



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И.
Носова»



УТВЕРЖДАЮ:
Директор ИЭиАС
С.И. Лукьянов
26.02.2020 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

ПРОГРАММИРОВАНИЕ НА PYTHON

Направление подготовки (специальность)
38.03.05 Бизнес-информатика

Уровень высшего образования – бакалавриат

Программа подготовки – академический бакалавриат

Форма обучения
очная

Институт
Кафедра
Курс
Семестр

Институт энергетики и автоматизированных систем
Бизнес-информатики и информационных технологий
1, 2
2, 3

Магнитогорск
2020 г.

Рабочая программа составлена на основе ФГОС ВО по направлению подготовки 38.03.05 Бизнес-информатика (уровень бакалавриата) утвержденного приказом МОиН РФ от 11.08.2016 г. № 1002.

Рабочая программа рассмотрена и одобрена на заседании кафедры Бизнес-информатики и информационных технологий 11.02.2020 г., протокол № 6.

Зав. кафедрой  / Г.Н. Чусавитина /

Рабочая программа одобрена методической комиссией Института энергетики и автоматизированных систем 26.02.2020 г., протокол № 5.

Председатель  / С.И. Лукьянов /

Рабочая программа составлена:

доцент каф. БИиИТ, к.п.н, доцент

 / Е.В. Карманова /

Рецензент:

Руководитель Группы анализа
ИТ-проектов ЗАО "Консом СКС"

 / В.А. Ошурков /

Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2021 - 2022 учебном году на заседании кафедры Бизнес-информатики и информационных

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ Г.Н. Чусавитина

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2022 - 2023 учебном году на заседании кафедры Бизнес-информатики и информационных

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ Г.Н. Чусавитина

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2023 - 2024 учебном году на заседании кафедры Бизнес-информатики и информационных

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ Г.Н. Чусавитина

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2024 - 2025 учебном году на заседании кафедры Бизнес-информатики и информационных

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ Г.Н. Чусавитина

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2025 - 2026 учебном году на заседании кафедры Бизнес-информатики и информационных

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ Г.Н. Чусавитина

1 Цели освоения дисциплины (модуля)

Целями освоения дисциплины «Программирование на Python» являются: формирование теоретических знаний и практических умений, развитие профессиональных компетенций в области программирования для решения практических и научно-исследовательских задач, используемых в дальнейшей профессиональной деятельности.

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Программирование на Python входит в базовую часть учебного плана образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

- Информационные процессы, системы и сети
- Теоретическая информатика
- Основы алгоритмизации и программирования

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

- Анализ больших данных
- Разработка Web-приложений
- Основы машинного обучения
- Нейронные сети

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Программирование на Python» обучающийся должен обладать следующими компетенциями:

Структурный элемент компетенции	Планируемые результаты обучения
ОПК-1 - способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	
Знать	<ul style="list-style-type: none">• синтаксис языка программирования Python;• принципы объектно-ориентированного программирования и их реализацию в Python;• основные стандартные модули, библиотеки языка Python.
Уметь	<ul style="list-style-type: none">• разрабатывать программы на языке программирования Python, создавая собственные классы, а также использовать классы и модули из библиотек этого языка
Владеть	<ul style="list-style-type: none">• навыками разработки консольных приложений в стиле объектно-ориентированного программирования на языке программирования Python;• навыками разработки приложений с внешними источниками данных (текстовыми файлами, xml-файлами, базами данных);• использовать набор библиотек языка Python для научных вычислений и научной визуализации.

4 Структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 7 зачетных единиц 252 акад. часов, в том числе:

- контактная работа – 146,05 акад. часов:
 - аудиторная – 141 акад. часов;
 - внеаудиторная – 5,05 акад. часов
- самостоятельная работа – 70,25 акад. часов;
- подготовка к экзамену – 35,7 акад. часа

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
1. Обработка данных в Python								
1.1. Концепция типов данных в Python		2	2		5	Выполнение лабораторных заданий	Отчет по лабораторной работе	ОПК-1 - зув
1.2. Основные алгоритмические структуры в Python		2	4/2И		5	Выполнение лабораторных заданий	Отчет по лабораторной работе	ОПК-1 - зув
1.3. Функции в Python. Основы функционального подхода		2	2/2И		5	Выполнение лабораторных заданий	Отчет по лабораторной работе	ОПК-1 - зув
1.4. Форматы хранения данных. Технологии парсинга json, xml, html-документов. Библиотеки Python для обработки данных. Регулярные выражения. Форматированный вывод данных.	2	4	8/4И		5	Выполнение лабораторных заданий	Отчет по лабораторной работе	ОПК-1 - зув
1.5. Сетевое программирование. Сокеты. Http-клиент. Анализ Url		2	4/2И		5	Выполнение лабораторных заданий	Отчет по лабораторной работе	ОПК-1 - зув
1.6. Работа с БД в Python. Язык запросов SQL.	2	3	8/4И		10	Выполнение лабораторных заданий	Отчет по лабораторной работе	ОПК-1 - зув
1.7. Библиотеки, реализующие шаблоны веб-страниц.	2	2	6		11,05	Выполнение лабораторных заданий	Отчет по лабораторной работе	ОПК-1 - зув
Итого по разделу	2	17	34/14И		56,05		Зачет	

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
Итого за семестр	2	17	34/14И		56,05		Зачет	
2. Объектно-ориентированное программирование в Python								
2.1. Основы ООП. Классы и объекты.	3	4	8/4И		2	Выполнение лабораторных заданий, изучение литературы	Отчет по лабораторной работе	ОПК-1 -зув
2.2. Наследование и иерархия наследования в Python. Полиморфизм.	3	6	12/4И		2	Выполнение лабораторных заданий, изучение литературы	Отчет по лабораторной работе	ОПК-1 -зув
2.3. Паттерны разработки приложений.	3	4	8/4И		2	Выполнение лабораторных заданий, изучение литературы	Отчет по лабораторной работе	ОПК-1 -зув
2.4. Создание приложений с GUI. Обзор графических библиотек в Python. Принципы построения интерфейса с использованием библиотек Python	3	4	12/4И		2	Выполнение лабораторных заданий, изучение литературы	Отчет по лабораторной работе	ОПК-1 -зув
Итого по разделу	3	18	40/16И		8		Тестирование	
3. Работа с геоданными в Python								
3.1. Обзор библиотек для получения геоданных. Yandex, folium и др. Извлечение статистических данных. Работа с API. Работа с Postman.	3	12	10/8И		4	Выполнение лабораторных заданий, изучение литературы	Отчет по лабораторной работе	ОПК-1 -зув
3.2. Создание интерактивных карт в Python.	3	6	4/4И		2,2	Выполнение лабораторных заданий, изучение литературы	Отчет по лабораторной работе	ОПК-1 -зув
Итого по разделу		18	14/12И		6,2		Тестирование	
Итого за семестр	3	36	54/28И		14,2		Экзамен	

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
Итого по дисциплине	3	53	88/42И		70,25		Экзамен	

5 Образовательные технологии

В ходе проведения занятий предусматриваются следующие образовательные технологии:

1. Информационная лекция – последовательное изложение материала в дисциплинарной логике, осуществляемое преимущественно вербальными средствами (монолог преподавателя).
2. Лабораторная работа – организация учебной работы с реальными материальными и информационными объектами, экспериментальная работа с аналоговыми моделями реальных объектов.
3. Технологии проблемного обучения – организация образовательного процесса, которая предполагает постановку проблемных вопросов, создание учебных проблемных ситуаций для стимулирования активной познавательной деятельности студентов.
4. Лабораторное занятие на основе кейс-метода – обучение в контексте моделируемой ситуации, воспроизводящей реальные условия научной, производственной, общественной деятельности. Обучающиеся должны проанализировать ситуацию, разобраться в сути проблем, предложить возможные решения и выбрать лучшее из них. Кейсы базируются на реальном фактическом материале или же приближены к реальной ситуации.
5. Информационно-коммуникационные образовательные технологии – организация образовательного процесса, основанная на применении специализированных программных сред и технических средств работы с информацией.
6. Формы учебных занятий с использованием информационно-коммуникационных технологий:
7. Лекция-визуализация – изложение содержания сопровождается презентацией (демонстрацией учебных материалов, представленных в различных знаковых системах, в т.ч. иллюстративных, графических, аудио- и видеоматериалов).
8. Лабораторное занятие в форме презентации – представление результатов проектной или исследовательской деятельности с использованием специализированных программных сред.

В рамках практических занятий предусматривается использование средств вычислительной техники при выполнении индивидуальных заданий. Используется существующий образовательный портал университета (newlms.magt.u.ru) для размещения ЭУМК по дисциплине. Текущий, промежуточный и рубежный контроль проводится на образовательном портале университета.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины (модуля)

а) Основная литература:

1. Федоров, Д. Ю. Программирование на языке высокого уровня Python: учебное пособие для среднего профессионального образования / Д. Ю. Федоров. — 2-е изд. — Москва: Издательство Юрайт, 2020. — 161 с. — (Профессиональное образование). — ISBN 978-5-534-11961-9. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/454101>
2. Гниденко, И. Г. Технологии и методы программирования: учебное пособие для вузов / И. Г. Гниденко, Ф. Ф. Павлов, Д. Ю. Федоров. — Москва: Издательство Юрайт, 2020. — 235 с. — (Высшее образование). — ISBN 978-5-534-02816-4. — Текст: электронный // ЭБС Юрайт [сайт]. — URL: <https://urait.ru/bcode/450999>

б) Дополнительная литература:

1. Гуриков, С. Р. Основы алгоритмизации и программирования на Python : учебное пособие / С.Р. Гуриков. — Москва : ФОРУМ : ИНФРА-М, 2020. — 343 с. — (Высшее образование: Бакалавриат). - ISBN 978-5-00091-487-8. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1206074>. - Режим доступа: по подписке.
2. Шелудько, В. М. Основы программирования на языке высокого уровня Python : учебное пособие / В. М. Шелудько ; Южный федеральный университет. - Ростов-на-Дону ; Таганрог : Издательство Южного федерального университета, 2017. - 146 с. - ISBN 978-5-9275-2649-9. - Текст : электронный. - URL: <https://znanium.com/catalog/product/1021662> – Режим доступа: по подписке.
3. Журнал «Программные продукты и системы» [Электронный ресурс]. Научно-исследовательский институт «Центрпрограммистем» — Режим доступа: https://e.lanbook.com/journal/2276#journal_name — Загл. с экрана.

в) Методические указания:

Методические указания по выполнению индивидуальных домашних заданий представлены в приложении 3

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

Наименование ПО	№ договора	Срок действия лицензии
MS Windows 7 Professional(для классов)	Д-1227-18 от 08.10.2018	11.10.2021
MS Office 2007 Professional	№ 135 от 17.09.2007	бессрочно
7Zip	свободно распространяемое ПО	бессрочно
Anaconda Python, Jupyter Notebook (anaconda)	свободно распространяемое ПО	бессрочно
Google Chrome	свободно распространяемое ПО	бессрочно
Браузер Mozilla Firefox	свободно распространяемое ПО	бессрочно

Профессиональные базы данных и информационные справочные системы

Название курса	Ссылка
Поисковая система Академия Google (Google Scholar)	URL: https://scholar.google.ru/
Информационная система - Единое окно доступа к информационным ресурсам	URL: http://window.edu.ru/

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

Учебные аудитории для проведения занятий лекционного типа

Специализированная (учебная) мебель (столы, стулья, доска аудиторная), мультимедийное оборудование (проектор, компьютер, экран) для презентации учебного материала по дисциплине;

Учебные аудитории для проведения лабораторных занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации

Специализированная (учебная) мебель (столы, стулья, доска аудиторная), персональные компьютеры объединенные в локальные сети с выходом в Internet и с доступом в электронную информационно-образовательную среду университета, оснащенные современными программно-методическими комплексами

Аудитории для самостоятельной работы (компьютерные классы; читальные залы библиотеки)

Специализированная (учебная) мебель (столы, стулья, доска аудиторная), персональные компьютеры объединенные в локальные сети с выходом в Internet и с доступом в электронную информационно-образовательную среду университета, оснащенные современными программно-методическими комплексами

Помещение для хранения и профилактического обслуживания учебного оборудования

Мебель (столы, стулья, стеллажи для хранения учебно-наглядных пособий и учебно-методической документации), персональные компьютеры.

Учебно-методическое обеспечение самостоятельной работы обучающихся

По дисциплине «Программирование на Python» предусмотрена аудиторная и внеаудиторная самостоятельная работа обучающихся.

Аудиторная самостоятельная работа студентов предполагает выполнение заданий по лабораторным работам, подготовка к тестированию.

Лабораторная работа №1

Тема «Обработка данных. Регулярные выражения»

Задания:

1. Напишите регулярное выражение для поиска в строке первого слова, которое содержит только буквы (английские или русские). Сохраните строку с выражением в переменной `pattern` и выведите её на экран.
2. Напишите регулярное выражение, которое получает информацию внутри тегов `<title> </title>`, и выведите результат его применения.
3. Напишите регулярное выражение, которое выводит все названия продуктов, начинающихся со слова Масло. Не включайте в название жирность и массу продукта.
4. Напишите регулярное выражение, которое найдет все названия, начинающиеся со слова Молоко. Сохраните результат в переменной `products`. Выведите сначала количество таких продуктов, а на следующей строке - сами продукты.

Лабораторная работа №2

Тема «Библиотеки, реализующие шаблоны веб-страниц»

Задания:

1. Разработать сайт с набором html-страниц. Реализуйте регистрацию и авторизацию пользователей, сохранение данных с формы контактов.
2. Разработайте программного бота, работающего по принципу клиент-серверного взаимодействия. 1. Идея бота: переводчик иностранных слов, бот-анекдотов и пр. (можно предлагать собственные идеи). 2. Разработайте систему команд для общения с ботом. 3. Реализацию необходимо построить с использованием шаблона MVC. 4. Оконный интерфейс tkinter.

Лабораторная работа №3

Тема «Работа с БД»

Задания:

1. Необходимо создать БД по заданной теме. БД должна содержать не менее трех таблиц. Должна присутствовать возможность экспорта/импорта из/в xml-файл, а также не менее трех статистических запросов.
 1. БД «Направления музыки».
 2. БД «Направления танца».
 3. БД «Наземный транспорт».
 4. БД «Вычислительная техника».
 5. БД «Фото-видео техника».

Лабораторная работа №4

Тема «ООП в Python»

Задания:

Создать класс с полями, указанными в индивидуальном задании. Реализовать в классе методы: конструктор по умолчанию; функции обработки данных (1 и 2) функцию формирования строки информации об объекте. Создать класс-потомок с полями, указанными в индивидуальном задании Реализовать в классе-потомке методы: конструктор; функцию обработки данных; функцию формирования строки информации об объекте.

Вариант 1. Класс 1: Дата (три числа): день, месяц, год Определить, является ли год високосным (кратным 4)

Увеличить дату на 5 дней

Класс 2: Человек: ФИО, телефон, дата рождения, Количество дней до дня очередного рождения

Вариант 2. Класс 1: Дата (три числа): день, месяц, год Увеличить год на 1 Уменьшить дату на 2 дня

Класс 2: Работник: ФИО, дата поступления на предприятие Количество лет работы на предприятии

Вариант 3. Класс 1: Дата (три числа): день, месяц, год Определить, совпадают ли номер месяца и число дня Увеличить дату на один месяц

Класс 2: Лекарство: наименование, дата выпуска, фирма Сколько прошло дней от изготовления лекарства

Вариант 4. Класс 1: Время (три числа): часы, минуты, секунды Вычислить количество секунд в указанном времени Увеличить время на 5секунд

Класс 2: Данные по движению поезда: номер поезда, направление, время отправления Количество минут до отправления поезда с указанным номером и введенное время

Вариант 5. Класс 1: Время (три числа): часы, минуты, секунды Вычислить количество полных минут в указанном времени Уменьшить время на 10 минут

Класс 2: Абонент мобильной связи: фамилия, оператор, текущее время Определить, является ли время льготным для абонента (время от 0 до 8 часов)

Лабораторная работа №5

Тема «Создание приложений с GUI.»

Задания:

1. Дана последовательность натуральных чисел, завершающаяся числом 0. Определите, какое наибольшее число подряд идущих элементов этой последовательности равны друг другу. Реализовать графический интерфейс для задачи.

2. Дана последовательность натуральных чисел, завершающаяся числом 0. Определите наибольшую длину монотонного фрагмента последовательности (то есть такого фрагмента, где все элементы либо больше предыдущего, либо меньше). Реализовать графический интерфейс для задачи.

3. Дана последовательность натуральных чисел, завершающаяся числом 0. Определите количество строгих локальных максимумов в этой последовательности. (Элемент последовательности называется локальным максимумом, если он строго больше предыдущего и последующего элемента последовательности. Первый и последний элемент последовательности не являются локальными максимумами.) Реализовать графический интерфейс для задачи.

4. Определите наименьшее расстояние между двумя локальными максимумами последовательности натуральных чисел, завершающейся числом 0. Если в последовательности нет двух локальных максимумов, выведите число 0. Реализовать графический интерфейс для задачи.

5. В списке все элементы различны. Поменяйте местами минимальный и максимальный элемент этого списка. Реализовать графический интерфейс для задачи.

Лабораторная работа № 6

Тема «Работа с API»

1. Получите прогноз погоды на 5 дней для региона Moscow.
2. Получите координаты расположения города Магнитогорска.
3. Создайте карту с метками точек общепита в городе Магнитогорске.
4. Получите данные с отзывами по точкам общепита города Магнитогорска, выведите данные под картой.

7 Оценочные средства для проведения промежуточной аттестации

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
ОПК-1 - способностью решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности		
Знать	<ul style="list-style-type: none"> • синтаксис языка программирования Python; • принципы объектно-ориентированного программирования и их реализацию в Python; • основные стандартные модули, библиотеки языка Python. 	<p><i>Вопросы к зачету:</i></p> <ol style="list-style-type: none"> 1. Области применения Python. 2. Основные алгоритмические конструкции языка Python. 3. Основные встроенные структуры данных в языке Python. Для чего они используются? Есть ли альтернатива для них? 4. Применение регулярных выражений. 5. Создание приложений для взаимодействия с базами данных в языке Python. 6. Технологии парсинга json, xml, html-документов, основные библиотеки Python. 7. Сетевое программирование на Python. 8. Библиотеки веб-разработки Python 9. Визуализация данных средствами библиотеки matplotlib. <p><i>Вопросы к экзамену:</i></p> <ol style="list-style-type: none"> 1. Области применения Python. 2. Основные алгоритмические конструкции языка Python. 3. Основные встроенные структуры данных в языке Python. Для чего они используются? Есть ли альтернатива для них? 4. Применение регулярных выражений. 5. Создание приложений для взаимодействия с базами данных в языке Python. 6. Технологии парсинга json, xml, html-документов, основные библиотеки Python. 7. Сетевое программирование на Python. 8. Библиотеки веб-разработки Python 9. Визуализация данных средствами библиотеки matplotlib. 10. Особенности ООП в Python. 11. Интеграция Python с другими языками программирования. 12. Создание графического интерфейса на Tkinter. 13. Паттерны разработки приложений. 14. Извлечение статистических данных. Работа с API. 15. Folium - Создание интерактивных карт в Python.
Уметь	<ul style="list-style-type: none"> • разрабатывать программы на языке программирования Python, создавая собственные классы, а также использовать классы и модули из 	<p><i>Практические задания:</i></p> <ol style="list-style-type: none"> 1. Дан список целых чисел. Требуется “сжать” его, переместив все ненулевые элементы в левую часть списка, не меняя их порядок, а все нули - в правую часть. Порядок ненулевых элементов

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
	библиотек этого языка	<p>изменять нельзя, дополнительный список использовать нельзя, задачу нужно выполнить за один проход по списку. Распечатайте полученный список. Пример ввода: 4 0 5 0 0 3 2 0 5 0 Пример вывода: 4 5 3 2 5 0 0 0 0 4)</p> <ol style="list-style-type: none"> Составить программу, моделирующую алгоритм группового сжатия данных RLE – замену нескольких подряд стоящих одинаковых элементов списка – парой вида: элемент, количество, а также программу обратного преобразования. В заданном списке слов найти все цепочки слов, в которых каждое слово (кроме первого) отличается от предыдущего ровно одним символом: Пример ввода: [муха, дуб, муза, кус, куб, сук, лупа, луза, вино, лето, кэб, лото] Пример вывода: [[муха, муза, луза, лупа], [дуб, куб, кус, кэб], [сук], [вино], [лето, лото]] Создать класс с полями, указанными в индивидуальном задании. Реализовать в классе методы: конструктор по умолчанию; функции обработки данных (1 и 2) функцию формирования строки информации об объекте. Создать класс-потомок с полями, указанными в индивидуальном задании Реализовать в классе-потомке методы: конструктор; функцию обработки данных; функцию формирования строки информации об объекте. Класс 1: Время (три числа): часы, минуты, секунды Вычислить количество полных минут в указанном времени Уменьшить время на 10 минут Класс 2: Абонент мобильной связи: фамилия, оператор, текущее время Определить, является ли время льготным для абонента (время от 0 до 8 часов)
Владеть	<ul style="list-style-type: none"> навыками разработки консольных приложений в стиле объектно-ориентированного программирования на языке программирования Python; навыками разработки приложений с внешними источниками данных (текстовыми файлами, xml-файлами, базами данных); использовать набор библиотек языка Python для научных вычислений и научной визуализации. 	<p>Проектное задание:</p> <ol style="list-style-type: none"> Разработка приложения на Python с GUI. Разработка программы на Python для Web. Разработка программы для анализа данных на Python.

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация по дисциплине «Программирование на Python» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений, проводится в форме зачета (2 семестр) и экзамена (3 семестр).

Показатели и критерии оценивания на зачет (в соответствии с формируемыми компетенциями и планируемыми результатами обучения):

– на оценку **«зачтено»** – обучающийся показывает высокий уровень сформированности компетенций, т.е. выполняет тренировочные, практические и лабораторные работы в установленные сроки; разрабатывает проектные задания по дисциплине с учетом заявленных требований, владеет терминологическим аппаратом, демонстрирует глубокое теоретическое знание вопроса в области программирования, грамотно определяет логико-структурные связи, обосновывает свое решение и формулирует необходимые выводы.

– на оценку **«не зачтено»** – результат обучения не достигнут, обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач в области программирования.

Экзамен по данной дисциплине проводится в устной форме по экзаменационным билетам, каждый из которых включает 2 теоретических вопроса и одно практическое задание.

Показатели и критерии оценивания экзамена:

– на оценку **«отлично»** (5 баллов) – обучающийся демонстрирует высокий уровень сформированности компетенций, всестороннее, систематическое и глубокое знание учебного материала, свободно выполняет практические задания, свободно оперирует знаниями, умениями, применяет их в ситуациях повышенной сложности.

– на оценку **«хорошо»** (4 балла) – обучающийся демонстрирует средний уровень сформированности компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

– на оценку **«удовлетворительно»** (3 балла) – обучающийся демонстрирует пороговый уровень сформированности компетенций: в ходе контрольных мероприятий допускаются ошибки, проявляется отсутствие отдельных знаний, умений, навыков, обучающийся испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

– на оценку **«неудовлетворительно»** (2 балла) – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.

– на оценку **«неудовлетворительно»** (1 балл) – обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач.

Методические по выполнению индивидуальных домашних заданий

Цель методических рекомендаций - обеспечить студенту оптимальную организацию процесса изучения дисциплины, а также выполнения различных форм самостоятельной работы.

Студентам необходимо ознакомиться: с содержанием рабочей программы дисциплины (далее - РПД), с целями и задачами дисциплины, ее связями с другими дисциплинами образовательной программы, методическими разработками по данной дисциплине, имеющимся на образовательном портале и сайте кафедры, с графиком консультаций преподавателей кафедры.

Инструкции по работе с Anaconda

Anaconda — это дистрибутивы Python и R. Он предоставляет все необходимое для решения задач по анализу и обработке данных (с применимостью к Python).

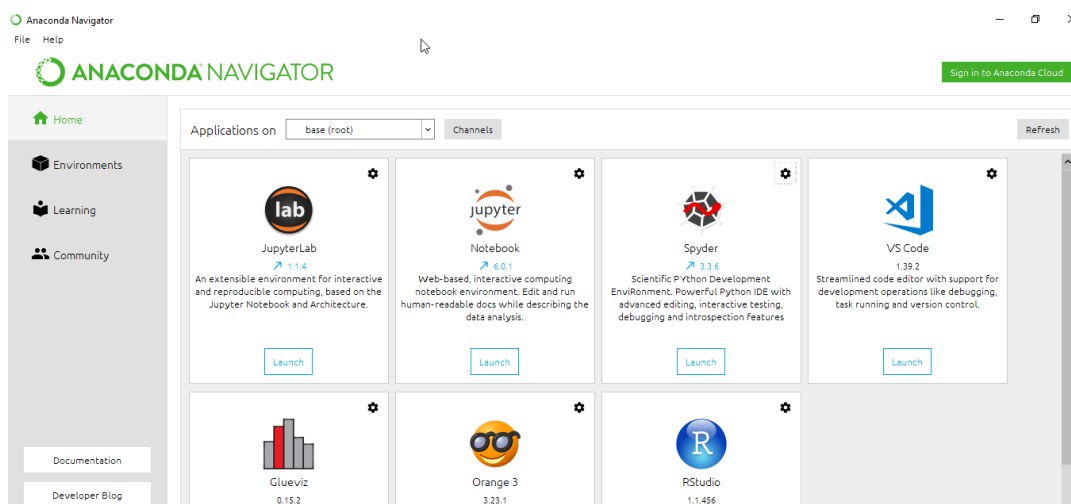
Anaconda — это набор бинарных систем, включающий в себя Scipy, Numpy, Pandas и их зависимости.

Scipy — это пакет статистического анализа.

Numpy — это пакет числовых вычислений.

Pandas — уровень абстракции данных для объединения и преобразования данных.

Anaconda Navigator — это графический интерфейс пользователя на рабочем столе (GUI), включенный в дистрибутив Anaconda, который позволяет запускать приложения и легко управлять пакетами, средами и каналами conda без использования команд командной строки. Навигатор может искать пакеты в Anaconda Cloud или в локальном репозитории Anaconda. Он доступен для Windows, MacOS и Linux.



JupyterLab — это интерактивная среда разработки для работы с блокнотами, кодом и данными.

Jupyter Notebook — удобный инструмент для создания красивых аналитических отчетов, позволяет хранить вместе код, изображения, комментарии, формулы и графики. Работа ведется в браузере.

Spyder — интерактивной IDE для научных расчетов на языке Python. Данная IDE позволяет писать, редактировать и тестировать код. Spyder предлагает просмотр и редактирование переменных с помощью GUI, динамическую интроспекцию кода, нахождение ошибок на лету и многое другое. Также, по необходимости, можно интегрировать Anaconda с другими Python IDE, включая PyCharm и Atom.

VS Code — это оптимизированный редактор кода с поддержкой таких операций разработки, как отладка, запуск задач и контроль версий.

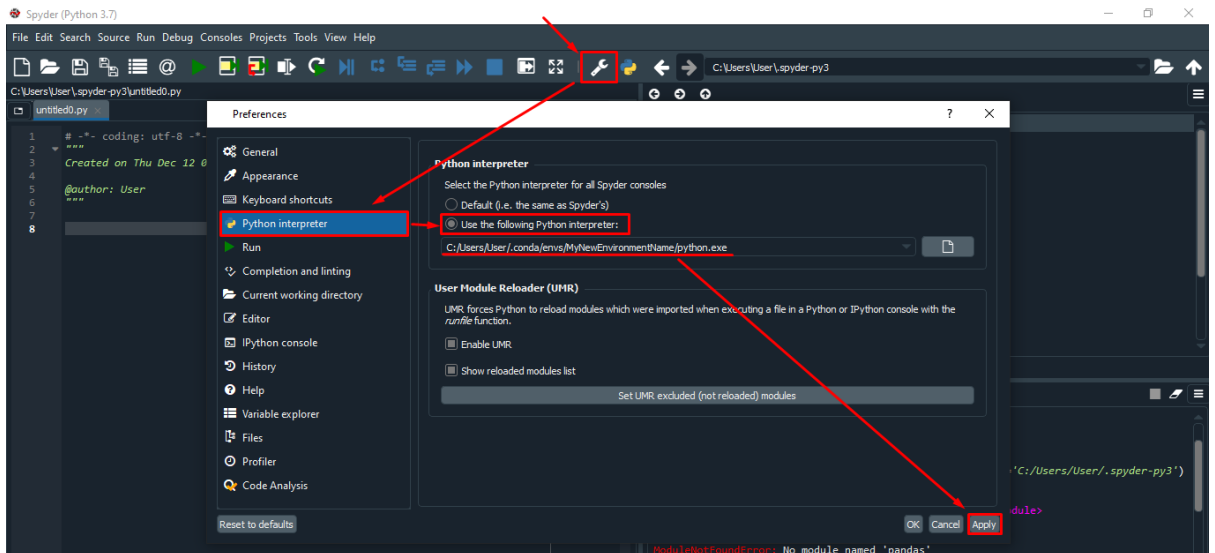
Glueviz — используется для визуализации многомерных данных в файлах. Он исследует отношения внутри и между связанными наборами данных.

Orange 3 — это основанная на компонентах структура интеллектуального анализа данных. Это может быть использовано для визуализации данных и анализа данных. Рабочие процессы в Orange 3 очень интерактивны и предоставляют большой набор инструментов.

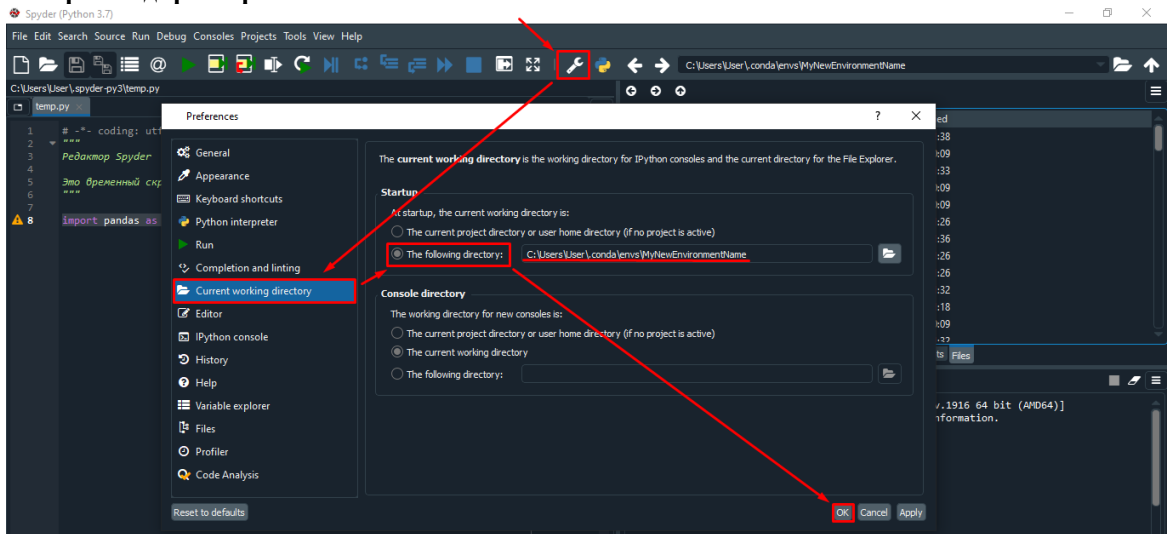
RStudio — это набор интегрированных инструментов, предназначенных для повышения продуктивности работы с R. Он включает в себя основы R и Notebooks.

Настройка среды для Spyder

1. Настройка интерпретатора

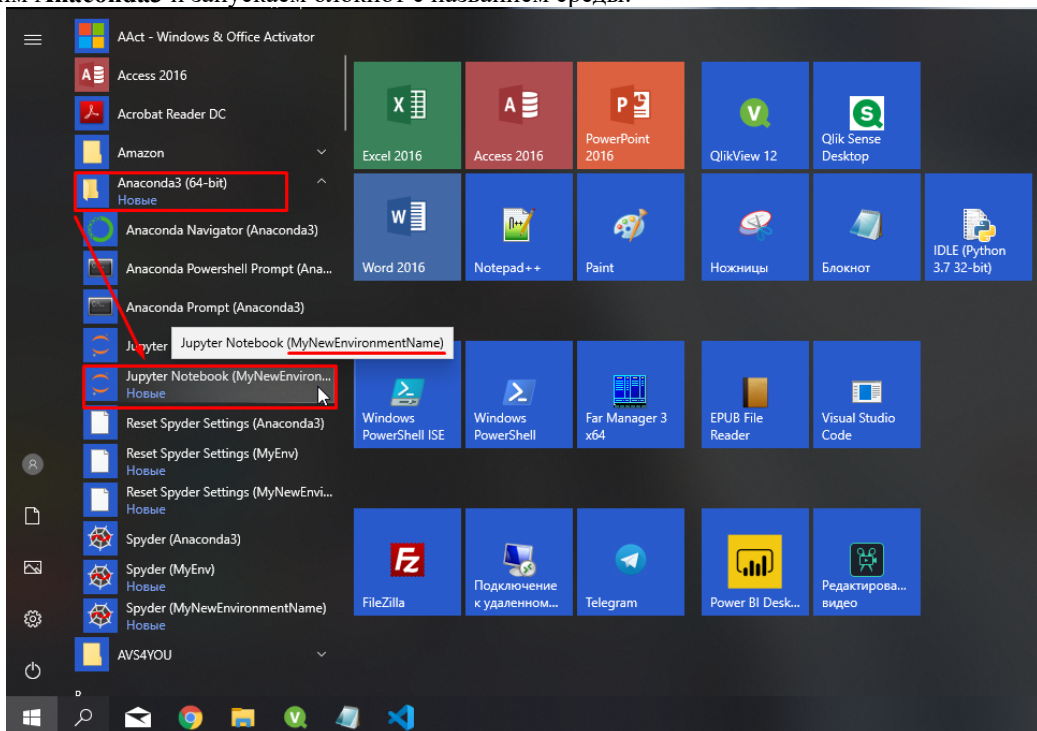


Настройка директории



Как открыть Jupyter Notebook в новой среде MyNewEnvironmentName

Для того, чтобы запустить **Jupyter Notebook** в созданной среде **MyNewEnvironmentName**, в пуске находим **Anaconda3** и запускаем блокнот с названием среды:

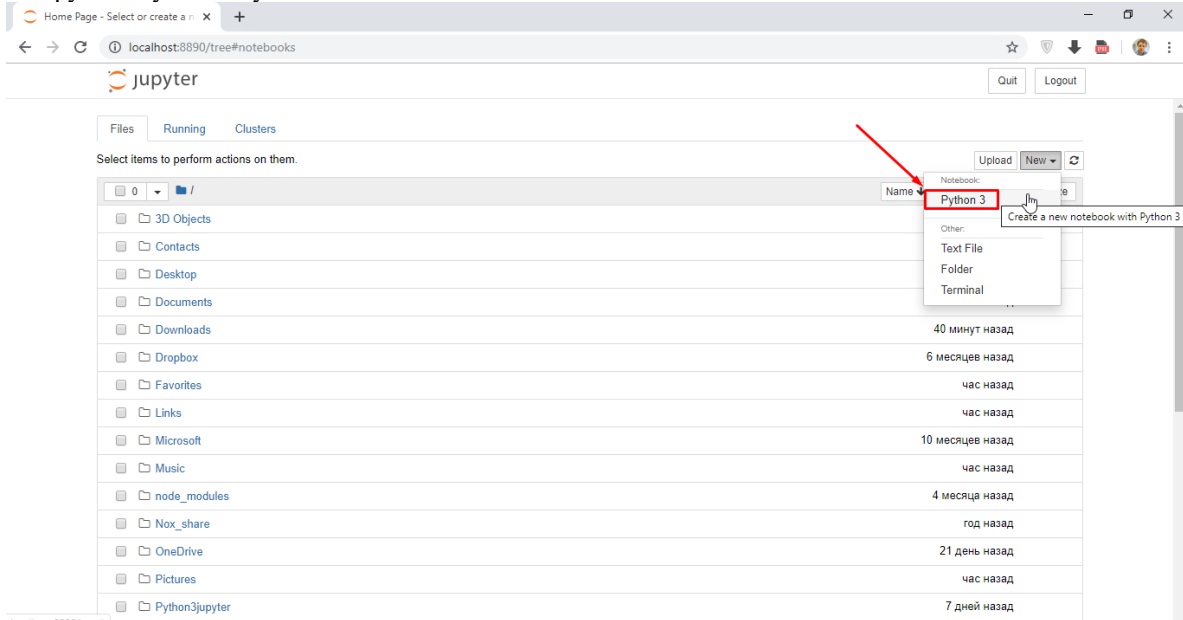


Появится консольное окошко — это движок Jupyter Notebook, который работает в фоновом режиме:


```
Jupyter Notebook (MyNewEnvironmentName)
[I 23:57:47.710 NotebookApp] The port 8888 is already in use, trying another port.
[I 23:57:47.710 NotebookApp] The port 8889 is already in use, trying another port.
[I 23:57:47.757 NotebookApp] Serving notebooks from local directory: C:\Users\User
[I 23:57:47.757 NotebookApp] The Jupyter Notebook is running at:
[I 23:57:47.757 NotebookApp] http://localhost:8890/?token=57cb326d88828682f9384226df11d58fff936a7547108cc6
[I 23:57:47.757 NotebookApp] or http://127.0.0.1:8890/?token=57cb326d88828682f9384226df11d58fff936a7547108cc6
[I 23:57:47.757 NotebookApp] Use Control-C to stop this server and shut down all kernels (twice to skip confirmation).
[C 23:57:47.969 NotebookApp]

To access the notebook, open this file in a browser:
file:///C:/Users/User/AppData/Roaming/jupyter/runtime/nbserver-9088-open.html
Or copy and paste one of these URLs:
http://localhost:8890/?token=57cb326d88828682f9384226df11d58fff936a7547108cc6
or http://127.0.0.1:8890/?token=57cb326d88828682f9384226df11d58fff936a7547108cc6
```

В Jupyter запускаем Python 3:



Для того, чтобы убедиться в какой среде мы работаем, можно вбить ряд команд (ниже приведен текст этих команд для Python 3):

```
Jupyter MyNewEnvironmentName_Python_Example Last Checkpoint: несколько секунд назад (unsaved changes)
Python 3

In [15]: import sys
         print(sys.version)
         print(sys.base_prefix)

3.7.5 (default, Oct 31 2019, 15:18:51) [MSC v.1916 64 bit (AMD64)]
C:\Users\User\.conda\envs\MyNewEnvironmentName

In [16]: print('\n'.join(sys.modules.keys()))

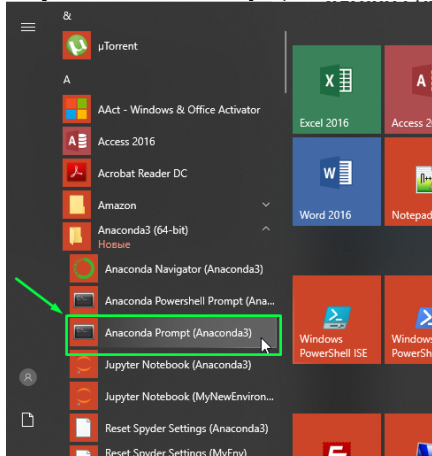
sys
builtins
_frozen_importlib
_imp
_itertools
_warnings
_weakref
zipimport
_frozen_importlib_external
_io
marshal
nt
winreg
encodings
codecs
_codecs
encodings.aliases
encodings.utf_8
signal
```

Узнать среду, в которой работает Jupyter Notebook:

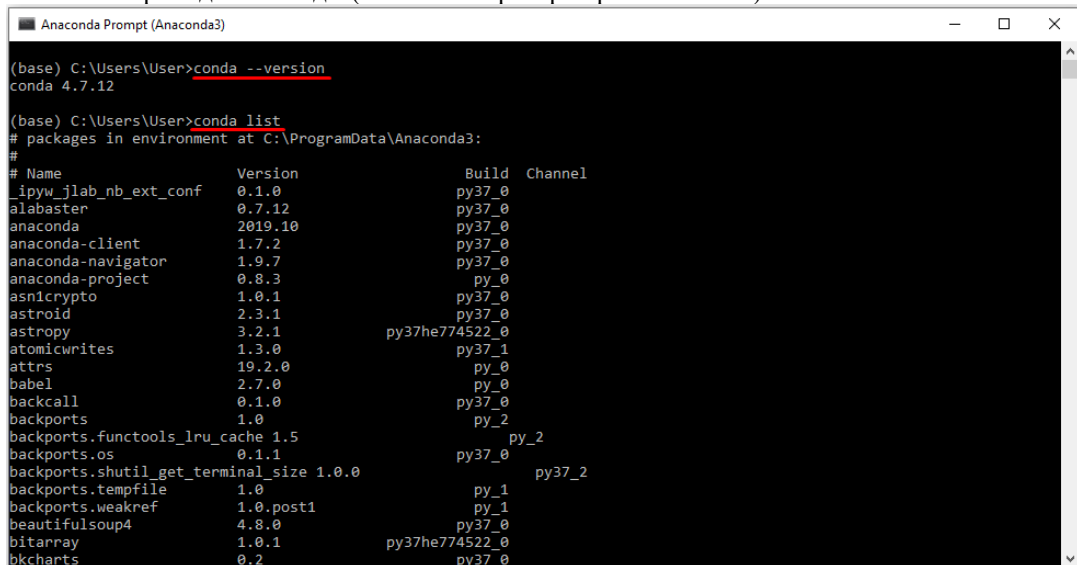
```
import sys
print(sys.version)
print(sys.base_prefix)
Получить список модулей, доступных в Env:
print('\n'.join(sys.modules.keys()))
```

Anaconda3 Prompt cmd Conda Command — Запуск команд через консоль

Для того, чтобы использовать команды conda через командную строку (cmd), необходимо запустить программу **Anaconda Prompt (Anaconda3)**

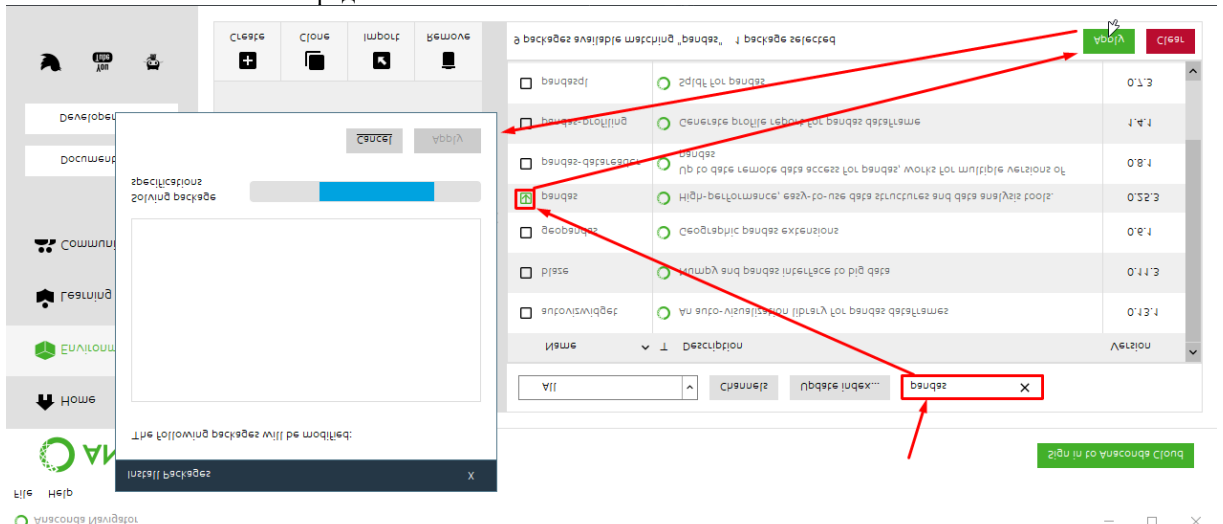


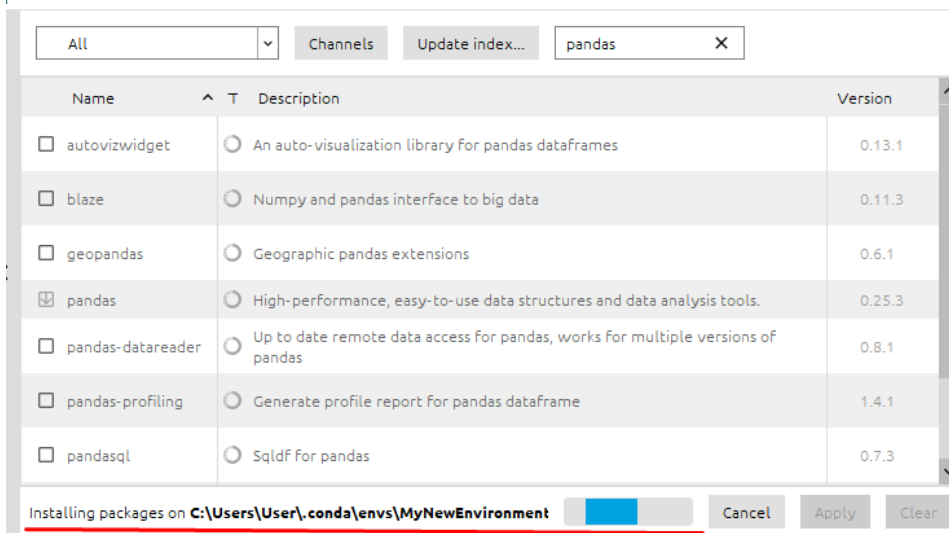
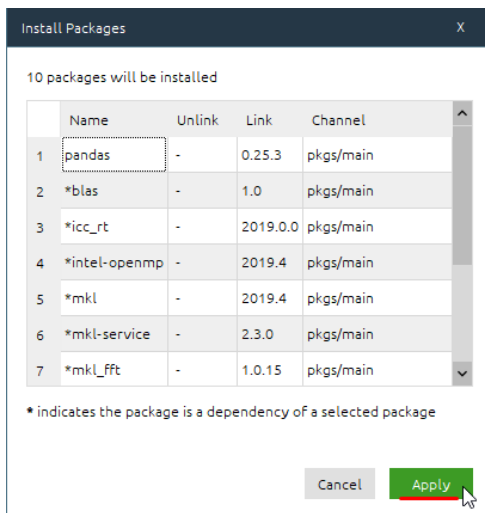
Можете набрать две команды (в качестве проверки работы conda):



Установка новой библиотеки (пакета) в среду

Пакеты управляются отдельно для каждой среды. Изменения, которые вы вносите в пакеты, применяются только к активной среде.





Исполняемые файлы в среде Conda

python.exe — исполняемый файл Python для приложений командной строки. Так, например, если вы находитесь в каталоге Example App, вы можете выполнить его: `python.exe exampleapp.py`

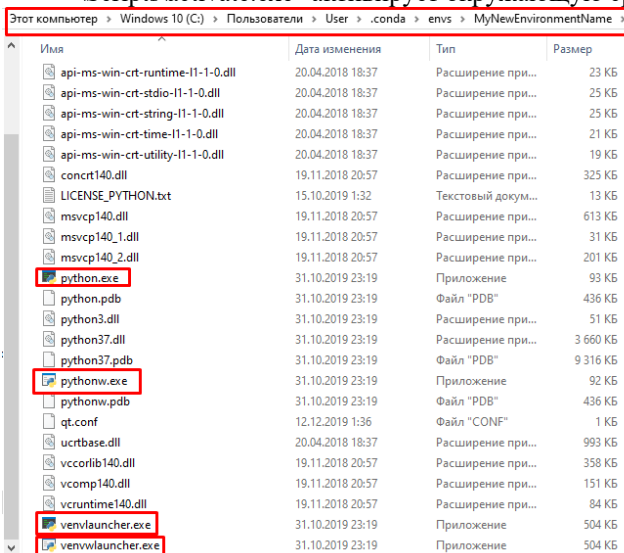
pythonw.exe — исполняемый файл Python для приложений с графическим интерфейсом или приложений без интерфейса пользователя

venvlauncher.exe —

venvwlauncher.exe —

`\Scripts` — исполняемые файлы, являющиеся частью установленных пакетов. После активации среды этот каталог добавляется в системный путь, поэтому исполняемые файлы становятся доступными без их полного пути.

`\Scripts\activate.exe` - активирует окружающую среду



Источники:

- <https://protostar.space/why-you-need-python-environments-and-how-to-manage-them-with-conda>
- https://kapeli.com/cheat_sheets/Conda.docset/Contents/Resources/Documents/index

Библиотека Tkinter

Tkinter – это кроссплатформенная библиотека для разработки графического интерфейса на языке Python . Tkinter расшифровывается как Tk interface, и является интерфейсом к [Tcl/Tk](#). Tkinter входит в стандартный дистрибутив Python

Tkinter (от англ. tk interface) - это графическая библиотека, позволяющая создавать программы с оконным интерфейсом и может быть использована в большинстве распространённых операционных систем (Windows, Linux, Mac OS X и др.).

Начиная с версии python-3.0 библиотека переименована в соответствии с PEP 8 в tkinter (с маленькой буквы). Импортируется она как и любая другая библиотека:

```
# для версии python 2.7 и ниже
```

```
import Tkinter
```

```
# для версии python 3.0 и выше
```

```
import tkinter
```

В Tkinter визуальные контроллы называются виджетами (widget, от англ. window gadget) - стандартизированный компонент графического интерфейса, с которым взаимодействует пользователь. К ним относятся: Button – кнопки с управлением , Label - виджет, предназначенный для отображения какой-либо надписи без возможности редактирования пользователем. Имеет те же свойства, что и кнопки, List, radioButton и т.д.

Класс Tk Создание простых окон Свойства окон

Tk является базовым классом любого Tkinter приложения. При создании объекта этого класса запускается интерпретатор tcl/tk и **создаётся базовое окно приложения**.

Tkinter является событийно-ориентированной библиотекой. В приложениях такого типа имеется главный цикл обработки событий. В Tkinter такой цикл запускается методом mainloop. Для явного выхода из интерпретатора и завершения цикла обработки событий используется метод quit.

Таким образом минимальное приложение на Tkinter будет таким:

Пример1 Создание окна через root.mainloop()

```
from tkinter import *
```

```
root = Tk()
```

```
root.mainloop()
```

Пример 1 Свойства окна

```
from tkinter import *
```

```
def window_deleted():
```

```
print ('Окно закрыто')
```

```
root.quit() # явное указание на выход из программы
```

```
root=Tk()
```

```
root.title('Пример приложения')
```

```
root.geometry('500x400+300+200')
```

```
# ширина=500, высота=400, x=300, y=200
```

```
root.protocol('WM_DELETE_WINDOW', window_deleted)
```

```
# обработчик закрытия окна
```

```
root.resizable(True, False) # размер окна может быть изменён только по горизонтали
```

```
root.mainloop()
```

В приложении можно использовать несколько интерпретаторов tcl/tk. Так как после вызова метода mainloop дальнейшие команды python исполняться не будут до выхода из цикла обработки событий, необходимо метод mainloop всех интерпретаторов кроме последнего осуществлять в фоновом режиме.

Пример 2. Запуск двух интерпретаторов:

```
from tkinter import *
```

```
root1 = Tk()
```

```
root2 = Tk()
```

```
# первый цикл запускаем в фоне
```

```
root1.after(500, root1.mainloop)
```

```
root2.mainloop()
```

При использовании двух и более интерпретаторов необходимо следить, чтобы объекты, созданные в одном интерпретаторе, использовались только в нём. Например, изображение, созданное в первом интерпретаторе, может быть использовано много раз в этом же интерпретаторе, но не может быть использовано в других интерпретаторах.

Пример 3. Создание окна через window.mainloop()

```
from tkinter import *
```

```
window = Tk()
```

```
window.title("Добро пожаловать
```

```
в приложение PythonRu")
```

```
window.mainloop()
```