Основы Python

Структура материалов курса

- Курс состоит из 8 уроков по 2 часа.
- Практические задания.
- Видеозапись.
- Методичка, презентация и исходные коды.
- Примеры выполнения каждого задания.

Цели курса

- Изучить основы Python.
- Проникнуться философией Python.
- Научиться писать правильный Python-код.
- Набить руку на задачах различной сложности.

План курса

Урок 1. Знакомство с Python

Урок 2. Встроенные типы и операции с ними

Урок 3. Функции

Урок 4. Импорт, модули и полезные возможности языка

Урок 5. Работа с файлами

Урок 6. ООП. Введение

Урок 7. ООП. Продвинутый уровень

Урок 8. ООП. Полезные дополнения

Практические задания

- ★ Рекомендуемый формат сдачи: ссылка на пулл-реквест в GitHub (или в виде архива).
- ★ В начале каждого урока обсуждаем ПЗ.
- ★ Выполняем задания не позднее 4 часов до начала.

Вопросы участников

GeekBrains

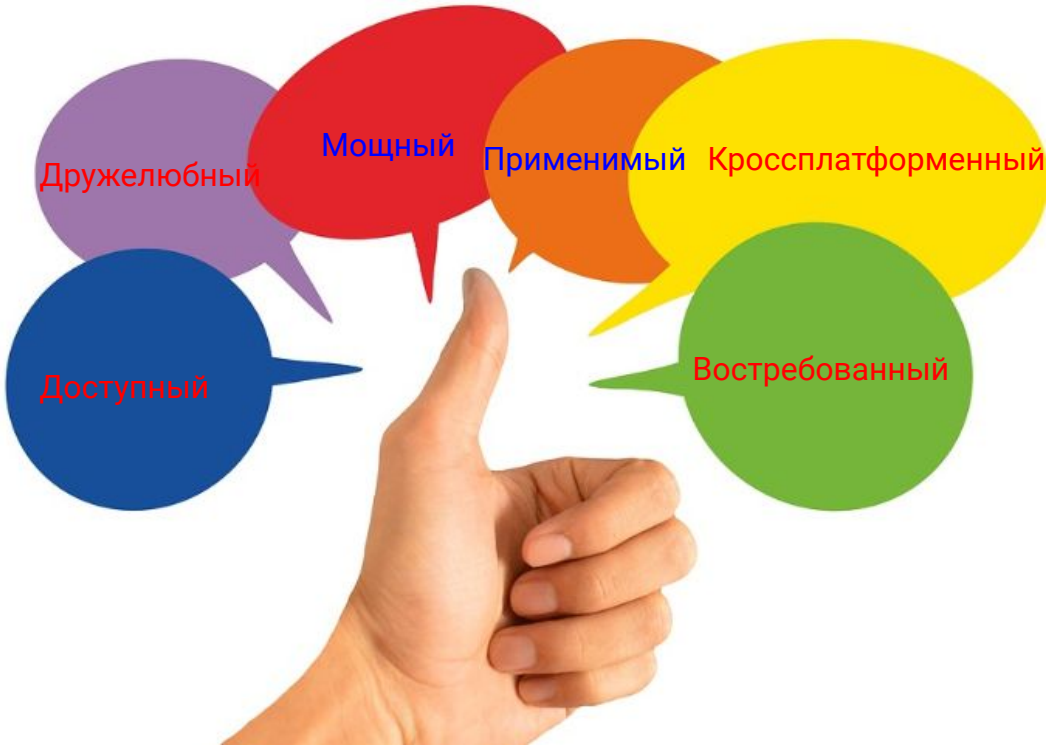
Урок 1

Знакомство с Python

На этом уроке

1. Что такое язык программирования Python, его преимущества.
2. Области применения.
3. Установка интерпретатора.
4. Что такое IDE.
5. Особенности программирования на Python.
6. Из чего состоит программа.
7. Динамическая типизация.
8. Реализация ввода/вывода.
9. Арифметические и логические операции.
10. Следования, ветвления, циклы.
11. Способы форматирования строк.
12. Частые ошибки начинающих разработчиков.

Python и его преимущества



Интерпретируемый, высокоуровневый

Где применяется

Веб-приложения

Игровые приложения

Сложные вычисления

Алгоритмы машинного обучения

Приложения с GUI

Проекты в области ИИ

Системы анализа и визуализации данных

Системные утилиты

Приложения для работы с БД

Примеры проектов с Python

BitTorrent

Центр приложений Ubuntu

BLENDER

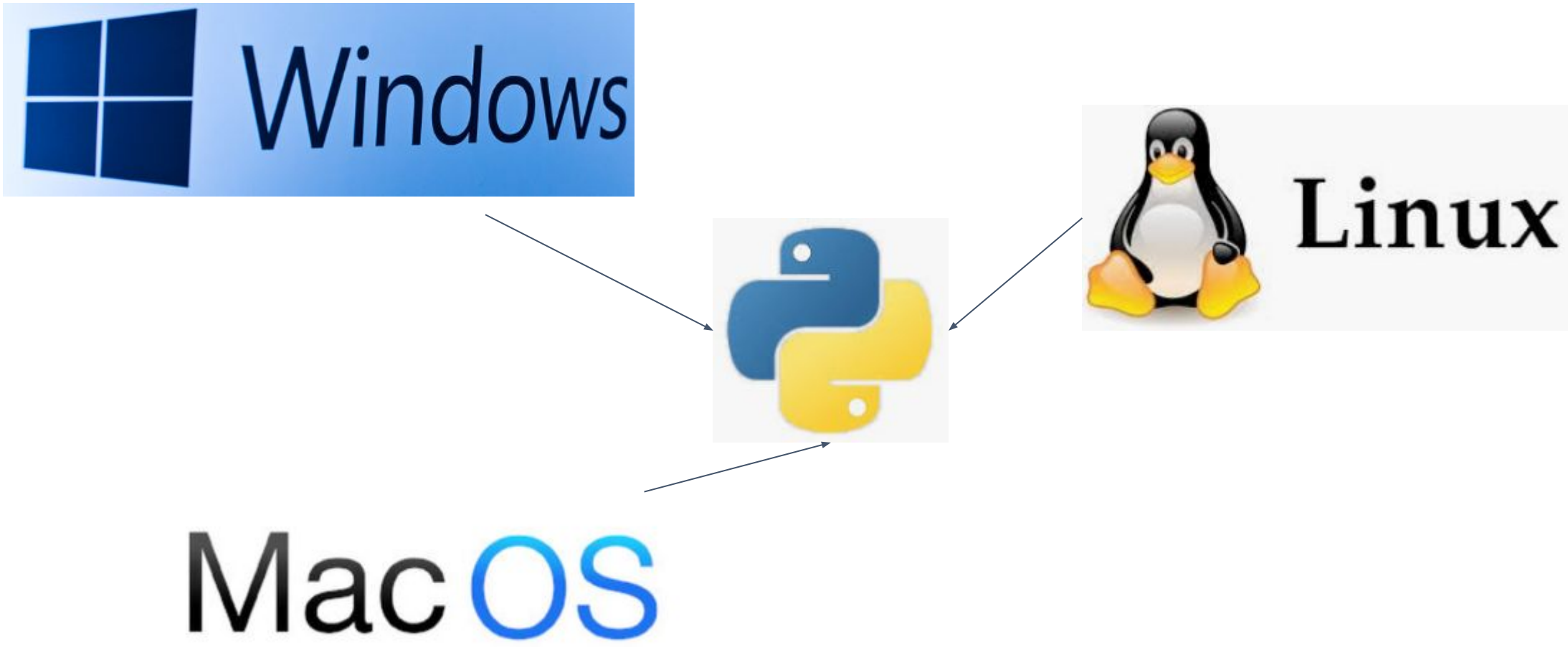
GIMP

YouTube

iRobot

Dropbox

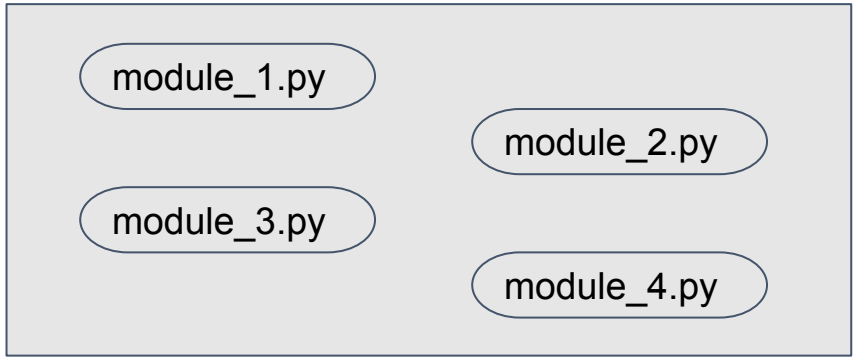
Интерпретатор с Python



IDE PyCharm

Программа на Python

Программа



Инструкции

```
a + b == b + a
```

Выражения

```
a + b
```

Объекты

```
a, b
```

Динамическая типизация

Java — статическая типизация

String price; — У переменной есть тип

price = “thousand”; — У переменной есть тип

~~price = 1000;~~ — Изменить тип переменной невозможно

Python — динамическая типизация

price = “thousand” — У переменной есть тип

price = 1000 — Тип переменной динамически изменился

Базовые встроенные типы

Тип	Пример
int	2, 4, 8, -10, -2
float	2.6, -5.2
str	“my_text”
bool	True, False
list	[2, 2.4, “Hello”]
tuple	(2, 2.4, “Hello”)
dict	{“name”: “Вася”, “age”: 10}

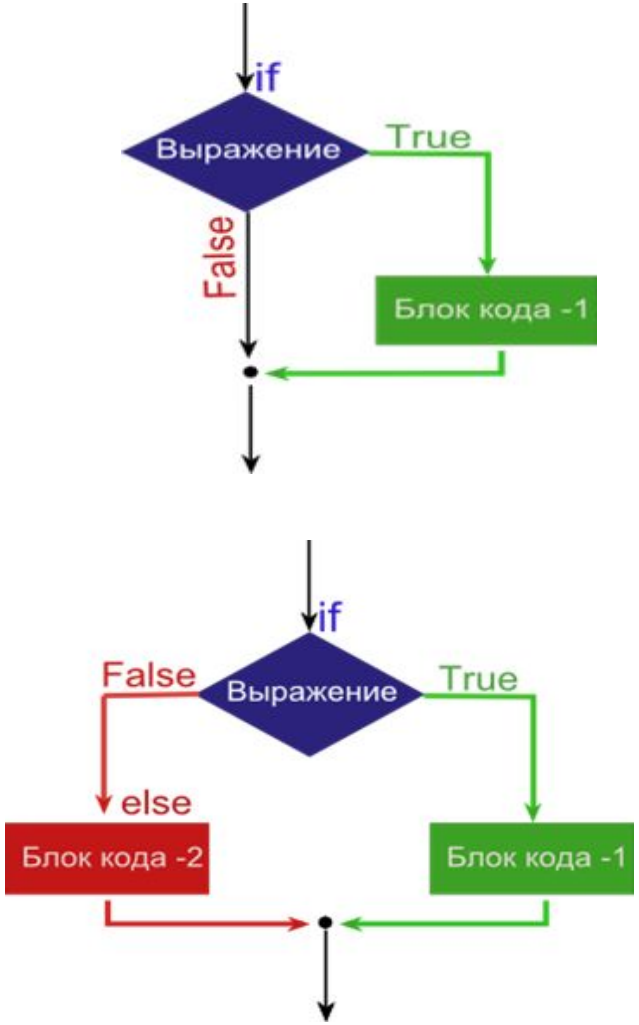
Арифметические операции

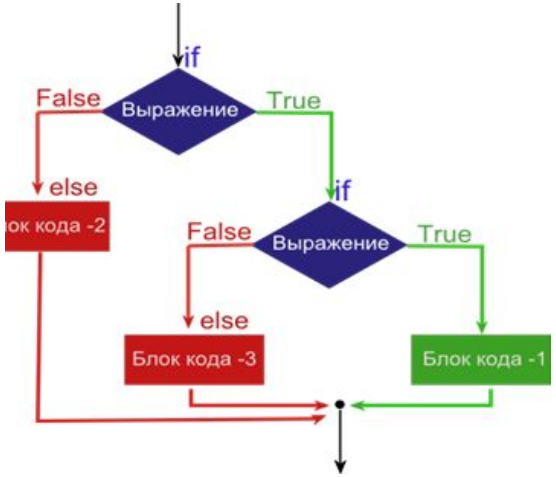
Тип	Пример
+	print(398 + 20) -> 418
-	print(200 - 50) -> 150
*	print(34 * 7) -> 238
/	print(36 / 6) -> 6.0
//	print(36 // 6) -> 6
%	print(36 % 6) -> 0
**	print(2 ** 16) -> 65536

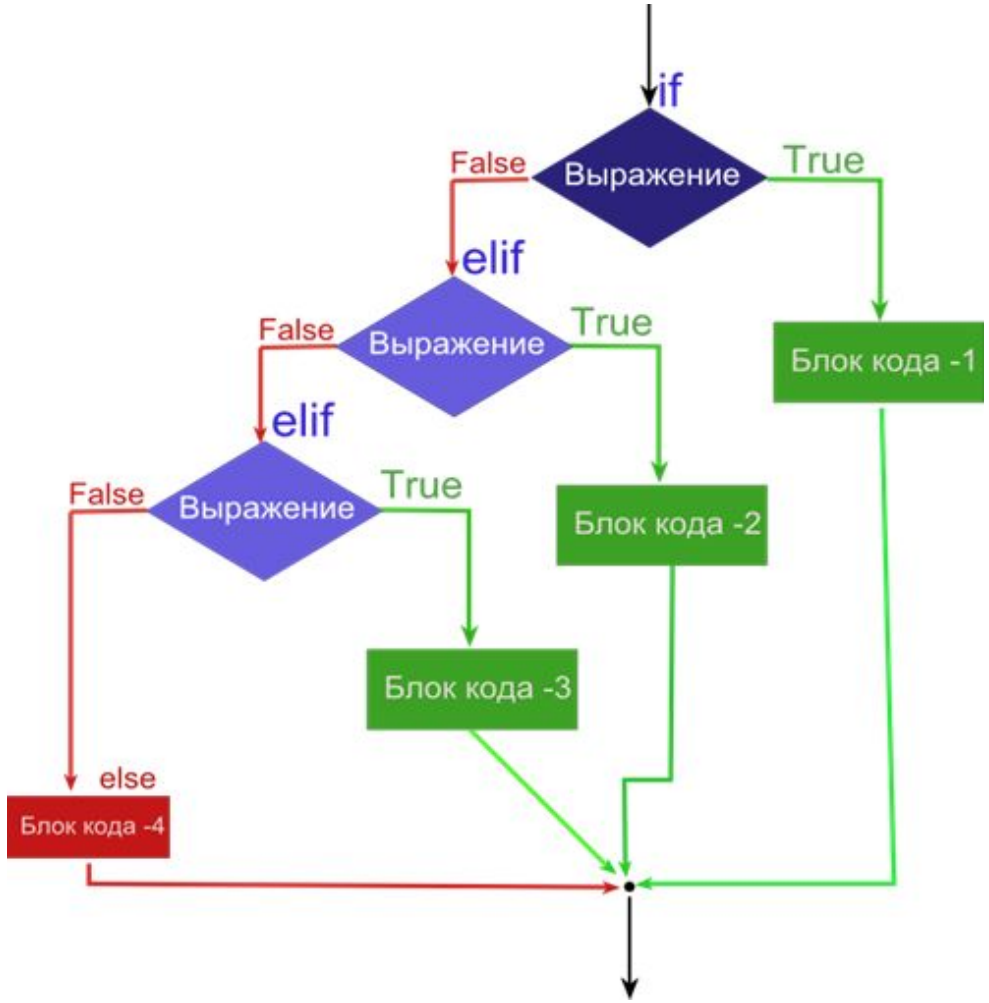
Базовые логические операции

Тип	Пример
>	print(40 > 40) -> False
<	print(3 < 9) -> True
==	print(10 == 10) -> True
!=	print(2 != 2) -> False
>=	print(40 >= 1) -> True
<=	print(3 <= 1) -> False
and	print(True and False) -> False
or	print(True or False) -> True

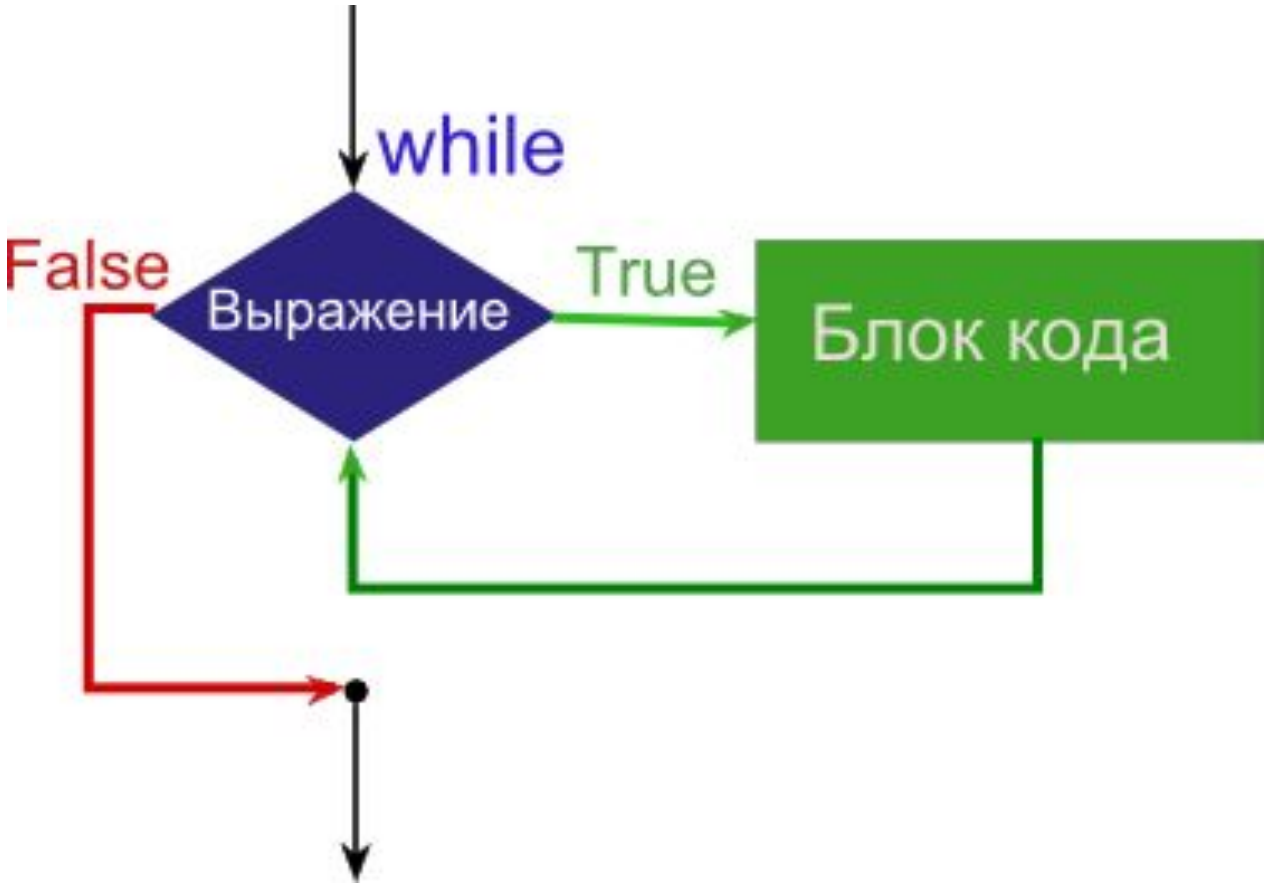
Ветвления







Циклы



Способы форматирования строк

Форматирование через f-строки

Форматирование через метод format

Форматирование через оператор %

"Старый стиль" форматирования строк (оператор %)

Зарезервированные слова в Python. Часть 1

Название	Описание
`False`	Значение «Ложь»
`None`	«Не определено», пустой объект
`True`	Значение «Истина»
`and`	Логическое «И»
`as`	Определение псевдонима для объекта
`assert`	Генерация исключения, если условие ложно
`async`	Обозначение функций как сопрограмм для использования циклом событий
`await`	
`break`	Выход из цикла

Зарезервированные слова в Python. Часть 2

Название	Описание
`class`	Пользовательский тип (класс), содержащий атрибуты и методы
`continue`	Переход на очередную итерацию цикла
`def`	Определение функции
`del`	Удаление объекта
`elif`	Ещё иначе, если
`else`	Иначе, если
`except`	Перехват исключения
`finally`	Выполнение инструкций, независимо были ли исключения или нет
`for`	Начало цикла перебора элементов набора

Зарезервированные слова в Python. Часть 3

Название	Описание
`from`	Указание пакета или модуля, из которого выполняется импорт
`global`	Значение переменной, присвоенное ей внутри функции, становится доступным вне этой функции
`if`	Если
`import`	Импорт модуля
`in`	Проверка на вхождение
`is`	Проверка, ссылаются ли два объекта на одно и то же место в памяти
`lambda`	Определение анонимной функции
`nonlocal`	Значение переменной, присвоенное ей внутри функции, становится доступным в объемлющей функции
`not`	Логическое «НЕ»

Зарезервированные слова в Python. Часть 4

Название	Описание
`pass`	Заглушка для функции или класса. Используется, когда код класса и функции ещё не определён
`raise`	Генерация исключения
`return`	Вернуть результат
`try`	Выполнить инструкции с перехватом исключения
`while`	Начало цикла «ПОКА»
`with`	Использование менеджера контекста
`yield`	Определение функции-генератора

Лучшие онлайн-компиляторы Python

Итоги

1. Познакомились с языком Python, особенностями программирования на этом языке.
2. Узнали, как установить интерпретатор и среду разработки.
3. Рассмотрели конструкции ветвления и циклы.
4. Узнали, как запросить данные у пользователя и выполнить форматирование строк.

На втором уроке мы остановимся на встроенных типах и операциях с ними.