# Язык программирования «Python»

Как, куда, зачем и почему

# Язык Python

- Современные тенденции таковы, что пусть даже если язык Python и не является самым популярным на сегодня, то так или иначе масштабы его меньше не становятся. Наоборот, они растут всё выше и выше. Как, соответственно, и спрос на программистов, работающих с данным языком.
- Основная причина возрастания популярности языка заключается в том, что он отличается как минимум тремя вещами: простота, красота, эффективность.

# Красота Python?

- Что особенного в языке Python?
- Среди наиболее важных характеристик языка Python можно выделить следующие:

 Язык интерпретируемый. Что это значит? Это значит, что при первом запуске программы на выполнение для неё создаётся промежуточный код. Именно промежуточный код используется при выполнении программы. Если впоследствии в программу мы вносим изменения, то при очередном запуске программы создаётся новый промежуточный код.

# Памятка

- Языки программирования как таковые бывают либо интерпретируемыми, либо компилируемыми. Если программа компилируется, то на основе исходного кода создаётся машинный (исполнительный) код, который и выполняется при запуске программы. Если же речь идёт об интерпретируемом языке, то программа, написанная на нём, выполняется построчно, без предварительной компиляции.
- Существует и промежуточный вариант: нечто среднее между компилированием и интерпретированием.

# Памятка

- В таком случае промежуточного варианта исходный код преобразуется в промежуточный код, который уже затем интерпретируется при выполнении.
- Интерпретируемые языки позволяют себе гораздо больше вольности, если так можно выразиться. Это проявляется как в описании, так и в обработке данных.
- Программы, написанные на компилируемых языках, характеризуются высокой скоростью выполнения.

# Возвращаемся к красоте Python

- Если рассматривать Python в плане синтаксиса, то он достаточно прост и лаконичен. Он не содержит громоздких избыточных конструкций. Но в то же время он является и не менее строгим языком по отношению к синтаксису: даже лишний пробел в программном коде может привести к ошибке.
- Язык Python поддерживает парадигму объектно-ориентированного программирования (ООП). Тем не менее он позволяет создавать программы, не использующие ни классов, ни объектов.

# Возвращаемся к красоте Python

- Язык Python удобен для создания приложений с графическим интерфейсом (и не только их).
- Ну и наконец еще одним фактором популярности языка является большое и дружное сообщество разработчиков, использующих данный язык. А так как сообщество огромное и дружное, то, соответственно, недостатка в свободно распространяемых программных продуктах, облегчающих жизнь питонистов, не выявлено.

# На что будет упор?

- Цель нашего обучения – научиться программировать на языке Python. Но учиться можно по разному.
- Опыт показывает, что проще всего усвоить основные концепции программирования и подходы, когда они проиллюстрированы конкретными примерами.

# Среда разработки

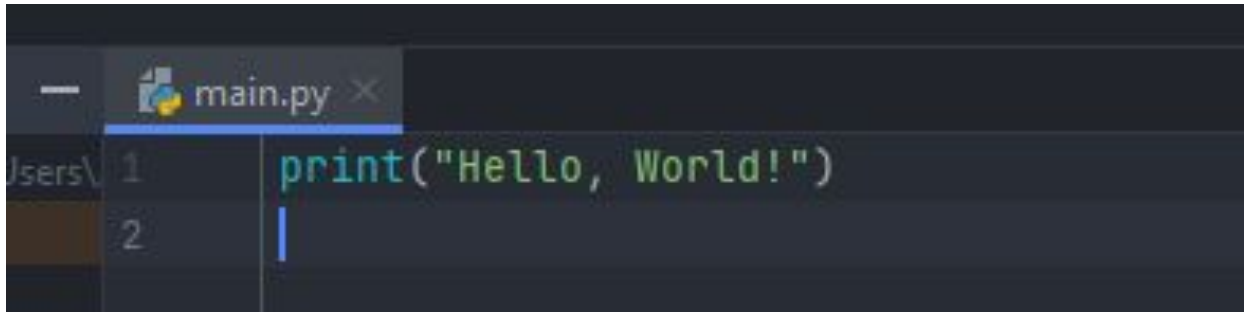
- Ну тут импровизация.

# Первая программа

- Для написания первой программы нам необходимо знать, во-первых, каковы правила составления программ на языке Python и, во-вторых, как и где набрать программный код и что с ним затем делать. Мы дадим ответы на оба вопроса.
- Собственно, программа – это набор команд и инструкций. Создание программы, отображающей в окне вывода определённое сообщение – простейший пример, с которого обычно начинается изучение любого языка. Так уж сложилось.
- В Python подобная программа очень проста и состоит всего из одной команды.

# «Hello, World!»

- Команда, из которой состоит программа, представляет собой инструкцию вызова встроенной функции print(). Аргументом функции передаётся текстовое значение (текстовый литерал) “Hello, World!”
- Выглядит это следующим образом:




- Памятка: Текстовые литералы в Python заключаются в двойные или одинарные кавычки.

# «Hello, World!»

- Текстовый аргумент, переданный в функцию print(), при выполнении соответствующей команды отображается в окне вывода интерпретатора Python. Поэтому при выполнении соответствующей команды отображает око вывода интерпретатора Python.
- Вывод выглядит следующим образом:

# Знакомство с переменными

- В программе переменная нужна для того, чтобы присвоить ей значение или чтобы прочитать значение переменной. Переменные можно передавать в качестве аргументов функциям, использовать их в выражениях, да и вообще, программа без переменных всё равно что песня без слов. Извиняюсь.

# Собственно, к чему это я

- Практически любая программа оперирует с данными. Данные в программе сохраняются, кроме прочего, с помощью переменных. Переменная, в общем случае, отождествляется с некоторой областью памяти. В эту область можно записать значение и из такой области можно считать значение. У переменной, как это принято, водится имя. Имя переменной используется для получения доступа к значению, записанному в памяти.
- В свою очередь, существуют разные механизмы реализации переменных. Переменная может как содержать значения, так и ссылаться на них.

# Собственно, к чему это я

- Переменную, которая содержит значение, удобно представить в виде коробочки, в которую можно что-то положить, а затем, при необходимости, посмотреть, что там лежит, ну либо положить туда что-то новое.
- Продолжая эту аналогию, можем отождествить название переменной с именем, написанным на коробочке. Переменная может ссылаться на значение. В таком случае переменная сама по себе значение не содержит. Ее фактическим значением является адрес другой ячейки, в которой содержится значение или в которую значение записывается. В таком случае переменная играет роль посредника. Но в большинстве случаев иллюзия такая, как если бы переменная сама содержала значение, а не ссылалась на него.

# Использование переменных в языке

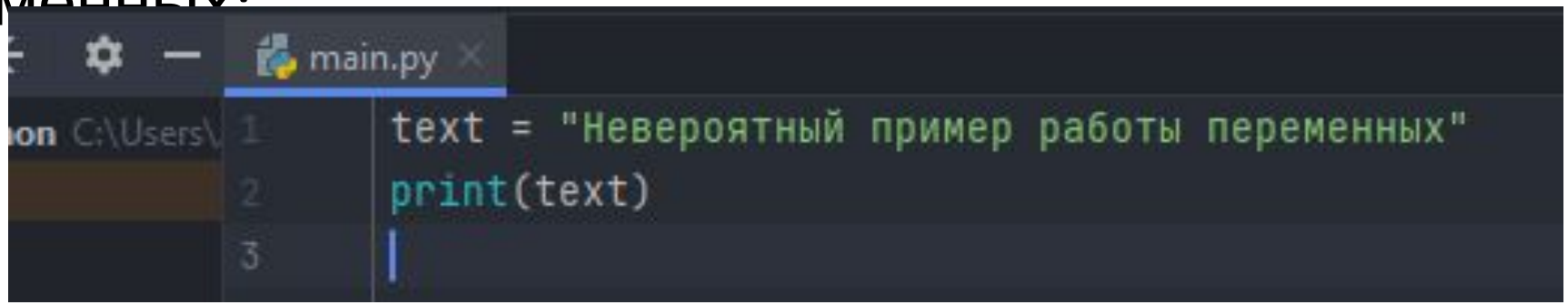
- Использовать переменные в программе на языке Python очень просто. Для этого не нужно даже объявлять переменную. Нет необходимости указывать её тип. Достаточно присвоить значение идентификатору, определяющему имя переменной. Как следствие, в программе появится переменная с соответствующим значением.

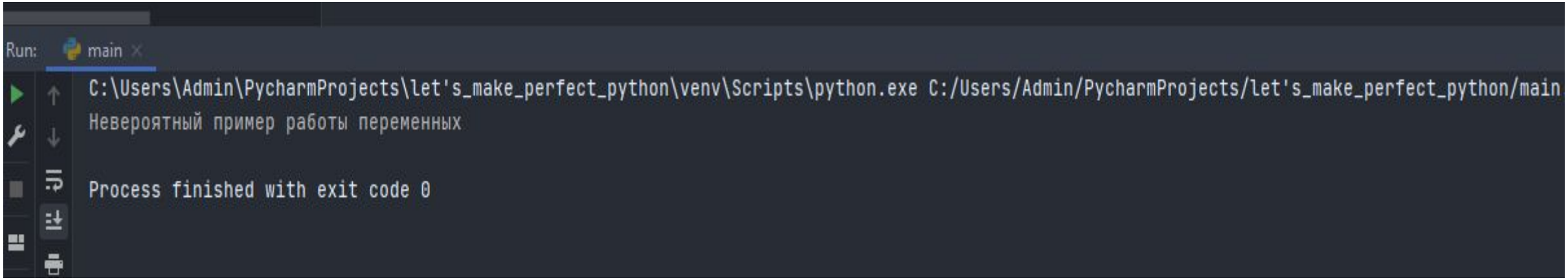
# Небольшое отступление об важности вышесказанного

- Во многих языках программирования необходимо объявлять переменные перед использованием. Объявление переменной обычно подразумевает указание её типа. Тип переменной влияет на размер области памяти, в которой хранится значение переменной. Для языка Python эти правила не действуют. (Правила созданы, чтобы их нарушать, лол).
- В Python переменные ссылаются на значения. Объявлять переменные не нужно. У переменной как такового типа нет. (Но есть тип у значения, на которое ссылается переменная).

# Погнали к примерам

- Небольшой пример программы с использованием переменных:



# Комментарии в программном коде

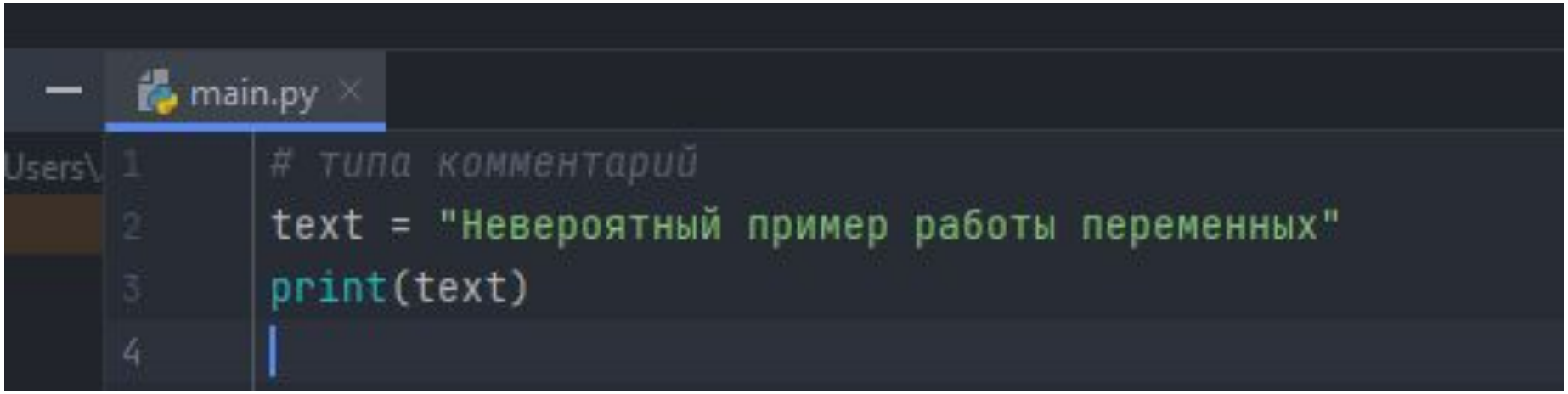
- В программе, кроме, собственно, переменных, мы знакомимся с еще несколькими новыми «конструкциями». Так, в нашей программе появились комментарии. Комментарий – текст, который игнорируется при выполнении программы. Комментарий предназначен для тех, кто работает с программным кодом. Обычно комментарии используют для пояснения тех или иных операций, осуществляемых в программном коде. Создать комментарий просто: он начинается с символа #. Всё, что находится справа от символа игнорируется при выполнении программного кода.

# И снова памятка к синтаксису Python

- В языке Python пробелы используются для структурирования программного кода. Проще говоря, если пробел перед командой не нужен, то его ставить нельзя. В противном случае последствия могут быть драматическими. В этом отношении Python сильно отличается от прочих языков программирования, в которых наличие пробелов перед командами имеет исключительно эстетический эффект. Что касается рассматриваемой программы, то каждая команда размещается в новой строке, без отступов слева.

# Что ш…

- Кроме комментариев наша программа содержит еще две команды:



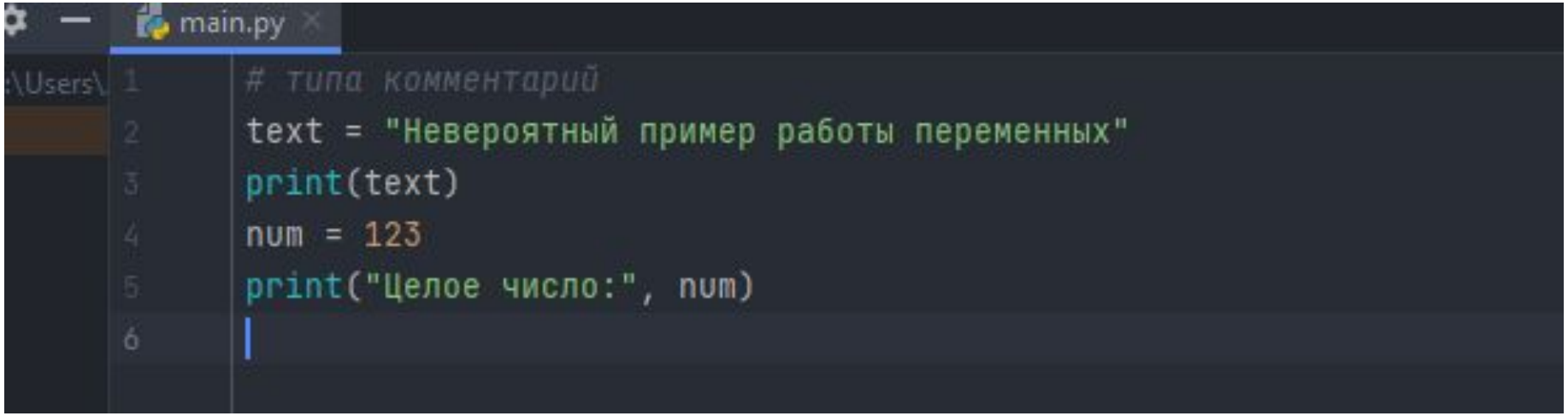

- Стоит отметить, что мы никаким специальным образом не объявляли переменную text. А просто для того, чтобы присвоить значение.

# Памятка

- Если в программном коде в команде присваивания нового значения уже существующей переменной по ошибке неправильно указать имя этой переменной, то случайно можно объявить новую переменную.

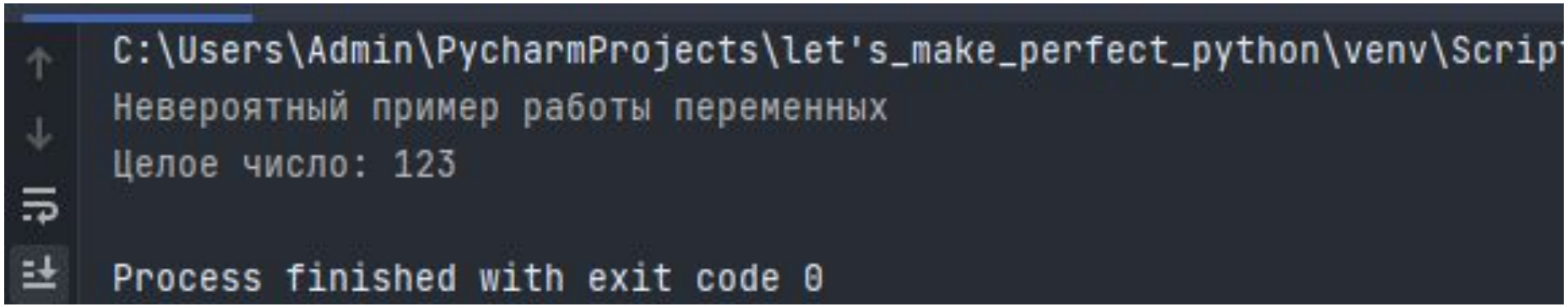
# Пара примеров…

- Добавим еще одну переменную:




- Особенность команды к новой переменной в том, что функции print() передаётся два аргумента: текстовый литерал и переменная num.

# Собственно, вывод



# Финальное

- Как мы могли наблюдать из примеров, у переменных в Python нет типа. Поэтому теоретически на разных этапах выполнения программы одна и та же переменная может ссылаться не просто на разные значения, но на значения разных типов.
- P.S.:Не следует путать тип данных, на которые ссылается переменная, с типом переменной. У данных тип есть. У переменной типа нет. Мы не можем говорить о типе переменной, но можем говорить о типе данных, на которые переменная ссылается. Целые числа относятся к типу int. Текстовые значения относятся к типу str.
- Ну я еще пример какой-нибудь покажу…