

Lesson 2

Повторение

1. Теория из урока 1
2. Разбор домашнего задания:

```
# Запрашиваем у пользователя его любимое животное
favorite_animal = input("Введите ваше любимое животное: ")

# Запрашиваем, как пользователь хочет назвать это животное
animal_name = input("Как вы хотите назвать это животное? ")

# Выводим результат
print("Ваше любимое животное – " + favorite_animal + ". Вы назвали его " +
      animal_name + ".")
```

Legend

Правила наименования переменных в Python

1. Зарезервированные слова

Зарезервированные слова (или ключевые слова) – это специальные слова, которые имеют особое значение в языке программирования и не могут быть использованы как имена переменных, функций, классов или других объектов в программе.

Примеры зарезервированных слов в Python:

```
if, else, def, class, import, for, while, return, try, except.
```

2. Виды регистров



camelCase



kebab-case



snake_case

В python:

- **snake_case** используется для переменных, функций и пакетов.
- **CamelCase** используется для имен классов.

Также стоит отметить, что **Python** является регистрозависимым (case-sensitive) языком, то есть в **Python** `full_name` и `ful_nAme` — это разные переменные.

3. Запрещенные символы в имени переменной

Пробелы: Не допускается использование пробела в имени переменной. Вместо пробела обычно используется подчеркивание (стиль **snake_case**), например: `my_variable`, `user_data`.

- Пример запрещенного: `my variable`

- **Специальные символы:** Символы, такие как `@`, `#`, `!`, `$`, `%`, `^`, `&`, `*`, `(`, `)`, `{`, `}`, `[]`, `=`, `+`, `-`, `|`, `,`, `:` и другие, запрещены в именах переменных.

- Пример запрещенного: `my@variable`, `total$amount`

- **Числа в начале:** Имя переменной не может начинаться с цифры. Переменная должна начинаться с буквы или подчеркивания.
- Пример запрещенного: `2variable`

Типы данных

Примитивные типы данных — это базовые типы данных, которые не могут быть разделены на более простые составляющие.

1. Числовые типы данных:

Числа — это примитивные типы данных, которые могут быть целыми или с плавающей запятой

- **int (целое число):**

Целые числа, как положительные, так и отрицательные, являются примитивным типом данных.

```
x = 5      # целое число
y = -42    # целое отрицательное число
```

- **float (число с плавающей запятой):**

Числа с дробной частью. Они представляют собой числа с плавающей точкой, такие как 3.14, -0.001, 2.0.

```
pi = 3.14159
temperature = -5.2
```

2. Логический тип данных (bool):

Тип данных `bool` используется для представления логических значений — `True` (истина) и `False` (ложь). Логические значения широко используются для условий, циклов и логических операций.

```
is_active = True
is_valid = False
```

3. Строки (str):

Строки — это последовательности символов, заключенные в одинарные или двойные

кавычки. Строки в Python являются неизменяемыми (immutable), то есть после их создания нельзя изменить их содержимое. Это позволяет эффективно управлять памятью и гарантировать, что строка не будет случайно изменена.

```
name = "Alice"  
greeting = 'Hello, World!'
```

Примечание: В Python используется динамическая типизация

Динамическая типизация — это особенность языков программирования, когда тип переменной не указывается явно при ее объявлении, и может быть изменен во время выполнения программы. В Python тип переменной определяется на основе значения, которое в нее присваивается, и может изменяться в процессе выполнения программы.

Математические операции

1. Основные арифметические операции:

- **Сложение (+):** Складывает два числа.
- **Вычитание (-):** Вычитает одно число из другого.
- **Умножение (*):** Умножает два числа.
- **Деление (/):** Делит одно число на другое и возвращает число с плавающей точкой.
- **Целочисленное деление (//):** Делит два числа и возвращает целую часть результата (округляет вниз).
- **Остаток от деления (%):** Возвращает остаток от деления.
- **Возведение в степень:** оператор `**`.

2. Операции с типами данных:

- **Преобразование в целое число (int()):** Преобразует строку или число с плавающей точкой в целое число.

```
int("42") # 42  
int(42.5) # 42
```

- **Преобразование в число с плавающей точкой (float()):** Преобразует строку или целое число в число с плавающей точкой.

```
float("42.5") # 42.5  
float(42) # 42.0
```

3. Операции с математическими функциями:

Python предоставляет стандартный модуль `math`, который содержит многие математические функции.

- **Квадратный корень:** Используется `math.sqrt()`.

```
import math
math.sqrt(16) # 4.0
```

- **Факториал:** Используется `math.factorial()`.

```
math.factorial(5) # 120
```

- **Возведение в степень:** Используется `math.pow()` или оператор `**`.

```
math.pow(2, 3) # 8.0
```

- **Пи (`math.pi`):** Константа Пи.

```
math.pi # 3.141592653589793
```

4. Работа с числами с плавающей точкой:

Для чисел с плавающей точкой доступны дополнительные операции:

- **Округление:**
- `round()` — округляет число до заданного количества знаков после запятой.

```
round(3.14159, 2) # 3.14
```

- **Преобразование в целое число (округление вниз):** `math.floor()` — округляет вниз до ближайшего целого.

```
import math
math.floor(3.7) # 3
```

- **Округление вверх:** `math.ceil()` — округляет вверх до ближайшего целого.

```
import math
math.ceil(3.1)    # 4
```

5. Пример с комбинацией операций:

```
a = 10
b = 20
c = 3

result = (a + b) * c / 2    # (10 + 20) * 3 / 2 = 45.0
print(result)
```

Задачи

****1. Сложение чисел****

Запросите два числа у пользователя и выведите их сумму.

****2. Конкатенация строк****

Запросите два текста у пользователя и выведите их, соединяя через пробел.

****3. Возведение числа в квадрат****

Запросите число у пользователя и выведите его квадрат.

****4. Преобразование строки в число****

Запросите строку, представляющую число, и выведите его целочисленное значение.

****5. Объединение строк****

Запросите три строки у пользователя и выведите их соединенные в одну.

****6. Деление чисел****

Запросите два числа у пользователя и выведите результат их деления.

****7. Преобразование числа в строку****

Запросите число и преобразуйте его в строку.

****8. Умножение чисел****

Запросите два числа и выведите их произведение.

****9. Конкатенация строки с числом****

Запросите строку и число у пользователя и выведите их как одну строку.

ДЗ-2

1. Вычисление периметра прямоугольника

Запросите у пользователя длину и ширину прямоугольника в виде чисел, затем вычислите и выведите его периметр.

Пример:

Ввод:

Введите длину прямоугольника: 5
Введите ширину прямоугольника: 3

Вывод:

Периметр прямоугольника: 16

2. Вычисление площади прямоугольника

Запросите у пользователя длину и ширину прямоугольника в виде чисел, затем вычислите и выведите его площадь.

Пример:

Ввод:

Введите длину прямоугольника: 5
Введите ширину прямоугольника: 3

Вывод:

Площадь прямоугольника: 15

Необходимые формулы:

1. Периметр прямоугольника вычисляется по формуле:

$$P = 2 * (a + b)$$

2. Площадь прямоугольника вычисляется по формуле:

$$S = a * b$$