

```
1 # Przypominajka - słowniki (klucz-wartość)
2 # Tworzenie słownika
3 student = {
4     'name': 'Jan Kowalski',
5     'age': 22,
6 }
7
8 # -----
9
10 # FORMAT JSON (JavaScript Object Notation)
11 # - lekki format wymiany danych,
12 # - łatwy do odczytu i zapisu dla ludzi
13 # - łatwy do interpretacji i generowania przez maszyny
14 # - format tekstowy
15 # - używany do przesyłania danych między serwerem, a aplikacją internetową
16
17
18 # STRUKTORA
19 # - pary klucz - wartość,
20 #       - klucz: napis,
21 #       - wartość: napis, liczba, obiekt-zagnieżdżony JSON, tablica-lista, bool, null
22 # - uporządkowane listy wartości,
23
24
25
26 # PODOBIĘŃSTWA
27 # -----
28 # 1. Mają klucz i wartość (np. "Imię": "Ala").
29 # 2. Mogą zawierać zagnieżdżone dane (pudełka w pudełkach).
30 # 3. Python łatwo zamienia słownik ↔ JSON.
31
32 # RÓŻNICE
33 # -----
34 # 1. Słownik: działa tylko w Pythonie.
35 #   JSON: działa wszędzie – jest światowym standardem.
36 # 2. Słownik: używany wewnętrz programu.
37 #   JSON: używany do wysyłania danych między komputerami.
38 # 3. JSON trzeba zamienić na tekst i z powrotem (serializacja/deserializacja).
39
40
41 # Zapoznanie z metodami obu struktur:
42 # * wrzucić do folderu:
43 # https://drive.google.com/file/d/1DUM6Ikkna_k59fIaWE8yt-WbB35nqDky/view
44
45
46 # ----- METODY SŁOWNIKA -----
47 #       klucze : wartość
48 gra = {"nazwa_gry" : "CS",
49         "data_wydania" : 1999,
50         "wydawca" : "valve",
51         "gatunek" : "strzelanka"}
```

```
54 # - odwoływanie się do elementów:  
55 # print(gra.get("nazwa_gry"))  
56 # print(gra["nazwa_gry"])  
57  
58 # print(gra.get("nazwa_gy")) #None  
59 # print(gra["nazwa_gy"]) # Błąd  
60  
61 print("\n")  
62 # - iterowanie po wartościach:  
63 for value in gra.values():  
64     print(value)  
65  
66  
67 print("\n")  
68 # - iterowanie po kluczach:  
69 for key in gra.keys():  
70     print(key)  
71  
72  
73 print("\n")  
74 # - iterowanie po parach klucz-wartość:  
75 for item in gra.items():  
76     print(item)  
77  
78  
79 print("\n")  
80 # - dodanie pary klucz-wartość:  
81 gra.setdefault("PEGI", 18)  
82 print(gra)  
83  
84  
85 print("\n")  
86 # - usunięcie i zwrócenie pary kluczy (spod podanego klucza)  
87 delated = gra.pop("wydawca")  
88 print(delated)  
89 print(gra)  
90  
91  
92 print("\n")  
93 # - usunięcie i zwrócenie ostatniej pary klucz-wartość  
94 last_item = gra.popitem()  
95 print(last_item)  
96 print(gra)  
97  
98  
99 print("\n")  
100 # - usunięcie pary klucz-wartość spod podanego klucza  
101 del gra["gatunek"]  
102 print(gra)  
103  
104  
105  
106  
107
```

```
108 print("\n")
109 # - usunięcie wszystkich par klucz-wartość
110 gra.clear()
111 print(grat)
112
113
114 print("\n")
115 # - biblioteka pprint (pretty printer) estetyczniejsze wyświetlanie danych
116 import pprint
117 pprint.pprint(grat)
118
119 # -----
120
121 # =====
122 # 📚 BIBLIOTEKA json
123 # =====
124 # Biblioteka json pomaga:
125 # - zamieniać obiekty Pythona na tekst w formacie JSON (serializacja),
126 # - zamieniać tekst JSON na obiekty Pythona (deserializacja).
127 #
128 # To jest potrzebne np. gdy:
129 # - zapisujemy dane do pliku,
130 # - wysyłamy dane przez internet (np. do API),
131 # - chcemy, żeby inne programy mogły odczytać nasze dane.
132
133 # -----
134 # PODSTAWOWE FUNKCJE BIBLIOTEKI json
135 # -----
136 # 1. json.dumps()
137 #     - zamienia obiekt Pythona (np. słownik) na łańcuch znaków (string)
138 #         w formacie JSON.
139 #
140 # 2. json.loads()
141 #     - zamienia łańcuch znaków w formacie JSON na obiekt Pythona
142 #         (np. słownik).
143 #
144 # 3. json.dump()
145 #     - zapisuje obiekt Pythona do pliku w formacie JSON.
146 #
147 # 4. json.load()
148 #     - wczytuje dane JSON z pliku i zamienia je na obiekt Pythona.
149 #
150
151
152
153 # ----- ĆWICZENIE -----
154 import pprint
155 import json
156
157 gra = {"nazwa_gry" : "CS",
158         "data_wydania" : 1999,
159         "wydawca" : "valve",
160         "gatunek" : "strzelanka"}
```

```
162 # otwiera plik w trybie read (czytanie)
163 with open("11.json", "r") as file:
164
165     # W słowniku szukamy listy pod kluczem "spis_gier"
166     spis_gier = json.load(file)
167
168 # Dodajemy do listy nową grę którą wcześniej stworzyliśmy
169 spis_gier["spis_gier"].append(gra)
170
171 # Wyświetlamy zaktualizowaną listę gier
172 pprint.pprint(spis_gier["spis_gier"])
173
174
175 # Tworzymy lub otwieramy plik 11_2.json w trybie zapisu
176 with open("11_2.json", "w") as file:
177
178     # Zapisujemy cały słownik do pliku w formacie JSON
179     json.dump(spis_gier, file, indent = 4, sort_keys = True)
180
181
182 # ----- CIEKAWOSTKA -----
183 # ŁACZENIE SŁOWNIKÓW
184
185 dict1 = {"a": 4, "b": 3}
186 dict2 = {"c": 1, "d": 2}
187
188 # sposób 1
189 dict3 = {**dict1, **dict2}
190
191 # sposób 2
192 dict4 = dict1 | dict2
193
194 print(dict3)
195 print(dict4)
196
197 # ----- ZADANIE DODATKOWE -----
198 # CIĄG FIBONACCIEGO
199
200 fib_json = {0:0, 1:1}
201 def fibonacci(n):
202     if n in fib_json:
203         return fib_json[n]
204     else:
205         result = fibonacci(n-1) + fibonacci(n-2)
206         fib_json[n] = result
207         return result
208
209 fibonacci(20)
210 for item in fib_json.items():
211     print(item)
```