

Pracujemy w sesji interaktywnej Pythona.

1. Użyj sesji interaktywnej Pythona jako kalkulatora. Czy Python to dobry kalkulator?
2. Oblicz  $0.1 + 0.1$ ,  $0.1 + 0.1 + 0.1$ ,  $3.1415 * 5$ ,  $2/3000000$ .
3. Wprowadź zmienne  $a = 6$ ,  $b = 8$ ,  $c = 0.3$ ,  $d = -2$  (jeśli chcesz to wprowadź również inne zmienne), uwaga możesz to zrobić w osobnych liniach lub w jednej rozdzielając poszczególne przypisania znakiem `;`. Wykonaj różne działania na tych zmiennych ( $+$ ,  $*$ ,  $/$ ,  $//$ ,  $\%$ ,  $**$ ), co to za działania? (Warto sprawdzić  $5//3$ ,  $-5//3$ .)
4. Wpisz i zinterpretuj `0xa`, `0xB + 0x1`, `0xF * 0x10`, `0x123a4`.
5. Wpisz i zinterpretuj `0b10101`, `0b11 + 0b1010`, `0b100000000`.
6. Wbudowane funkcje `hex(l.całkowita)`, `oct(l.całkowita)`, `bin(l.całkowita)` przekształcają podaną liczbę całkowitą w jej reprezentację w odpowiednim systemie. Wyświetl liczbę 10237 w systemie dwójkowym, ósemkowym i szesnastkowym.
7. Wpisz `bin(13) + bin(8)` oraz `bin(13 + 8)`. Czy są to polecenia równoważne? Które z nich ma sens?
8. Wpisz `1j * 1j`. Zinterpretuj wynik.
9. Liczbę zespoloną zapisujemy jako  $a + bj$ , gdzie  $a$  i  $b$  są liczbami rzeczywistymi. Wprowadź dwie zmienne  $x = 1 - 2j$ ,  $y = -3 + 1j$  wykonaj działania  $+$ ,  $*$ ,  $/$ ,  $//$ ,  $\%$ .
10. Wpisz i zinterpretuj `2 < 10`, `11 > -5`, `-100 > 2`, `2 < 2`, `4 >= 4`, `3 < 6 < 0`, `2 * *10 < 3 * *7`, `0x1a > 0b10111`, `0xa4f5 == 0b1010010011110101`. Poproś i innymi wartościami. Co się stanie, gdy wpisujemy `1j * 1j < 5`?
11. W sesji interaktywnej wpisz `"pies"`, `'kot'`, `żyba`". Uwaga w potrójnym lub `'` można wstawiać enter. W pojedynczych lub `'` używamy `\n`.
12. Zdefiniuj `s1 = ' Pies i kot'` oraz `s2 = ' ryba'`. Wykonaj i zinterpretuj działania `s1 + s2`, `s2 * 5`. Czy łańcuchy uległy zmianie po wykonaniu tych poleceń?
13. Funkcja `len(łańcuch znaków)` zwraca długość łańcucha. Wpisz `len(s1)`, `len(s2)`. Czy argumentem funkcji może być liczba? Sprawdź.
14. Sprawdź i zinterpretuj polecenia. `s1[2]`, `s1[0]`, `s1[4]`, `s2[5]`, `s1[2 : 7]`, `s1[: 6]`, `s1[3 :]`, `s1[: 2]`, `s1[1 : 7 : 2]`, `s1[-2]`.
15. Sprawdź i zinterpretuj polecenia (metody dla łańcuchów znaków): `s2.find('a')`, `s1.find('ot')`, `s1.find('je')`, `s1.upper()`, `s1.lower()`
16. Sprawdź i zinterpretuj `s1.replace('kot', s2)`. Czy łańcuch `s1` uległ zmianie po wykonaniu tego polecenia?
17. Wpisz

```
s =''' Math is fun
      math is cool'''
```

Wyświetl `s`, następnie wyświetl `s` za pomocą funkcji `print` (`print(s)`). Widzisz jakąś różnicę?

18. Stwórz nowy łańcuch  $p = ' \textit{Math is easy}'$ . Sprawdź następujące polecenia:  $\textit{print}(p, s)$ ,  $\textit{print}(p, s, sep = '\n')$ ,  $\textit{print}(p, s, sep = '???)$ ,  $\textit{print}(p, s, sep = ' ')$ ,  $\textit{print}(p, s, sep = '')$ ,  $\textit{print}(p, s, sep = '\n', end = '!!!)$ ,  $\textit{print}(p, s, sep = '\n', end = '\n!!!)$ .
19. Czy Python wyświetli błąd po wpisaniu  $\textit{True} + 6$  lub  $\textit{False} * 25$ . Zinterpretuj wyniki.