

## Занятие 1. «Введение».

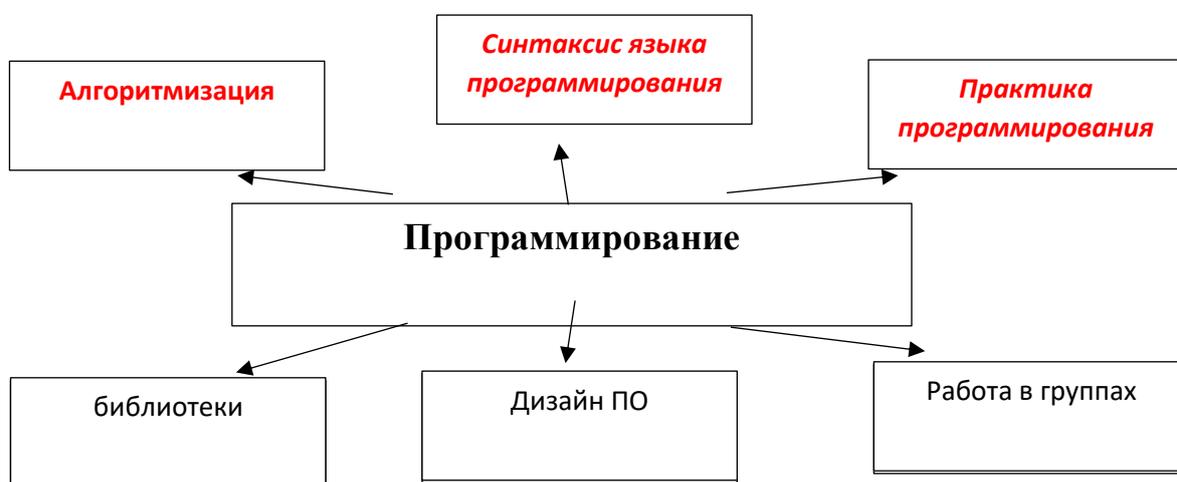
### План

1. Основные задачи курса.
2. Установка программы Python.
3. Первые программы на языке Python.

### 1. Основные задачи курса.

Чтобы научиться программировать, нужно параллельно освоить ряд направлений:

- Основы алгоритмизации, знание алгоритмических структур;
- Синтаксис языков программирования;
- Практика программирования – решение задач;
- Знание библиотек языка – то есть готовых функций;
- Оформление программ, их дизайн;
- Работа в группах.



Немного о каждом направлении.

Программирование, это не просто знание современного языка, это умение решать задачи, строить алгоритмы. Язык программирования – это всего лишь способ записи алгоритма. Поэтому, для первичного изучения основ алгоритмизации выбор языка не столь важен. Языки со временем устаревают, на их место приходят новые. Поэтому нужно понимать, что то, что сегодня мы изучаем и используем, завтра может уйти на второй план. Но, это, на самом деле, не важно. Важно научиться алгоритмически мыслить.

Но мы будем идти в ногу со временем и будем изучать язык Python. Безусловно, язык очень мощный, имеет огромный спектр библиотек, востребованный сегодня на рынке труда программиста.

Курс занятий направлен на изучение:

- основ алгоритмизации,
- изучение синтаксиса языка Python,
- развитие навыков решения задач начиная от постановки задачи, составления алгоритма до непосредственно написания программ.

При этом, конечно, будем использовать некоторые библиотеки и методы, которые реализованы для многих функций.

Работа в группах – это не только обсуждение проблем, мозговой штурм сложных ситуаций. Умение оформлять программы стилистически правильно, подчиняясь единым правилам PEP8 позволит научиться единым требованиям. Они нужны для того, чтобы легче читать (и если надо, исправлять) чужие программы, а во время работы быть взаимозаменяемыми.

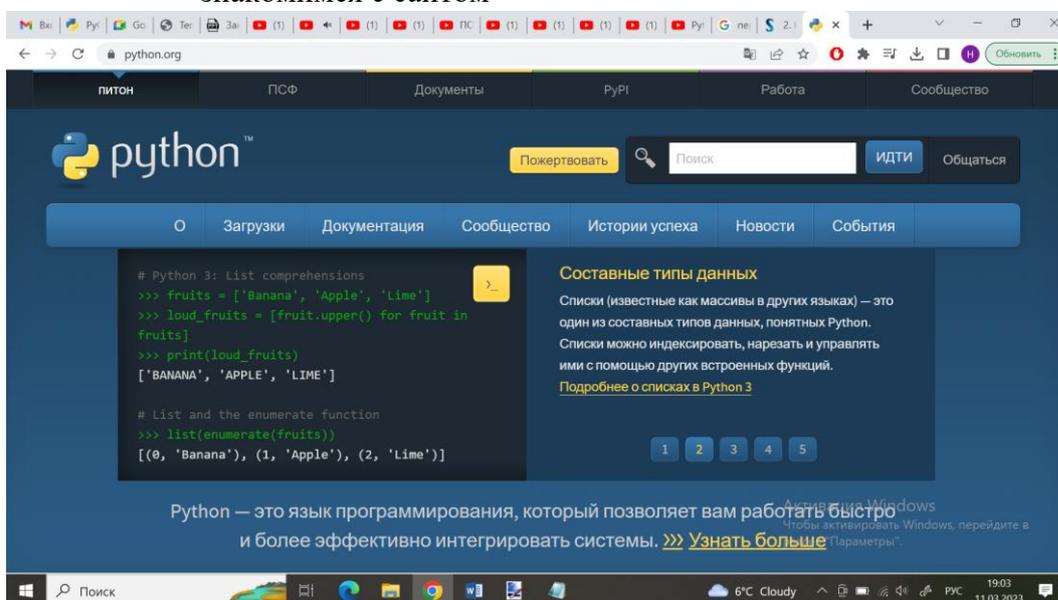
Вопрос оформления программ в текстовом виде – это умение строить диалог между программой и пользователем. Написание программ, которые комментируют ввод данных и вывод результатов подготовит нас к тому, что пользователь должен понимать что от него хочет программа, и что выводит. Это правила хорошего тона. А вот реализацию графического интерфейса оставляем на потом.

Первоначальная задача – научиться алгоритмически мыслить и разбирать задачи на их постановку, математическую основу, алгоритм и только потом – программирование.

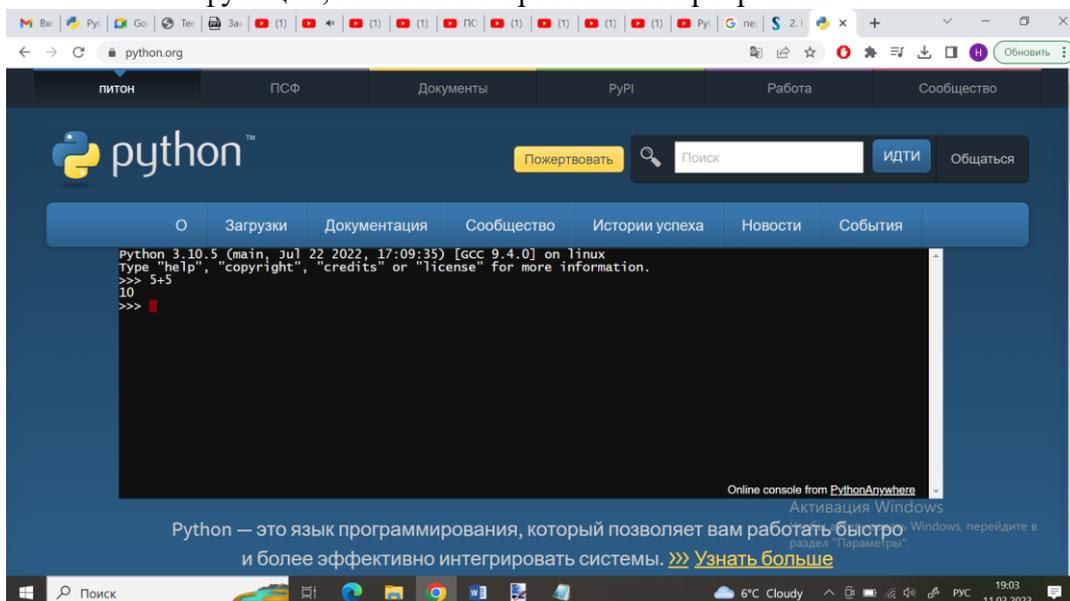
### 3. Инсталляция программы Python

На официальном сайте <https://www.python.org/> есть встроеное окно прямых команд Python, ссылка на бесплатное скачивание среды программирования, документация и многое другое. Сайт постоянно обновляется.

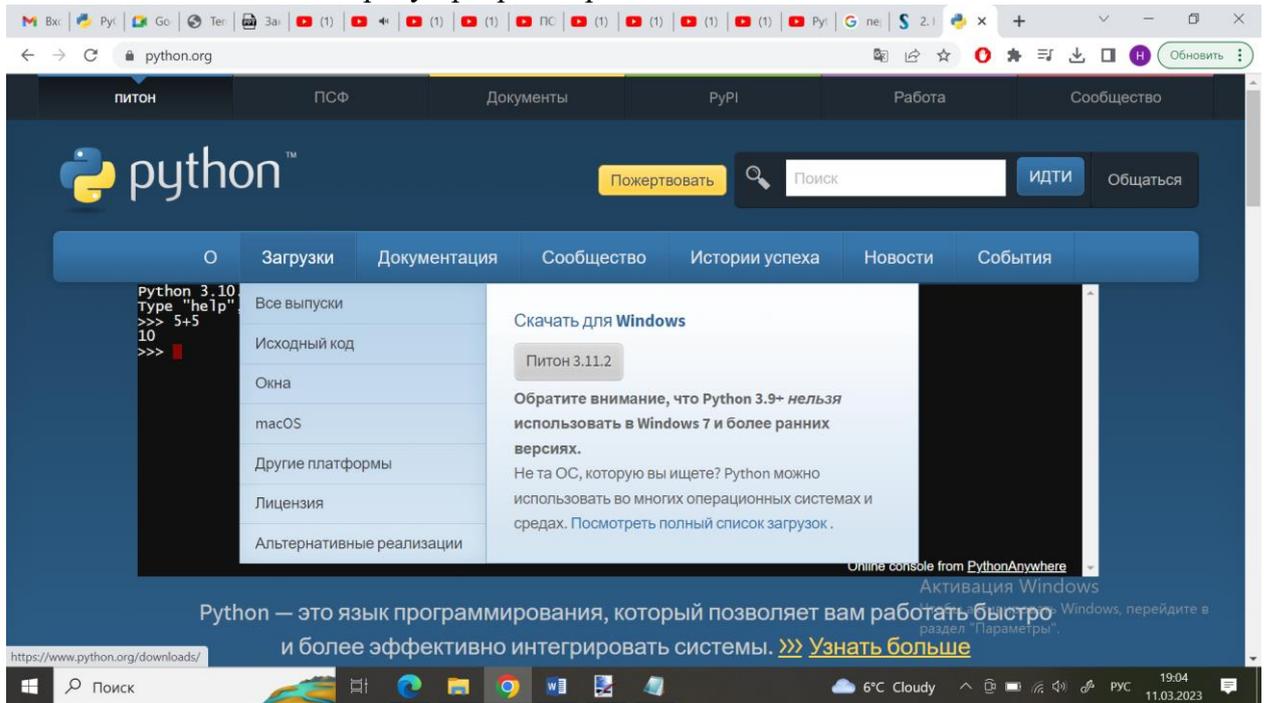
- знакомимся с сайтом



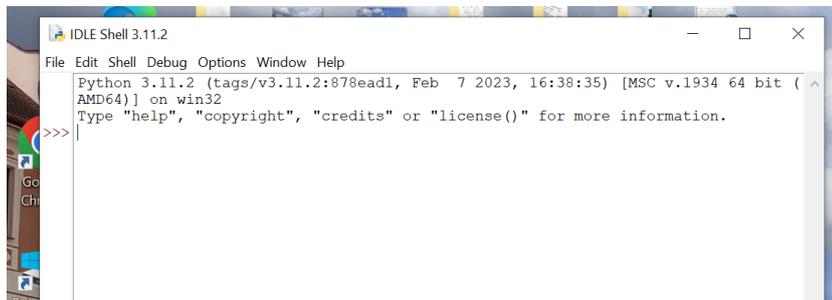
- работаем в интерактивной среде. Можно набрать любое арифметическое выражение 5+5, среда выведет ответ. Здесь можно использовать переменные, функции, записывать простейшие программы.



- скачиваем среду программирования



- устанавливает ПО на своем компьютере, выставляем ярлык на рабочий стол IDLE (Python 3.11 64-bit)
- Запуск программы IDLE (работа в интерактивной среде), изучение основных пунктов меню.



В режиме прямых команд значок >>> означает готовность среды к новой команде.

Если команда неправильная, то Python напишет ошибку (сообщение красным цветом). НЕ БОЙТЕСЬ ОШИБОК! НАУЧИТЕСЬ ИХ ЧИТАТЬ (переводите их)!

Набираем	Ответ среды IDLE	Комментарии
Привет	Среда сообщает об ошибке	Не правильная команда и тип данных
'Привет'	'Привет'	Введена строка
5	5	Введено число
5+2	7	Введено арифметическая операция, среда выполняет ее. Результат – целое число.
'5'+ '2'	52	Арифметическая операция невозможна, так как числа взяты в кавычки и являются строками. Операция называется склеиванием строк. Результат - строка.
'2' + '5'	25	склеивание строк
'Привет ' + 5	Среда сообщает об ошибке	Попытка выполнить арифметическое сложение или

		склеивание невозможно, так как происходит смешивание типов данных
$5 * (20 - 13)$	-15	Выполнена арифметическая операция, выводится ее результат. Результат целое число.
$6 / 3$	2.0	Результат вещественный (дробь)
$5 / 2$	2.5	Результат вещественный (дробь)
$12 / 0$	Среда сообщает об ошибке	Делить на 0 нельзя. Ошибка деления на ноль.

```

Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>> Привет
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    Привет
NameError: name 'Привет' is not defined
>> "Привет"
'Привет'
>> баркадабра
Traceback (most recent call last):
  File "<pyshell#2>", line 1, in <module>
    баркадабра
NameError: name 'баркадабра' is not defined
>> "баратроор"
'баратроор'
>> 5 + 10
15
>> 5 * (20 - 23)
-15
>> 6 / 2
3.0
>> 5 / 2
2.5
>> "Привет" + "абракадабра"
'Приветабракадабра'
>> "5" + "2"
'52'
>> "2" + "5"
'25'
>> 2 + 5
7

```

### 3. Типы данных. Первые команды Python

Существует несколько типов данных. Мы уже познакомились с тремя:

Тип данных	Обозначение на Python	Пример	Комментарий
Целые числа	int	5 2 12 -15	Целые числа знаковые, могут быть положительные и отрицательные
Вещественные числа (с дробной частью)	float	2.5 0.3333378 -7.876543	Числа представленные в виде десятичной дроби. Целая часть числа отделяется от дробной точкой (в математике запятой)
строка	str	'Привет' '5'	

Есть другие типы данных, но о них пойдет речь дальше.

Чтобы проверить какой тип данных, нужно воспользоваться встроенной функцией `type`

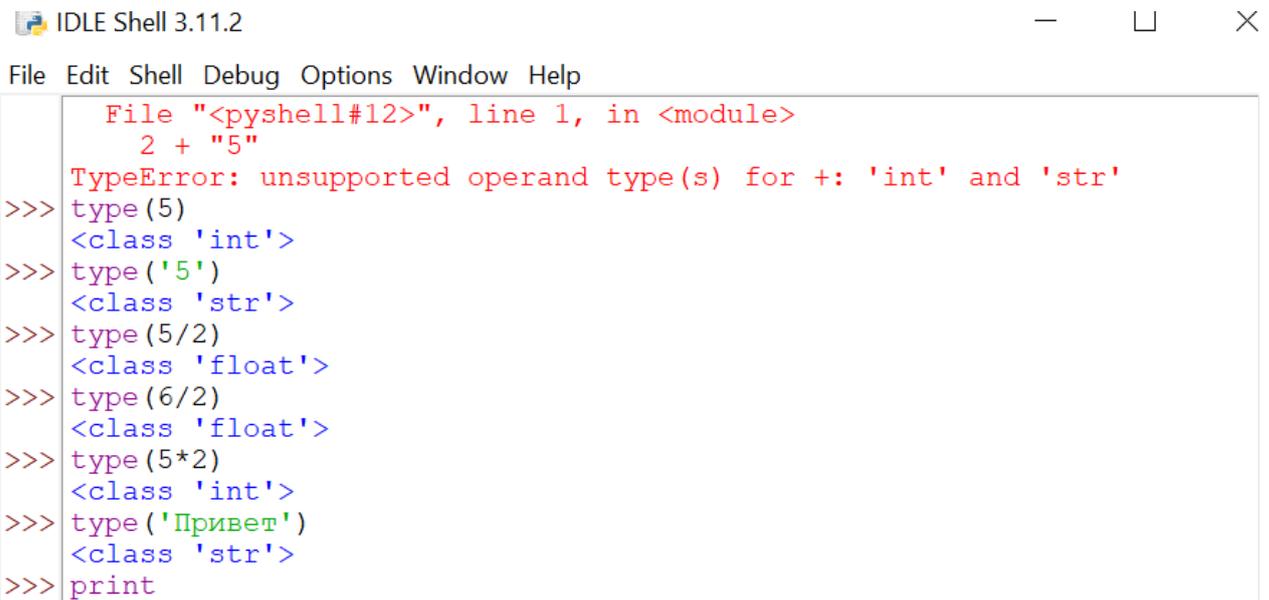
`type(5)` - тип `int` (целое число)

`type('5')` - тип `str` (строка)

`type(5 + 3)` - тип `int` (целое число)

`type(5 / 2)` - тип `float` (вещественное, дробное число)

Проверить все это можно в режиме прямых команд



```
File "<pyshell#12>", line 1, in <module>
  2 + "5"
TypeError: unsupported operand type(s) for +: 'int' and 'str'
>>> type(5)
<class 'int'>
>>> type('5')
<class 'str'>
>>> type(5/2)
<class 'float'>
>>> type(6/2)
<class 'float'>
>>> type(5*2)
<class 'int'>
>>> type('Привет')
<class 'str'>
>>> print
```

**Домашнее задание.** Повторить тот материал, который обсуждался. Поработать в режиме прямых команд: выполнить несколько вычислений, склеиваний, пробовать делать действия, которые приводят к ошибкам. Проверить типы данных с помощью функции `type`.