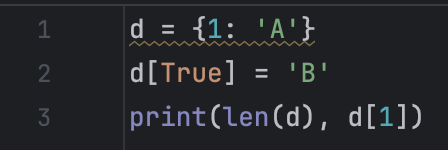
**Урок 3.** **Проектная деятельность. Работа со словорями. Что такое JSON.**

ФИО\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

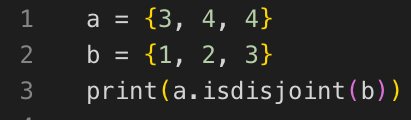
Тест:

1. *Что выведет данная программа?*

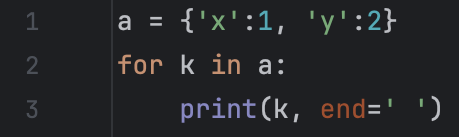


*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

1. *Что* ***нельзя*** *использовать как ключ?*
2. ('x', 1)
3. 3.14
4. ['a', 'b']
5. frozenset({1,2})
6. *Что выведет данная программа?*

**

1. *Что выведет данная программа?*

**

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

1. *Как получить* ***новое множество*** *— объединение a и b*?
2. a.update(b)
3. a | b
4. set(a, b)
5. a + b

***Теория:***

## *Словари (dict)*

**Словарь** — это коллекция, в которой значения хранятся не по индексу, а по ключу. Словарь сам по себе неупорядоченная структура данных — это значит, что все пары «ключ — значение» хранятся в произвольном порядке.

Правила задания ключа:

* Разрешено: int, str, bool, tuple(из хешируемых), frozenset.
* Нельзя: list, dict, set (изменяемые).
* True и 1 — «одинаковые» ключи

#Создание и инициализация  
d = {} # пустой словарь  
d = {'a': 1, 'b': 2}  
d = dict([('a', 1), ('b', 2)]) # из пар  
d = dict(a=1, b=2) # через именованные аргументы  
d = dict.fromkeys(['a','b'], 0) # {'a':0, 'b':0}  
d = {k: k\*k for k in range(5)} # dict-comprehension

#Чтение и запись  
d[k] = v # присвоение (создаст/перезапишет)  
x = d[k] # KeyError, если ключа нет  
x = d.get(k, default) # безопасно, не создаёт ключ, вовращяет default если не нашел   
x = d.setdefault(k, default) # вернёт значение, а при отсутствии запишет default и вернёт его  
k in d # проверка по ключу

#Обход и представление  
for k in d: ... # по ключам (в порядке вставки)  
for k, v in d.items(): ...  
d.keys(), d.values(), d.items() # «живые» представления

#Обновление и удаление  
d.update(other) # перезаписывает существующие ключи  
del d[k] # KeyError, если нет  
v = d.pop(k, default)   
k, v = d.popitem() # удаляет ПОСЛЕДНЮЮ вставленную пару   
d.clear()

dict\_sample = { "a": 1, "b": 2, "c": 3 }  
x = dict\_sample.copy() # поверхностная копия (вложенные объекты общие)  
# Для полной независимости: import copy; deep = copy.deepcopy(d).

## *Словарь и JSON: как связаны*

**Сериализация данных в Python** — это процесс преобразования объектов в формат, пригодный для хранения или передачи. В языке есть встроенные библиотеки для сериализации данных в разные форматы: например JSON.

**JSON** (JavaScript Object Notation) — текстовый формат обмена данными.

| **JSON** | **Python** |
| --- | --- |
| object | dict |
| array | list |
| string | str |
| number | int/float |
| true/false | True/False |
| null | None |

* В **JSON ключи — только строки**. В dict ключи могут быть любыми хешируемыми.
* Типы вроде set, tuple, datetime напрямую в JSON **не сериализуются** — их нужно преобразовывать.

import json  
  
data = {'name': 'Оля', 'scores': [5, 4, 5], 'ok': True, 'note': None}  
  
s = json.dumps(data, ensure\_ascii=False, indent=2) # dict -> JSON-строка  
back = json.loads(s) # JSON-строка -> dict

**Задачи:**

1. **Количество слов.** По строке s (низкий регистр, сплит по пробелам) вывести по каждому слову из строки его частоту попадания в эту строку. Используйте get в качестве счетчика.
2. **Группировка учеников.** Список пар (имя, класс) → {'7А': [...], '8Б': [...]}. Используйте setdefault.
3. **Инверсия с коллизиями.** По словарю user→city получить city→set(users).
4. **Слияние прайс-листов.** Два словаря товар→цена/кол-во. Объединить, **складывая** значения одинаковых ключей.
5. **Анаграммы.** По списку слов сгруппировать анаграммы: ключ — «подпись» (''.join(sorted(word))), значение — список слов.
6. **Two-Sum индексы.** По массиву a и S вернуть **любую** пару индексов i ≠ j, где a[i] + a[j] == S.
7. **Счётчик пар по сумме.** Для массива посчитать, сколько пар (i<j) даёт каждую возможную сумму. Вернуть словарь sum→count.
8. **JSON-отчёт по оценкам.** Дана JSON-строка со списком учеников:

[{"name":"Ann","scores":[5,4]}, {"name":"Bob","scores":[3,4,5]}].

Распарсить и получить словарь name → средняя\_оценка.