Назначение лекции

Вы узнаете, где применяется язык Python и как выглядит программный код. Познакомитесь со средой разработки и формированием простейшего приложения. Вы узнаете, что такое данные в программе, переменные, их тип и действия по записи в переменные различных значений.

Новые слова

**Python** – высокоуровневый язык программирования.

**Тип данных** – понятие в программировании, определяющее количество и структуру памяти для размещения данных.

**Переменная** – символьный адрес места в памяти для данных.

**Присваивание** – запись в переменную конкретных данных.

# Введение

Создатель языка Python — нидерландский программист Гвидо ван Россум. Он был участником проекта по написанию ABC, языка для обучения программированию. В конце 1989 года Гвидо приступил к разработке нового языка и задумал его как потомк ABC. Так и получился Python.

Откуда такое название? Многие разработчики считают, что язык назван в честь семейства змей, но это не так. Когда Гвидо работал над проектом, он любил смотреть комедийное шоу «Летающий цирк Монти Пайтона», поэтому и нарёк своё творение в честь британской комик-группы. Так что правильно произносить название языка как «Пайтон».

Python свободно распространялся через интернет и со временем у него появились последователи — люди, заинтересованные в развитии этого языка программирования. В настоящее время свежая версия — Python 3.10 и выше

Python подходит для решения широкого круга задач и применяется на всех популярных платформах. Росту популярности Python способствовала и его эффективность в стремительно развивающихся сферах [Machine Learning](https://practicum.yandex.ru/blog/chto-takoe-mashinnoe-obuchenie/) и Data Science.

Одной из главных достоинств языка – простота стнтаксиса. Код языка чистый и понятный, без лишних символов и выражений.

Главный недостаток - Python требует высоких вычислительных мощностей серверов и компьютеров. Это делает его не таким быстрым, как хотелось бы, он отстаёт от других языков по части производительности. В условиях сильного развития мощностей это не так заметно, как раньше, но всё равно даёт о себе знать. Чтобы нивелировать эту проблему, разработчики обращаются к C и С++ для реализации проблемного участка кода.

# Код Python

**n=5**

**string="Hello World "**

**print(string \* n)**

Результат:

**Hello World Hello World Hello World Hello World Hello World**

# Среда разработки

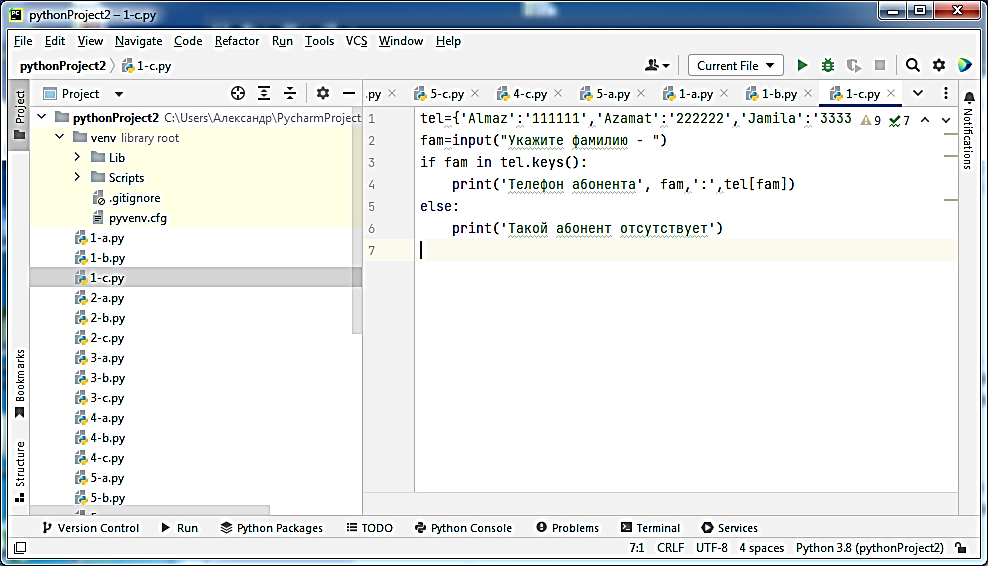
Прежде чем браться за разработку приложений на Python, необходимо загрузить среду разработки (*IDE- Integrated Development Environment*). Их несколько. Одну из них мы будем использовать: PyCharm. Это бесплатная версия.

Для простого кода можно использовать знакомую online среду: <https://www.onlinegdb.com/>

# PyCharm 2022.2.3 (Community Edition)

Рассмотрим интерфейс системы, в объеме, достаточном для выполнения лабораторных работ. Более подробно эту информацию можно получить в сети Интернет. Там этой информации более, чем достаточно.

Главное окно системы:



В редакторе (справа) сразу можно писать код и выполнить (даже по одной строке). Левое поле содержит список всех программ проекта.

# Комментарии

Комментарии — исключительно полезный инструмент программиста.

Комментарии служат двум целям:

1. объясняют, как работает программа;
2. временно блокируют работу отдельных частей программы, чтобы вы могли запустить и протестировать другие части программы.

Комментарии начинаются с символа **#**.

Например,

print("Это пример программы")

print() #Выдает пустую строку для удобства чтения

name = input("Пожалуйста введите ваше имя: ") #Запрашивает ввод имени

print("Привет", name) #Соединяет слово "Привет" и введенное имя

Результат:

Это пример программы

Пожалуйста введите ваше имя: Александр

Привет Александр

В реальных условиях вы не будете сопровождать комментариями строки, содержащие очевидный код. Добавляйте комментарии только там, где они необходимы.

Python **различает регистр символов**, поэтому очень важно использовать правильный регистр; в противном случае ваш код работать не будет.

При выборе имен **переменных** (то есть ячеек памяти, в которых вы хотите хранить данные) нельзя использовать специальные слова (**print**, **input** и т. д.), иначе ваш код работать не будет.

# Примеры кода.

num1 = 93

Присваивает значение переменной. Если переменная еще не существует, она будет создана. Переменная представляет собой ячейку памяти для хранения значения (в данном случае переменная называется «num1», и в ней хранится значение 93.) Значение, хранящееся в переменной, может изменяться во время выполнения программы. Переменной можно присвоить практически любое имя (кроме зарезервированных слов Python — print, save и т. д.); это имя должно начинаться с буквы (но не с цифры или знака) и не может содержать пробелов.

answer = num1 + num2

Складывает **num1** и **num2** и сохраняет результат в переменной с именем **answer**.

answer = num1 - num2

Вычитает **num2** из **num1** и сохраняет результат в переменной с именем **answer**.

answer = num1 \* num2

Умножает **num1** на **num2** и сохраняет результат в переменной с именем **answer**.

answer = num1 / num2

Делит **num1** на **num2** и сохраняет результат в переменной с именем **answer**.

answer = num1 // num2

Выполняет целочисленное деление (например, **9//4 = 2**) и сохраняет результат в переменной с именем **answer**.

print ("Это сообщение")

Выводит сообщение, заключенное в скобки. Так как выводимое значение содержит текст, оно заключается в кавычки, которые не будут отображаться при выводе. Если же вы хотите вывести числовое значение или содержимое переменной, кавычки не понадобятся.

print ("First line\nSecond line")

Последовательность **"\n"** переход на следующую строку.

Например,

First line

Second line

print ("Ответ = ", answer)

Выводит сообщение **"Ответ= "** и значение переменной **answer** в одну строку.

textValue = input("Введите текст: ")

Выводит сообщение **"Введите текст: "** и сохраняет значение, введенное пользователем, в переменной с именем **textValue**. Пробел после двоеточия нужен для того, чтобы вводимые пользователем данные отделялись от сообщения. Без этого все символы будут сцеплены в одну длинную строку, а это некрасиво.

numValue = int(input("Введите число: "))

Выводит сообщение **"Введите число: "** и сохраняет введенное значение в виде целого числа в переменной **numValue**. Целые числа можно спользовать

в вычислениях, а переменные, хранимые в текстовом виде, — нельзя.

numValue = float(input("Введите дробное число: "))

Выводит сообщение **"Введите дробное число: "** и сохраняет введенное значение в виде вещественного числа в переменной **numValue**. Дробные числа можно спользовать в вычислениях, а переменные, хранимые в текстовом виде, — нельзя.

print(round(numValue \*2.5,3))

Выводит значение вычисления **numValue \*2.5** с округлением до 3 знаков после запятой (функция **round**).

На начальном уровне, для обучения можно использовать online сервисы, например, <https://www.onlinegdb.com/online_python_compiler>

Пока все