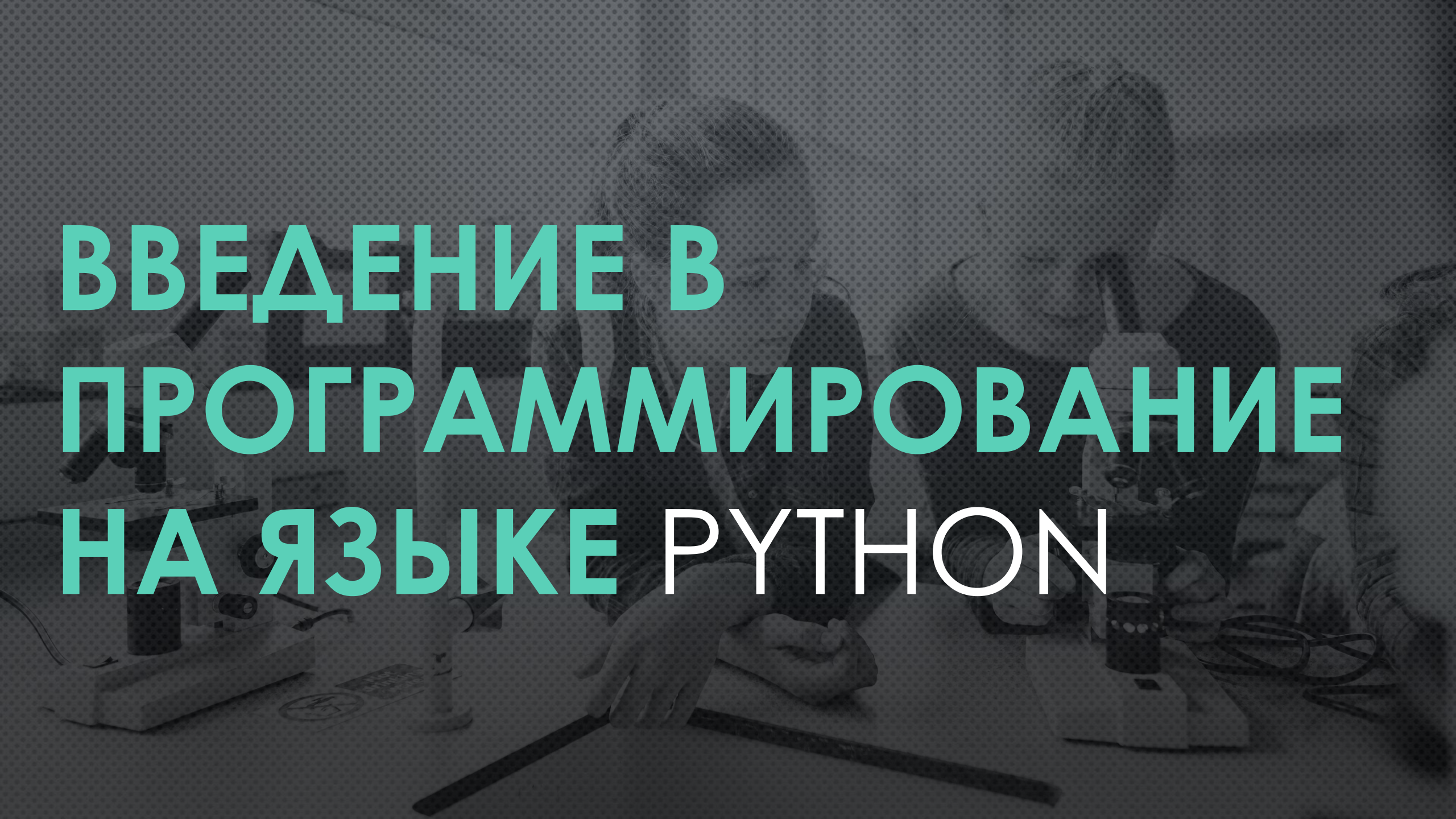
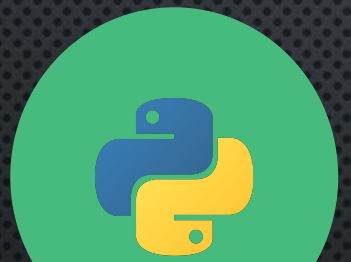

ВВЕДЕНИЕ В
ПРОГРАММИРОВАНИЕ
НА ЯЗЫКЕ PYTHON

ПРИНЦИПЫ ПРОГРАММИРОВАНИЯ



Понятие компьютерной программы

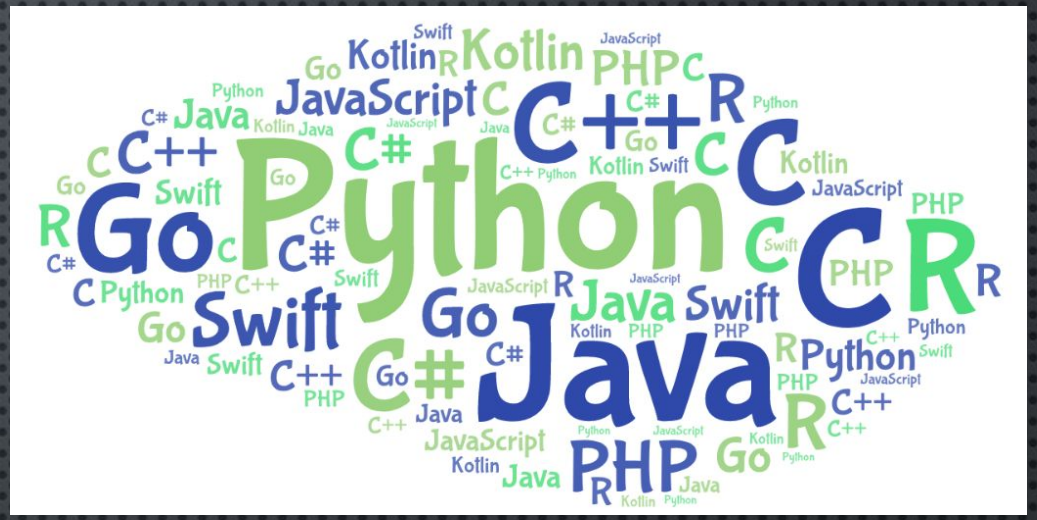
Понятие языка программирования

Понятие алгоритма

КОМПЬЮТЕРНАЯ ПРОГРАММА

- **Компьютерная программа** – это набор инструкций, которые могут выполняться на компьютере для выполнения конкретной задачи.
- **Компьютерная программа** обычно пишется программистом на языке программирования. Из программы в её удобочитаемой форме исходного кода, компилятора или других средств может получить машинной код – форму, состоящую из инструкции, которые может выполнять напрямую.

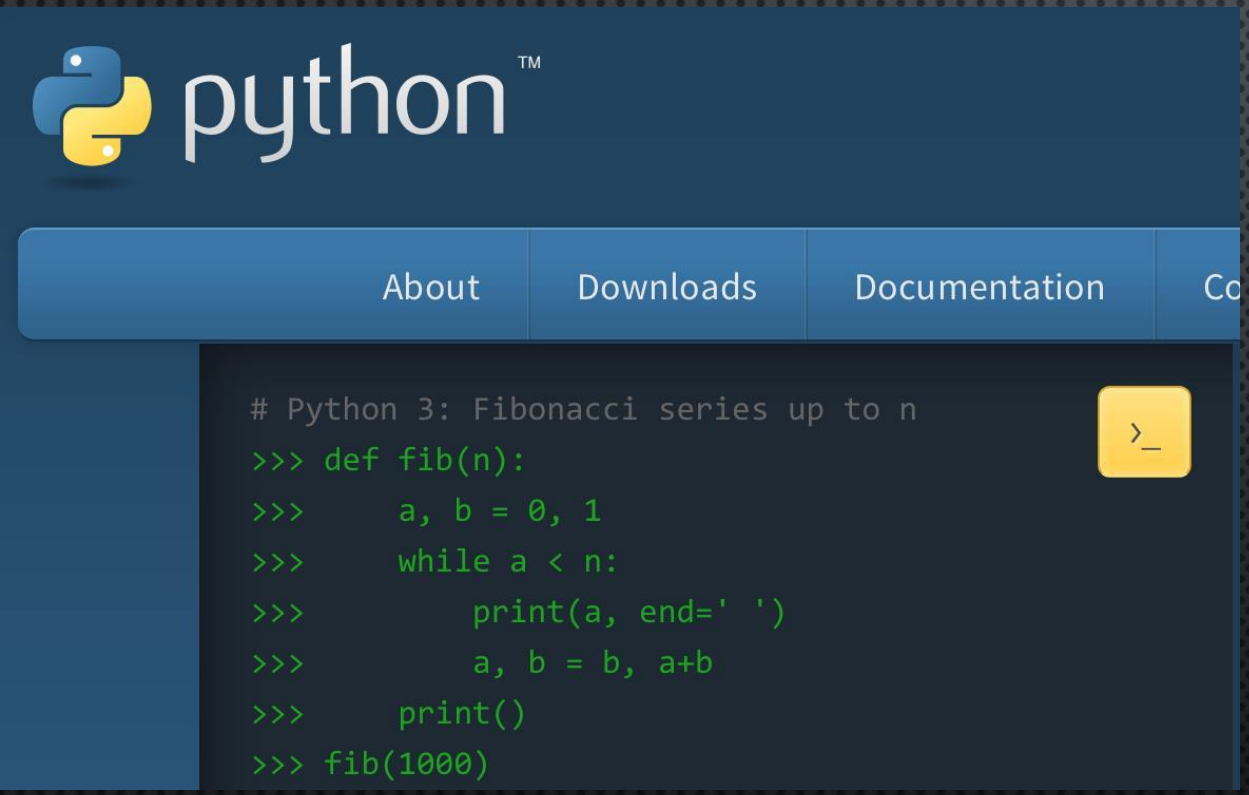
Язык программирования



- Язык программирования — это набор формальных правил, по которым пишут программы. Обычный язык нужен для общения людей, а язык программирования — для общения с компьютером. Как и в любом естественном языке, тут есть лексика — слова, функции и операторы, из которых по правилам синтаксиса составляются выражения. Они имеют чёткий, вполне определённый смысл, понятный компьютеру, — семантику.

ВИДЫ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ

ПРИМЕРЫ ПРОГРАММ НА ЯЗЫКАХ ПРОГРАММИРОВАНИЯ




```
program qq;
var x: integer;
begin
   writeln('Введите возраст');
   read ( x );
   if (x >= 25) and (x <= 40) then
        writeln ('Подходит')
   else writeln ('Не подходит')
end.
```

сложное условие

Алгоритмизация и программирование, язык C++, 10 класс

Сложение чисел: простое решение

```
#include <iostream>
using namespace std;
main()
{
   int a, b, c;
   cin >> a >> b;
   c = a + b;
   cout << c;
   cin.get(); cin.get();
}
```

ждём нажатия на клавишу

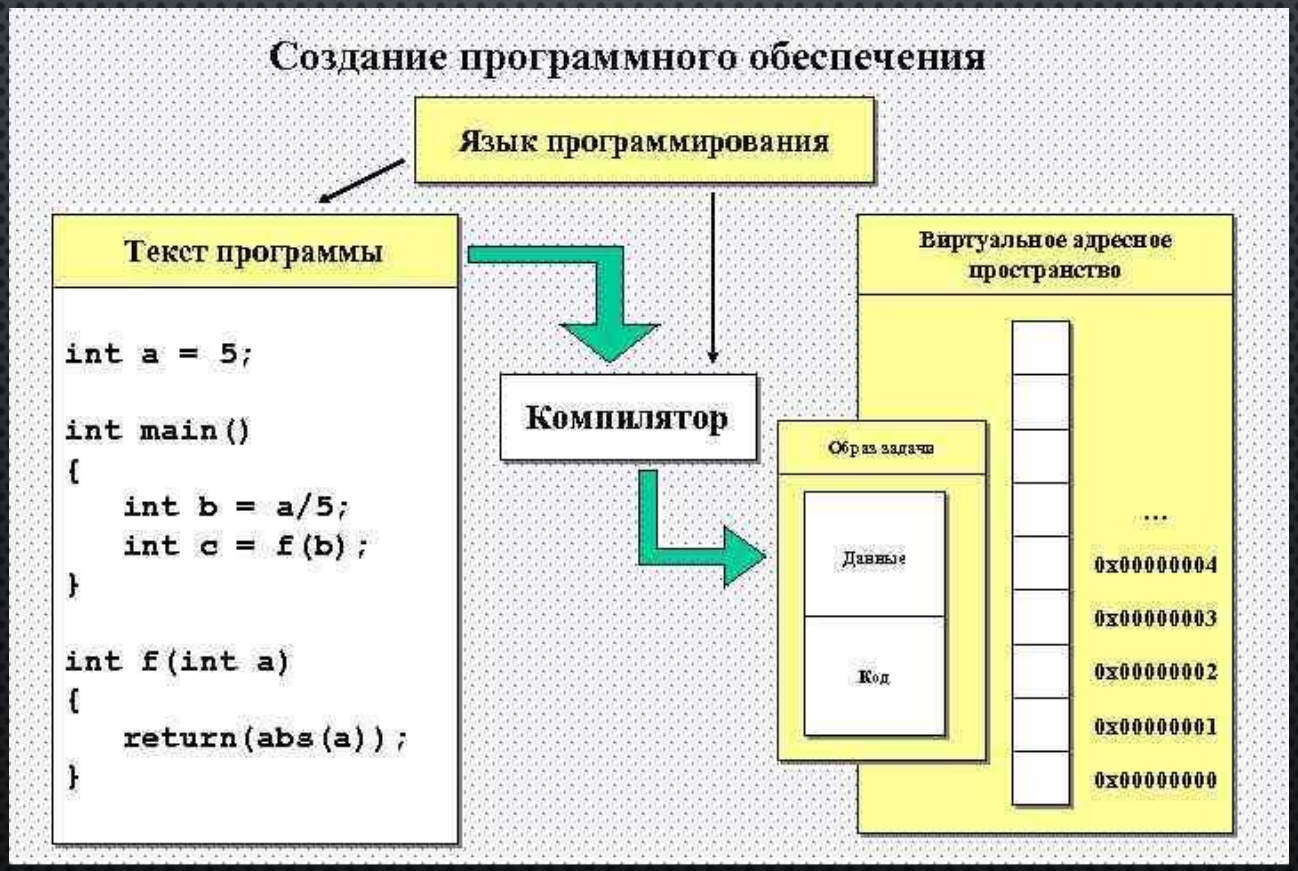
читаем остатки входного потока после ввода

? Что плохо?

Как компьютер понимает разные языки программирования

- На самом деле язык программирования — это не язык компьютера. Машина понимает последовательности нулей и единичек: есть напряжение в цепи — единица, нет — ноль. Поэтому любую программу сначала надо перевести в набор таких машинных команд.
- Для этого есть два инструмента — компилятор и интерпретатор. Компилятор работает как бюро переводов: вы отдаёте ему весь текст программы, а он превращает его в исполняемый код, набор команд для процессора. Интерпретатор больше похож на переводчика-синхрониста: сказали фразу — синхронист тут же её перевёл, а компьютер выполнил.
- Внутри компиляторов и интерпретаторов — сложные наборы правил по превращению языка программирования в машинный код, понятный компьютеру. Это тоже программы. Их пишут создатели нового языка — на каком-то другом, уже существующем. Например, интерпретатор *Python* написан на C, а сам C — на ассемблере, практически машинном коде.

ПРИМЕР ПОНИМАНИЕ КОМПЬЮТЕРА К ПРОГРАММЕ



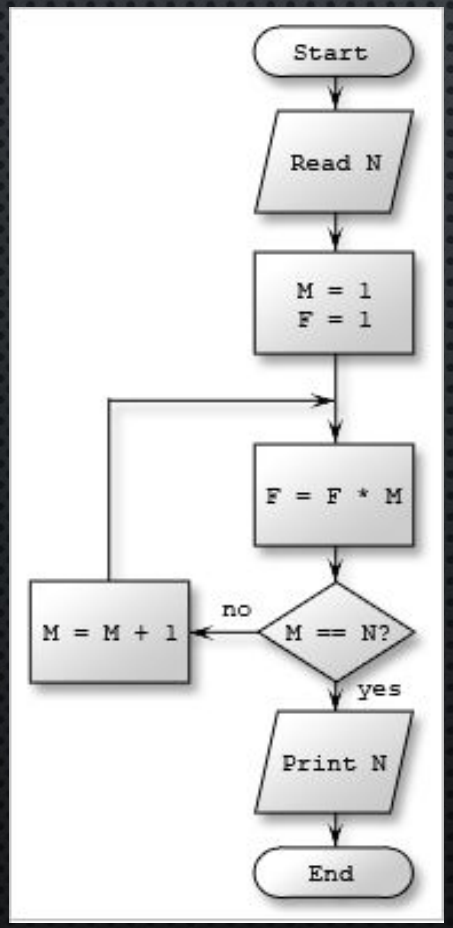
Алгоритм

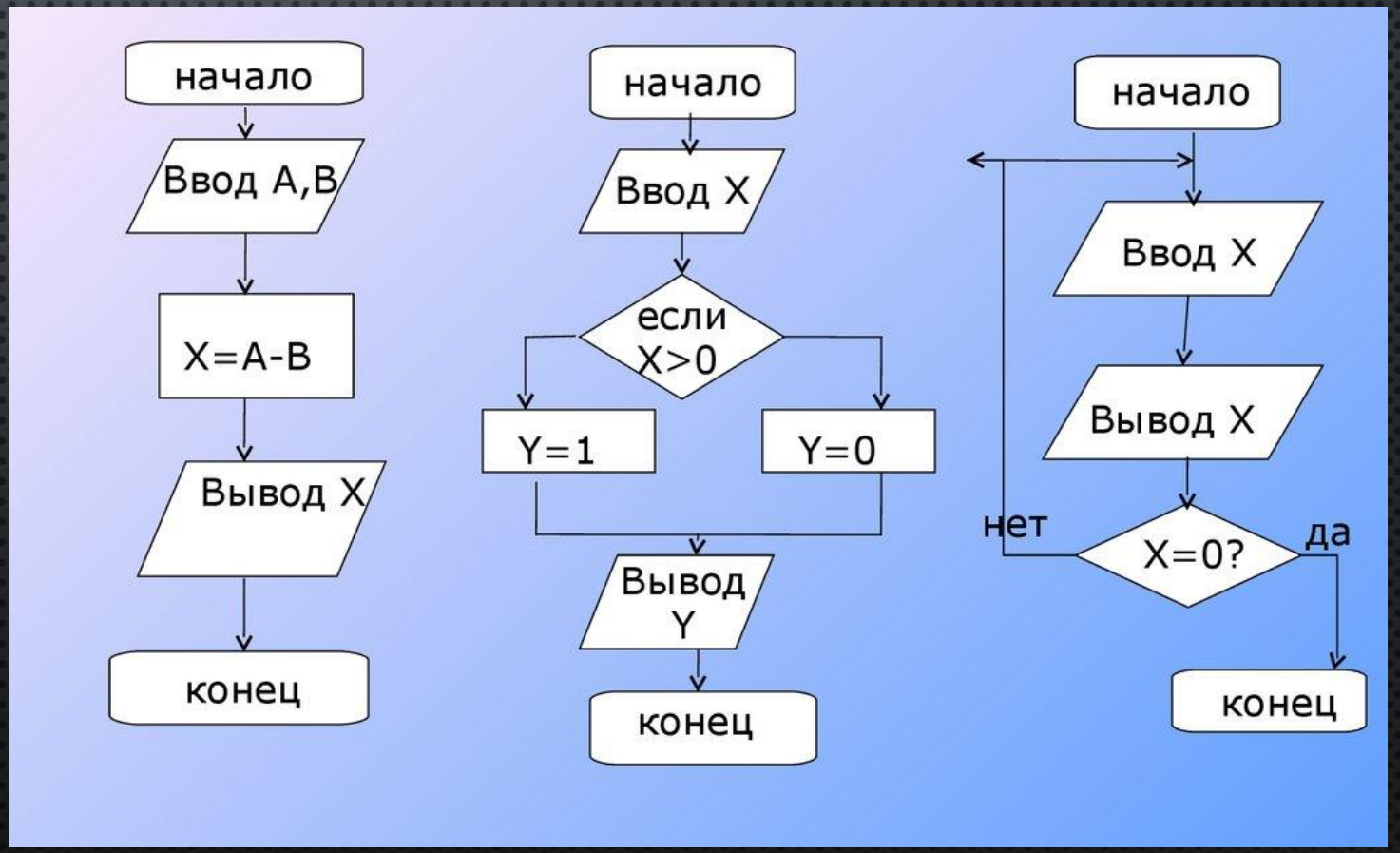
- **Алгоритм** — это последовательность команд, предназначенная исполнителю, в результате выполнения которой он должен решить поставленную задачу. Алгоритм должен описываться на формальном языке, исключающем неоднозначность толкования.
- **Исполнитель** — это человек, компьютер, автом. устройство и т. д. Исполнитель должен уметь выполнять все команды, составляющие алгоритм. Множество возможных команд конечно и изначально строго задано. Действия, которые выполняет исполнитель по этим командам называются **элементарными**. Запись алгоритма на формальном языке называется **программой**.

Свойства алгоритма

- **Дискретность.** Процесс решения задачи должен быть разбит на последовательность отдельных шагов-команд, которые выполняются одна за другой. Только после завершения одной команды начинается выполнение следующей.
- **Понятность**. Алгоритм должен содержать только те команды, которые известны исполнителю.
- **Детерминированность**. Каждый шаг и переход от шага к шагу должны быть точно определены, чтобы его мог выполнить любой другой человек или механическое устройство. У исполнителя нет возможности принимать самостоятельное решение (алгоритм исполняется *формально*).
- **Конечность**. Обычно предполагают, что алгоритм заканчивает работу за конечное число шагов. Результат работы алгоритма также должен быть получен за конечное время. Можно расширить понятие алгоритма до понятия процесса, который по различным каналам получает данные, выводит данные и потенциально может не заканчивать свою работу.
- **Массовость**. Алгоритм должен решать не одну частную задачу, а класс задач. Не имеет смысла строить алгоритм нахождения наибольшего общего делителя только для чисел 10 и 15.

ПРИМЕР АЛГОРИТМА

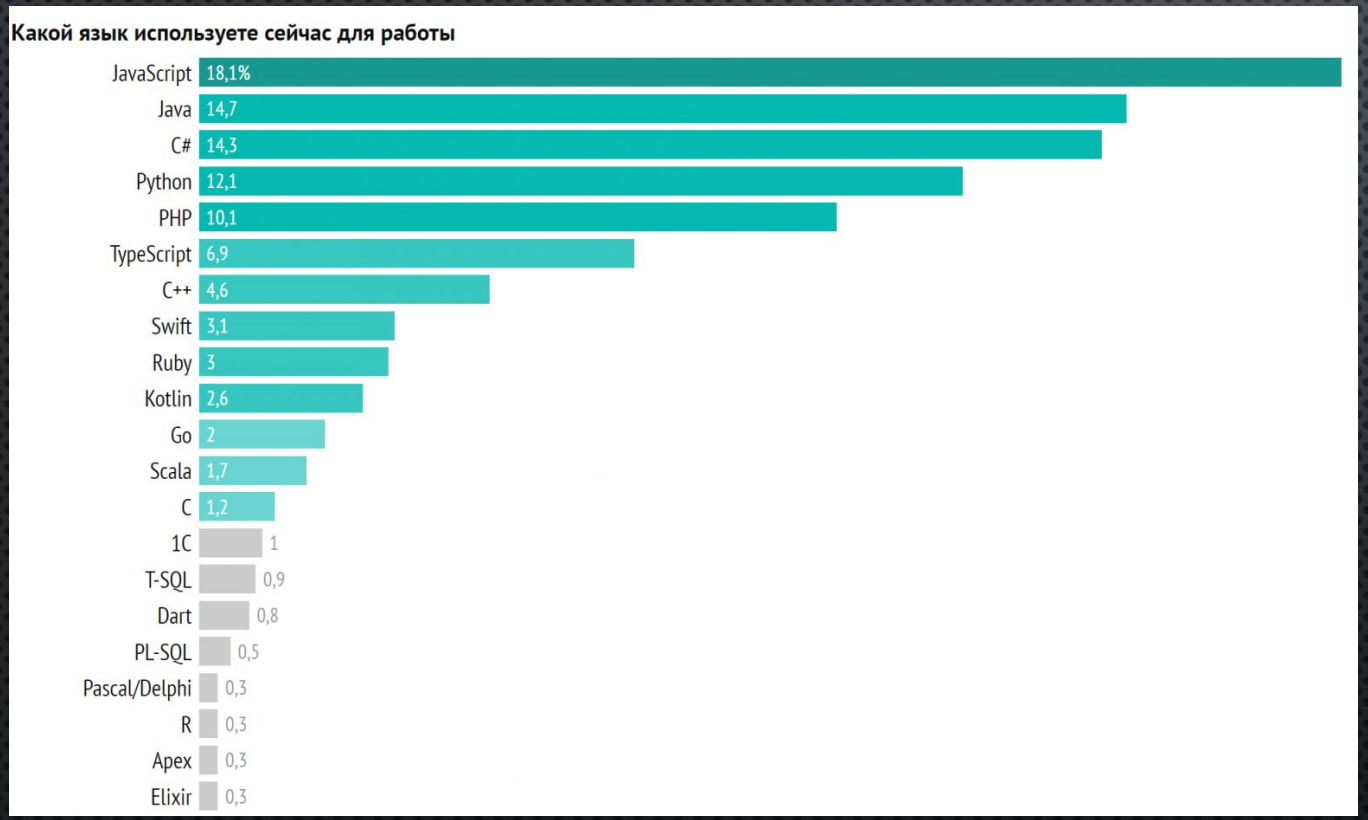




Чем языки программирования отличаются от алгоритмов

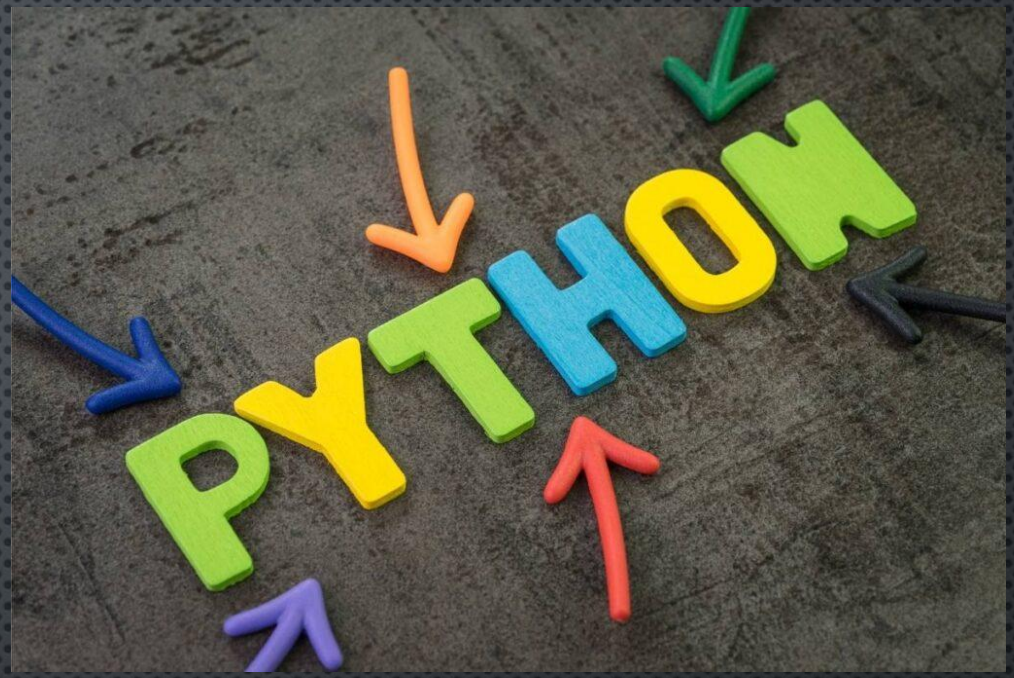
- Программы нужны для того, чтобы машина сделала что-то полезное. Это невозможно, если нет чёткого порядка действий и правил их выполнения — алгоритма.
- Алгоритм работает как маршрут в навигаторе: *«Из пункта А едем в пункт Б, поворот через 150 метров»*. Англичанин понимает его по-английски, китаец — по-китайски, а мы с вами — по-русски. Языки разные, а порядок действий один и все должны добраться до нужного места.
- Любая программа начинается с алгоритма, но на разных языках это может выглядеть по-разному.

РЕЙТИНГ ЯЗЫКОВ ПРОГРАММИРОВАНИЯ



Mar 2021	Mar 2020	Change	Programming Language	Ratings	Change
1	2	↑	C	15.33%	-1.00%
2	1	↓	Java	10.45%	-7.33%
3	3		Python	10.31%	+0.20%
4	4		C++	6.52%	-0.27%
5	5		C#	4.97%	-0.35%
6	6		Visual Basic	4.85%	-0.40%
7	7		JavaScript	2.11%	+0.06%
8	8		PHP	2.07%	+0.05%
9	12	↑	Assembly language	1.97%	+0.72%
10	9	↓	SQL	1.87%	+0.03%
11	10	↓	Go	1.31%	+0.03%
12	18	⇈	Classic Visual Basic	1.26%	+0.49%
13	11	↓	R	1.25%	-0.01%
14	20	⇈	Delphi/Object Pascal	1.20%	+0.48%
15	36	⇈	Groovy	1.19%	+0.94%
16	14	↓	Ruby	1.18%	+0.13%
17	17		Perl	1.15%	+0.24%
18	15	↓	MATLAB	1.04%	+0.05%
19	13	⇊	Swift	0.95%	-0.28%
20	19	↓	Objective-C	0.91%	+0.17%

О языке Python



- Язык программирования Python был создан к 1991 году голландцем Гвидо ван Россумом
- После того, как Россум разработал язык, он выложил его в Интернет, где сообщество программистов присоединилось к его улучшению

Основные особенности языка

- Python – это полноценный во многом универсальный язык программирования, используемый в различных сферах
- Основная, но не единственная, поддерживаемая им установка, – объектно-ориентированное программирование
- Однако в данном курсе мы будем изучать структурное программирование, так как оно является базой
- Без знания основных типов данных, ветвлений, циклов, функций нет смысла изучать более сложные парадигмы, т. к. в них все это используется

Основные особенности языка

- Python – интерпретируемый язык программирования. Это значит, что исходный код частями преобразуется в машинный в процессе его чтения специальной программой – интерпретатором
- Python характеризуется ясным синтаксисом, т. к. в мало используются такие вспомогательные синтаксические элементы как скобки, точки с запятыми
- Правила языка заставляют программистов делать отступы для обозначения вложенных конструкций

ПРИМЕРЫ ПРОГРАММ НА PYTHON

```
a = int(input("Введите число:"))
if a < 0:
    print(a," меньше нуля")
elif a == 0:
    print(a," равно нулю")
else:
    print(a," больше нуля")
```

```
a = 10
b = 20
c = 30
if (a + b) / c == 1 and c - b - a == 0:
    print('yes')
else:
    print('no')
```

ПЕРВАЯ ПРОГРАММА

```
1  print("Hello, World!")
```

```
Hello, World!
```

Задание №1

- С помощью метода print() выведите свое имя на русском языке.

Задание №2

- С помощью метода print() выведите свое имя на английском языке.

ВТОРАЯ ПРОГРАММА

```
print('Андрей', 'Косов', sep='\n')
```

Аргумент sep

- Параметр sep контролирует то, какой разделитель будет использоваться между элементами.

Управляющий символ

- **Управляющие** последовательности. Применение **символа** “\” (обратный слеш) - **это** способ помещения специальных **символов** в строку.

1) \n – перевод строки
2) \t – табуляция горизонтально
3) \\ - обратный слеш

Задание №3

- Напишите свое ФИО на русском языке, где каждый элемент должен начинаться с новой строки.

Задание №4

- Напишите свое ФИО через специальный символ слеш.

Задание №5

- Напишите свое ФИО на английском языке через символ табуляции.

ТРЕТЬЯ ПРОГРАММА

```
print(96, end='(')
print(192, end=')(')
print(90, end=')')
```

Аргумент end

- Параметр end контролирует то, какое значение выведется после вывода всех элементов.

Задание №6

- Напишите своё ФИО, где каждый элемент будет находиться в новом Print с использованием аргумента end, перед каждым элементом поставьте ***.

Задание №7

- Напишите сою дату рождения используя end и спец. символ /. Каждая цифра даты должна писать в новом print.

ЗАНЯТИЕ ОКОНЧЕНО!