**Информация об уроке**

**Модуль: 1**

**Номер занятия: 4**

**Тип занятия: Лекция**

**Название занятия/ Тема занятия:** Введение в Python. Синтаксис, консольные вывод и ввод

**Цель занятия:** изучить базовые настройки Python.

**Образовательные результаты:** понятие алгоритма и алгоритмического языка программирования. Научиться писать первые программы на языке Python.

**Глоссарий:**

**Конспект занятия**

1. **Приветствие**

Приветствуем вас на новом курсе по Python!

1. **Тема урока и целеполагание**

Сегодня мы пройдём: **Введение в язык программирования Python, синтаксис, консольные ввод и вывод**

Эти темы являются основополагающими не только в Python, но и в целом во всех языках программирования. Поняв логику работы программ, перейти с одного языка на другой достаточно просто, ведь всё изучение будет упираться в отличия синтаксиса между разными языками программирования.

1. **Актуализация**

На прошлом уроке мы говорили о том, что существуют различные языки программирования. Сегодня мы приступаем непосредственно к изучению одного из таких языков программирования - Python.

**язык программирования** – знаковая система, позволяющая программисту работать с компьютером.

Высокоуровневый язык программирования - язык, в котором написание программ сводится к написанию кода при помощи слов, которые затем переводятся в машинный код, который исполняет компьютер.

Программа - это алгоритм, описанный кодом.

1. **Основное содержание**

**1. Введение в язык программирования Python, синтаксис.**

Для того, чтобы понять, что такое алгоритм, нам надо разобраться с тем, что такое инструкция.

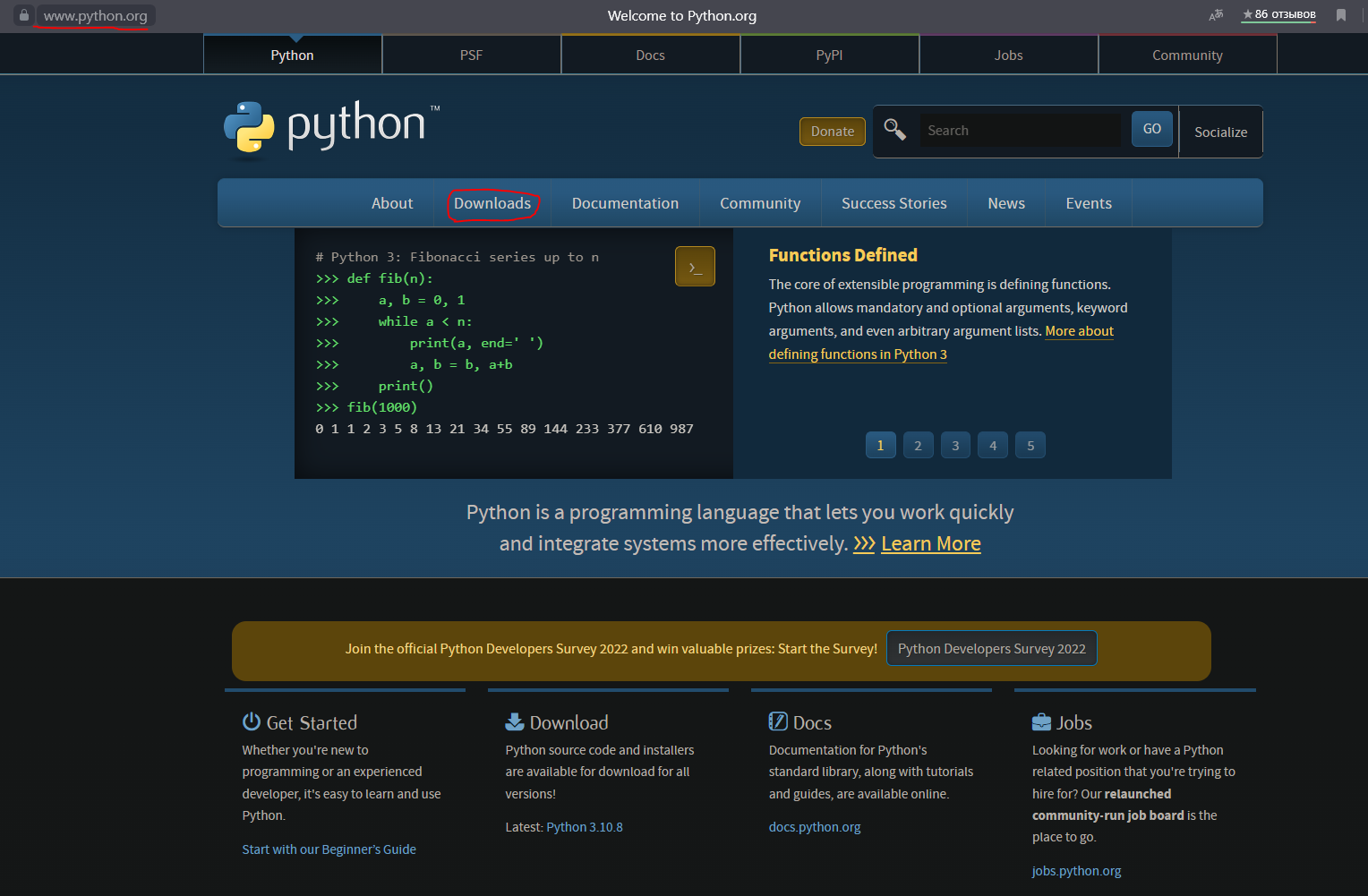
**Инструкция** - некоторое действие. Оно может означать математическую операцию, сравнение или сохранение / удаление данных из переменной. Программы состоят из последовательного набора таких инструкций (действий).

Первая ассоциация со словом алгоритм – что-то, что повторяется постоянно, последовательно одно за другим.

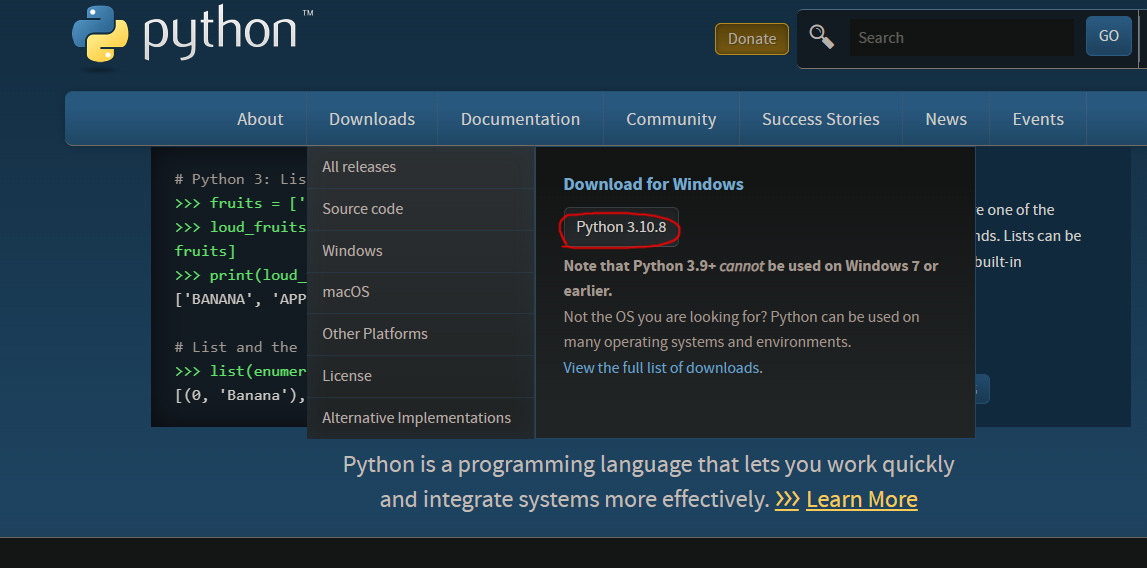
**Алгоритм** – последовательная совокупность заданных инструкций, которая приводит к решению задачи.

Теперь мы знаем, что алгоритм это всего лишь набор инструкций, который выполняет компьютер.

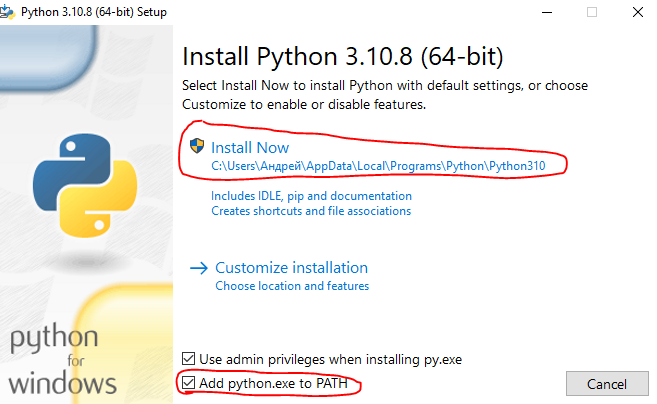
Для написания программ будем использовать оригинальный язык Python с официального сайта:



нажимаем на кнопку download и нажимаем на кнопку (обведена красным):

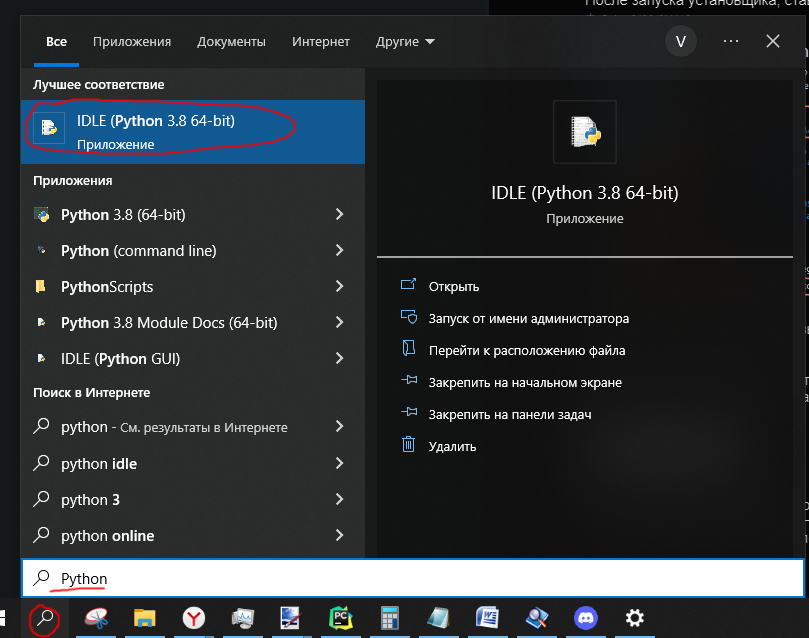


После запуска установщика, ставим галочку напротив Add launcher to the path:

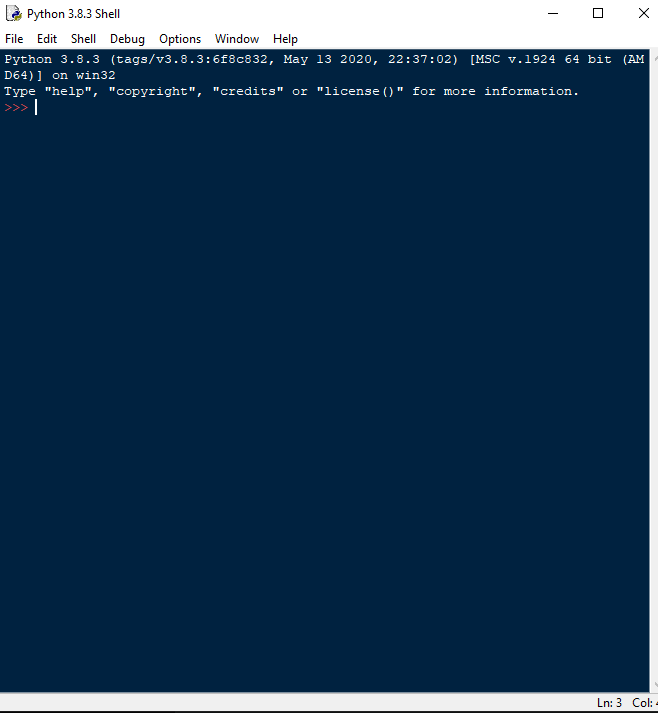


Поздравляем! Вы установили язык Python

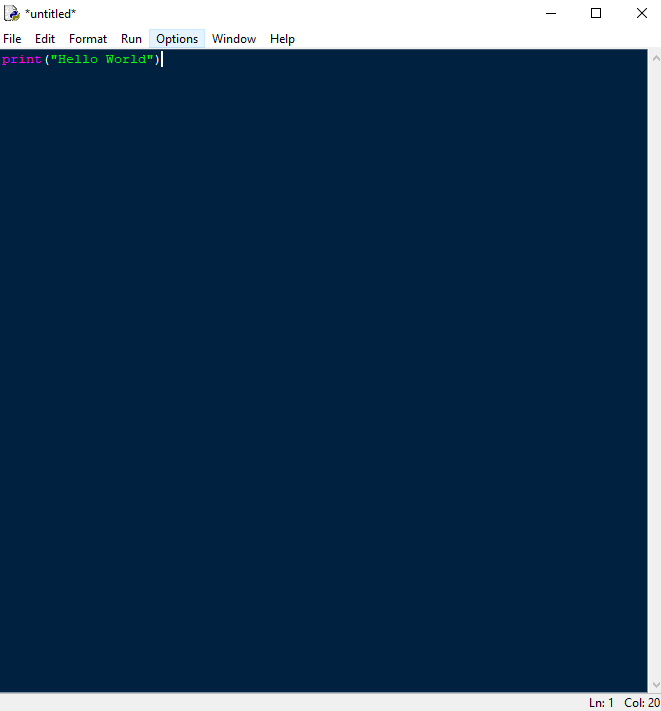
Ищем в поиске windows Python:



Только что мы запустили консоль Python:



Для создания программы, нажимаем слева сверху на file - new file:



Все программы, мы будем писать здесь.

Программа на языке Python состоит из набора инструкций. Каждая инструкция помещается на новую строку. Например:

https://lh6.googleusercontent.com/RDnl5ceG3UJKrcP1YNdrcHoS0Ri_cW7fwT0qKzu0QxHh3ygRjJpiRaaBZXYPXFqSQpqD_W_0z9Yl7f_UR1j1hRKFzpnKXNhabCod5XXA4ifIc5gVKWKa0g4hB0zerBP0N6zJwIKePuNR0UwTDu6slkHkOxOcXr_MrFEPNoby6kz4SMURHX5e7NuZiQa_PWEntBY4Pw

Большую роль в Python играют отступы. Неправильно поставленный отступ фактически является ошибкой. Например, в следующем случае мы получим ошибку, хотя код будет практически аналогичен приведенному выше:

https://lh3.googleusercontent.com/ESITY-AcsFQhqkC8Hmwr5u7u1OkSoEaFlqGcugdsOOixLEiFCXseAEocN5yjVUukuPZudTzEfJhUNzxJTpF4heG72K3hR6C5xLcHJFOGeGGu5M5HtQmPwJC5s8j-MdrRO9VBg7K9QpKaubjknIhqFIlySrjhegaaDR-xYOWolFE4_o2zPhURhuoQygnTnDPFZuRTvg

Поэтому стоит помещать новые инструкции сначала строки. В этом одно из важных отличий Python от других языков программирования, как C# или Java.

Однако стоит учитывать, что некоторые конструкции языка могут состоять из нескольких строк. Например, условная конструкция if:

https://lh4.googleusercontent.com/OWeoQMIyIiVQt0PHWLp6WDV7XXcvrZ5X20Gb2-EYD3UgQRXaJssoU2ma4Q4knnCf7Rhr5EWp9HHCpO8rlOvGocEqidtzAz2lHegCTDPtHHH5G8POReidJsvAmWjD0UYNlyk6CO5GeGYRyx35jXPHl6wcNzSFrVhRm4dLtl72-zrUQo0_4wp4xNs2ROGA-yNYxccJTw

В данном случае если 1 меньше 2, то выводится строка "Hello". И здесь уже должен быть отступ, так как инструкция print("Hello") используется не сама по себе, а как часть условной конструкции if. Причем отступ, согласно руководству по оформлению кода, необходимо делать из такого количество пробелов, которое кратно 4 (то есть 4, 8, 16 и т.д.). 1 нажатие клавиши Tab = 4 нажатиям пробела.

Таких конструкций не так много, поэтому особой путаницы по поводу где надо, а где не надо ставить пробелы, не должно возникнуть.

**Регистрозависимость**

Python - регистрозависимый язык, поэтому выражения print и Print или PRINT представляют разные выражения. И если вместо метода print для вывода на консоль мы попробуем использовать метод Print:

https://lh4.googleusercontent.com/eXHLdZC4UgvNevlzCHnANKOfJqNQ3GY8V4b9-eM39EWPf2AJAYWN7wvC9F8xxa1Nn-uY9MGjpeFmK0SyhFpv0SpLjPS4koUmUTrX-Bl9uppHOGT9mLEQkQcc9EsbqsaHymY7-fA5GKJQLcLgfq8aTj93__r_xkPFN6wGWDfvusPw0Rz5CkMQBdDZhL1ynYA3T7eUVw

то у нас ничего не получится.

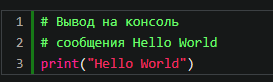
**Комментарии**

Для отметки, что делает тот или иной участок кода, применяются комментарии. При трансляции и выполнении программы интерпретатор игнорирует комментарии, поэтому они не оказывают никакого влияния на работу программы. Комментарии в Python бывают блочные и строчные.

**Интерпретатор –** подпрограмма, которая собирает текстовый код в исполняемый файл. Интерпретатор в процессе выполнения кода (последовательном переходе от одной инструкции к другой) переводит текст в машинный код.

Существует также **компилятор** (применяется в языках со строгой типизацией данных – C, C++, C#, Java, Golang…). Компилятор в отличие от интерпретатора переводит код в исполняемый файл до запуска программы, если в коде будут ошибки, программа не запустится в то время, как интерпретатор будет работать до тех пор, пока не встретит ошибку в коде.

Строчные комментарии предваряются знаком решетки - #. Они могут располагаться на отдельной строке:

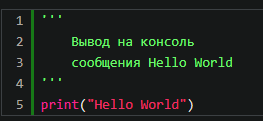


Любой набор символов после знака # представляет комментарий. То есть в примере выше первые две строки кода являются комментариями.

Также они могут располагаться на той же строке, что и инструкции языка, после выполняемых инструкций:

https://lh4.googleusercontent.com/FPmlFcsEjOEKNwjJTtVZ3p0nPesAY8LwG4JXenzDycRaCnPoDMV_6d5r4Bw48-GlviaG8i3glCRLbQ-b2HOFmMK6X5eM1bG7gAJ-teSk625OtcBgxV1om3wNZK89H_8TWCI6reNx6TLSabrhf08yhE3rJ5KEO89fAbZcqYUFg3eFOrnxBv0hrpbLhoq-74L-9sFykw

В блочных комментариях до и после текста комментария ставятся три одинарные кавычки: '''текст комментария'''. Например:



**Консольный ввод и вывод**

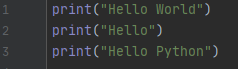
**Вывод на консоль**

Для вывода информации на консоль предназначена встроенная функция **print()**. При вызове этой функции ей в скобках передается выводимое значение:

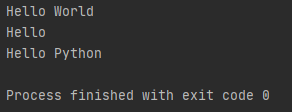


Данный код выведет нам на консоль строку "Hello World".

Отличительной особенностью этой функции является то, что по умолчанию она выводит значение на отдельной строке. Например:



Здесь три вызова функции **print()** выводят некоторое сообщение. Причем при выводе на консоль каждое сообщение будет размещаться на отдельной строке:



Такое поведение не всегда удобно. Например, мы хотим, чтобы все значения выводились на одной строке. Для этого нам надо настроить поведение функции с помощью параметра **end**. Этот параметр задает символы, которые добавляются в конце к выводимой строке. При применении параметра **end** вызов функции **print()** выглядит следующим образом:

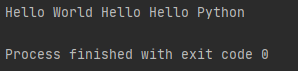


По умолчанию end равен символу "\n", который задает перевод на следующую строку. Собственно поэтому функция print по умолчанию выводит передаваемое ей значение на отдельной строке.

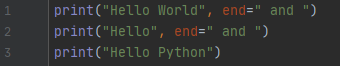
Теперь определим, чтобы функция не делала перевод на следующую строку, а выводила значение на той же строке:



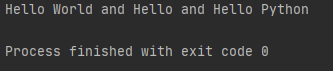
То есть теперь выводимые значения будут разделяться пробелом:



Причем это может быть не один символ, а набор символов:



данном случае выводимые сообщения будут отделяться символами " and ":

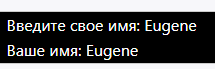


**Консольный ввод**

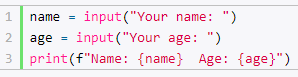
Наряду с выводом на консоль мы можем получать ввод пользователя с консоли, получать вводимые данные. Для этого в Python определена функция **input()**. В эту функцию передается приглашение к вводу. А результат ввода мы можем сохранить в переменную. Например, определим код для ввода пользователем имени:



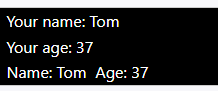
В данном случае в функцию **input()** передается приглашение к вводу в виде строки "Введите свое имя: ". Результат функции - результат ввода пользователя передается в переменную name. Затем мы можем вывести значение этой переменной на консоль с помощью функции **print()**. Пример работы кода:



Еще пример с вводом нескольких значений:



Пример работы программы:



Стоит учитывать, что все введенные значения рассматриваются как значения типа str, то есть строки. И даже если мы вводим число, как в втором случае в коде выше, то Python все равно будет рассматривать введенное значение как строку, а не как число.

Промежуточный итог по части:

* **Инструкция –** действие в программе
* **Алгоритм** – последовательная совокупность заданных инструкций, которая приводит к решению задачи.
* **язык программирования** – знаковая система, позволяющая программисту работать с компьютером.
* Программа на языке Python состоит из набора инструкций. Каждая инструкция помещается на новую строку.
* Большую роль в Python играют отступы. Неправильно поставленный отступ фактически является ошибкой.
* Python - регистрозависимый язык, поэтому выражения print и Print или PRINT представляют разные выражения.
* Для отметки, что делает тот или иной участок кода, применяются комментарии. При трансляции и выполнении программы интерпретатор игнорирует комментарии
* Основной функцией для вывода информации на консоль является функция **print()**
* функция **input** отвечает за ввод информации через консоль

1. **Подведение итогов/Рефлексия**

На этом занятии мы узнали что такое программа, из чего она состоит. Какой синтаксис в языке программирования Python, на что обращать внимание при написании кода. Что такое переменные и типы данных, какие операции существуют в языке и как они работают.