ТЕХНОЛОГИЧЕСКАЯ КАРТА ЗАНЯТИЯ

Тема занятия: Переменные и операции над ними.

Аннотация к занятию: на данном уроке обучающиеся знакомятся с переменными в Python, типами переменных и приведением типов. В первой части урока они учатся создавать и применять переменные разных типов. Во второй части урока обучающиеся пробуют преобразовать данные одного типа в другой. Обучающиеся изучат операции над переменным в Python, разберут основные математические операции и операции с переменными в Python, изучат ввод данных с клавиатуры.

Цель занятия: знакомство обучающихся с понятием переменной, основными типами переменных, способами объявления переменных в языке программирования Python.

Задачи занятия:

- научить создавать (инициализировать) переменные;
- познакомить с основными типами переменных в Python;
- узнать, как преобразовать один тип переменных в другой;
- научить вводить данные с клавиатуры для программы в Python;
- познакомить с основными математическими и логическими операциями с переменами в Python;
- узнать, какие типы простых математических задач решаются с помощью операций над переменными в Python.

Ход занятия

Этап занятия	Время	Деятельность педагога	Комментарии, рекомендации для педагогов
Организационный этап	2 мин.	Добрый день! Сегодня у нас будет интересное занятие!	
Постановка цели и задач занятия. Мотивация учебной деятельности обучающихся	10 мин.	**Вопрос для обсуждения** Есть ли необходимость хранить данные? **Возможные ответы учеников:** Да, это важно. Как исходные данные на начальном этапе, так и промежуточные. **Вопрос для обсуждения** Какие этапы при работе с переменными могут быть? **Возможные ответы учеников** Создание переменных, оперирование с переменными, вывод и вывод. **Вопросы для обсуждения** Какие типы данных нам потребуются? **Возможные ответы учеников** Целые числа, дробные числа, строки и другие типы.	Обсуждается необходимость существования переменных для решения задач анализа данных и машинного обучения. Задаётся вопрос о переменных в других языках программирования, если кто-то из присутствующих владеет другими языками Программирования.

		Озвучивается план, в соответствии с которым нужно разобраться, как создавать переменные, как присваивать в них некие значения, какие операции могут потребоваться при работе с переменными, как вводить данные от пользователя. Если публика что-то слышала о переменных в Python, можно обсудить, есть ли разница между переменными в этом языке и в других языках программирования.	
Изучение нового материала	35 мин.	В процессе занятия происходит постепенный спуск внизу по блокноту, где обязательно запускаются все ячейки. **Порядок тем для изучения:** ● Переменная как контейнер. Каким может быть имя переменной ● Создание (инициализация) переменной ● Основные типы переменных: int, float, str ● Преобразование одного типа данных в другой. Проблемы при приведении типов Создание переменных "=" — оператор присвоения значения. Как присвоить значение: имя_переменной = значение или выражение, результат которого хотим хранить в переменной Правила именования переменных Есть несколько правил при определении переменных: ● имена переменных не должны повторять уже встроенные в Python имена, например, print, str, in и другие;	Отдельно стоит обратить внимание, что в конце есть несколько задач, которые можно успеть решить на уроке, а можно воспринимать как домашнее задание. Ссылка на Google Colab: https://colab.research.google.com/drive/1rTaNcFn_H01-9i0m1iaz7GsynVP67SXW?usp=sharing Ссылка на Google Colab:

		• имена переменных могут содержать только буквы, цифры и нижнее подчёркивание; • имена переменных не должны начинаться с цифры; • хорошим тоном считается, чтобы имена переменных выражали суть переменной и не были слишком длинными. **Важно!** Python чувствителен к регистру букв при написании кода: это значит, что переменные Text, text и TEXT— абсолютно разные для интерпретатора кода. **Типы переменных** int — тип для хранения целых чисел. float — тип для хранения вещественных чисел. str (string) — тип для хранения строковых (текстовых) значений. **Важно!** Вы можете управлять тем, как Python интерпретирует число, — как целое или как вещественное. Например, если вы хотите, чтобы students_count имело тип float, то можете задать ему значение с точкой: students_count = 35.0 **Приведение типов** type() — возвращает тип аргумента, указанного в скобках. str() — возвращает аргумент, переданный в скобках, в виде строки (текста). int() — возвращает аргумент, переданный в скобках, в виде целого числа.	https://colab.research.google.com/drive/1jHE12HdFJ9u3mbN6d1naaHu3T9q8EuM?usp=sharing

float() — возвращает аргумент, переданный в скобках, в виде вещественного числа.

Важно!
Не всякую строку можно перевести в число. Если вы попытаетесь перевести
строку, которая содержит в себе буквы, это приведёт к ошибке.

Ранее мы познакомились с типами данных переменных и узнали, что над строчным типом можно применять всего два вида операций. Это:

- умножение на число,
- склеивание строк.

Сейчас мы рассмотрим операции над численным типом данных и узнаем, зачем программистам три вида деления.

К численному типу, будь то целочисленное или вещественное значение, можно применять 4 алгебраических оператора:

Что нужно сделать?	Оператор в Python	Пример
Сложить	+	2 + 3 = 5
Вычесть	–	5 - 2 = 3
Умножить	*	2 * 3 = 6
Разделить	/	6 / 3 = 2

Конечно, стоит упомянуть и об операции возведения числа в степень. Программисты — весьма практичные люди. Если операция умножения в Python — это звездочка (*), то операция степени — это две звездочки (**). Удобно, не правда ли?

		Пример возведения в степень: 2**3 = 8 Операцию извлечения корня можно представить, как взятие числа в степени меньше единицы. Получается, двумя звездочками можно извлекать корни. В программировании работают все математические законы, которые вы изучали на школьных уроках. Правда, на 5 стандартных математических операциях разработчики не остановились и добавили в язык два дополнительных способа деления. Они не встречаются в классической математике, поэтому мы рассмотрим их подробнее. Целочисленное деление записывается как два знака деления подряд "//". Как следует из названия, цель такого деления — отбросить дробную часть результата. Оператор возвращает int, если оба операнда — целые числа, и float, если хотя бы один операнд является вещественным числом. В любом из вариантов дробная часть будет отброшена, а результат округлён в меньшую сторону. В противоположность прошлой операции существует остаток от деления. Он записывается знаком процента и возвращает остаток. Важно: операция возвращает не дробную часть результата, а именно остаток изначального числа.	

Остаток от деления — не самая простая операция, ведь её аналогии не существует в математике.

Итак, алгебраические операции — не единственное, что пришло в программирование из математики. Здесь можно встретить булеву логику, элементы тригонометрии и стереометрию. Это понятно: первыми программистами были профессора математики, которые хотели ускорить свои вычисления за счёт машин. Но программированию ничего не мешает вносить нововведения, например, целочисленное деление, которое мы рассмотрели в этом видео.

https://www.youtube.com/watch?time_continue=45&v=wVgLnreMpwk&feature=emb_logo

Подведём итоги

Алгебраические операции в Python

Для строк

Что нужно сделать?	Оператор в Python	Пример
Склеить	+	"a" + "b" = "ab"
Продублировать	*	"ab" * 2 = "abab"

Для чисел

Что нужно сделать?	Оператор в Python	Пример
Сложить	+	2 + 3 = 5

Вычесть	-	5 - 2 = 3	
Умножить	*	2 * 3 = 6	
Разделить	/	6 / 3 = 2	

Получить остаток от деления	%	7 % 3 = 1
Получить целую часть от деления	//	7 // 3 = 2

Логические выражения

Введение

В жизни мы часто соглашаемся с утверждениями или отрицаем их. Если вам скажут «На улице идет дождь», вы можете согласиться или отвергнуть довод. В такой формулировке утверждения не существует промежуточного этапа: дождь либо есть, либо нет.

В математике такие утверждения называют высказываниями — предложениями на любом языке, содержание которых можно однозначно определить, как истинное или ложное.

Какие из следующих утверждений можно назвать высказываниями?

- Температура плавления свинца — 1500 градусов
- Русский писатель А.С. Пушкин родился в 1799 году

		• Все мандарины вкусные • Второй закон Ньютона выражается формулой F=m•a • Встань и закрой дверь Какие из следующих утверждений можно назвать высказываниями? • Земля плоская • Это предложение является ложным • 13 + 24 = 37 • 12**2 = 120 **Важно:** выражения существуют в любом языке, в том числе в математике и языках программирования. Оценим выражение 15 + 7 > 20. Оно правдиво, ведь число 22 больше 20. Запишем то же самое, но уже в Python. Как видно, Python с нами согласен, результат выражения — True, то есть правда. Результат программы False означает ложь. Вне зависимости от длины, сложности и количества элементов, в логическом выражении всего два конечных результата: • True, • False. Операторы сравнений Для создания алгебраического выражения необходимо использовать специальные операторы: сложение, вычитание и другие.	

		Как следствие, для создания логического выражения требуются логические операторы или операторы сравнения. Они используются для оценки двух объектов, не всегда являющимся числами. Результат оценки — логическое значение True или False. Python поддерживает шесть логических операторов. Познакомимся с ними поближе. Первая тройка знакома вам ещё с начальной школы: «<» (меньше) — условие верно, если первый операнд меньше второго; «>» (больше) — условие верно, если первый операнд больше второго; «==» (равенство) — условие верно, если два операнда равны. Комбинируя их между собой, можно получить ещё три оператора: «<=» (меньше или равно) — условие верно, если первый операнд меньше второго или они равны; «>=» (больше или равно) — условие верно, если первый операнд больше второго или они равны; «!=» (неравенство) — условие верно, если два операнда не равны. **Важно!** Оператор «==» (в отличие от привычного «равно») используется не для присваивания значений, а для сравнения двух объектов между собой.	

		Результат сравнения любой пары объектов — это логическое значение True или False. Это не число и не строка, а особый вид данных boolean или логический тип (в некоторых курсах его называют булевым типом). А раз это тип данных, то его можно записать в переменную. Переменная, хранящая boolean, занимает всего один бит информации. Для сравнения: int занимает 8 бит, а float — 16. Сравнивать можно не только числа, но и строки. Самый распространённый вариант сравнения строк это знак равенства. Стоит отметить, что отличие хотя бы на один символ вызывает несовпадение строк, даже если различается регистр одного символа. print(“apple” == “Apple”) print(“apple” == “appl”) Операторы сравнения в Python можно объединять в цепочки. Например: a == b == c или 1 <= x <= 10.	

Закрепление изученного материала	30 мин.	Для создания переменной нам понадобится знак "=". Он же — оператор присвоения значения. Чтобы поместить в переменную некоторое значение, нужно придерживаться структуры: имя_переменной = значение или выражение, результат которого мы хотим хранить в переменной. Имя переменной: ● не должно повторять уже встроенные в Python имена: print, str, int и другие; ● может содержать только латинские буквы, цифры и нижнее подчёркивание; ● имена переменных не должны начинаться с цифры; ● хорошим тоном считаются короткие и понятные имена. При работе с переменными важно учитывать их тип данных: ● int — хранит целые числа; ● float — меньший диапазон значений, хранит вещественные числа; ● str (string) — тип для хранения текстовых значений. От типа данных зависит результат некоторых математических операций. Если вам нужно изменить тип данных в переменной, на помощь придут следующие функции: ● type() — вернёт тип данных переменной, ● str() — изменит тип данных аргумента на текстовый, ● int() — изменит тип данных аргумента на целочисленный,	Ссылка на Google Colab: https://colab.research.google.com/drive/17qYFLpm7UFhTJ0WWxLuiG78mRyvn4lqJ?usp=sharing

		• float() — изменит тип данных аргумента на вещественный. Язык Python сильно упростил типизацию переменных: это облегчает разработчику задачи, но требует бдительности. Пример этому — функция input(), которая всегда возвращает данные в формате строки. Если вам нужно считать число, необходимо изменить тип данных значения. Одни и те же алгебраические операторы дают различные результаты на разных типах данных, об этом поговорим чуть позже. **Задача 1** Дана переменная типа int: students_count = 45 Напишите код, с помощью которого можно привести хранящееся в данной переменной значение к типу str и присвоить это значение новой переменной str_students_count. Выведите при помощи функции print() переменную str_students_count на экран дважды. Код в редакторе: students_count = 45 Решение: str_students_count = str(students_count) print(str_students_count*2) Логические выражения стоят особняком от остальной математики. В программировании они часто встречаются в составе условных операторов — с ними мы познакомимся в следующем видео.	

		Итак, мы познакомились с новым типом данных — boolean. Существует два типа логических значений: True (истина) или False (ложь). Для создания переменной с таким типом данных, в неё можно вручную записать значение: ● a = True ● b = False Или создать выражение с помощью логических операторов: ● a = 25*2 > 40 ● b = “Привет” != “Hello” Python поддерживает шесть логических операторов. Первая тройка знакома вам ещё с начальной школы: «<» (меньше) — условие верно, если первый операнд меньше второго; «>» (больше) — условие верно, если первый операнд больше второго; «==» (равенство) — условие верно, если два операнда равны. Комбинируя их между собой, можно получить ещё три оператора : «<=» (меньше или равно) — условие верно, если первый операнд меньше второго или они равны; «>=» (больше или равно) — условие верно, если первый операнд больше второго или они равны; «!=» (неравенство) — условие верно, если два операнда не равны.	

Этап подведения итогов занятия (рефлексия)	8 мин.	Вопросы для обсуждения • Что тебе на уроке более всего понравилось? • Что было трудным или непонятным?	Педагог способствует размышлению обучающихся над вопросами
Информация о домашнем задании, инструктаж по его применению	5 мин.	Домашнее задание представляет из себя решение нескольких задач по программированию. Программы проверяются в автоматическом режиме на платформе Stepik или платформе Академии искусственного интеллекта.	

Рекомендуемые ресурсы для дополнительного изучения:

1. ПИТОНТЬЮТОР. [Электронный ресурс] – Режим доступа: http://pythontutor.ru/.
2. Онлайн игра на программирование CodeCombat:. [Электронный ресурс] – Режим доступа: https://codecombat.com/.
3. Прямая ссылка на начало игры. [Электронный ресурс] – Режим доступа: https://codecombat.com/play.