

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
<	mniej?	x < y
<=	mniej lub równe?	x <= y

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

Porównania tekstów

O dziwo w Pythonie porównania tekstów są dozwolone np.

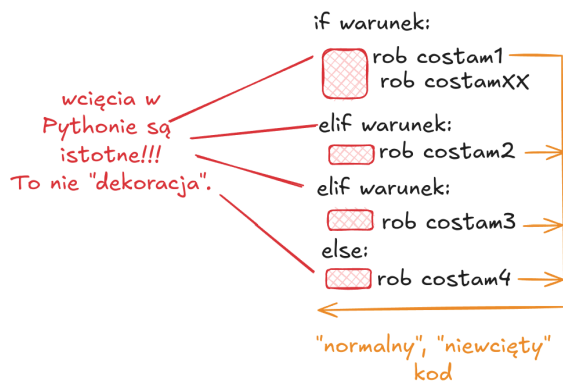
```
"pies" > "kot"
```

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:



jeżeli warunek spełniony (zwraca prawdę) to
 rob costam1
jeżeli inny warunek spełniony to
 rob costam2
jeżeli inny warunek spełniony to
 rob costam3
w przeciwnym wypadku
 rob costam4

Zapiszmy przykład w "pseudokodzie":

A = 12

B = 6

jeżeli A > B:

 wykonaj coś1

a jeżeli A == B:

 wykonaj coś2

w przeciwnym wypadku:

 wykonaj coś3

Tłumacząc to na Pythona wygląda to tak:

```
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)`.

Pamiętajcie też, że `else` ("w przeciwnym wypadku") wykona się tylko wtedy jeśli żaden z powyższych warunków nie zwróci prawdy.

Czemu warto korzystać 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` porównania wykonują się do momentu uzyskania w którymś odpowiedzi Prawda (`True`) - pozostałe wtedy są pomijane. Dzięki temu minimalizujemy ilość niepotrzebnie wykonywanych operacji.

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

Nie skorzystanie z `elif` może mieć też dość bolesne konsekwencje jeśli dodamy na końcu `else` - gdyż w takim wypadku `else` będzie dotyczył JEDYNE ostatniej instrukcji z `if`-em, nie całego bloku decyzyjnego.

Odnosiniki

„3.12.5 Documentation”. Dostęp 18 sierpień 2024. <https://docs.python.org/3/>.