

**РЕЗУЛЬТАТЫ ОСВОЕНИЯ КУРСА ВНЕУРОЧНОЙ ДЕЯТЕЛЬНОСТИ**

***Личностные результаты.*** К личностным результатамосвоения информационных и коммуникационных технологий как инструмента в учёбе и повседневной жизни можно отнести:

* критическое отношение к информации и избирательность её восприятия;
* уважение к информации о частной жизни и информационным результатам других людей;
* осмысление мотивов своих действий при выполнении заданий с жизненными ситуациями;
* начало профессионального самоопределения, ознакомление с миром профессий, связанных с информационными и коммуникационными технологиями.

***Метапредметными результатами*** изучения курса «Программирование в Python 3» являются формирование следующих универсальных учебных действий:

***Регулятивные УУД*:**

* планирование последовательности шагов алгоритма для достижения цели;
* поиск ошибок в плане действий и внесение в него изменений.

***Познавательные УУД*:**

* моделирование – преобразование объекта из чувствен­ной формы в модель, где выделены существенные характе­ристики объекта (пространственно-графическая или знаково-символическая);
* анализ объектов с целью выделения признаков (суще­ственных, несущественных);
* синтез – составление целого из частей, в том числе самостоятельное достраивание с восполнением недостающих компонентов;
* выбор оснований и критериев для сравнения, сериации, классификации объектов;
* подведение под понятие;
* установление причинно-следственных связей;
* построение логической цепи рассуждений.

***Коммуникативные УУД*:**

* аргументирование своей точки зрения на выбор оснований и критериев при выделении признаков, сравнении и классификации объектов;
* выслушивание собеседника и ведение диалога;
* признавание возможности существования различных точек зрения и права каждого иметь свою.

Python – серьёзный язык программирования. В то же время учащиеся в полной мере могут раскрыть свои творческие таланты, так как с его помощью можно легко создавать игры и другие приложения.

**содержание курса внеурочной деятельности**

**Знакомство учащихся с устройством языка Python.**

Техника безопасности на занятии. Знакомство с интерфейсом программы.

**Установка Python. Среда разработки IDLE. Сохранение программы. Ошибки в окне программы. Ошибки в окне консоли. Поиск ошибок.**

Скачивание и установка на компьютер Python 3. Знакомство со средой IDLE – это ориентированная на начинающих среда разработки, в которой есть несложный текстовый редактор для написания и отладки Рython- программ.

**Проект №1. «Дом с привидениями». Структура программы: подготовка к игре, основной цикл, конец игры.**

В этой игре показаны некоторые основные возможности Python. Программа разбирается по частям для понимания её устройства: подготовка к игре, основной цикл, конец игры. Основные понятия: ввод программы, запуск программы, структурирование программы, понятие переменных, отображение текста.

**Ветвление на языке Python. Использование ветвления в основном цикле игры.**

Понятие «ветвления». Программа выполняет разные действия в зависимости от того, выполняется ли условие. Оператор «ветвления».

**Два вида окон в IDLE: окно программы и окно консоли.**

Два вида окон в IDLE: окно программы и окно консоли. Сравнение этих окон и их возможностей.

**Переменные в Python. Создание переменной. Использование переменных.**

Создание переменной, изменение значения переменной, имена переменных, использование переменных, печать переменных.

**Типы данных: числа, строки, булевы значения. Преобразование типов данных.**

Данные, которыми пользуются компьютеры (и языки программирования), представлены несколькими типами, и это вполне очевидно. Числа бывают дробными, целыми, могут состоять из множества цифр или быть весьма массивными из-за дробной части. Чтобы интерпретатору было проще работать с ними, и он мог понять, с чем имеет дело, следует задать определенный тип. Более того, он необходим, чтобы числа поместились в отведенную ячейку памяти. Наиболее распространенные типы данных, которым пользуется язык программирования Python: Integer. Речь идет о целых числах, имеющих как отрицательное, так и положительное значение. Ноль также входит в данный тип. Для того чтобы интерпретатор понял, что работает с дробными частями, следует задать тип float point. Как правило, им пользуются в случае использования чисел с варьирующейся точкой. Следует помнить, что при написании программы нужно придерживаться записи «3.25», а не использовать запятую «3,25». В случае добавления строк язык программирования Python позволяет добавить тип string. Зачастую слова или фразы заключаются в одинарные или двойные кавычки.

**Вычисления в Python. Простые вычисления. Использование скобок. Результат в переменной. Случайные числа.**

Простые вычисления: сложение, вычитание, умножение, деление. Использование скобок, результат в переменной, случайные числа. Функция randint().

**Строки в Python. Создание строк. Сложение строк. Длина строки. Нумерация символов. Апострофы.**

 Строки в переменных. Сложение двух строк. Добавление третьей строки в середине. Длина строки. Нумерация символов. Отсчёт символов. Срез символов. Апострофы. Экранирование апострофа.

**Ввод и вывод. Использование input(). Функция print(). Вывод в окне консоли.**

Команда input(). Использование input(). Вывод в окне консоли. Функция print(). Использование print(). Два способа разделять вывод. Три способа завершить вывод: точка в конце, вывод без перевода строки, с новой строки. Пустые строки в конце.

**Логические операции. Виды операций сравнения. Проект № 2 «Проверка на день рождения».**

Понятие логических операций. Виды операций сравнения. Проект 2: программа запрашивает номер дня и месяца, а затем с помощью логических операций выясняет, не пора ли праздновать. Сравнение строк. Строковые операции (операция in).

**Ветвление. Условие if. Условие if-else. Условие if-elif-else**

Условие if. Программа: спрашивается у пользователя, не день ли рождения у него. Если тот ответит «Y», программа напечатает поздравление Условие if-else. Программа: если ввести «Y» программа напечатает новогоднее поздравление, если ввести что-нибудь другое – появится другое сообщение. Условие if-elif-else.

**Циклы в Python. Повтор команд. Цикл for. Переменные цикла. Вложенные циклы.**

Управление Черепашкой. Используем цикл for. Переменные цикла: отсчёт, отсчёт по двойкам, обратный отсчёт. Вложенные циклы: цикл внутри цикла.

**Цикл while. Блок repeat until. Бесконечный цикл. Выход из цикла.**

Понятие цикла while. Использование цикла while. Бесконечный цикл : создание бесконечного цикла. Остановка цикла. Команда break внутри цикла. Команда continue(продолжить).

**Списки. Что такое списки? Пример списка. Использование списков.**

Понятие списка. Пример списка. Использование списков. Списки списков. Списки и циклы. Сложение списков.

**Функции. Полезные функции. Создание и вызов функций. Передача данных в функцию. Получение данных из функции**.

Понятие функции. Создание и вызов функций: определение функции, вызов функции. Параметры функции. Передача значений. Получение данных из функции. Определение функции, которая возвращает число.

**Индивидуальный творческий проект.**

Составление плана. Поиск информации. Реализация.

**Методическое обеспечение дополнительной образовательной программы**

**Формы занятий:** лекции и практические занятия на компьютере и планшете.

**Приёмы и методы организации учебно-воспитательного процесса:**

индивидуальная работа за компьютером, парная работа, работа в микрогруппах.

**Формы подведения итогов:** промежуточные итоги – небольшие проекты с целью закрепления полученных на уроке знаний, окончательные итоги – игры, демонстрирующего все знания, полученные учащимся на уроках.

**ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ**

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **темы занятий** | **Всего часов** | **В том числе** |
| **Теория** | **Практика** |
| **1** | Знакомство учащихся с устройством языка Python | **3** | **1** | **2** |
|  | Знакомство учащихся с устройством языка Python. (Дистанционно) | **1** | **1** | **0** |
| **2** | Установка Python. Среда разработки IDLE. Сохранение программы. Ошибки в окне программы. Ошибки в окне консоли. Поиск ошибок. | **5** | **1** | **4** |
|  | Установка Python. Среда разработки IDLE. Сохранение программы. Ошибки в окне программы. Ошибки в окне консоли. Поиск ошибок. (Дистанционно) | **1** | **1** | **0** |
| **3** | Проект №1. «Дом с привидениями». Структура программы: подготовка к игре, основной цикл, конец игры. | **5** | **1** | **4** |
|  | Проект №1. «Дом с привидениями». Структура программы: подготовка к игре, основной цикл, конец игры. (Дистанционно) | **1** | **1** | **0** |
| **4** | Ветвление на языке Python. Использование ветвления в основном цикле игры. | **6** | **2** | **4** |
| **5** | Два вида окон в IDLE: окно программы и окно консоли.  | **4** | **0** | **4** |
|  | Два вида окон в IDLE: окно программы и окно консоли. (Дистанционно) | **2** | **2** | **0** |
| **6** | Переменные в Python. Создание переменной. Использование переменных. | **5** | **1** | **4** |
|  | Переменные в Python. Создание переменной. Использование переменных. (Дистанционно) | **1** | **1** | **0** |
| **7** | Типы данных**:** числа, строки, булевы значения. Преобразование типов данных. | **5** | **1** | **4** |
|  | Типы данных**:** числа, строки, булевы значения. Преобразование типов данных. (Дистанционно) | **1** | **1** | **0** |
| **8** | Вычисления вPython. Простые вычисления. Использование скобок. Результат в переменной. Случайные числа. | **5** | **1** | **4** |
|  | Вычисления вPython. Простые вычисления. Использование скобок. Результат в переменной. Случайные числа. (Дистанционно) | **1** | **1** | **0** |
| **9** | Строки вPython. Создание строк. Сложение строк. Длина строки. Нумерация символов. Апострофы. | **5** | **1** | **4** |
|  | Строки вPython. Создание строк. Сложение строк. Длина строки. Нумерация символов. Апострофы. (Дистанционно) | **1** | **1** | **0** |
| **10** | Ввод и вывод. Использование input(). Функция print(). Вывод в окне консоли. | **5** | **1** | **4** |
|  | Ввод и вывод. Использование input(). Функция print(). Вывод в окне консоли. (Дистанционно) | **1** | **1** | **0** |
| **11** | Логические операции. Виды операций сравнения. Проект № 2 «Проверка на день рождения». | **5** | **1** | **4** |
|  | Логические операции. Виды операций сравнения. Проект № 2 «Проверка на день рождения». (Дистанционно) | **1** | **1** | **0** |
| **12** | Ветвление. Условие if. Условие if-else. Условие if-elif-else | **5** | **1** | **4** |
|  | Ветвление. Условие if. Условие if-else. Условие if-elif-else. (Дистанционно) | **1** | **1** | **0** |
| **13** | Циклы в Python. Повтор команд. Цикл for. Переменные цикла. Вложенные циклы. | **5** | **1** | **4** |
|  | Циклы в Python. Повтор команд. Цикл for. Переменные цикла. Вложенные циклы. (Дистанционно) | **1** | **1** | **0** |
| **14** | Цикл while. Блок repeat until. Бесконечный цикл. Выход из цикла. | **5** | **1** | **4** |
|  | Цикл while. Блок repeat until. Бесконечный цикл. Выход из цикла. (Дистанционно) | **1** | **1** | **0** |
| **15** | Списки. Что такое списки? Пример списка. Использование списков. | **5** | **1** | **4** |
|  | Списки. Что такое списки? Пример списка. Использование списков. (Дистанционно) | **1** | **1** | **0** |
| **16** | Функции. Полезные функции. Создание и вызов функций. Передача данных в функцию. Получение данных из функции. | **5** | **1** | **4** |
|  | Функции. Полезные функции. Создание и вызов функций. Передача данных в функцию. Получение данных из функции. (Дистанционно) | **1** | **1** | **0** |
| **17** | **Индивидуальный творческий проект** | **40** | **0** | **40** |

**Всего 134 ч.**