**Практическое задание № 1: Работа с переменными, функциями и циклами**

1. Изменяемый остаток total

total = 100  
  
while total > 0:  
 n = int(input('Снятие: '))  
 if n > total:  
 print('Недостаточно средств, операция отменена.')  
 break  
 total -= n  
  
print('Финальный остаток:', total)

1. Степени двойки от 0 до 20

i = 0  
while i <= 20:  
 print(2 \*\* i)  
 i += 1

1. Случайный список

from random import randint  
  
nums = []  
for \_ in range(10): # любое нужное количество  
 nums.append(randint(-50, 50)) # пример диапазона  
print(nums)

1. Диапазон → список шагом 17

lst = list(range(0, 101, 17))  
print(lst) # [0, 17, 34, 51, 68, 85]

1. Считаем отрицательные элементы

data = [12, -4, 7, -9, 0, -1]  
negatives = 0  
for x in data:  
 if x < 0:  
 negatives += 1  
print('Отрицательных:', negatives)

1. Список слов и их длин

words = []  
lengths = []  
  
for \_ in range(5):  
 w = input('Введите слово: ')  
 words.append(w)  
 lengths.append(len(w))  
  
print('Слова: ', words)  
print('Длины: ', lengths)

1. Вызов функций из функции

def positive():  
 print('Положительное')  
  
def negative():  
 print('Отрицательное')  
  
def test():  
 n = int(input('Число: '))  
 if n >= 0:  
 positive()  
 else:  
 negative()  
  
# вызов главной функции  
test()

В Python объявление функции должно встретиться до её первого вызова. Поэтому расположение positive() и negative() относительно test() не критично, если сам вызов test() расположен последним.

Ответы на контрольные вопросы

|  |
| --- |
|  |
| int, float, str, bool, list, tuple, dict, set, NoneType. Объявляем присваиванием: x = 10, name = "Ann". |
| Инкапсулируют повторяющийся код. Создаются через def; параметры позиционные/именованные; результат через return. |
| while — пока условие истинно; for — перебор последовательности. |
| Обрабатываем try … except; предотвращаем падение программы при ошибках. |
| Счётчики ссылок + сборщик мусора; использовать with, генераторы, del для крупных объектов. |