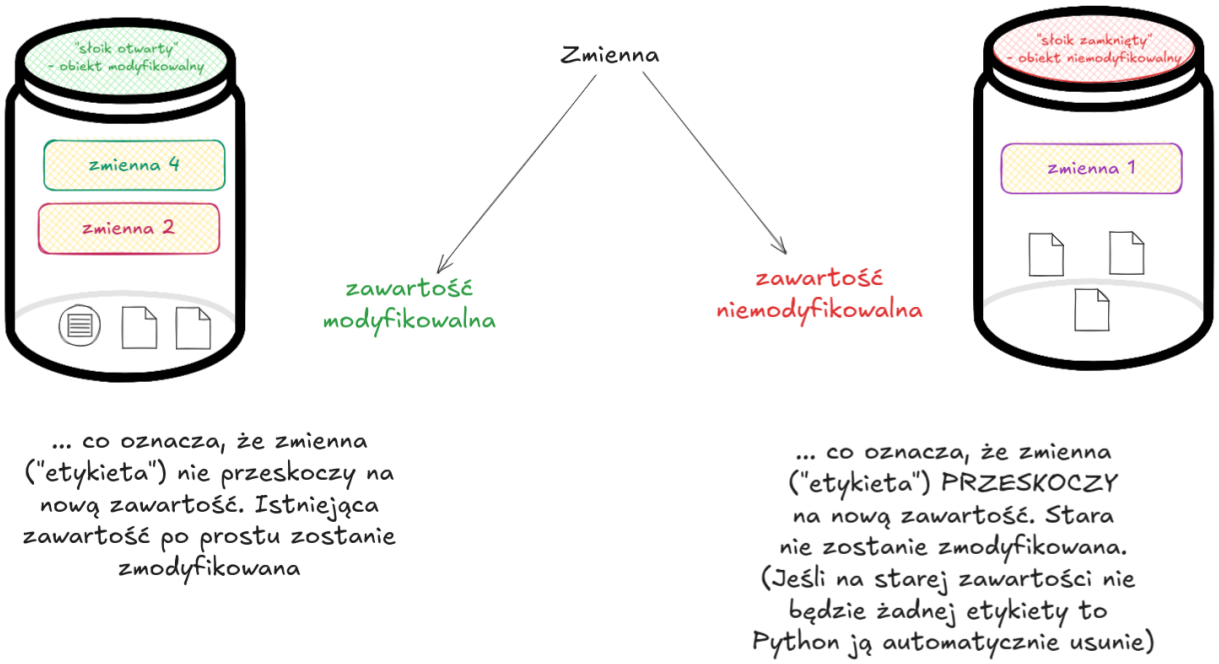
Modyfikacja zmiennej



Niektóre zmienne przekazywane są po wartości - co oznacza, że wartość jednej zmiennej jest KOPIOWANA do drugiej. Dotyczy to wszystkich typów, które są niemodyfikowalne - czyli np. liczb, ciągu znaków (teksty - string), wartości logicznych, krotek.

np.

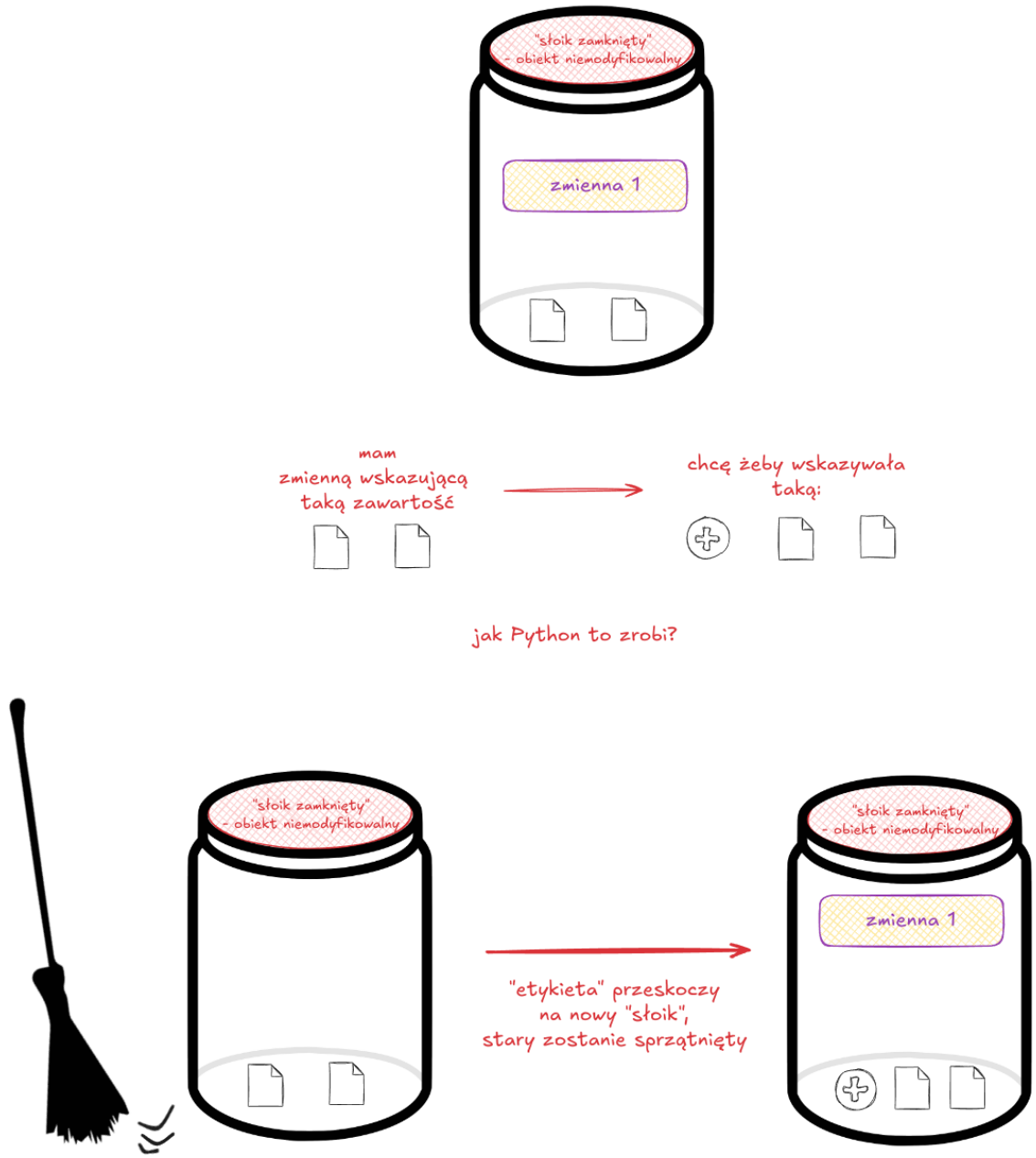
```
a = 4       # liczba jest "niemodyfikowalna" sama z siebie
b = a       # do zmiennej b jest kopiowana 4 przechowywana pod a
b = b + 2   # do wartości przechowywanej pod b dodawane jest 2
print(a)
print(b)
# wynikiem jest:
# 4
# 6
# wynika to z tego, że b miało wartość SKOPIOWANĄ z a
```

W Pythonie istnieją też typy, które podlegają modyfikacji - np. listy czy słowniki. W ich wypadku to przekazanie nie robi kopii - przekazywana jest tzw. referencja do obiektu (okropny anglicyzm). Czyli zamiast kopiować wartość przekazywany jest wskaźnik informujący gdzie w pamięci komputera przechowywana jest interesująca nas wartość.

```
a = [4,2]       # liczba jest "niemodyfikowalna" sama z siebie
b = a       # zmienna b przechowuje ten sam wskaźnik do obiektu co a
b.pop()   # z listy na którą wskazuje b usuwana jest ostatnia wartość
print(a)
print(b)
# wynikiem jest:
# [4]
# [4]
# wynika to z tego, że b i a wskazują na tę samą listę (nie jest robiona kopia)
```

Skopiowanie warto?ci

W przypadku gdy mamy do czynienia z obiektem **niemodyfikowalnym** (np. krotką) jedyne co Python może zrobić to "przekleić" naszą etykietę (zmienną) na nowy obiekt. Jeśli ten pierwotny nie będzie miał na sobie innej etykietki (nie wskazuje na niego inna zmienna) to zostanie on automatycznie usunięty (przez tzw. garbage collector).



Zmiana zawarto?ci

W przypadku gdy obiekt jest **modyfikowalny** (np. lista) i dołożymy do niej kolejny element zmienna nie będzie musiała przeskoczyć na nowy obiekt z nową zawartością - po prostu zostanie dodany nowy element (zmieni się zawartość obiektu, etykieta będzie wskazywać ten sam obiekt).

... a czemu to istotne?

Powody są co najmniej dwa:

1. Należy pamiętać, że gdy zmieniamy jakiś obiekt modyfikowalny na który wskazuje kilka zmiennych to zmienimy wartości dla każdej z nich.
2. Tworzenie obiektów (rezerwowanie miejsca w pamięci dla nich) to proces względnie czasochłonny - stąd też proces kopiowania całego obiektu i dodawania do niego jakiegoś dodatkowego elementu (czyli wariant z obiektem niemodyfikowalnym) może spowolnić działanie tworzonej aplikacji.

-> stąd też tak istotne jest świadome dobieranie struktur danych :) .

Co jest modyfikowalne, a co nie?

typ	modyfikowalny	niemodyfikowalny
liczba całk.		tak
liczba zmiennop.		tak
wart.log		tak
ciąg znaków		tak
krotka		tak
lista	tak	
słownik	tak	
zbiór	tak	

Wersja #4
Utworzono 2024-10-02 13:49:38 UTC przez Przemek Jeske
Zaktualizowano 2026-04-28 20:31:32 UTC przez Przemek Jeske