

Stałe w Pythonie

M@я3k Pүđ€£kØ

Programowanie w Pythonie

Spis treści

- Stałe
- Zalety stosowania stałych
- Stałe w Pythonie
- Nazwy stałych
- Przykładowe stałe
 - moduł math
 - moduł string

Stała

- Stała to obiekt programowania, którego wartość nie może być zmieniona w czasie pracy programu.

PI = 3.14159

EULER = 2.718281828459

Zalety stosowania stałych

Poprawiona czytelność	Opisowa nazwa reprezentująca daną wartość w całym programie jest zawsze bardziej czytelna i jednoznaczna niż sama podstawowa wartość. Łatwiej odczytać i zrozumieć nazwę stałej PRICE niż samą wartość ceny.
Jasna komunikacja intencji	Zazwyczaj 3.14 odnosi się do stałej Pi. Jednak użycie nazwy PI będzie bardziej zrozumiałe niż użycie samej wartości.
Lepsza łatwość utrzymania	Ta sama identyfikuje tą samą wartość w całym kodzie. Przy aktualizacji wartości stałej, wystarczy zmienić ją w jednym miejscu: definicji stałej.
Niższe ryzyko błędów	Stała reprezentująca daną wartość w całym programie jest mniej podatna na błędy niż kilka jawnych wystąpień tej wartości. Przy zmianie precyzji w obliczeniach, zastąpienie wartości może być podatne na błędy. Bezpieczniej utworzyć różne stałe dla różnych poziomów precyzji i zmienić kod w jednym miejscu.
Zmniejszone potrzeby debugowania	Stałe pozostaną niezmiennione przez cały okres istnienia programu. Nie powinny powodować błędów. Funkcja kluczowa w dużych projektach z udziałem wielu programistów.
Bezpieczne dla wątków przechowanie danych	Do stałych można jedynie uzyskać dostęp, a nie je zapisać. Obiekty bezpieczne dla wątków - kilka wątków może jednocześnie korzystać ze stałej bez ryzyka uszkodzenia lub utraty danych bazowych.

Stałe w Pythonie

- Język Python nie ma zdefiniowanych stałych.
- Przyjęto konwencję, że stałe są zapisywane dużymi literami.

```
PI = 3.14
```

```
IMIE = "JANEK"
```

- Jest to tylko informacja dla programistów, że dana wartość ma być traktowana jako stała.
 - Używanie wielkich liter to tylko umowa i nie blokuje przypisywania nowych wartości stałym.
 - Programista powinien zachować ostrożność i nigdy nie pisać kodu, który zmienia wartości stałych.

Użycie stałych w Pythonie

```
pi = 3.14  
print (pi)  
  
pi = 6.28  
print (pi)  
  
PI = 3.14159  
print (PI)  
  
PI = 2.71  
print (PI)
```

```
3.14  
6.28  
  
3.14159  
2.71
```

Nazwy stałych

- Stałe w Pythonie to po prostu zmienne i podlegają tym zasadom nazewnictwa.
 - Składają się tylko z wielkich liter (A– Z)
 - Są dowolnej długości
 - Mogą mieć cyfrę (0– 9) - nie jako pierwszy znak
 - Do oddzielenia słów można stosować znak podkreślenia (_)
- Zalecaną praktyką jest definiowanie stałych na początku każdego programu, tuż po poleceniach `import`. W ten sposób programista zaznaczy przeznaczenie stałych i oczekiwane traktowanie.

Przykładowe stałe – moduł **math**

```
import math
```

```
math.e          # Euler's number (e)
print (math.e)  #2.718281828459045
math.pi         # Pi (π)
print (math.pi) #3.141592653589793
math.inf        # Infinite (∞)
print (math.inf) # inf
math.nan        # Not a number (NaN)
print (math.nan) # nan
math.tau        # Tau (τ)
print (math.tau) # 6.283185307179586
```

```
import math

print ("math.e", math.e)      math.e 2.718281828459045
print ("math.pi", math.pi)   math.pi 3.141592653589793
print ("math.inf", math.inf)  math.inf inf
print ("math.nan", math.nan)  math.nan nan
print ("math.tau", math.tau)  math.tau 6.283185307179586
```

Przykładowe stałe – moduł **string**

Nazwa	Wartość
ascii_lowercase	abcdefghijklmnopqrstuvwxyz
ascii_uppercase	ABCDEFGHIJKLMNOPQRSTUVWXYZ
ascii_letters	ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
digits	0123456789
hexdigits	0123456789abcdefABCDEF
octdigits	01234567
punctuation	!"#\$%&'()*+,-./:;<=>?@[\\]^_`{ }~
whitespace	Połączenie znaku spacji, tabulatora poziomego i pionowego, nowego wiersza, powrotu karetki i nowego formularza
printable	Połączenie digits, ascii_letters, punctuation i whitespace

```
import string
print (string.ascii_lowercase[12])      #m
print (string.punctuation[5])          #&
print (string.hexdigits[15])            #f
print (string.digits[7])                #7
```

Kasowanie stałej

- Do skasowania stałej używa się instrukcji „del”.
- Jeśli po skasowaniu zmiennej będziemy chcieli wyświetlić jej wartość, pojawi się błąd.

```
PI= 3.14159
```

```
print (PI)
```

```
del PI
```

```
print (PI)
```

```
PI= 3.14159
print(PI)

del PI
print(PI)
```

```
3.14159
Traceback (most recent call last):
  File "/home/main.py", line 5, in <module>
    print(PI)
    ^^
NameError: name 'PI' is not defined
```

Ćwiczenie

1. Napisz wzór na pole koła. Do przedstawienia π użyj stałej.
 - a) Możesz użyć jej z modułu **math**.
2. Napisz program, który wczytuje wiek i podaje w którym roku się urodził. Aktualny rok zapisz jako stałą.
3. Stwórz aplikację wczytującą prędkość w m/s i przeliczającą ją na km/h. Przelicznik przedstaw jako stałą.
4. Napisz program, przeliczający wartość kwoty w złotych na dolary. Kurs dolara poszukaj w aktualnych danych NBP. Kurs walutowy ustaw jako stałą.

Powtórzenie

1. Co to jest stała?
2. Jak tworzy się stałą w Pythonie
3. Jakie reguły musi spełnić nazwa stałej w Pythonie?
4. Jak skasować stałą?