

АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ
ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
«Ес Программирование»

«Утверждаю»

Директор АНО ДО

«Ес Программирование»

И.В. Полоников

« 01 » Сентябрь 2024г



**ДОПОЛНИТЕЛЬНАЯ ОБЩЕОБРАЗОВАТЕЛЬНАЯ
ПРОГРАММА – ДОПОЛНИТЕЛЬНАЯ
ОБЩЕРАЗВИВАЮЩАЯ ПРОГРАММА
ПО ПРОГРАММИРОВАНИЮ ДЛЯ ДЕТЕЙ
ШКОЛЬНОГО ВОЗРАСТА**

**«Основы программирования на языке Python»
(11-16 ЛЕТ)**

Екатеринбург – 2024

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

Дополнительная общеобразовательная программа «Основы программирования на языке Python» разработана на основе следующих нормативно-правовых документов:

1. Федеральный Закон «Об образовании в Российской Федерации» от 29.12.2012 № 273-ФЗ.
2. Концепция развития дополнительного образования детей (утверждена распоряжением Правительства РФ от 31 марта 2022 г. № 678-р).
3. Постановлением Главного государственного санитарного врача РФ от 28.09.2020 года №28 «Об утверждении санитарных правил СП 2.4. 3648-20 «Санитарно-эпидемиологические требования к организациям воспитания и обучения, отдыха и оздоровления детей и молодежи» (зарегистрированного от 18.12.2020 №61573).
4. Письмо Минобрнауки России от 18.11.2015 № 09-3242 «О направлении информации» (вместе с «Методическими рекомендациями по проектированию дополнительных общеразвивающих программ (включая разноуровневые программы)).

Направленность дополнительной общеобразовательной программы – *техническое*

Актуальность программы состоит в том, что активизация познавательного процесса позволяет учащимся более полно выражать свой творческий потенциал и реализовывать собственные идеи в изучаемой области знаний, создаёт предпосылки по применению информационных компетенций в других учебных курсах, а также способствует возникновению дальней мотивации, направленной на освоение профессий, связанных с разработкой программного обеспечения.

Курс служит средством внутрипрофильной специализации в области новых информационных технологий, что способствует созданию дополнительных условий для проявления индивидуальных образовательных интересов учащихся, их дальнейшей профессиональной ориентации.

Отличительные особенности программы:

Ключевой особенностью курса является его направленность на формирование у учащихся навыков поиска собственного решения поставленной задачи, составления алгоритма решения и его реализации с помощью средств программирования.

Целью программы является обучение программированию на ЯП Python, через создание творческих и прикладных проектов.

Задачи:

1. Изучить базовые понятия: алгоритм, блок-схема, переменная, цикл, условия, вычисляемая функция;
2. Сформировать навыки выполнения технологической цепочки разработки программ средствами языка программирования Python;
3. Изучить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками, кортежами, словарями, множествами);
4. Научить применять навыки программирования на конкретной учебной ситуации
5. Привить навыки проектной деятельности.
6. Способствовать расширению словарного запаса.
7. Способствовать развитию памяти, внимания, технического мышления, изобретательности.
8. Способствовать развитию алгоритмического мышления.
9. Способствовать формированию интереса к техническим знаниям.
10. Способствовать формированию умения практического применения полученных знаний.
11. Сформировать умение формулировать, аргументировать и отстаивать своё мнение.
12. Сформировать умение выступать публично с докладами и презентациями.
13. Воспитывать аккуратность и дисциплинированность при выполнении работы.

14. Способствовать формированию положительной мотивации к трудовой деятельности.
15. Способствовать формированию опыта совместного и индивидуального творчества при выполнении командных заданий.

Программа адресована для детей от 11 лет до 16 лет, через индивидуальные, групповые занятия или в формате онлайн-обучения. Рекомендуемая частота занятий – 1 раз в неделю по 2 академических часа.

Формы организации образовательного процесса

- индивидуальная
- групповая
- онлайн

Виды занятий:

- теоретические
- практические

Ожидаемые результаты и способы определения их *результативности*:

Знать:

- отдельные способы планирования деятельности;
- составление плана предстоящего проекта в виде рисунка, схемы;
- базовый синтаксис языка программирования Python.

Уметь:

- составить план проекта, включая: выбор темы; анализ предметной области; разбиение задачи на под-задачи; проанализировать результат и сделать выводы; найти и исправить ошибки; - подготовить небольшой отчет о работе; публично выступить с докладом; - наметить дальнейшие пути развития проекта;
- реализовывать простые алгоритмы с помощью языка программирования Python;

иметь первичные навыки:

- работы в группе;
- ведения дискуссии;
- донесения своих мыслей до других.

Ожидаемые результаты обучения по программе

Учащиеся овладевают следующими знаниями, умениями и способами деятельности:

- знают принципы и структуру Python проектов, формы представления и управления информацией в проектах;
- умеют спроектировать, изготовить и запустить проект на языке программирования Python;
- владеют способами работы с изученными программами;
- знают и умеют применять при создании Python проектов основы организации кода.
- способны осуществлять рефлексивную деятельность, оценивать свои результаты, корректировать дальнейшую деятельность по разработке Python проектов.

Программа рассчитана на 68 часов (2 часа в неделю). Занятия разделены на академические часы (40 минут) с перерывами между ними по 10 минут. Занятия индивидуальные, групповые или в формате онлайн обучения.

Форма подведения итогов реализации программы:

Защита творческих работ, созданных с помощью языка программирования Python. Представление проектов через размещение на портале GitHub.

Календарный учебный график

№	Месяц	Форма занятия	Кол-во часов	Раздел, тема занятия	Форма контроля
				Раздел I. Введение в программирование на Python	
1	Сентябрь	Учебное занятие	2	Введение в курс. Инструктаж по ТБ. Знакомство со средой PyCharm IDE	Тестирование, практическое задание
2	Сентябрь	Учебное занятие	2	Знакомство с историей Python. Установка Python и запуск PyCharm IDE	Наблюдение, практическое задание
3	Сентябрь	Учебное занятие	2	Программа №1 «Бесполезные факты». Основы ввода-вывода данных и типы данных	Наблюдение, практическое задание
4	Сентябрь	Учебное занятие	2	Переменные и строковые методы. Создание программы «Бесполезные факты»	Тестирование, практическое задание
				Раздел II. Условные конструкции и циклы	
5	Октябрь	Учебное занятие	2	Программа №2 «Угадай число». Генерирование случайных чисел	Наблюдение, практическое задание
6	Октябрь	Учебное занятие	2	Условные конструкции: if, else, elif и составные условия	Тестирование, практическое задание
7	Октябрь	Учебное занятие	2	Цикл while и создание программы «Угадай число»	Наблюдение, практическое задание
8	Октябрь	Учебное занятие	2	Итоговая отладка программы «Угадай число»	Наблюдение, ответы на вопросы
				Раздел III. Циклы и работа со строками	
9	Ноябрь	Учебное занятие	2	Программа №3 «Анаграммы». Циклы for и работа со строками	Наблюдение, практическое задание
10	Ноябрь	Учебное занятие	2	Индексация и срезы строк. Кортежи и их использование	Наблюдение, ответы на вопросы
11	Ноябрь	Учебное занятие	2	Создание игры «Анаграммы». Итоговая отладка	Тестирование, практическое задание
12	Ноябрь	Учебное занятие	2	Подведение итогов работы с играми: «Анаграммы»	Наблюдение, практическое задание
				Раздел IV. Работа со списками и словарями	
13	Декабрь	Учебное занятие	2	Программа №4 «Загадочное слово». Введение в списки и списочные методы	Тестирование, практическое задание
14	Декабрь	Учебное занятие	2	Словари и их использование в Python	Наблюдение, практическое задание
15	Декабрь	Учебное занятие	2	Создание игры «Загадочное слово». Итоговая отладка	Тестирование, практическое задание
16	Декабрь	Учебное занятие	2	Подведение итогов работы с «Виселицей»	Наблюдение, практическое задание

				Раздел V. Функции в Python	
17	Январь	Учебное занятие	2	Программа №5 «Крестики-нолики». Функции, параметры и возвращаемые значения	Тестирование, практическое задание
18	Январь	Учебное занятие	2	Локальные и глобальные переменные. Создание плана игры	Наблюдение, ответы на вопросы
19	Январь	Учебное занятие	2	Написание функций для игры «Крестики-нолики». Итоговая отладка	Наблюдение, практическое задание
20	Январь	Учебное занятие	2	Подведение итогов работы с «Крестиками-ноликами»	Тестирование, практическое задание
				Раздел VI. Работа с файлами и обработка исключений	
21	Февраль	Учебное занятие	2	Программа №6 «Викторина». Чтение и запись файлов	Тестирование, практическое задание
22	Февраль	Учебное занятие	2	Обработка исключений. Создание плана игры	Наблюдение, практическое задание
23	Февраль	Учебное занятие	2	Реализация игры «Викторина». Тестирование и отладка	Наблюдение, ответы на вопросы
24	Февраль	Учебное занятие	2	Подведение итогов работы с «Викториной»	Практическое задание
				Раздел VII. Программные объекты.	
25	Март	Учебное занятие	2	Основы объектно-ориентированного подхода	Практическое задание
26	Март	Учебное занятие	2	Применение конструкторов	Наблюдение, практическое задание
27	Март	Учебное занятие	2	Знакомство с программой «Зверюшка с конструктором»	Наблюдение, ответы на вопросы
28	Март	Учебное занятие	2	Создание и вызов статических методов	Практическое задание
				Раздел VII. Проектная деятельность	
25	Апрель	Учебное занятие	2	Проблематизация и целеполагание. Поиск решения и планирование проекта	Наблюдение, обсуждение
26	Апрель	Учебное занятие	2	Реализация замысла проекта. Начальный этап	Наблюдение, практическая работа
27	Апрель	Учебное занятие	2	Реализация замысла. Основной этап	Наблюдение, практическая работа
28	Апрель	Учебное занятие	2	Завершение проекта. Презентация и рефлексия	Презентация проекта, рефлексия
				Итоговая аттестация	
29	Май	Учебное занятие	2	Подготовка к защите проектов, добавление результатов на GitHub	Практическое задание
29	Май	Учебное занятие	2	Защита проектов	Презентация, защита проекта

Учебный план

№	Наименование тем	Кол-во часов	Теория	Практика	Формы аттестации/контроля
	Раздел I. Введение в программирование на Python				
1	Введение в курс. Инструктаж по ТБ. Знакомство со средой PyCharm IDE	2	1	1	Тестирование, решение практических задач
2	Знакомство с историей Python. Установка Python и запуск PyCharm IDE	2	1	1	Наблюдение, практическое задание
3	Программа №1 «Бесполезные факты». Основы ввода-вывода данных и типы данных	2	1	1	Наблюдение, практическое задание
4	Переменные и строковые методы. Создание программы «Бесполезные факты»	2	1	1	Тестирование, практическое задание
	Раздел II. Условные конструкции и циклы				
5	Программа №2 «Угадай число». Генерирование случайных чисел	2	1	1	Наблюдение, практическое задание
6	Условные конструкции: if, else, elif и составные условия	2	1	1	Тестирование, решение практических задач
7	Цикл while и создание программы «Угадай число»	2	1	1	Наблюдение, практическое задание
8	Итоговая отладка программы «Угадай число»	2		2	Наблюдение, ответы на вопросы
	Раздел III. Циклы и работа со строками				
9	Программа №3 «Анаграммы». Циклы for и работа со строками	2	1	1	Наблюдение, практическое задание
10	Индексация и срезы строк. Кортежи и их использование	2	1	1	Наблюдение, ответы на вопросы
11	Создание игры «Анаграммы». Итоговая отладка	2		2	Тестирование, решение практических задач
12	Подведение итогов работы с играми: «Анаграммы»	2		2	Наблюдение, практическое задание
	Раздел IV. Работа со списками и словарями				
13	Программа №4 «Загадочное слово». Введение в списки и списочные методы	2	1	1	Тестирование, решение практических задач
14	Словари и их использование в Python	2	1	1	Наблюдение, практическое задание
15	Создание игры «Загадочное слово». Итоговая отладка	2		2	Тестирование, решение практических задач
16	Подведение итогов работы с «Виселицей»	2		2	Наблюдение, практическое задание

	Раздел V. Функции в Python				
17	Программа №5 «Крестики-нолики». Функции, параметры и возвращаемые значения	2	1	1	Тестирование, решение практических задач
18	Локальные и глобальные переменные. Создание плана игры	2	1	1	Наблюдение, ответы на вопросы
19	Написание функций для игры «Крестики-нолики». Итоговая отладка	2		2	Наблюдение, практическое задание
20	Подведение итогов работы с «Крестиками-ноликами»	2		2	Тестирование, решение практических задач
	Раздел VI. Работа с файлами и обработка исключений				
21	Программа №6 «Викторина». Чтение и запись файлов	2	1	1	Тестирование, решение практических задач
22	Обработка исключений. Создание плана игры	2	1	1	Наблюдение, практическое задание
23	Реализация игры «Викторина». Тестирование и отладка	2		2	Наблюдение, ответы на вопросы
24	Подведение итогов работы с «Викториной»	2		2	Практическое задание
	Раздел VII. Программные объекты				
25	Основы объектно-ориентированного подхода	2	1	1	Практическое задание
26	Применение конструкторов	2	1	1	Наблюдение, практическое задание
27	Знакомство с программой «Зверюшка с конструктором»	2		2	Наблюдение, ответы на вопросы
28	Создание и вызов статических методов	2		2	Практическое задание
	Раздел VIII. Проектная деятельность				
29	Проблематизация и целеполагание. Поиск решения и планирование проекта	2	1	1	Наблюдение, обсуждение
30	Реализация замысла проекта. Начальный этап	2		2	Наблюдение, практическая работа
31	Реализация замысла. Основной этап	2		2	Наблюдение, практическая работа
32	Завершение проекта. Презентация и рефлексия	2		2	Презентация проекта, рефлексия
	Итоговая аттестация				
33	Подготовка к защите проектов. Добавление результатов на GitHub	2		2	Практическое задание
34	Защита проектов	2		2	Презентация, защита проекта

Содержание изучаемого курса:

1. Введение в курс. Инструктаж по ТБ. Знакомство со средой программирования PyCharm IDE

- **Теория:** Знакомство с историей Python, объектами его применения.
 - **Практика:** Установка Python и запуск PyCharm IDE.
2. **Программа №1. «Бесполезные факты»**
 - **Теория:** Основы ввода-вывода данных, типы данных, переменные, строковые методы.
 - **Практика:** Создание программы «Бесполезные факты».
 3. **Программа №2. «Угадай число»**
 - **Теория:** Генерирование случайных чисел, условные конструкции if, else, elif, составные условия, цикл while.
 - **Практика:** Создание программы «Угадай число» и её итоговая отладка.
 4. **Программа №3. «Анаграммы»**
 - **Теория:** Циклы for, работа со строками (индексация, срезы), кортежи.
 - **Практика:** Создание игры «Анаграммы» и итоговая отладка.
 5. **Программа №4. «Загадочное слово»**
 - **Теория:** Списки, списочные методы, словари, их использование в Python.
 - **Практика:** Создание игры «Загадочное слово» и её отладка.
 6. **Программа №5. «Крестики-нолики»**
 - **Теория:** Функции, параметры и возвращаемые значения, локальные и глобальные переменные.
 - **Практика:** Написание функций для игры «Крестики-нолики», запуск и отладка игры.
 7. **Программа №6. «Викторина»**
 - **Теория:** Чтение и запись в текстовые файлы, обработка исключений.
 - **Практика:** Создание и отладка игры «Викторина».
 8. **Программные объекты**
 - **Теория:** Основы объектно-ориентированного программирования, применение конструкторов, создание статических методов.
 - **Практика:** Разработка программы «Зверюшка с конструктором» и отладка.
 9. **Проектный модуль**
 - **Теория:** Проблематизация, целеполагание, поиск решения и планирование проекта.
 - **Практика:** Реализация, завершение проекта и его презентация.
 10. **Итоговая аттестация**
 - **Практика:** Подготовка к защите проекта, защита и добавление результатов на GitHub.

Методическое обеспечение дополнительной общеобразовательной программы дополнительного образования детей:

1. Доусон М. Программируем на Python. – СПб.: Питер, 2014 г. – 416 с..

ТЕХНИЧЕСКОЕ ОСНАЩЕНИЕ

- Ученический стол со стульями.
- Стол и стул (для преподавателя).
- Ноутбук (для преподавателя).
- Мышь компьютерная 3шт.
- Ноутбук (для ученика) 2шт.

Ожидаемые результаты и способы определения их результативности

По итогам реализации программы, учащиеся **будут знать:**

- основы дизайнерского оформления созданных программ;
- принципы программирования на языке Python.

По итогам реализации программы, учащиеся **будут уметь:**

- производить чтение и запись программ на языке Python;
- запускать и отлаживать программу.

Формы определения результативности обучения

Результаты освоения программы отслеживаются по итогам опросов, выполнения практических заданий.

Условия реализации программы

Для успешной реализации программы в очном формате необходим ноутбук с установленным на нем любым современным браузером и компьютерная мышь.

Для успешной реализации программы с дистанционным форматом необходим компьютер с доступом к сети Интернет (желательно со скоростью подключения не меньше 50мб/сек), подключенными колонками (или наушниками). На компьютере должен быть установлен любой современный браузер и платформа Discord.

Организационно-педагогические условия

При проведении занятия с использованием дистанционных образовательных технологий, электронного обучения, в водной его части следует обозначить правила работы и взаимодействия (объяснить учащимся технические особенности работы и правила обмена информацией). В процессе занятия педагогу необходимо четко давать инструкции выполнения заданий.

Список литературы или Интернет источники

1. Доусон М. Програмируем на Python. – СПб.: Питер, 2014 г. – 416 с.
2. Официальный сайт программы [Электронный ресурс] – Режим доступа: <https://docs.python.org/>, свободный.
3. Сайт, среда разработки для языка Python. [Электронный ресурс] – Режим доступа: <https://www.jetbrains.com/pycharm/?fromMenu>, свободный.

Тест №1 по дисциплине Python. «Переменные, ввод/вывод и условные конструкции»

Введите Ваше ФИО:

Ответы, которые Вы считаете верными, необходимо обвести в круг.

Что такое переменная?

- 1) Пустая ячейка памяти
- 2) Именованная ячейка памяти, которая хранит какую-то информацию
- 3) Символ
- 4) Ячейка для изменения данных

Выберите правильные названия переменных в python? (2 ответа)

- 1) myName
- 2) my_name"
- 3) 13MyName
- 4) my_name
- 5) ?Name
- 6) my name
- 7) my/name

Как создать переменную в Python?

- 1) С помощью ключевого слова var
- 2) С помощью конструкции [тип данных] [имя переменной]
- 3) С помощью ключевого слова const
- 4) Необходимо просто указать имя для переменной и присвоить ей значение с помощью знака «=»

При помощи какой команды пользователь может вводить данные с клавиатуры?

- 1) Print
- 2) Return
- 3) Elif
- 4) Input

Какая команда отвечает за простейший вывод данных в Python.

- 1) Print
- 2) Return
- 3) Elif
- 4) Input

Какой тип возвращаемого значения у команды input?

- 1) String
- 2) Float
- 3) Integer
- 4) Bool

Какие операторы из перечисленных принадлежат к математическим?

- 1) +
- 2) ;
- 3) *
- 4) :
- 5) ^
- 6) >
- 7) <

Какие операторы из перечисленных принадлежат к операторам сравнения?

- 1) +
- 2) ;
- 3) >=
- 4) >
- 5) <

Какие операторы принадлежат к условиям (ветвлению)?

- 1) While
- 2) For
- 3) If
- 4) Elif
- 5) Return

Что выведет данная программа

```
a = 6
b = 6
if a==b:
    print(2*a)
else:
    print(b)
```

Ответ:

Что выведет данная программа

```
if 3>4:
    print(2)
else:
    print(1)
```

Ответ:

Что выведет данная программа

```
a = True
b = False
if a or b:
    print(1)
else:
    print(2)
```

Ответ:

Тест №2 по дисциплине Python. «Циклы While и For»

Введите Ваше ФИО:

Ответы, которые Вы считаете верными, необходимо обвести в круг.

Какие операторы принадлежат к циклам?

- 1) While
- 2) For
- 3) If
- 4) Elif
- 5) Return

Какой из циклов не содержит итератор?

- 1) while
- 2) for

До какого момента будет работать цикл while?

- 1) Пока итератор больше нуля
- 2) Пока не будет вызвана функция остановки
- 3) Пока значение предусловия равно True

После какого оператора указывается перебираемая последовательность в цикле for?

- 1) of
- 2) for
- 3) do
- 4) in

Сколько раз повториться следующий цикл:

```
for i in range(0. 50. 5)
```

- 1) 9
- 2) 50
- 3) 10
- 4) 11

Что выведет данная программа

```
for i in range(2,7,2):  
    print(i)
```

Ответ:

Что выведет данная программа

```
for i in range(5,9,4):  
    print(i)
```

Ответ:

Что выведет данная программа

```
for i in range(3):  
    print(i)
```

Ответ:

Что выведет данная программа

```
a = 2
while(a<5):
    a+=2
print(a)
```

Ответ:

Что выведет данная программа

```
a = 2
while(a<5):
    a+=2
print(a)
```

Ответ:

Что выведет данная программа

```
a = 3
b = 0
while(a<6):
    a+=4
    b+=a
print(b)
```

Ответ:

Тест №3 по дисциплине Python. «Кортежи, списки, словари»

Введите Ваше ФИО:

Ответы, которые Вы считаете верными, необходимо обвести в круг.

1. Что такое списки в Python?

- 1) Программа-конструктор для создания списка покупок
- 2) Модель хранения данных одного типа
- 3) Модель хранения данных разных типов
- 4) Модель хранения набора любых данных

2. Какие символы помогают получить определённый элемент списка?

- 1) {n}
- 2) -n-
- 3) (n)
- 4) [n]

3. Что выведет данная программа

```
L = [7,2,3,1]  
print(L[2])
```

Ответ:

4. Какие особенности есть у кортежей?

- 1) После определения кортежа, данные можно изменить
- 2) После определения кортежа, данные нельзя изменить
- 3) Особенности нет

5. Какие особенности есть у словарей?

- 1) После определения словаря, его значения изменять нельзя
- 2) В словарь можно складывать только данные одного типа
- 3) Данные в словаре хранятся в формате ключ: значение
- 4) Особенности нет

6. В списке

```
colors = ["blue", "red", "Purple"]
```

под каким индексом стоит элемент "red"?

- 1) 1
- 2) 0
- 3) 3
- 4) 2

7. Какой метод используется для возвращения ключей словаря?

- 1) keys()
- 2) get_keys()
- 3) keys_list()
- 4) key()

8. Что выведет данная программа

```
d = {'foo': 100, 'bar': 200, 'baz': 300}
print(d['foo'])
```

Ответ:

9. Как получить доступ к элементу 'qux' списка:

```
a = ['foo', 'bar', 'baz', 'qux', 'quux', 'corge']
```

Ответ:

Тест №4 по дисциплине Python. «Итоговый»

Введите Ваше ФИО:

1. Что такое переменная?

- ☐ Переменная – это значение результата работы программы
- ☐ Переменная – часть кода, в которой указывается смысл программы
- ☐ Переменная – название для операторов в коде
- ☐ Переменная – ячейка памяти в компьютере, в которой хранится значение

2. Какой тип данных обозначает целое число?

- ☐ Int
- ☐ Str
- ☐ Bool
- ☐ Float

3. Какой тип данных обозначает логическую переменную?

- ☐ Int
- ☐ Str
- ☐ Bool
- ☐ Float

4. Выберите специальный символ, который обозначает комментарий в python?

- ☐ /*
- ☐ //
- ☐ #
- ☐ <!--

5. Выберите все варианты где имя переменной создано без ошибок

- ☐ myName
- ☐ my_name"
- ☐ 13MyName
- ☐ my_name
- ☐ ?Name
- ☐ my name
- ☐ my/name

6. После запуска программы, что будет выведено на экран?

```
1  x = 5
2  y = 10
3
4  if x < y:
5      y = x
6  print('x =', x, 'y =', y)
```

- ☐ x = 5 y = 10
- ☐ x = 10 y = 5
- ☐ x = 5 y = 5
- ☐ x = 10 y = 10

7. Выберите математический оператор, который позволит получить остаток от деления?

- ☐ /
- ☐ %
- ☐ //
- ☐ /\

8. Какое выражение верно?

- ☐ "x = x * 3" то же самое, что и "x *= x * 3"
- ☐ "x = x * 3" то же самое, что и "x *= x"
- ☐ "x = x + 3" то же самое, что и "x += 3"
- ☐ "x = x - 3" то же самое, что и "-= x - 3"

9. Вы вводите два числа: $x = 5$ $y = 5$ Что будет выведено на экран?

```
1  x = input()
2  y = input()
3
4  print(x + y)
```

☐ 10

☐ 55

☐ 25

☐ x+y

10. С помощью какой функции можно перевести строку в целое число?

☐ str()

☐ int()

☐ bool()

☐ float()

11. С помощью какой функции можно перевести целое число в строку?

☐ str()

☐ int()

☐ bool()

☐ float()

12. Какой оператор обозначает условие?

- ☐ for
- ☐ if
- ☐ while
- ☐ int

13. Выберите все операторы, которые принадлежат операторам сравнения?

Отметьте все подходящие варианты.

- ☐ +
- ☐ ==
- ☐ >
- ☐ *=
- ☐ !=
- ☐ !

14. Что будет выведено на экран?

```
user_input = ""
password = "secret"

if user_input == password:
    print('Доступ открыт')
elif not user_input:
    print('Вы не ввели пароль')
else:
    print('Доступ закрыт')
```

- ☐ Secret
- ☐ Вы не ввели пароль
- ☐ доступ открыт
- ☐ доступ закрыт

15. Какой вариант условной конструкции создан правильно?

☐ `x = 10`
`y = 5`

```
if x < y:
    print(x + y)
else:
    print(x ** y)
```

☐ `x = 10`
`y = 5`

```
if x < y
    print(x + y)
else:
    print(x ** y)
```

☐ `x = 10`
`y = 5`

```
if x < y then
    print(x + y)
else:
    print(x ** y)
```

☐ `x = 10`
`y = 5`

```
if x < y:
    print(x + y)
else:
    print(x ** y)
```

16. Что такое цикл?

- ☐ Остановка программы
- ☐ Ничего не выполнять
- ☐ Блок кода с именем
- ☐ Многократное выполнение код

17. Что означает оператор `or`?

- ☐ логическое ИЛИ
- ☐ проверка условия
- ☐ логическое РАВНО
- ☐ Логические И

18. Что означает логический оператор ==

- ☐ Не равно
- ☐ Больше
- ☐ Равно
- ☐ Меньше

19. Что делает функция len

- ☐ преобразовывает строку в нижний регистр
- ☐ очищает строку
- ☐ преобразует строку в верхний регистр
- ☐ она принимает любую последовательность в качестве аргумента и возвращает её длину

20. Как получить элемент из списка

- ☐ по ключу
- ☐ по индексу
- ☐ с помощью метода .get()

21. В чем разница между title() и upper()?

- ☐ title() первую букву переводит в верхний регистр, а все остальные в нижний, а upper() - переводит все буквы в верхний регистр
- ☐ title() – возвращает количество символов в первом слове, upper() – количество символов в последнем слове
- ☐ нет разницы
- ☐ title() преобразует строку в верхний регистр, upper() – в нижний

22. Каким ключевым словом обозначают функцию?

- ☐ с помощью инструкции func
- ☐ с помощью инструкции function
- ☐ с помощью инструкции def
- ☐ с помощью инструкции f

23. Что делает `import math`?

- ☐ Ничего не делает
- ☐ подключает модуль math
- ☐ инициализирует программу