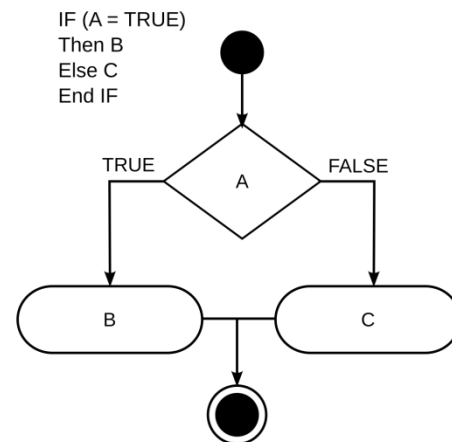




Instrukcja IF w Pythonie

M@я3k Pцđ€£kØ

Programowanie w Pythonie

Spis treści

- Instrukcja alternatywy
- Alternatywa w Pythonie
- Operatory porównania
- Zagnieżdżony warunek
- Instrukcja **elif**

Instrukcja alternatywy

- Instrukcja warunkowa to element programu pozwalający na podjęcie decyzji i wykonanie różnych fragmentów kodu w zależności od spełnienia lub niespełnienia określonych warunków logicznych (**prawda/fałsz**).
- Zazwyczaj używa się kluczowego słowa IF (lub IF-ELSE) do wyboru wykonywanego kodu.

Alternatywa w Pythonie

- Instrukcja alternatyw w języku Python ma postać:

```
if (warunek) :
```

```
    Instrukcje gdy warunek jest  
    spełniony
```

```
else:
```

```
    Instrukcje gdy warunek nie jest  
    spełniony
```

- Na końcu liniiki `if` należy napisać : (dwukropek)
- Wykonywane instrukcje zaczynają się od wcięcia
- `else` umożliwia wykonanie instrukcji, które mają być realizowane w wypadku gdy nie są spełnione warunki instrukcji IF.

Przykład alternatywy

```
'''  
alternatywa  
'''  
x = int (input ('Podaj stan konta '))  
  
if x > 0:  
    print ('konto jest dodatnie')  
    print ("możesz iść na zakupy")  
else:  
    print ('masz debet na koncie')  
    y = int (input ("Ile chcesz wpłacić?"))  
    x +=y  
    print ('masz teraz na koncie', x)
```

Przykład wykonania alternatywy

```
'''  
alternatywa  
'''  
x = int (input ('Podaj stan konta '))  
  
if x > 0:  
    print ('konto jest dodatnie')  
    print ("możesz iść na zakupy")  
else:  
    print ('masz debet na koncie')  
    y = int (input ("Ile chcesz wpłacić?"))  
    x +=y  
    print ('masz teraz na koncie', x)
```

```
Podaj stan konta 450  
konto jest dodatnie  
możesz iść na zakupy
```

```
Podaj stan konta -450  
masz debet na koncie  
Ile chcesz wpłacić?100  
masz teraz na koncie -350
```

Instrukcja **if** bez **else**

- Możliwe jest użycie instrukcji wyboru bez części wykonywanej, gdy warunek nie jest spełniony.

```
if wiek >= 18:  
    print ('jesteś pełnoletni')
```

Operatory porównania

Operator	Opis
==	czy dwie wartości są równe
!=	czy dwie wartości są różne
>	czy pierwsza wartość jest większa od drugiej
<	sprawdza, czy pierwsza wartość jest mniejsza od drugiej
>=	czy pierwsza wartość jest większa lub równa drugiej
<=	czy pierwsza wartość jest mniejsza lub równa drugiej

- Operator **is** służy do sprawdzania, czy dwie zmienne wskazują na dokładnie ten sam obiekt w pamięci
- Nie sprawdza samej zawartości jak operator **==**.

Przykład operatorów porównania

```
x = 10
y = 10

if x > y:
    print ('x większe od y')

if x >= y:
    print ('x nie jest mniejsze od y')

if x != y:
    print ('x i y nie są sobie równe')

if x == y:
    print ('x równe y')

if x <= y:
    print ('x nie jest większe od y')

if x < y:
    print ('x jest mniejsze od y')

if x is y:
    print ('x równe y')
```

Zagnieżdżony warunek

- Może zajść sytuacja, gdy trzeba uwzględnić kilka warunków realizowanych w zależności od tego jak zostały spełnione inne.
- Można to zrealizować w postaci zagnieżdżania kolejnych warunków.

```
if wiek >= 18:
    print ('jesteś pełnoletni')
    if wiek >= 65:
        print ('jesteś emerytem')
    else:
        print ('jesteś dorosły')
else:
    if wiek > 11:
        print ('jesteś młodzieżą')
    else:
        print ('jesteś dzieckiem')
```

Instrukcja elif

- Możliwe jest użycie instrukcji **elif** = **else if**, wykonywanej gdy wcześniejszy warunek nie został spełniony.
- Dzięki niej możliwe jest sprawdzenie większej liczby warunków.

```
if wiek >= 18:  
    print ('jesteś pełnoletni')
```

```
if wiek >= 65:  
    print ('jesteś emerytem')  
elif wiek >=18:  
    print ('jesteś dorosły')  
elif wiek >=11:  
    print ('jesteś młodzieżą')  
else:  
    print ('jesteś dzieckiem')
```

Ćwiczenie 1

1. Napisz program wczytujący liczbę i podający czy jest parzysta czy nieparzysta.
2. Masz trzy liczby **a**, **b**, **c**.
 - a) Jeśli **a** jest większe od **b**, zwiększ **a** o wartość **c**.
 - b) W przeciwnym razie pomnóż **b** przez **c**.
3. Masz dwie liczby **a** i **b**. Zwiększ **a** o 10%, zmniejsz **b** o 10%. Sprawdź, która liczba jest większa po zmianach.
4. Użytkownik podaje dochód. Oblicz podatek 19% dla dochodów do 120 000 zł, a 32% dla wyższych.
5. Pobierz dwie liczby **a** i **b**. Oblicz:
 - kwadrat sumy: $(a + b)^2$
 - sumę kwadratów: $a^2 + b^2$

Porównaj wyniki i wypisz, który jest większy.

Operatory logiczne

- Pozwalają utworzyć złożone warunki logiczne złożone z wielu elementów.

Operator	Opis
and	Oba warunki muszą być spełnione (True and True)
or	Wystarczy, że jeden warunek jest spełniony (True or True)
not	Odwraca wartość logiczną wyrażenia (True → False, False → True)

- Współpracują ze zmiennymi logicznymi (typ bool)
= **True** lub **False**.

Operatory logiczne – przykład użycia

```
if wiek>=18 and wiek<65:  
    print ('jesteś dorosły')
```

```
if wiek<18 or wiek>=65:  
    print ('Chyba nie pracujesz')
```

```
if not wiek>=18:  
    print ('Jesteś nieletni')
```

```
if czy_podzielono == True:  
    print ('Dokonano dzielenia')
```

```
if czy_otwarto == False:  
    print ('Drzwi są zamknięte')
```

Ćwiczenie 2

1. Napisz program wczytujący liczbę i podający:
 - a) czy jest parzysta i ujemna.
 - b) czy jest nieparzysta i dodatnia.
2. Uczeń zdał, jeśli ma ocenę powyżej 50 punktów i obecność powyżej 75%. Wczytaj dane (ocena i frekwencja) i sprawdź wynik.
3. Napisz program sprawdzający czy użytkownik może dostać zniżkę.
 - Warunki: jest studentem lub ma powyżej 65 lat i ma kartę lojalnościową.
4. Wczytaj 3 liczby **a**, **b**, **c** i sprawdź czy da z nich zbudować trójkąt.
 - Możliwe, gdy każda para boków jest większa od trzeciego.
5. Pobierz dwie liczby **a** i **b**. Oblicz:

Powtórzenie

1. Co robi instrukcja alternatywy?
2. Jak jest realizowana w języku Python?
3. Jak zrealizować więcej niż 2 warunków?
4. Omów operatory porównania.
5. Omów operatory logiczne.