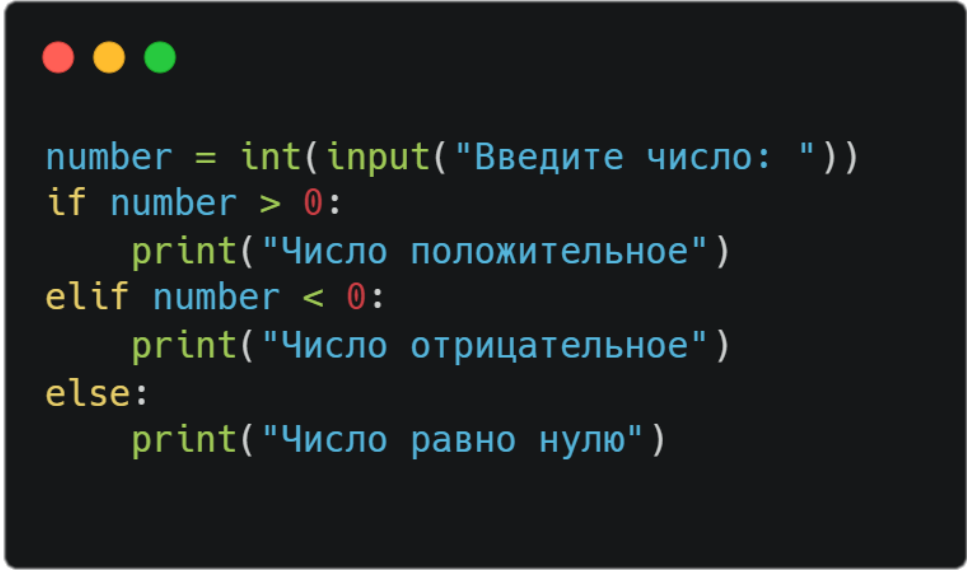


День 1: Введение в условные операторы

Условие: Написать программу, которая принимает число и проверяет, положительное оно или отрицательное.

Решение:



```
number = int(input("Введите число: "))
if number > 0:
    print("Число положительное")
elif number < 0:
    print("Число отрицательное")
else:
    print("Число равно нулю")
```

Задача 1

Условие: Написать программу, которая принимает три числа и выводит наибольшее из них.

Методические указания:

- Используйте вложенные условные операторы if-elif-else;
- Продумайте все возможные случаи сравнения трёх чисел;
- Попробуйте также решить задачу с использованием встроенной функции max()

Задача 2

Условие: Напишите программу, которая определяет, является ли год високосным.

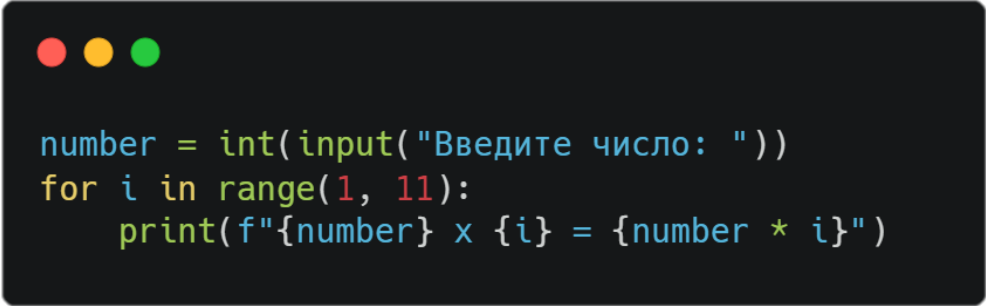
Методические указания:

- Год является високосным, если он делится на 4;
- Но если год делится на 100, он не високосный;
- Однако, если год делится на 400, он високосный;
- Используйте операторы and и or для составления условия.

День 2: Простые циклы

Условие: Написать программу, которая выводит таблицу умножения для введённого числа.

Решение:

A dark-themed code editor window with three colored window control buttons (red, yellow, green) in the top-left corner. It contains Python code for a multiplication table.

```
number = int(input("Введите число: "))  
for i in range(1, 11):  
    print(f"{number} x {i} = {number * i}")
```

Задача 1

Условие: Написать программу, которая находит сумму всех чётных чисел от 1 до N.

Методические указания:

- Используйте цикл for и range();
- Проверяйте чётность числа с помощью оператора %;
- Организуйте накопление суммы в отдельной переменной.

Задача 2

Условие: Написать программу, которая выводит все делители введённого числа.

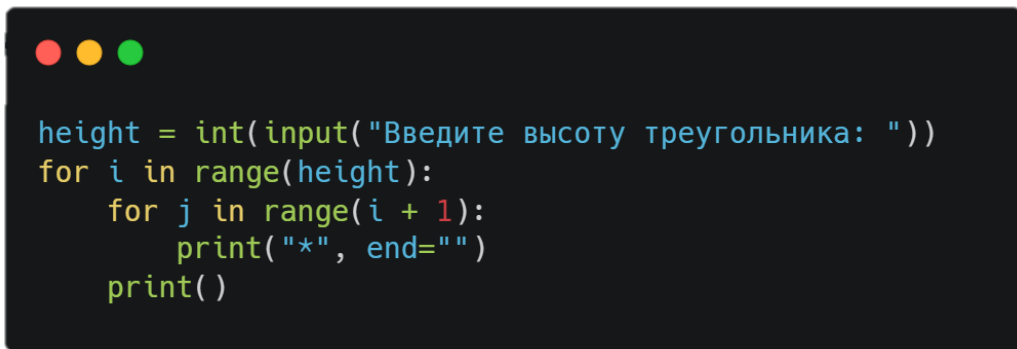
Методические указания:

- Используйте цикл for для перебора чисел от 1 до введённого числа;
- Используйте оператор % для проверки делимости;
- Сохраняйте делители в список для последующего вывода.

День 3: Вложенные циклы

Условие: Нарисовать прямоугольный треугольник из звёздочек.

Решение:

A screenshot of a terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The terminal displays a Python script that takes a height as input and prints a right-angled triangle of asterisks. The code is as follows:

```
height = int(input("Введите высоту треугольника: "))
for i in range(height):
    for j in range(i + 1):
        print("*", end="")
    print()
```

Задача 1

Условие: Написать программу, которая выводит таблицу умножения от 1 до 9.

Методические указания:

- Используйте два вложенных цикла for;
- Продумайте форматирование вывода для красивого отображения таблицы;
- Используйте f-строки для форматирования.

Задача 2

Условие: Найти все простые числа от 1 до N.

Методические указания:

- Используйте вложенные циклы: внешний для перебора чисел, внутренний для проверки делителей;
- Создайте функцию is_prime() для проверки числа на простоту;
- Используйте список для хранения найденных простых чисел.

День 4: Циклы с условиями

Условие: Написать программу, которая находит первое число-палиндром после введенного числа.

Решение:

```
def is_palindrome(n):  
    return str(n) == str(n)[::-1]  
number = int(input("Введите число: "))  
while True:  
    number += 1  
    if is_palindrome(number):  
        print(f"Следующий палиндром: {number}")  
        break
```

Задача 1

Условие: Написать программу, которая находит наибольший общий делитель (НОД) двух чисел.

Методические указания:

- Используйте алгоритм Евклида;
- Реализуйте как с помощью цикла while, так и с помощью рекурсии;
- Проверьте работу программы на различных парах чисел.

Задача 2

Условие: Написать программу для игры "Угадай число" с компьютером.

Методические указания:

- Используйте модуль random для генерации случайного числа;
- Организуйте цикл while для процесса угадывания;
- Добавьте подсказки "больше"/"меньше" после каждой попытки.

День 5: Работа со строками и циклы

Условие: Проверить, является ли строка палиндромом (с учётом только букв и цифр).

Решение:

```
def is_palindrome(text):  
    cleaned_text = ''.join(c.lower() for c in text if c.isalnum())  
    return cleaned_text == cleaned_text[::-1]  
text = input("Введите текст: ")  
if is_palindrome(text):  
    print("Это палиндром")  
else:  
    print("Это не палиндром")
```

Задача 1

Условие: Найти самое длинное слово в введённом предложении.

Методические указания:

- Разбейте строку на слова методом `split()`;
- Используйте цикл для перебора слов;
- Храните информацию о самом длинном слове и его длине;
- Учтите случай, когда самых длинных слов несколько.

Задача 2

Условие: Зашифровать текст шифром Цезаря с заданным сдвигом.

Методические указания:

- Используйте ASCII-коды символов (`ord()` и `chr()`);
- Учтите, что буквы должны "переворачиваться" по кругу (z -> a);
- Сохраняйте регистр букв и не изменяйте небуквенные символы.

День 6: Продвинутые задачи

Условие: Реализовать программу для построения ромба из символов с заданной диагональю.

Решение:

```
def print_diamond(n):
    for i in range(n):
        print(" " * (n - i - 1) + "*" * (2 * i + 1))
    for i in range(n - 2, -1, -1):
        print(" " * (n - i - 1) + "*" * (2 * i + 1))
size = int(input("Введите размер ромба: "))
print_diamond(size)
```

Задача 1

Условие: Написать программу для решения sudoku 3x3 (упрощённая версия).

Методические указания:

- Создайте функцию для проверки корректности расстановки чисел;
- Используйте вложенные циклы для перебора ячеек;
- Реализуйте рекурсивный алгоритм для решения;
- Добавьте проверку на наличие решения.

Задача 2

Условие: Реализовать игру "Жизнь" Конвея на поле 10x10.

Методические указания:

- Используйте двумерный список для хранения состояния клеток;
- Реализуйте функции для подсчёта соседей и определения следующего состояния;
- Организуйте красивый вывод игрового поля;
- Добавьте возможность задать начальное состояние.