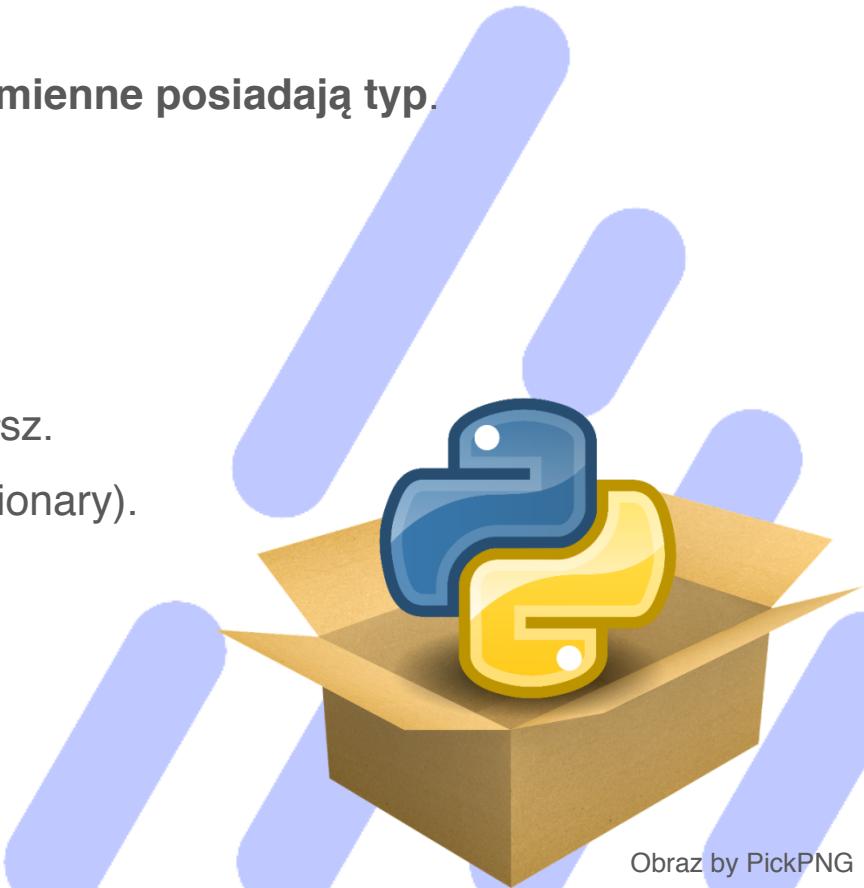
znaków specjalnych za wyjątkiem podkreślnikiem (_)

Typy danych

W Pythonie to wartości, a nie zmienne posiadają typ.

Najczęściej używane typy to:

- ✓ **Liczby** (Numbers) - integer albo float
- ✓ **String** – jedna litera lub ciąg znaków.
- ✓ **Typ logiczny** (Boolean) – tak albo nie, prawda/fałsz.
- ✓ **Kolekcja** – listy (List), krotki(Tuple), słowniki (Dictionary).
- ✓ **Funkcje**.
- ✓ **None**, czyli nic.



Liczby

```
#!/usr/bin/env python
a = 2
b = 5
print(a+b)
print(a*b)
print(a/b)
print(a%b)
```

Int

Liczby całkowite dowolnej wielkości

float

Liczby zmiennoprzecinkowe

String

- ✓ Typ tekstowy od jednej litery do sekwencji znaków.
- ✓ Można je zapisać w cudzysłowie pojedynczym " lub podwójnym "" – zawsze jednak zamknięcie musi być takie same jak otwarcie.

Niemutowalne

Stringi są niezmienne. Raz stworzonego tekstu nie można bezpośrednio modyfikować.

True

Bool

- ✓ To typ logiczny true/false w Pythonie istnieje dopiero od wersji 2.3
- ✓ Służy do sprawdzania warunków i zawsze wraca wartość prawda lub fałsz.

```
>>> 0 < 1  
True  
>>> 0 < 1  
False
```

- ✓ Używamy tego typu w instrukcjach warunkowych i pętlach.



==

Porównanie, czy dwie wartości są takie same.

!=

Sprawdzenie twierdzenie „nie równa się”, czyli 0 nie równa się 1 to prawda.

If, else i elif

if

- ✓ Po słowie If następuje warunek, którego rezultat jest prawdą lub fałszem.
- ✓ Jeżeli warunek jest prawdziwy, to wykona się polecenie.

else

- ✓ Jeżeli warunek if nie został spełniony, można użyć else do wykonania alternatywnego polecenia.

elif

- ✓ Jeżeli do sprawdzenia są więcej niż dwie możliwości używa się elif(if else)
- ✓ Wykona się, gdy warunek z if jest wyrażeniem fałszywym.

```
#!/usr/bin/env python
```

```
b = 2
```

```
a = 1
```

```
if a > b:
```

```
    print b
```

```
else:
```

```
    print a
```

```
c = 2
```

```
if a > b and a > c:
```

```
    print a
```

```
elif c == b:
```

```
    print „C I B ROWNE”
```

```
else:
```

```
    print b
```

Instrukcje warunkowe

- ✓ Instrukcje warunkowe sprawdzają, czy dane wyrażenie jest prawdziwe.
- ✓ Określają warunek, po spełnieniu którego wykona się pewien fragment kodu.



operatory logiczne

and oba warunki muszą być prawdziwe.

or jeden z dwóch warunków ma być spełniony.

operatory porównania

in sprawdza, czy zmienna jest zawarta w innym obiekcie.

is sprawdza, czy dwie wartości wskazują na ten sam obszar w pamięci.

not dodaje zaprzeczenie (zmienia wartość wyrażenia na przeciwną).

Pętle

- ✓ Pętle pozwalają daną część kodu wykorzystać kilkakrotnie.
- ✓ W Pythonie używamy dwa rodzaje pętli **for** i **while**
- ✓ Pętle można „kontrolować” za pomocą instrukcji **brake** i **continue**

brake

Porównanie, czy dwie wartości są takie same.

continue

Sprawdzenie twierdzenie „nie równa się”, czyli 0 nie równa się 1 to prawda.



Pętla for i while

for

Wykonuje czynność z góry określoną liczbę powtórzeń.

for

while

Porównanie, czy dwie wartości są takie same.

Sprawdzenie twierdzenia „nie równa się”, czyli 0 nie równa się 1 to prawda.

while

```
#!/usr/bin/env python
```

```
for i in range(10):  
    print(i)
```

```
print "Petla 2:"  
for i in range(3, 5):  
    print(i)
```

```
print "Petla 3:"  
for i in range(10, 100, 10):  
    print(i)
```

```
#!/usr/bin/env python
```

```
licznik = 10  
wartosc = 15
```

```
While licznik <= wartosc:  
    Licznik += 1  
    print „Jestem w while.”
```

operatory

- >> **+** dodawanie.
- >> **-** odejmowanie (odejmowanie wartości od lewej).
- >> ***** mnożenie (mnoży wartości).
- >> ****** znak potęgowania (lewą wartość podnosi do potęgi prawej wartości).
- >> **/** znak dzielenia (dzieli lewą wartość przez prawą).
- >> **//** dzielenie do całkowitej (dzieli lewą wartość przez prawą, wynikiem jest liczba całkowita).
- >> **%** modulo, czyli reszta z dzielenia.

Listy

- ✓ To złożone typy danych.
- ✓ Przechowują różne elementy znajdujące się w nawiasie kwadratowym []
- ✓ Elementy na liście mogą mieć różne typu.
- ✓ Pozwalają na modyfikację elementów.

numerowanie

Listy są indeksowane od 0

Pierwszy element -> lista[0]

wymiana elementów

Elementy można prosto podmienić:

Lista[1] = 7

indeksy

Listy działają w oparciu o indeksy



Tuple (krotki)

- ✓ Tuple to uporządkowane kolekcje dowolnych obiektów.
- ✓ Tworzymy za pomocą nawiasów okrągłych
()
- ✓ Nie można zmieniać zawartych w niej elementów i mają ustalony rozmiar.

odwołanie do pierwszego elementu

Krotki tak, jak listy są indeksowane od 0

Pierwszy element -> tup[0]

łączenie

Tuple można jednak ze sobą łączyć.

tup_new = tup2 + tup1

powielanie

Można je też powielać za pomocą operatora *



Słowniki

- ✓ Podobnie jak listy są wbudowanym typem Pythona i można je modyfikować.
- ✓ Pracuję nie w oparciu o parę: **klucz-wartość**.
- ✓ Wartością może być dowolny obiekt pythonowy.
- ✓ Kluczem może być dowolny niezmienny typ np. liczby lub Stringi.

tworzenie

Wkładając parę klucz-wartość w nawias klamrowy:

```
Dict = {klucz1:wartosc1, klucz2:wartosc2}
```

unikatowość

Klucze w słowniku muszą być niepowtarzalne.



Operacje na słownikach

Przykładowy słownik: `Dict = { "Adam": "123456789", "Ania": "234567890"}`

Dostęp do elementów – przywołujemy nazwę jego klucza w nawiasach kwadratowych
`x = Dict["Adam"]`

Zmianianie wartości – przez odwołanie się do klucza określonego elementu
`Dict["Ania"] = 890123456`

Dodawanie elementu – stworzenie nowego klucza i dodanie do niego wartości.
`Dict["Bartek"] = 567123456`

Usuwanie elementu – są trzy sposoby:
metoda `pop()` – usuwa element o określonej nazwie klucza `Dict.pop("Adam")`
metoda `popitem()` – usuwa ostatni wstawione element `Dict.popitem()`
słowo kluczowe `del` – usuwa element o określonej nazwie klucza `del dict["Adam"]`

A teraz do dzieła!

