Podejmowanie decyzji

Porównania

Żebyśmy mogli podjąć w programie jakąś decyzję musimy najpierw sformułować pytanie na które da się jednoznacznie odpowiedzieć tak/nie (prawda / fałsz) .

Najprostszym podejściem do tego jest wykonanie jakiegoś porównania np. czy 12 jest większe od 6.

Podstawowe porównania

Operator	Pytanie	Przykład
`==`	równe?	`x == y`
`!=`	nierówne?	`x != y`
`>`	większe?	`x > y`
`>=`	większe lub równe?	`x >= y`
`<`	mniejsze?	`x < y`
`<=`	mniejsze lub równe?	`x <= y`

W Pythonie jest podwójny znak równości (`==`) służy do porównania czy dane wartości są równe, podczas gdy pojedynczy `=` służy do przypisania wartości.

Porównania tekstów

O dziwo w Pythonie porównanie tekstów są dozwolone np.

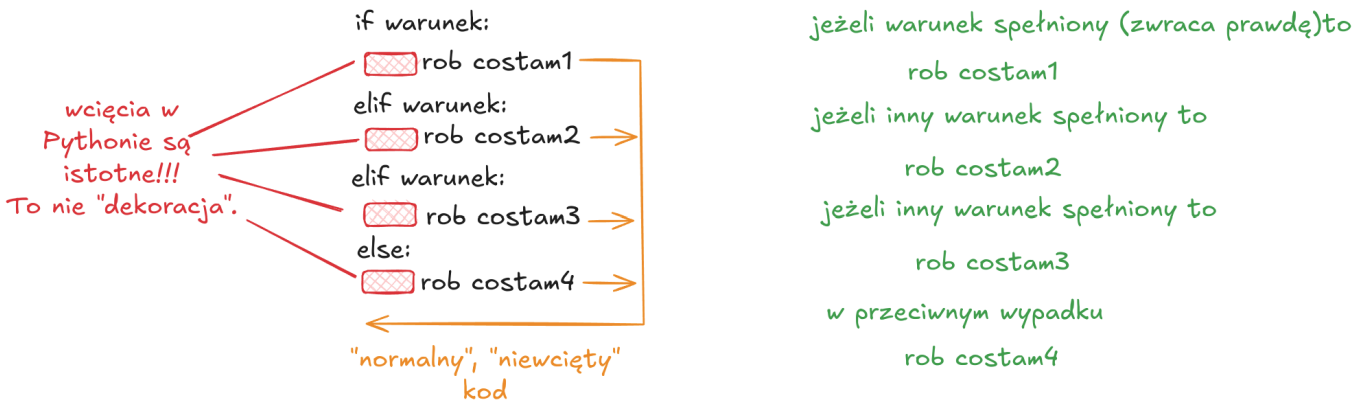
```
"pies" > "kot"
```

Takie porównanie zwróci nam wartość `True`.

Jak Python doszedł do takiego wniosku? Każdy znak jest jakąś liczbą w ściśle określonej tablicy znaków (w tym wypadku unicode). Python porównuje numer danej litery dla każdego znaku w tekście i na tej podstawie podejmuje decyzję.

Warunki i bloki kodu

Mając przygotowane pytanie na które da się odpowiedzieć tak/nie, prawda/fałsz możemy wykonać jakąś akcję wg schematu:




```
A = 12
B = 6

jeżeli A > B:
    wykonaj coś1
a jeżeli A == B:
    wykonaj coś2
w przeciwnym wypadku: (jeśli żaden inny warunek nie został spełniony)
    wykonaj coś3
```

Tłumacząc to na Pythona wygląda to tak:

```
# to pseudokod - nie zadziała bo nie poinformowaliśmy Pythona co oznacza zrob_cos :D
A = 12
B = 6

if A > B:
    zrob_cos(1)
elif A==B:
    zrob_cos(2)
else:
    zrob_cos(3)
```

Należy pamiętać o **wcięciach(!**) - decydują one o tym co Python wykona po podjęciu decyzji np. w powyższym przykładzie jeśli A jest większe od B to uruchomi się jedynie funkcja (fikcyjna) `zrob_cos(1)`.

np.

```
wiek_uzytkownika = 12


if wiek_uzytkownika < 18 :
    print("Jesteś niepełnoletni/-a")
elif wiek_uzytkownika > 65 :
    print("Hmmm... Senior?")
else:
    print("Pełnoletni.")
```

Czemu warto korzysta? z `elif`

Teoretycznie można by było w powyższym przykładzie użyć tylko i wyłącznie `if` - użytkownik programu nie zauważyłby raczej różnicy.

Dla komputera jednak różnica jest spora - używając tylko `if` wykonałby każde z porównań niezależnie od wyniku poprzedniego.

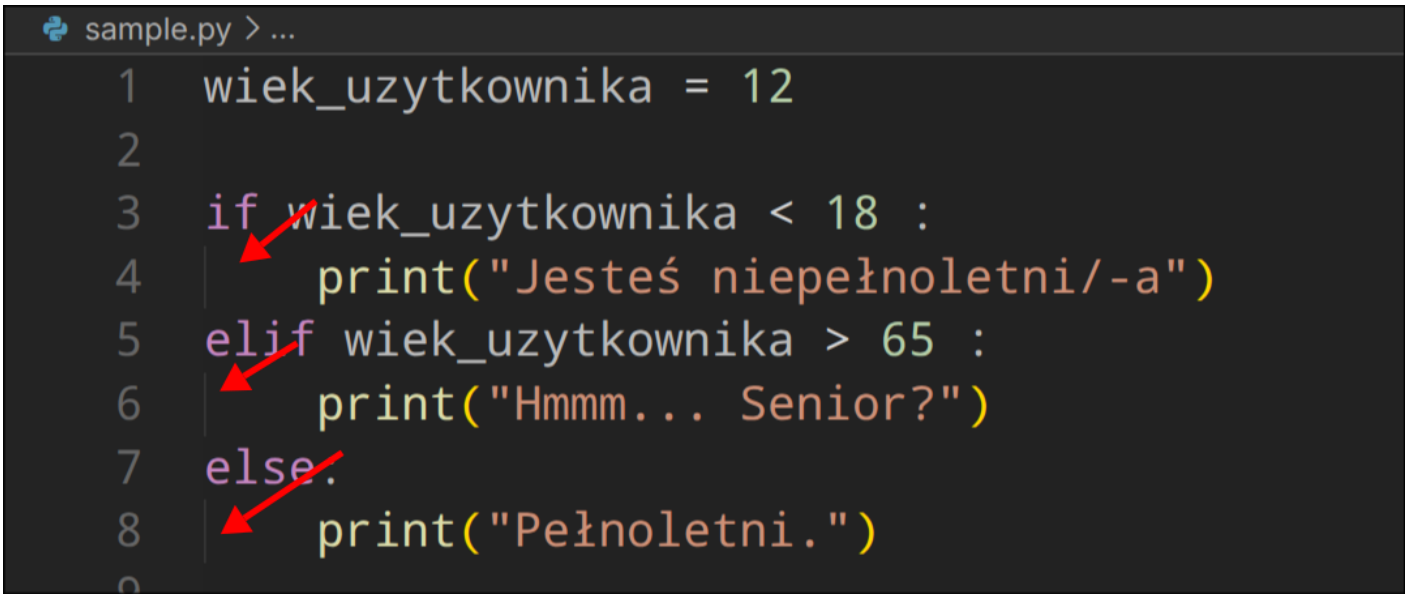
Jeśli korzystamy z `elif` / `else` porównania wykonują się do momentu uzyskania w którymś odpowiedzi Prawda (`True`) - pozostałe wtedy są pomijane. Dzięki temu minimalizujemy ilość niepotrzebnie wykonywanych operacji.

Korzystając z `if` bez `elif` należy też uważać na ew. dodanie sekcji `else` - w takiej sytuacji `else` będzie się odnosic wyłącznie(!) do ostatniego warunku (ostatniego `if`-a).

Stąd też jeśli wykonujemy jakieś porównania, które są z sobą powiązane (zależą od siebie) należy korzystać z `elif` i `else`.

Wci?cia w edytorach kodu

Edytoru kodu starają się pomóc użytkownikom z wcięciami - w oczywistych miejscach od razu je dla nas robią. Oprócz tego w większości z nich są one oznaczana pionową linią:



Wersja #8
Utworzono 2024-08-25 16:17:23 UTC przez Przemek Jeske
Zaktualizowano 2026-04-28 19:25:34 UTC przez Przemek Jeske