ТЕХНОЛОГИЧЕСКАЯ КАРТА ЗАНЯТИЯ

Тема занятия: Введение в язык программирования Python.

Аннотация к занятию: данный урок является вводным в курс. На данном уроке обучающиеся знакомятся с понятием искусственный интеллект, языком программирования Python и средой разработки Anaconda. В первой части урока они узнают об искусственном интеллекте. Во второй части познакомятся со структурой этого курса. В третьей — познакомятся со средой разработки и её установкой. В заключительной части создается первая программа на Python.

Цель занятия: формирование у учеников представление об ИИ, о языке программирования Python.

Задачи занятия:

- познакомить обучающихся с ИИ и языком программирования Python;
- научить устанавливать и открывать Jupyter Notebook;
- создать и запустить первую программу;
- сформировать понятие «аргументы функций».

Ход занятия

Этап занятия	Время	Деятельность педагога	Комментарии, рекомендации для педагогов
Организационный этап	2 мин.	Добрый день! Сегодня у нас будет интересное занятие.	Педагог создает благоприятный эмоциональный фон перед началом занятия.
Постановка цели и задач занятия. Мотивация учебной деятельности обучающихся	10 мин.	**Вопросы для обсуждения:** Слышали ли вы об искусственном интеллекте? Интересна ли вам эта тема? **Возможные ответы учеников:** Да, эта тема обсуждается в СМИ. Тема очень интересная! **Вопросы для обсуждения:** Какие этапы могут быть в работе с переменными? **Возможные ответы учеников:** Создание переменных, оперирование с переменными, вывод и вывод. **Вопросы для обсуждения:** Какой язык программирования обычно используется для работы с ИИ? Почему?	Озвучивается тема занятия, цель занятия. Обсуждаются ожидания от кружка.

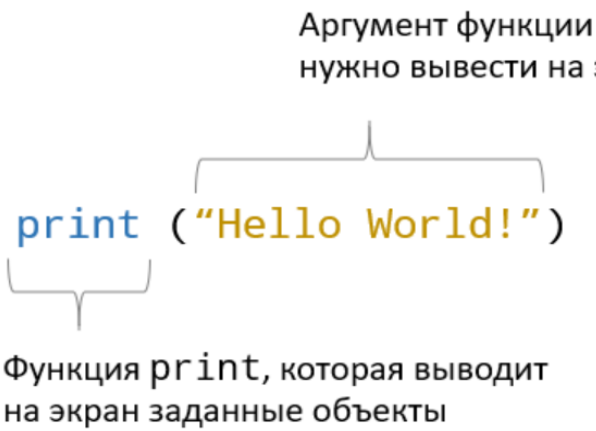
		Возможные ответы учеников: Это Python. Потому что он популярный /лёгок в освоении /имеет много возможностей. Озвучивается план, в соответствии с которым вначале нужно разобраться, что такое искусственный интеллект в целом, а потом обсудить, что мы будем изучать в течение курса.	
Изучение нового материала	25 мин.	Введение в искусственный интеллект Обсуждается искусственный интеллект, его важность и применимость. Можно запустить видео: https://youtu.be/EA-35ALskEY Длительность видео: 14,5 минут. Если с интернетом есть сложности, то видео можно заранее скачать по ссылке https://www.ssyoutube.com/watch?v=EA-35ALskEY Вместо просмотра видео можно обсудить важность темы ИИ на основе последних новостях об ИИ из СМИ. Искусственный интеллект — объёмная наука. Теоретический курс выглядел бы так, как вы видите на слайде. Понять эту структуру сложно. Важно проверять услышанное на практике с помощью программирования. За применение моделей искусственного интеллекта отвечает машинное обучение — о нём вы узнаете в этом курсе. Выделяют десяток наук, которые помогают разобраться в машинном обучении. Мы остановимся на двух: алгебре и программировании.	Дети работают в группах. Изучают материал, готовят ответы. Подготовка школьников к выполнению групповой работы: распределение по группам, ознакомление с последовательностью выполнения, текущий инструктаж, установка на сотрудничество

		Именно они необходимы для полного погружения в тему. Не удивляйтесь тому, что первые модули пересказывают школьные темы: они нужны для подготовки. Математика поможет понять принцип работы моделей машинного обучения. Программирование — закрепить навыки и самостоятельно поработать с искусственным интеллектом. Главным языком программирования ИИ стал Python — о нём и пойдёт речь в этом модуле. Говорить о преимуществах языка Python можно долго. Выделим главные. ● Python — отличный язык для знакомства с программированием. После него будет проще освоить другие языки. ● Python очень популярен. Если у вас появятся вопросы, большое сообщество пользователей поможет советом. ● Python используют в обработке больших данных и машинном обучении. Это первый шаг к пониманию принципов ИИ. ● На Python написано множество продуктов с открытым исходным кодом. Пока программисты на других языках пишут всё самостоятельно, вы можете использовать готовые решения. Подводя итог, отмечу: изучение Python в 2022 году — хорошее решение для человека, который только начинает погружаться в программирование. На этом уроке мы познакомимся с Python. Начнём с его главной силы — функций! Их примеры я продемонстрирую в коде.	

		Функция — это минимальная единица любой программы на Python, как клетка в организме. Каждая функция уникальна и умеет делать что-то своё. Например, извлекать корень, выводить текст на экран или даже двигать курсор. Выделяют два вида функций: • встроенные — их можно использовать сразу после того, как вы впервые скачали Python, • рукотворные — функции, которые вы создали сами для своих задач. Мы научимся создавать уникальные функции и использовать функции других программистов из интернета	
Закрепление изученного материала	40 мин.	Установка Anakonda Видео: https://youtu.be/QFnb_OvkaIQ Подробная инструкция: https://docs.google.com/document/d/1gJZGu43sO1RQpSqgyLtr3_oxEB3mQOmSjki_vywTZC0/edit?usp=sharing Примечание. Крайне желательно установить Anakonda на школьные компьютеры заранее, чтобы не было проблем. Если возникают сложности с установкой или что-то пошло не так, то есть резервный вариант. На занятиях можно воспользоваться сервисом Google Colab: https://colab.research.google.com/	Ссылка на колаб: https://colab.research.google.com/drive/1y9MMHhOhurNBS3j_f0JeKksXcKvpp_bl?usp=sharing Видео-инструкция: https://youtu.be/5XRuXLsBZ-o

		Важно: для работы и учителю, и ученикам потребуются учётные записи Google. Если учётные записи есть, то для организации процесса можно использовать Google Класс: https://classroom.google.com/h Плейлист по работе с Google Классом: https://youtube.com/playlist?list=PLGhq4RhYque4XOQr1CfyA6PLRUNVC8Hgo Чтобы говорить с компьютером на понятном ему языке, придётся выучить пару сотен функций. Не пугайтесь: после этого курса вы будете знать пару десятков — этого хватит на первое время. Отличительная черта функций — круглые скобки в конце. Это встречается и в других языках программирования. Если вы откроете код на любом языке и заметите круглые скобки, перед вами наверняка окажется функция. Помимо скобок, функции включают ключевые слова — это добавляет им уникальности. Рассмотрим нашу первую функцию — print(). Она состоит из: ● ключевого слова print, ● и, неожиданно, скобок. Print() — встроенная в Python функция, её можно использовать сразу. Всего таких функций 66. Создадим нашу первую программу — традиционный вывод фразы "Hello, World!".	

		Да, print() умеет выводить текст на экран. Но чтобы вывести текст, нужно как-то связать его с функцией . Для этого просто запишем нужную строчку внутри скобок и запустим программу. print(“Hello world!”) Запишем текст на русском и выведем строку длиннее. print(“Скажи-ка, дядя, ведь недаром Москва, спаленная пожаром”) Даже такая строчка — не проблема для print(). Думаю, вы заметили, что текст внутри скобок функции светится зеленым, а по краям у него кавычки. Это не случайно. Функции в программировании тоже состоят из букв. Чтобы Python понимал, что речь идёт не о функциях, а о тексте, текст заключают в кавычки (“ ”). Если этого не сделать, программа выдаст ошибку. Например: print(Скажи-ка, дядя, ведь недаром Москва, спаленная пожаром) Функция — это минимальная единица любой программы на Python, как клетка в организме. Давайте ещё раз рассмотрим, из чего она состоит:	Все базовые функции языка Python вы увидите по ссылке.



Аргументы функций

Введение
Значения, которые мы посылаем на вход функции, будь то текст или числа, называются аргументами функции. В каждой функции аргументы выполняют свои уникальные роли: где-то выступают в качестве числителя, где-то в качестве диапазона скорости робота.

При работе с функциями обратите внимание на допустимые значения аргументов. Если print() умеет работать со всеми типами данных в аргументах, то функция извлечения корня из числа sqrt() работает только с числами.

Ссылка на Google Colab:

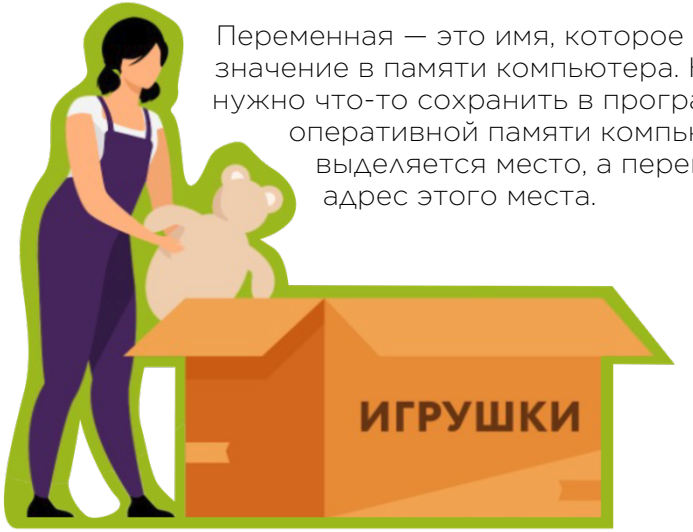
https://colab.research.google.com/drive/1Cp7SXFWUFNCO0XS1NP3c9OllBrnpZTIz?usp=sharing

		Чтобы избежать ошибок при написании функции, нужно изучить её документацию. Посмотрим на документацию функции print(): `print(*items, sep=' ', end='\n', file=sys.stdout, flush=False)` Оказывается, у нее есть несколько параметров вывода — мы их пока опустим. Нам важна вот эта часть — *items. Она говорит о том, что функция print() принимает на вход неограниченное число аргументов: это может быть как одна строка, так и целая книга «Война и мир»! Чтобы задать функции несколько аргументов, их достаточно разделить запятой внутри скобок. Возможность обрабатывать неограниченное число аргументов позволяет комбинировать в функции разные типы данных. print(“440*30/168 - 68 =”, 440*30/168 - 68) Функция input() Познакомимся с главным напарником print() — функцией input(). input() — это функция чтения данных с клавиатуры. Если раньше мы только выводили заранее заданную информацию на экран, то при помощи input() мы можем запросить её у пользователя в любой момент. Input() позволяет программе считывать данные и использовать их, а не просто выводить на экран заданные значения.	

		Аргументы функции input() будут выведены на экран в качестве подсказки. После этого вам предложат ввести данные в программу. Попробуем считать имя пользователя. Чтобы нам было понятно, что требует программа, в качестве аргумента запишем «Введи своё имя». Запустим программу и посмотрим, что произойдет дальше: input(”Введи свое имя”) Когда исполнение программы доходит до команды input(), ее аргумент — фраза «Введи своё имя» — выводится на экран, а ход программы останавливается до тех пор, пока пользователь не введёт данные через командную строку. Вводить данные нужно с клавиатуры, а в конце нажать клавишу Enter. После того, как мы ввели имя, программа закончила своё выполнение. Логично, ведь у input() была задача принять значение, про другие действия речи не шло. Нужно придумать, как повторно использовать считанное значение, в нашем случае — имя пользователя. Попробуем вложить одну функцию в другую, т.е. поместить функцию input() внутрь print(). Как это будет работать? Команда input() умеет принимать данные. После выполнения команды на её месте окажутся данные, которые ввёл пользователь. print(“Привет”,input(”Введи свое имя: ”))	

Если не сохранить результат работы функции, он будет безвозвратно утерян. Если мы его сохраним, то сможем использовать в дальнейшем.

На такой случай в программировании существуют переменные. Что это такое и как они работают?



Переменная — это имя, которое ссылается на значение в памяти компьютера. Когда нам нужно что-то сохранить в программе, в оперативной памяти компьютера выделяется место, а переменная хранит адрес этого места.

Проще говоря, переменная — это коробочка, в которую можно записать одно значение, и в нужный момент считать его.

Чтобы создать переменную, ей нужно дать имя. Хороший тон — создавать переменные с «говорящим» названием. Создадим переменную с именем number и сохраним в ней число 9.

```
number = 9
```

		Для связи переменной и значения, которое она будет хранить, используют оператор «=». Запись значения в переменную выглядит следующим образом: имя_переменной = значение или выражение, результат которого мы хотим хранить в переменной. Посмотрим, на что способны переменные. Они отлично чувствуют себя в качестве аргументов функции: number = 9 print(number) Во время вызова функций или математических операций переменные заменяют себя своими значениями. В отличии от print(), input() возвращает значение, введённое пользователем в программу. Почему бы не сохранить его в переменную? name = input(“Введи свое имя: ”) secondname = input(“Введи свою фамилию: ”) print(“Привет”,name ,secondname ,”неплохой день для программирования ?”) Кроме того, переменные отлично ведут себя в качестве элементов уравнений. Над ними можно совершать все известные математические операции: a = 5 b = 12	

		print(a+b) Кстати, в программировании существует целых три вида операции деления! Детально мы рассмотрим их в следующем видео. При создании переменной важно помнить о правилах оформления имён. Они: ● могут содержать только цифры и латинские буквы, ● не должны начинаться с цифры, ● не должны повторять уже встроенные в Python имена, например, print, str, int и т.д. Важно! Python чувствителен к регистру букв при написании кода. Это значит, что переменные Text, text и TEXT — абсолютно разные для интерпретатора кода. Помимо этих правил, хорошим тоном считается не слишком длинное и понятное имя переменной. Из-за того, что вы создали переменные a,b,c,d, программа не перестает работать, но через пару часов вы и сами не вспомните, что хранится в этих переменных. Заключение На этом уроке мы начали знакомство с языком программирования Python, написали первую программу, рассмотрели функцию input() с интересным функционалом: она умеет считывать значения от пользователя и передавать их в	

		программу. Это даёт нам возможность работать с запросами пользователя. Помимо этого, мы рассмотрели новую структуру — переменные. Цель переменных в программировании — хранить в себе значения. Благодаря им мы можем хранить ответ пользователя на функцию input() или результат работы некоторого выражения. В следующем уроке мы познакомимся с типами данных, которые могут находиться в переменных.	
Этап подведения итогов занятия (рефлексия)	8 мин.	Вопросы для обсуждения • Что больше всего вдохновляет в этой теме? • Есть ли что-то, что кажется непонятным?	Педагог способствует размышлению обучающихся над вопросами.
Информация о домашнем задании, инструктаж по его применению	5 мин.	В этом домашнем задании вам предстоит освоиться с простейшим синтаксисом библиотеки Python: операторами ввода-вывода, простыми математическими операторами, логическими выражениями, условным оператором. Домашнее задание представляет из себя решение нескольких задач по программированию. Программы проверяются в автоматическом режиме на платформе Stepik или на платформе Академии искуственного интеллекта.	

Рекомендуемые ресурсы для дополнительного изучения:

1. ПИТОНТЬЮТОР. [Электронный ресурс] – Режим доступа: http://pythontutor.ru/.
2. Онлайн-курс «Поколение Python»: курс для начинающих. [Электронный ресурс] – Режим доступа: https://stepik.org/course/58852/syllabus.
3. Онлайн-игра на программирование CodeCombat:. [Электронный ресурс] – Режим доступа: https://codecombat.com/.
4. Прямая ссылка на начало игры. [Электронный ресурс] – Режим доступа: https://codecombat.com/play.