

План обучения с ключевыми темами, кратким объяснением, базовыми конструкциями и практическими задачами (с решениями) для каждой темы.

1. Ввод-вывод данных

Суть: обмен информацией между программой и пользователем.

Основные функции:

- `input()` — считывает строку с клавиатуры;
- `print()` — выводит данные на экран;
- форматирование строк: f-строки, `.format()`, `%`.

Примеры конструкций:

```
name = input("Ваше имя: ")
print(f"Привет, {name}!")
price = 99.99
print("Цена: %.2f руб." % price) # Цена: 99.99 руб.
```

Задачи с решениями

Задача 1. Запросить у пользователя два числа и вывести их сумму.

Решение:

```
a = float(input("Первое число: "))
b = float(input("Второе число: "))
print(f"Сумма: {a + b}")
```

Задача 2. Ввести имя и возраст, вывести: «Привет, [имя]! Тебе [возраст] лет».

Решение:

```
name = input("Имя: ")
age = input("Возраст: ")
print(f"Привет, {name}! Тебе {age} лет.")
```

Задача 3. Ввести температуру в градусах Цельсия, перевести в Фаренгейты по формуле $F = C * 9/5 + 32$.

Решение:

```
c = float(input("Температура в °C: "))
f = c * 9/5 + 32
print(f"{c}°C = {f}°F")
```

2. Структуры данных

Суть: способы хранения и организации данных.

Основные типы:

- **Списки** (list) — упорядоченные изменяемые коллекции: `[1, 2, 3]`;
- **Строки** (str) — неизменяемые последовательности символов: `"hello"`;
- **Словари** (dict) — пары «ключ-значение»: `{"name": "Alice", "age": 16}`;
- **Множества** (set) — неупорядоченные уникальные элементы: `{1, 2, 3}`;
- **Кортежи** (tuple) — неизменяемые последовательности: `(1, 2, 3)`.

Методы:

- списки: `.append()`, `.remove()`, индексация `[0]`;
- строки: `.upper()`, `.split()`, срезы `::2`;
- словари: `.keys()`, `.values()`, доступ по ключу `d["key"]`.

Задачи с решениями

Задача 1. Создать список из 5 чисел, вывести первый и последний элементы.

Решение:

```
numbers = [10, 20, 30, 40, 50]
print("Первый:", numbers[0])
print("Последний:", numbers[-1])
```

Задача 2. Ввести строку, вывести её в обратном порядке.

Решение:

```
text = input("Строка: ")
print("Обратно:", text[::-1])
```

Задача 3. Создать словарь с данными о книге (название, автор, год), вывести все значения.

Решение:

```
book = {
    "название": "1984",
    "автор": "Джордж Оруэлл",
    "год": 1949
}
for key, value in book.items():
    print(f"{key}: {value}")
```

Задача 4. Объединить два списка, удалить дубликаты с помощью множества.

Решение:

```
list1 = [1, 2, 3]
list2 = [3, 4, 5]
unique = list(set(list1 + list2))
print("Без дублей:", unique) # [1, 2, 3, 4, 5]
```

3. Циклы

Суть: повторение действий.

Типы:

- for — перебор элементов коллекции;
- while — выполнение пока условие истинно.

Конструкций:

```
for i in range(5):
    print(i)
```

```
while x > 0:
    x -= 1
```

Задачи с решениями

Задача 1. Вывести числа от 1 до 10.

Решение:

```
for i in range(1, 11):
    print(i)
```

Задача 2. Вывести квадраты чисел от 1 до 5.

Решение:

```
for i in range(1, 6):
    print(f"{i}^2 = {i**2}")
```

Задача 3. Считать числа, пока не введён 0, вывести их сумму.

Решение:

```
total = 0
while True:
    num = int(input("Число (0 для выхода): "))
    if num == 0:
        break
    total += num
print("Сумма:", total)
```

Задача 4. Вывести таблицу умножения 5×5.

Решение:

```
for i in range(1, 6):
    for j in range(1, 6):
        print(f"{i}×{j}={i*j}", end="\t")
    print() # новая строка
```

4. Функции

Суть: блоки кода, которые можно вызывать многократно.

Конструкция:

```
def имя(аргументы):
    тело функции
    return результат
```

Примеры:

```
def square(x):
    return x ** 2

def greet(name="Дпур"):
    print(f"Привет, {name}!")
```

Задачи с решениями

Задача 1. Написать функцию, которая возвращает максимум из двух чисел.

Решение:

```
def max_of_two(a, b):
    if a > b:
        return a
    else:
        return b
print(max_of_two(10, 20)) # 20
```

Задача 2. Написать функцию, которая проверяет, является ли число чётным.

Решение:

```
def is_even(n):
    return n % 2 == 0

print(is_even(4)) # True
print(is_even(7)) # False
```

Задача 3. Написать функцию, которая считает факториал числа (через цикл).

Решение:

```
def factorial(n):
    result = 1
    for i in range(1, n + 1):
        result *= i
    return result
```

```
print(factorial(5)) # 120
```

Задача 4. Написать функцию, которая принимает список и возвращает сумму его элементов.

Решение:

```
def sum_list(lst):
    total = 0
    for item in lst:
        total += item
    return total
```

```
numbers = [1, 2, 3, 4, 5]
print(sum_list(numbers)) # 15
```

Задача 5. Написать функцию с аргументом по умолчанию: приветствие с именем.

Решение:

```
def hello(name="Ученик"):
    print(f"Привет, {name}!")

hello()      # Привет, Ученик!
hello("Алиса") # Привет, Алиса!
```

Общие рекомендации по отработке

1. **Сначала — разбор примера, потом — самостоятельная реализация.**
2. **Меняйте входные данные — проверяйте, работает ли код для разных случаев.**
3. **Ошибайтесь и исправляйте — это часть обучения.**
4. **Пишите комментарии к своему коду — так вы лучше поймёте логику.**
5. **Повторяйте задачи через 2–3 дня — закрепляйте навык (иначе все забудете через 2 дня).**

Как двигаться дальше

После освоения этих тем:

- комбинируйте их в проектах (калькулятор, список дел, игра «Виселица»);
- изучайте **условия** (if-elif-else) и **логические операторы**;
- переходите к **файлам** (open(), read(), write());
- осваивайте **библиотеки** (random, datetime, requests).

Главное: не спешите. Лучше решить 5 задач на одну тему, чем 1 задачу на 5 тем.

Онлайн ресурсы:

1. [PythonLib Теория, задачи, видеоуроки, книги, форум, онлайн компилятор, Python, курсы питон бесплатно программирование питон](#)
2. <https://umschool.net/library/informatika/osnovy-programmirovaniya-na-yazyke-python-chast-1/>
3. <https://education.yandex.ru/handbook/python/article/vvod-i-vyvod-dannykh-operatsii-s-chislami-strokami-formatirovaniye>
4. https://mmx2024.ucoz.net/programming_for_beginners.html
5. https://mmx2024.ucoz.net/pascaltasks/tasks_python.html
6. <https://vkvideo.ru/@club227978201/playlists>



У вас все получится на 5!