МИНИСТЕРСТВО ПРОСВЕЩЕНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Министерство образования и науки Удмуртской Республики

Управление образования администрации города Ижевска

МБОУ "СОШ №100"

РАССМОТРЕНО

Педагогическим советом МБОУ «СОШ №100»

Протокол №11 от 30.08.2023 г.

УТВЕРЖДЕНО

Приказом МБОУ «СОШ №100»

_____Помыткин Б.В.

Приказ директора №288-од от 30.08.2023 г.

РАБОЧАЯ ПРОГРАММА

учебного предмета «Введение в программирование»

для обучающихся 6-9 классов

Ижевск 2023

Раздел 1. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Актуальность программы

Язык Python – один из самый изучаемый и один из самых востребованных на рынке труда. Python достаточно широко используется при изучении основ алгоритмизации и программирования в школьном курсе информатики. В частности, используется в качестве базового языка для изучения в УМК по информатике авторов Поляков К. Ю., Еремин Е. А., допущенном к использованию в общеобразовательных организациях Министерством просвещения РФ (Приказ Министерства Просвещения РФ от 28 декабря 2018 года № 345).

Изучение Python в школе откроет ученикам возможности дальнейшего развития в области IT и поможет профориентации в старших классах, пригодится в олимпиадах по программированию и решении заданий ЕГЭ.

Курс предполагает смешанный формат обучения. Сочетание групповой работы с учителем в классе и индивидуальной работы в личном кабинете на онлайн-платформе позволяет ученикам выработать не только технические навыки программирования, но и навыки социального взаимодействия при работе над финальным проектом курса, а главное – научиться самостоятельно выстраивать свое профессиональное развитие.

Цель реализации программы – формирование у обучающихся навыков программирования на языке Python и самонаправленного обучения.

Направленность дополнительной общеобразовательной программы: техническая.

Задачи реализации программы:

Обучающие

1. Изучить основы программирования на языке Python;

Развивающие

2. Научиться применять полученные знания для решения практических задач.

Воспитательные

3. Научиться применять полученные знания для решения практических задач.
4. Повысить уровень самостоятельности в обучении

Организация курса:
Курс состоит из 68 уроков.
Методические материалы курса состоят из:

1. Методических указаний для учителя в текстовом виде,
2. Презентации с иллюстративным изложением теоретического материала;
3. Упражнений на платформе Stepik с теоретическим и практическим материалом языка Python;
4. Интерактивных проверочных заданий в приложениях Kahoot и Learning Apps;
5. Подвижных игр, направленных на закрепление знаний, полученных на занятии.
6. Инструкций для проведения рефлексии процесса обучения с учениками.

Планируемые результаты обучения:
Личностные
По окончании курса учащийся сможет:

- Программировать на языке Python.
- Использовать инструменты разработки среды Wing.
- Самостоятельно реализовывать проекты, связанные с разработкой игр.

Метапредметные

- Ставить учебные цели.
- Формулировать достигнутый результат.
- Планировать свою самостоятельную учебно-познавательную деятельность; выбирать индивидуальную траекторию достижения учебной цели.
- Определять подходы и методы для достижения поставленной цели.
- Отбирать необходимые средства для достижения поставленной цели.
- Осуществлять самооценку промежуточных и итоговых результатов своей самостоятельной учебно-познавательной деятельности.
- Проводить рефлексию своей учебно-познавательной деятельности.

Предметные
Практическим результатом работы служит финальный проект каждого ученика: 2D игра на движке Pygame, либо серия самостоятельно разработанных мини-проектов в консольном режиме языка Python.

Основные формы организации занятий программы «Python для начинающих»:

- Практические занятия с использованием онлайн-платформы Stepik;
- Работа в IDE;

- Прохождение опросов на сайте Kpolyakov.spb.ru;
- Домашние практические занятия с использованием онлайн-платформы Stepik, направленные на отработку навыков программирования на языке Python.

Категория обучающихся: ученики общеобразовательных школ от 12 до 18 лет в рамках внеурочной деятельности и дополнительного образования.

Срок освоения программы: в течение 3 учебных года, в объеме 72 часов.

Отличительные особенности программы: авторский подход к организации образовательного процесса. Реализация программы предполагает проведение аудиторных занятий с использованием вариативных дистанционных ресурсов:

- авторских онлайн уроков, размещенных на платформе Stepik;
- авторских интерактивных заданий (опросов, викторин, дидактических игр и др.), размещенных на бесплатных специализированных сервисах Kahoot и Learning Apps;
- авторских проектных заданий для создания учащимися игр на языке Python.

Раздел 2. Содержание программы

В рамках смешанного подхода к обучению каждая из тем, изложенных преподавателем, отрабатывается в уроках платформы Stepik и в практических заданиях. (https://stepik.org/course/58852/syllabus)

2.1. Учебный (тематический) план

№	Название темы	Количество часов	Теоретические	Практические
1	Знакомство с Python. Команды input() и print().	2	1	1
2	Параметры sep, end. Переменные. Комментарии. PEP 8	2	1	1
	Проверочная работа №1	2		
3	Работа с целыми числами	2	1	1
4	Условный оператор. Логические операции and, or, not	3	1	2
5	Вложенный и каскадный условный оператор	3	1	2
	Проверочная работа №2	2		
6	Типы данных int, float, str. Встроенные функции min(), max(), abs(). Оператор in.	2	1	1
7	Цикл for. Функция range().	3	1	2
	Проверочная работа №3	2		
8	Частые сценарии при написании циклов. Расширенные операторы присваивания.	2	1	1
9	Цикл с предусловием while	3	1	2
10	Операторы break, continue, else.	2	1	1
11	Вложенные циклы	3	1	2
	Проверочная работа №4	2		
12	Строковый тип данных: индексация и срезы	3	1	2
13	Методы строк	3	1	2
	Проверочная работа №5	2		
14	Резервное время. Введение в списки.	2	1	1

15	Основы работы со списками. Методы списков	2	1	1
16	Вывод элементов списка. Строковые методы split() и join()	2	1	1
17	Методы списков. Списочные выражения	2	1	1
18	Функции	3	1	2
	Проверочная работа №6	2		
19	Локальные и глобальные переменные. Функции возвращающие значения.	3	1	2
20	Функции возвращающие значения.	2	1	1
21	Работа над проектом	2	1	1
22	Работа над проектом	2	0	2
23	Работа над проектом	2	0	2
24	Работа над проектом	2	0	2
25	Работа над проектом	2	0	2
	Итого	72		

2.2. Рабочая программа

№	Название	Виды учебных занятий, учебных работ	Краткое описание

1	Знакомство с Python. Команды input() и print()	Интерактивные занятия	Знакомство с учениками. Сбор ожиданий учеников, пояснение программы курса, рефлексия. Объяснение темы.
		Работа за компьютером	Регистрация на платформе Stepik. Решение задач.
2	Параметры sep, end. Переменные. Комментарии. PEP 8	Интерактивные занятия	Продолжение знакомства в группе. “Что?Где?Когда?” по пройденной теме. Объяснение новой темы.
		Работа за компьютером	Отработка навыков работы с переменными, использования комментариев в коде.
3	Работа с целыми числами	Интерактивные занятия	Обсуждение правильной работы с ошибками. Практика взаимодействия в группе. Объяснение темы.
		Работа за компьютером	Отработка операций с целыми числами. Обработка цифр числа.
4	Условный оператор. Логические операции and, or, not	Интерактивные занятия	Приоритеты и планирование. Объяснение темы. Разбор задач на доске.
		Работа за компьютером	Решение задач с использованием условного оператора и логических операций.
5	Вложенный и каскадный	Интерактивные занятия	Повторение. Объяснение темы.

	условный оператор		Разбор задачи. Рефлексия “Повар”
		Работа за компьютером	Решение задач с использованием каскадного условного оператора и вложенных ветвлений.
6	Типы данных int, float, str. Встроенные функции min(), max(), abs(). Оператор in	Интерактивные занятия	Анонс нестандартных форм работы с классом. Объяснение темы. Групповая работа над презентацией.
		Работа за компьютером	Самостоятельное изучение темы в малых группах и создание презентации. Решение задач на платформе.
7	Цикл for. Функция range()	Интерактивные занятия	Повторение. Объяснение новой темы. Подвижная игра с функцией range(). Рефлексия.
		Работа за компьютером	Решение задач на платформе.
8	Частые сценарии при написании циклов. Расширенные операторы присваивания	Интерактивные занятия	Игровое повторение предыдущей темы. Самостоятельное изучение новой темы в командах. Объяснение новой темы.
		Работа за компьютером	Самостоятельное исследование частых сценариев программирования. Решение задач на платформе на отработку частых сценариев.

9	Цикл с предусловием while	Интерактивные занятия	Игра на вопросы с бинарной логикой “данетка”. Объяснение новой темы.
		Работа за компьютером	Решение задач на использование цикла с предусловием.Самостоятельное изучение процедуры обработки цифр натурального числа.
10	Операторы break, continue, else	Интерактивные занятия	Объяснение темы. Разбор устных задач. Работа в командах над ревью кода.
		Работа за компьютером	Решение задач на платформе: отработка применения оператора break в циклах.
11	Вложенные циклы	Интерактивные занятия	Объяснение темы “Вложенные циклы”. Разбор задач. Рефлексия.
		Работа за компьютером	Решение задач на платформе: отработка применения вложенных циклов.
12	Строковый тип данных: индексация и срезы	Интерактивные занятия	Повторение темы “Циклы”. Короткое объяснение темы. Рефлексия.
		Работа за компьютером	Самостоятельное изучение темы “Строковый тип данных”, решение задач на платформе.
13	Методы строк	Интерактивные занятия	Повторение темы “Строки”. Устный разбор методов строк и

			их функционала. Рефлексия.
		Работа за компьютером	Решение задач на платформе: использование методов строк.
14	Резервное время. Введение в списки	Интерактивные занятия	Индивидуальная работа с учениками. Объяснение новой темы “Списки”.
		Работа за компьютером	Решение задач на платформе: сначала всех пропущенных, потом - на использование списков.
15	Основы работы со списками. Методы списков	Интерактивные занятия	Повторение прошлой темы. Подведение итогов самостоятельной работы учеников, рефлексия.
		Работа за компьютером	Самостоятельное изучение теории. Решение задач на отработку методов работы со списками.
16	Вывод элементов списка. Строковые методы split() и join()	Интерактивные занятия	Повторение предыдущей темы. Подведение общих итогов самостоятельного изучения теории. Объяснение методов split и join. Игра в парах с образцами кода.
		Работа за компьютером	Самостоятельное изучение теории. Решение задач на

			работу со списками.
17	Методы списков. Списочные выражения	Интерактивные занятия	Разминка, повторение предыдущей темы. Групповое подведение итогов темы “Методы списков”. Соревновательное подведение итогов изучения темы “Списочные выражения”. Рефлексия командной работы.
		Работа за компьютером	Самостоятельное изучение темы “Методы списков”. В парах: изучение темы “Списочные выражения”
18	Функции	Интерактивные занятия	Объяснение темы “Функции без параметров”. Объяснение темы “Функции с параметрами”
		Работа за компьютером	Решение задач на платформе.
19	Локальные и глобальные переменные. Функции, возвращающие значения	Интерактивные занятия	Повторение предыдущей темы. Постановка личной цели на урок “Дерево цели”. Объяснение темы “Локальные и глобальные переменные. Рефлексия с оценкой процента выполнения поставленных целей.
		Работа за компьютером	Самостоятельное изучение темы “Функции,

			возвращающие значения". Решение задач на платформе.
20	Функции, возвращающие значения	Интерактивные занятия	Игра "Шляпа" на глобальное повторение. Дискуссия об использовании функций. Финализирующая рефлексия "Палитра". Игра на введение в проектную деятельность.
		Работа за компьютером	Решение задач на отработку темы "Функции, возвращающие значения"
21	Работа над проектом	Интерактивные занятия	Объяснение проектного подхода к заданиям.
		Работа за компьютером	Работа над общим проектом-образцом на платформе.
22	Работа над проектом	Работа за компьютером	Самостоятельная работа над проектом.
		Работа за компьютером	Самостоятельная работа над проектом.
23	Работа над проектом	Работа за компьютером	Самостоятельная работа над проектом.
		Работа за компьютером	Самостоятельная работа над проектом.
24	Работа над проектом	Интерактивные занятия	Объяснение принципов краткой презентации. Рефлексия.
		Работа за компьютером	Самостоятельная работа над проектом.

25	Работа над проектом	Работа за компьютером	Самостоятельная работа над проектом.
		Презентация проекта	Выступление с презентацией собственного проекта.

Раздел 3. Формы аттестации и оценочные материалы

Аттестация проводится в форме выполнения индивидуальных и групповых заданий по пройденному материалу. Контроль в указанной форме осуществляется как промежуточный, так и итоговый. Отметочная форма контроля отсутствуют.

Для заданий на онлайн-тренажере указан необходимый минимум — 70% выполненных заданий, чтобы тема считалась пройденной успешно и был открыт доступ к следующей теме.

После каждой темы в онлайн-курсе стоит итоговая работа: от ученика требуется в ограниченное время (три часа) решить набор задач по пройденной теме. В среднем, ученик справляется с решением за 30 минут. Преподаватели могут использовать эти итоговые работы в качестве промежуточных проверочных работ.

В конце курса, по итогам работы над групповыми и индивидуальными проектами проводится обсуждение результатов в коллективе с опорой на чек-лист, исправление ошибок и, тем самым, коррекция и закрепление полученных знаний.

Раздел 4. Организационно-педагогические условия реализации программы

4.1. Литература для педагога

1. **Васильев, А. Н. Python на примерах** [Текст]:практ. курс /А. Н. Васильев - Наука и Техника, 2019 - 432 с.
2. **Прохоренок, Н. А. Python 3: самое необходимое** [Текст]:практ. курс / Н. А. Прохоренок, В. А. Дронов - БХВ-Петербург, 2019 - 608 с.
3. **Гэддис, Т. Начинаем программировать на Python** [Текст]: учебник/Т. Гэддис - БХВ-Петербург, 2019 - 768 с.
4. **Седжвик, Р. Программирование на языке Python** /Р. Седжвик, К. Уэйн, Р. Дондеро - Вильямс, 2017 - 736 с.
5. **Харрисон, М. Как устроен Python.** [Текст]:практ. курс / М. Харрисон - Питер, 2002 - 272 с.

Литература для обучающихся: Не предусмотрена

Электронные ресурсы:

1. Курс Поколение Python: курс для начинающих на платформе Stepik. (https://stepik.org/course/58852/syllabus)

4.2. Материально-технические условия реализации программы

1. Обязательные

- помещение (предпочтительно, изолированное);
- 10—15 рабочих мест: стол, стул, розетка, компьютеры на каждое рабочее место;
- проектор, аудио колонки;
- Интернет-соединение, скорость загрузки не менее 2 Мбит/сек;
- меловая, магнитно-маркерная доска или флипчарт;
- общие условия в соответствии с СанПиНом 2.4.4.3172-14

Требования к ПО:

- Операционная система Windows 7 или моложе / MacOS / Unix-based системы с поддержкой протокола HTML5;
- Приложения Google Chrome, Gimp, Brackets;
- интерактивная оболочка (бесплатная IDE Wing101 или аналог).

Приложение 1. Образец методических указаний для преподавателя.

Методические указания

Урок 2. Параметры sep, end. Переменные. Комментарии. PEP 8

Задачи урока:

- Научиться настраивать команду print(): параметры sep, end
- Научиться работать с переменными
- Научиться работать с комментариями
- Узнать о стандарте PEP 8

Материалы, демонстрируемые на проекторе:

- Презентация

Материалы, необходимые для урока

- Распечатанные карточки с дилеммами

План урока:

№	Этап	Время этапа (мин.)
1	Фокусировка/разминка (warm up)	10 мин
2	Повторение print() / input()	5 мин
3	Параметры sep, end	10 мин
4	Что? Где? Когда?	5 мин
5	Практика на платформе Stepik	10 мин
6	Перерыв	5 мин
7	Фокусировка	5 мин
8	Переменные	10 мин
9	PEP 8 и комментарии	15 мин
10	Рефлексия	10 мин

0. Подготовка к уроку

До начала урока учителю необходимо:

1) Просмотреть в классе на платформе Stepik, как ученики справились с домашним заданием
2) Прочитать методичку
3) Решить задачи из курса на Stepik: урок 2.3

1. Создаем атмосферу на занятии (10 мин.)

Как предлагалось на предыдущем уроке, мы начинаем второй урок с настройки взаимодействия в группе и налаживания контактов между учащимися.

Учитель: Добрый день. Приятно вас видеть на втором занятии. Помните, мы обсуждали принципы работы в группе. Вот они:

1. **Все вопросы важны и ценны.**
2. **Ошибка - наш помощник.**
3. **Активность и ответственность - ваше всё.**
4. **Стремимся к максимуму.**
5. **Поддержка и взаимопомощь.**

Для того, чтобы эти принципы хорошо работали и нам было здорово учиться вместе, нам нужно получше узнать друг друга. Сейчас каждый из вас получит лист и карточку с дилеммой. Что такое дилемма? Правильно, это два противоположных утверждения, из которых нужно выбрать лишь одно. Например: риск или спокойствие? Каждая позиция по-своему хороша, и нужно внимательно взвесить, что лучше подходит лично вам.

Вы делите лист на 2 столбика, записываете сверху свою дилемму и отправляетесь собирать статистику по группе, спрашивая других что они выбирают и почему. На листик нужно коротко записать услышанные аргументы в пользу выбранной позиции. У вас на это 3 минуты.

А сейчас по очереди делимся результатами, сколько человек выбрали одну альтернативу, сколько - другую, рассказывая, почему люди выбирали то, что выбирали.

Можно просто поделиться статистикой без объяснения кто что выбрал.

Чтение или кино? Путешествие или отдых в лесу? Пойти в кафе или купить любимой еды в магазине? Осень, зима, весна или лето? Сова или жаворонок? Суперсила на один день или день в теле другого человека? Велосипед или ролики? Потратить деньги на развлечения или приобретение вещей? Пойти в гости или принимать гостей? Интуиция или логика? Суши или пицца? Самолет или поезд? Жить в большом городе или в маленькой деревне? Поехать в отпуск с родителями или остаться дома одному? Fanta или Cola? Instagram или VK?

2. Вспоминаем прошлый урок

(5 мин.)

Учитель: Теперь давайте вспомним, чему мы учились на прошлом уроке? *Покажите слайд с вопросами. Ребята отвечают, что помнят. Хвалите каждый толковый ответ.*

Приблизительные правильные ответы:

1. Компьютерная программа - это список инструкций для компьютера
2. Язык программирования позволяет общаться человеку и компьютеру.
3. Плюсы: работает на любой операционной системе, простой, пластичный. Минусы: медленный.
4. Для вывода данных на экран.
5. Для ввода данных.

3. Параметры sep и end

(10 мин.)

Учитель: Сегодня мы с вами разберемся с **дополнительными параметрами команды print()**, которые позволяют настраивать вывод данных так, как нам надо. Например, выводить данные не через пробел, а, скажем, через запятую или какой-либо другой символ.

Покажите слайд в котором объясняется принцип действия параметра sep.

Здесь вы видите, какой эффект имеет параметр sep. Попробуйте выдвинуть гипотезы, что такое sep. *После этого учитель, ссылаясь на ответы ребят, говорит, что sep происходит от слова* separate *англ. [сéпарэйт]*, что значит “разделять”. Это поможет ученикам не путать в дальнейшем параметры sep и end (*ученики часто путают их*).

Далее учитель демонстрирует слайд с принципом действия параметра end, обсуждает с ребятами этот параметр и поясняет, что он позволяет переопределить стандартный перенос строки после вывода print().

Учитель заостряет внимание, что команда print() разделяет свои аргументы одним пробелом при выводе и вставляет переход на новую строку. Когда в качестве разделителя между аргументам и в качестве окончания вывода нужно использовать другие символы мы используем необязательные параметры sep и end. (*от английского separator [сепарэйтор] - разделитель и end [энд] - конец*)

Следует сказать, что параметр end вставляется единожды по завершении вывода, в отличие от параметра sep, который вставляется между каждой парой аргументов команды print().

Учителю стоит сказать про управляющую последовательность символов **\n**, которая означает переход на новую строку.
Имеет смысл заострить внимание на значениях параметров по умолчанию:

```
sep = ' '    # пробел
end = '\n'  # перевод строки
```

Если поведение по умолчанию нас устраивает, то писать явно значения параметров sep и end не имеет смысла.

4. Что? Где? Когда?

(5 мин.)
Разделите группу на две команды, примерно по 6 человек в каждой. Объясните правила игры: на слайде будет показана задача, на решение которой у команды

ровно одна минута. Ответ нужно записать на один лист и сдать учителю сразу после сигнала окончания минуты.
После того, как все ответы сданы, учитель показывает слайд с правильным ответом и комментирует его, если ответы команд не верны. Затем на экран выводится следующая задача, и начинается новый отсчет времени.

Далее учитель показывает два слайда с устными задачами и спрашивает у учеников, что будет выведено. Учитель при необходимости наводит на правильный ответ.

5. Практика на платформе Stepik

(10 мин).

Далее ученики садятся за компьютеры и решают самостоятельно задачу на программирование “Кастомный разделитель” - вторая задача на программирование из блока 2.3

```
separator = input()
a = input()
b = input()
c = input()
print(a, b, c, sep=separator)
```

6. Перерыв

(5 мин.)

7. Разминка. Счет до 33

(5 мин).

Учитель: мы сейчас будем считать с вами до 33 (или 43, 53). Правда определенным образом. Мы считаем по очереди, по кругу: Один начинает, говорит “один”, другой продолжает, следующий говорит “два” и т.д. При этом, число, которое делится на 3, например, 6, 9, или содержит “тройку”, например, 13, 23, мы заменяем хлопком. Если кто-то не хлопнул или хлопнул на неверное число, например, 16, 29, мы начинаем сначала.

Лучше, чтобы каждый раз начинали разные люди, иначе все просто

запоминают свои позиции и уже не задумываются. Рекомендуем сделать ограничить количества попыток, например, до 5 или во времени. Это упражнение направлено на внимание и концентрацию.

8. Переменные

(10 мин.)

Учитель: Когда мы изучали команду input(), то имели дело с переменными. Давайте чуть подробнее поговорим о них.

Переменные - это именованные участки памяти в которых хранится какая-нибудь нужная для программы информация. Информация может быть абсолютно произвольной: текст, числа и т.д.

Далее учитель объясняет смысл кода указанного на слайде и говорит, что у переменной обязательно есть две составляющие: ИМЯ и ЗНАЧЕНИЕ.

ИМЯ: Учитель рассказывает о правилах именования переменных в Python.

Обязательно учитель говорит, что имя переменной должно быть осмысленным: если переменная содержит имя, то она может называться name, если возраст, то age и т. д.

Учитель напоминает, что Python - регистрозависимый язык программирования. Также стоит сказать, о стиле именования переменных принятому в Python **lower_case_with_underscores (маленькие буквы с подчеркиванием)**

ЗНАЧЕНИЕ: Учитель демонстрирует слайд со значением переменной.
Нужно обязательно сказать, что имя переменной находится слева от оператора присваивания:
<имя переменной> = <значение переменной> **Правильно.**
<значение переменной> = <имя переменной> **Это частая ошибка!**

Ученики решают две устные задачи, в которых демонстрируется переприсваивание переменных. Затем они сами придумывают в парах одну свою задачу и отдают ее учителю. (Если учитель понимает, что на это есть время. Иначе пропустить этот шаг.) Эти задачи можно использовать на следующем занятии для разогрева.

9. PEP 8 и комментарии

(15 мин)

Цель этой части - обсудить с ребятами хороший стиль языка Python. Предлагаем это сделать через игровой элемент. Ребятам предлагается представить себя в роли руководителей компании. И им нужно выбрать одного программиста для проекта. Они думают, кого они выберут и почему. Соответственно в каждом резюме есть есть позиция по поводу культуры кода.

Учитель объясняет задачу, что ребятам нужно выбрать программиста для проекта. Зачитывает резюме из презентации и проводит обсуждение, кого бы ребята выбрали и почему.

Это упражнение можно проводить также в малых группах. Тогда ребята сначала обсуждают в малых группах, а затем рассказывают свой выбор. Здесь интересно будет посмотреть, совпали ли мнения или нет, в чем разница аргументации.

После этого учитель делает презентацию о культуре кода.

Презентация:
Создатель языка Гвидо ван Россум и его соратник Барри Уорсо описали хороший стиль Python кода в документе PEP 8 (читается ПИП 8).

Далее учитель демонстрирует три слайда и показывает некоторые рекомендации.
Учитель должен сказать, что следование стандарту PEP 8 не является обязательным, однако это то, что отличает опытного Python программиста от начинающего. Во всех крупных компаниях в которых пишут на Python используют PEP 8.

Учитель: запоминать все правила из PEP 8 не имеет смысла, поскольку среда разработки Wing IDE их все знает. Мы можем включить в среде Wing IDE автоматическое форматирование под стандарт PEP 8.

Мы уже знаем, что программы состоят из команд (инструкций), которые понимает Python. Однако иногда нам бывает нужно вставить в программу

текст, который нужен только нам, людям, которые создают программу. Это может быть текстовая метка, поясняющая что-либо. Такая метка через некоторое время поможет вспомнить, что делает указанный код.

Комментарии также полезны, когда над кодом работает целая команда программистов. С их помощью один программист, может понять, что имел ввиду другой программист, если код не является очень простым. Учитель показывает слайды с комментариями.

Если осталось время - можно посадить учеников за компьютеры, чтобы они решили несколько тестовых заданий в уроке 2.3 на платформе.

10. Рефлексия

(5 мин.)

Предложите ученикам на небольших листочках ответить на вопросы чек-листа.

- ❑ Я понял/а как работает команда input() и могу применять ее.
- ❑ Я решил/а __ задач на Stepik.
- ❑ Мне понятны преимущества хорошего тона в написании кода.
- ❑ Я знаю два дополнительных параметра команды print()
- ❑ Я установил/а дома IDE Wing 101
- ❑ Мне еще нужно разобраться в ...

Скажите, что вам будет любопытно посмотреть на результаты этого опроса, но сдавать их не обязательно, если кто-то категорически не хочет этого делать. Подчеркните важность вовремя задать вопросы, если какая-то тема осталась непонятой.

Домашняя работа

В конце урока можно подвести итоги и дать домашнюю работу:

1. Ученики могут пройти уроки 2.3 из курса:
    a. прочитать теоретические конспекты
    b. решить тестовые задачи и задачи на программирование

Учителю стоит сказать ученикам, что домашняя работа не является обязательной, однако, решения задач на практике позволяет закрепить материал и понять то, что не удалось понять на занятии. В курсе есть дополнительная полезная информация.

Дополнительно

Если на уроке остается время, то ученикам можно предложить решить оставшиеся задачи на ввод и вывод данных. В таком случае им достанется меньше задач на самостоятельное решение дома.