

**Uruchom notatnik jupyter i w kolejnych komórkach wykonaj zadania:**

**1. zapytaj użytkownika o imię i wypisz je 7 razy w jednej linii a następnie 5 razy w osobnych liniach,**

```
# Pytamy użytkownika o imię
imie = input("Podaj swoje imię: ")

# Wypisanie imienia 7 razy w jednej linii
print((imie + " ") * 7)

# Wypisanie imienia 5 razy w osobnych liniach
for _ in range(5):
    print(imie)
```

**2. dla listy zawierającej różnych 10 liczb:**

**1. bez użycia pętli, wypisz te większe od 10,**

```
liczby = [3, 12, 7, 20, 8, 14, 22, 5, 18, 30]
print([x for x in liczby if x > 10])
```

**2. zapisz to zadanie w jednej linii,**

```
print([x for x in [3, 12, 7, 20, 8, 14, 22, 5, 18, 30] if x > 10])
```

**3. zapisz to samo zadanie z użyciem pętli,**

```
wynik = []
for x in liczby:
    if x > 10:
        wynik.append(x)
print(wynik)
```

**4. zapisz to samo zadanie z użyciem pętli w jednej linii (List Comprehension),**

```
print([x for x in liczby if x > 10])
```

**3. utwórz listę kolejnych liczb od 50 do 150, wypisz te podzielne przez 3 i przez 7,**

```
liczby = list(range(50, 151))
podzielne = [x for x in liczby if x % 3 == 0 and x % 7 == 0]
print(podzielne)
```

**4. utwórz słownik, przechowaj w nim 10 nazw krajów i ich stolic,**

```
kraje_i_stolice = {
    "Polska": "Warszawa",
    "Niemcy": "Berlin",
    "Francja": "Paryż",
    "Hiszpania": "Madryt",
    "Włochy": "Rzym",
    "Grecja": "Ateny",
    "Szwecja": "Sztokholm",
    "Australia": "Canberra",
    "Argentyna": "Buenos Aires",
    "Egipt": "Kair"
}
```

**1. wypisz wszystkie stolice,**

```
print(list(kraje_i_stolice.values()))
```

**2. wypisz tylko te kraje których stolice zaczynają się na A,**

```
kraje_z_stolicami_na_a = [kraj for kraj, stolica in kraje_i_stolice.items()
    if stolica.startswith("A")]
print(kraje_z_stolicami_na_a)
```

**3. wypisz tylko te kraje, w których nazwie występuje litera E,**

```
kraje_z_e = [kraj for kraj in kraje_i_stolice.keys() if 'e' in kraj.lower()]
print(kraje_z_e)
```

**5. dla listy 10 liczb całkowitych, wyświetl największą i najmniejszą z nich (bez użycia biblioteki numpy),**

```
liczby = [15, 3, 7, 22, 9, 13, 25, 4, 18, 10]
najwieksza = max(liczby)
najmniejsza = min(liczby)
print("Największa:", najwieksza, "Najmniejsza:", najmniejsza)
```

**6. Wylosuj 20 liczb rzeczywistych z zakresu 0 - 1, i wypisz średnią liczb oraz największą i najmniejszą z nich używając biblioteki numpy,**

```
import numpy as np

losowe_liczby = np.random.rand(20)
srednia = np.mean(losowe_liczby)
najwieksza = np.max(losowe_liczby)
najmniejsza = np.min(losowe_liczby)

print("Średnia:", srednia)
print("Największa:", najwieksza)
print("Najmniejsza:", najmniejsza)
```

**7. Wylosuj tabelkę 10 wierszy i 20 kolumn liczb całkowitych z zakresu 5 do 15 używając biblioteki numpy**

```
macierz = np.random.randint(5, 16, (10, 20))
print(macierz)
```

**1. policz średnie wartości w wierszach,**

```
srednie_w_wierszach = np.mean(macierz, axis=1)
print("Średnie w wierszach:", srednie_w_wierszach)
```

**2. wyznacz największe wartości w wierszach,**

```
najwieksze_w_wierszach = np.max(macierz, axis=1)
print("Największe wartości w wierszach:", najwieksze_w_wierszach)
```

**3. wyznacz pozycje (indeksy) największych wartości w wierszach,**

```
indeksy_najwiekszych_w_wierszach = np.argmax(macierz, axis=1)
print("Indeksy największych wartości w wierszach:",
      indeksy_najwiekszych_w_wierszach)
```

**8. zdefiniuj funkcję która dla przekazanej jako argument listy liczb zwróci tylko te dodatnie, przetestuj tę funkcję,**

```
def tylko_dodatnie(liczby):
    return [x for x in liczby if x > 0]

# Przetestowanie funkcji
przyklad = [-5, 3, -1, 7, -2, 10]
print(tylko_dodatnie(przyklad))
```

**9. zdefiniuj funkcję która dla słownika z krajami i stolicami, wypisze sumaryczną liczbę liter w nazwach krajów i w nazwach stolic**

```
def liczba_liter(kraje_stolice):
    suma_liter_kraje = sum(len(kraj) for kraj in kraje_stolice.keys())
```

```
    suma_liter_stolice = sum(len(stolica) for stolica in kraje_stolice.values())  
    return suma_liter_kraje + suma_liter_stolice  
  
# Przetestowanie funkcji  
print(liczba_liter(kraje_i_stolice))
```