Муниципальный бюджетный общеобразовательное учреждение
средняя общеобразовательная школа №1
имени Чернявского Якова Михайловича станицы Крыловской
муниципального образования Крыловский район

«Python»

Автор работы:
Голиков Матвей Леонидович
Ученик 9 «А» класс
МБОУ СОШ №1 ст. Крыловской
МО Крыловский район

Руководитель:
Кряжимский Артём Сергеевич
Учитель информатики
МБОУ СОШ № 1 ст. Крыловской

2023

Оглавление

Введение

При решении задач по информатике в формате ЕГЭ я наткнулся на задачи с использованием языков программирования. Поскольку меня давно интересовало программирование и перспективы работы в данной сфере я решил освоить какой-либо язык. Блуждая на просторах сети Интернет в поисках информации о различных языках, я остановился на языке Python. Он позиционировал себя как простой, многогранный язык. Эта тема актуальна, потому что многие задачи мы не можем выполнить без использования программ, найти ответы на вопросы без использования сайтов, которые пишутся на языках программирования, в том числе Python. Человек ежедневно огромное время проводит в сети Интернет, смотрит видеоролики на видеохостингах, читает онлайн книги, находит решение задач поставленных перед ним – все это происходит на сайтах, написанных в частности и на языке Python.

Объект исследования: язык Python

Предмет исследования: развитие языка, его особенности

Цель: Исследовать язык, происхождения, основы синтаксиса языка. На результате своих изучений подтвердить или опровергнуть утверждения о простоте и многогранности данного языка.

Задачи:

1) Познакомиться с происхождением языка Python
2) Узнать в каких сферах программирования он используется
3) Выявить необходимость языка в программировании
4) Проанализировать статистические показатели
5) Самому попробовать изучить базовые понятия языка

Гипотеза:Ускорению работы сайтов благодаря скачку веб-разработки в следствии развития языка.

1. ПРОШЛОЕ ЯЗЫКА PYTHON

1.1 История появления

Все началось всего лишь 30 лет назад, в конце 1980 году – голландский программист Гвидо ван Россум, работавший в таких компаниях как Google, Dropbox Inc, а после в Microsoft сформировал идею нового, идеального на его взгляд, языка Python. В декабре 1989 года он приступил к созданию языка, который должен был заменить язык программирования ABC, стал его потомком. Вряд ли кто-то кроме самого Россума видел перспективы в этом языке. Никто не предполагал тогда, что это «изобретение» переменит веб-программирование и программирование в целом в течение следующих 10-15 лет и станет лидирующим языком среди огромного множества программистов. Видя что-то новое, мы в редких случаях можем предсказать его влияние на нашу жизнь, тем более за такой короткий период.

Ван Россум является основным автором Python и продолжал выполнять центральную роль в принятии решений относительно развития языка вплоть до 12 июля 2018 года. По поводу названия сам создатель говорит, что оно заимствовано из названия юмористического шоу «Летающий цирк Монти Пайтона» (Monthy Piton). Как утверждает Гвидо Ван Россум, просмотр этого шоу помогало ему отвлечься и расслабиться при написании этого языка программирования. Но многие люди ассоциируется название языка со змеями, тем более что логотип представляет собой стилизованное изображение змеиной головы.

Python – простой язык. Отчасти это связано с тем, что язык программирования ABC, который был взят за основу Python, разрабатывался для работы на нем людьми, которые не слишком тесно связаны с программированием.

Многие международные IT компании используют Python. Помимо Google, этот язык используется в таких крупных компаниях как Dropbox, Facebook, Instagram. Некоторые части Youtube написаны на Python.

1.2 Этапы развития

Python имеет три основных ступени своего развития, но на каждой ступени имеет огромное количество «патчей» , которые исправляют мелкий ошибки, но не являются глобальными изменениями в языке.

Python 1.0

Python 1.0 появился в январе 1994 года. Основными новыми возможностями, включёнными в этот релиз, были средства функционального программирования: лямбда-исчисление(формализация и анализ вычислимости), map(функция предназначенная для списков языка, которая рассматривается как «применить ко всем»), filter(обрабатывает структуру данные в некотором порядке для создания новой последовательности) и свёртка списка(преобразовывает порядок данных к некоторому атомарному значению). Ван Россум утверждал: «Python приобрёл lambda, reduce(), filter() и map() благодаря любителю Lisp(обработка слов и списков) , которому их не хватало, и он предоставил патчи, реализующие эти функции».

Последней версией, выпущенной Ван Россумом во время работы в центре математики и информатики (CWI), был Python 1.2. С 1995 года Ван Россум продолжил работу над Python-ом в корпорации национальных исследовательских инициатив (CNRI) в городе Рестон, штат Вирджиния, где было выпущено несколько версий языка.

К версии 1.4 Python включал в себя множество новых функций, среди которых наиболее заметными были позаимствованные в Modula-3 именованные параметры (также подобные параметрам Common Lisp) и встроенная поддержка комплексных чисел. Также в 1.4 появилась простая форма сокрытия данных при помощи name mangling (которое, правда, легко обходится).

Во время пребывания в CNRI Ван Россум запустил проект "Программирование для всех" (англ. *Computer Programming for Everybody*, CP4E), предназначенный сделать программирование доступным для большего числа людей, на основе получения базовой "компьютерной грамотности", подобной базовому знанию языка и математики, требуемых большинству работающих. Python играл центральную роль в этой инициативе, благодаря своей нацеленности на ясный синтаксис. Проект CP4E финансировался DARPA, в данное время проект закрыт. И, хотя Python старается быть простым в изучении и не слишком магическим в синтаксисе и

семантике, простота его использования не-программистами не является главной задачей.

Python 2.0

В версии Python 2.0 появилось списковое включение — функция, заимствованная из функциональных языков программирования SETL и Haskell. Синтаксис в Python для этой конструкции очень похож на Haskell, за исключением того, что в Haskell предпочли использовать символы пунктуации, а в Python — ключевые слова. Также в Python 2.0 была добавлена система «сборки мусора»(система автоматического управления памятью) с поддержкой циклических ссылок.

Python 2.1 очень похож на Python 1.6.1 и Python 2.0. Лицензия, начиная с этой версии, была переименована в Python Software Foundation License. Начиная с альфа релиза Python 2.1 весь код, техническая документация и спецификации принадлежат некоммерческой организации Python Software Foundation (PSF), созданной в 2001 году по образцу Apache Software Foundation. Релиз включал изменение в спецификацию языка, поддерживающее вложенные области видимости, как в языках со статической (лексической) областью видимости. (Эта возможность была выключена по умолчанию и не потребовалась до Python 2.2.)

Главным нововведением в Python 2.2 было объединение базовых типов Python и классов, создаваемых пользователем, в одной иерархии. Это сделало Python полностью объектно-ориентированным языком. Тогда же были добавлены генераторы(системы управления циклом), идея которых заимствована из Icon.

В ноябре 2014 было объявлено, что Python 2.7 будет поддерживаться до 2020 года, и подтверждено, что релиза 2.8 не будет, так как предполагается, что пользователи должны переходить на версию 3.4+ при первой же возможности.

Python 3.0

Python 3.0 (называемый также "Python 3000" или "Py3K") разрабатывался с целью устранения фундаментальных изъянов в языке. Эти изменения не могли быть сделаны при условии сохранения полной обратной совместимости с 2.x версией, поэтому потребовалось изменение главного номера версии. Ведущим принципом разработки Python 3 было: «уменьшение дублирующейся функциональности устранением устаревших способов сделать это». Python 3.0 разрабатывался с той же философией, что и предыдущие версии. Однако, поскольку в Python скопились новые и, ставшие избыточными, старые способы решения одних и тех же задач, в Python 3.0 был сделан упор на удалении дублирующихся конструкций и модулей, следуя принципу: «должен существовать один и, желательно, только один очевидный способ сделать это».

Тем не менее, Python оставался «мультипарадигменным» языком. Программист всё ещё мог выбирать между объектно-ориентированным, структурным, функциональным программированием и другими парадигмами. Но, при таком широком выборе, особенности каждого подхода в Python 3.0 должны были быть более очевидны, чем в Python 2.x. Python 3.0 был выпущен 3 декабря 2008 года. Планировалось, что версии Python 2.x и Python 3.x будут сосуществовать параллельно на протяжении нескольких релизов. Версия 2.x, главным образом, для совместимости с существующими приложениями, и с портированием в неё некоторых возможностей 3.x. Python 2.6 был выпущен как соответствующий Python 3.0 и включал ряд его возможностей, а также режим «предупреждения», в котором подсвечивалась функциональность, удалённая в 3.x. Подобным же образом, Python 2.7 соответствовал и включал функциональность Python 3.1, выпущенного 26 июня 2009 года. Python 2.7 был последним релизом 2.x: параллельные релизы прекратились на Python 3.2.

Python 3.0 нарушил обратную совместимость. Не требовалось, чтобы код на Python 2.x выполнялся под Python 3.0 без изменений. Было сделано много принципиальных изменений, таких как превращение оператора print в функцию (так что прежнее использование оператора print будет приводить к невозможности запустить программу) и переход на Unicode для любых строк. Динамическая типизация языка вместе с планами изменить семантику некоторых методов (например, словарей) сделали очень трудной полную автоматическую трансляцию Python 2.x в Python 3.x. Однако, инструмент, названный "2to3", может выполнить основную работу по переводу с указанием мест с неоднозначностью, используя комментарии и

диагностические сообщения. Даже в альфа-версии инструмент "2to3" выполнял трансляцию довольно успешно. Для проектов, требующих совместимости как с 2.x, так и 3.x, разработчики Python рекомендуют поддерживать исходный текст в Python 2.x и выпускать релизы для 3.x

ГЛАВА 2. ПОЧЕМУ ИМЕННО PYTHON?

2.1 КАК ВЫБИРАЮТ ЯЗЫК

Почти все начинающие программисты задаются вопросом: " С какого языка начать? " Нужно понимать, что можно "написать" в процессе изучения, чтобы постоянно получать заряд мотивации. Для примера можно сравнить C++ и Python. Изучая первый, человек будет тратить много времени на понимание принципа работы памяти. Такими образом, перед написания первой программы нужно потратить месяц, а то и два. В тоже время на Пайтоне уже в первую неделю-две можно легко запрограммировать бота-Telegram или голосового помощника. Разница в том, что порог вхождения у этих языков принципиально разный. Если в C++ спустя неделю-две думаешь о том, как работают указатели или об особенностях типов данных, то на Пайтоне за то же время уже начнёшь писать какие-либо программы, получая приливы мотивации. Когда я спрашиваю на форумах : "С какого языка начать обучение?", мне 4 из 5 человек отвечают Python. Это не удивительно, так как по статистике(приложение 1) на компьютерах Science США около 30 из 35 университетов используют Пайтон для обучения своих студентов программированию. Python подходит везде. Создание сайтов, игр, программ. Язык не настолько сложен, значит не придётся погружаться в его особенности, что позволяет погрузиться в изучение алгоритмов. Даже в учебнике "Грокаем алгоритмы", в качестве примеров кода используется язык Пайтон. Этот язык крайне востребованный и современный язык, что позволяет легко найти вакансию для работы. На Пайтоне создают сайты, программы. Язык постоянно прогрессирует, потому что имеет огромное коммьюнити разработчиков, которое создаёт огромное количество модов и модулей для языка. Пайтон зарекомендовал себя, как один из лучших языков программирования для веб-разработки. В рейтинге TIOBE за 2021 год Пайтон занимает третью строчку(приложение2), а в рейтинге PyPl уверенное первое место(приложение 3) среди языков предназначенных для создания сайтов. Пайтон максимально продуманный язык: его синтаксис легко читается , а все функции, модули и методы прописаны предельно чётко.

2.2 ПЛЮСЫ И МИНУСЫ ПАЙТОН

2.2.1 ПЛЮСЫ

1) В арсенале языка Пайтон имеется огромное количество преимуществ. Особенностью его простого, не замысловатого синтаксиса является крайне короткий код, что позволяет занимать меньше оперативной памяти при работе программы, а также быстрее писать код. Python считается наиболее востребованным языком в Data Science. С его помощью пишут алгоритмы ML-программ и аналитические приложения. Также с его помощью обслуживают облачные сервисы и хранилища данных. Также пайтон используется при работе с нейронными сетями, в машинном обучении(Машинное обучение – исправление ошибок путём их совершения. Например, робот должен пробежать сложный маршрут и не выйти за его приделы. После выхода за пределы маршрута робот возвращается в исходное положение и сам исправляет свою ошибку, корректируя код. Таким образом сходя с маршрута, совершая ошибку, робот сам исправит их и будет проходить маршрут без проблем).

2) В отличии от того же PHP, Пайтон не испортит представление о коде, не научит вредным путям, как тот же PHP, а даже наоборот полностью погрузит в понимание алгоритмов, функций, методов.

3) Python является интерпретируемым языком программирования, который не компилируется. Таким образом, до запуска он представляет собой обычный текстовый файл. Соответственно, программировать можно почти на всех платформах, а сам язык логичен и хорошо спроектирован.

Таким образом язык Пайтон универсален, прост, лаконичен и акутален.

2.2.2 МИНУСЫ

1) Программы на Python считают недостаточно быстрыми. Для сравнения: софт для iOS, написанный на языке Swift, может работать в 8 раз быстрее, чем на Python. Также Python — не очень подходящее решение для задач, которые требуют большого объёма памяти, — такие задачи лучше решать с помощью C либо C++.

2) Высокий уровень зависимости от системных библиотек. В результате затрудняется перенос на другие системы. Да, проблема решается посредством Virtualenv, однако у этого инструмента свои недостатки: костыли, избыточность полных методов изоляции, дублирование системных библиотек.

3) Global Interpreter Lock не даёт возможности одновременно исполнять несколько потоков Python в реализации CPython. Но GIL мы можем на какое-то время отключить, как это реализовано в математическом пакете NumPy.

ГЛАВА 3. КАК Я УЧИЛ ПАЙТОН

3.1 Процесс изучения.

К сожалению, Питон не является встроенным языком, как и большинство других языков, поэтому первым делом я скачал эту утилиту с официального сайта языка (https://www.python.org/). Обучаться я начал на версии 3.8.7. На самом деле, это очень увлекательное занятие. Сначала я пробовал переводить и понимать информацию системы языка на самом сайте, но это требовало очень много времени и сил, не всегда всё было понятно. Исходя из сроков на изучение основы языка я искал более рациональный метод обучения, который бы не занял огромное количество времени. Я нашёл сайт под названием Питонтьютор (https://pythontutor.ru/) который позволил мне с легкостью понять основы языка, типы данных, переменные и простые действия, такие как вывод текста на экран или различные математические функции(сложение, вычитание, умножение, деление, возведение в степень и тп). На этом сайте предлагается моментальная практика на различных качественных задачах, подобные заданиям ЕГЭ, например, распределить яблоки по номерам в списках. В итоге за неделю я уже имел базовые познания в Пайтоне и имел практику с использованием этих самых знаний на примере задач, но не в написании программ, который действительно полезные и используются людьми в повседневной жизни. В итоге я решил написать цветной, многофункциональный калькулятор.

3.2. Создание калькулятора.

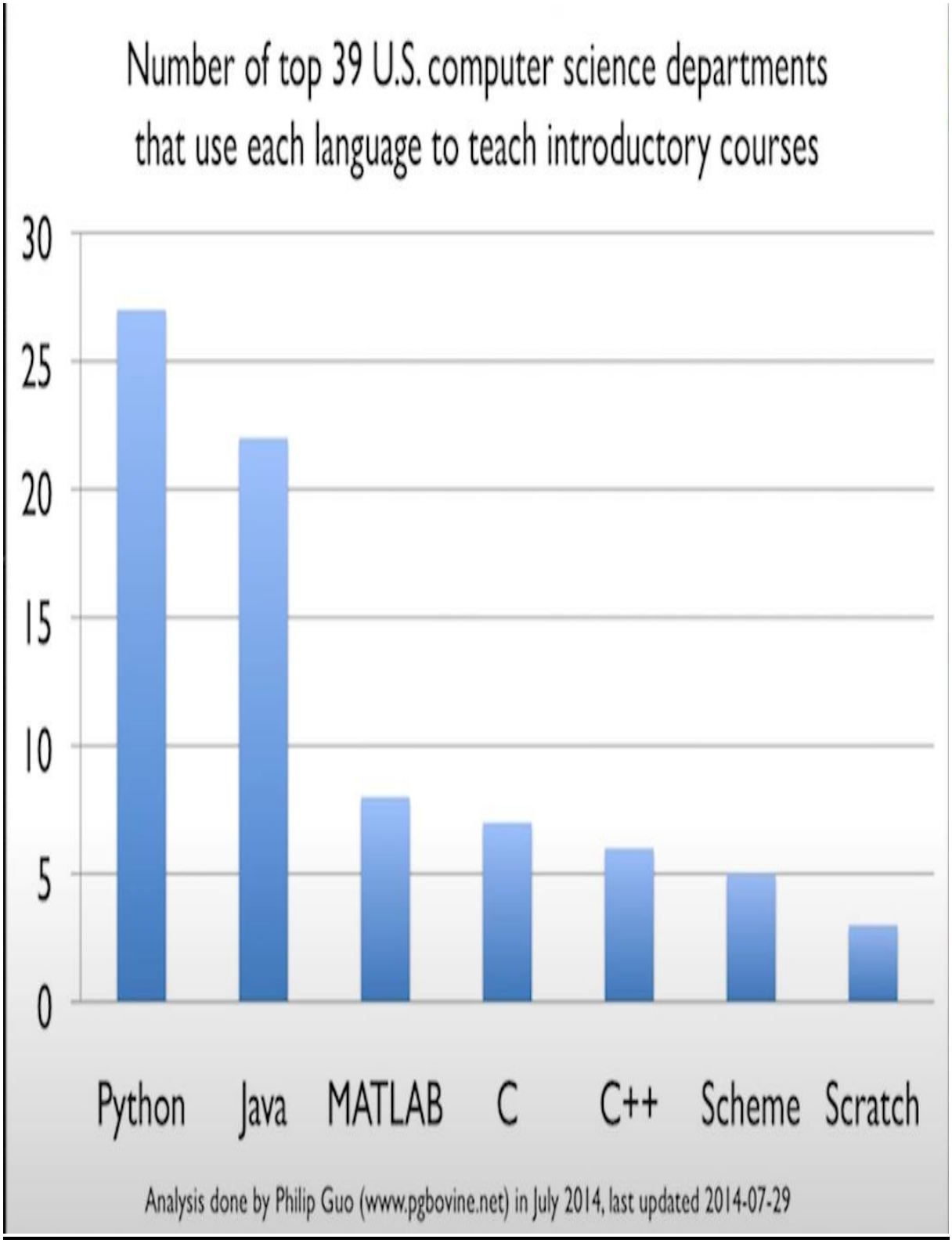
Первым делом я установил модуль языка для использования цветовой гаммы, который называется ColoramaPY. На выкладке модуля на GitHub, GitHub – хранилище модулей, созданных для различных языков программирования как официальными разработчиками, так и коммьюнити того или иного языка, я использовал документацию, чтобы разобраться в модуле, как его использовать(https://pypi.org/project/colorama/). Установка модуля(приложение4). Первым делом я импортировал сам модуль в нужный мне файл с кодом. После этого я добавил переменные с выбором функции, а также переменные с выбором самих чисел(приложение 5). Очень важно в выборе переменных задать тип данных(float – вещественные числа), потому

что сама функция input является исходным текстовым значением. Следующим шагом я задал список условий при помощи оператора if и его производных elif и else. Таким образом я давал условия переменной содержащей функцию для того, чтобы активировать следующие из нужных действий по выбранному запросу. Добавляем input чтобы избежать закрытия калькулятора сразу после совершения действия. (приложение 6). Последним шагом я добавил цвета(это прописано в строчках кода под номерами : 7, 8, 15, 19)(приложение 7). Начав тестировать, я заметил недочёты, которые появились из-за неопытности программирования. После их исправления, код стал выглядеть следующим образом(приложение 8). Проведя повторное тестирование, я пришёл к нужному результату(приложение 9) .

ЗАКЛЮЧЕНИЕ

В конце я хочу сказать, что изучение любого языка программирования это интересный процесс. Я, во-первых, нагрузил свой мозг, во-вторых, получил удовольствие изучая язык. Пайтон крайне прост для изучения, именно поэтому в будущем он будет только развиваться, количество разработчиков на Python расти, а работа сайтов и программ только улучшаться. Язык только развивается, поэтому невозможно представить какими будут его возможности в недалёком будущем. Я считаю, что подтвердил выдвинутую мной гипотезу.

ПРИЛОЖЕНИЕ 1



ПРИЛОЖЕНИЕ 2

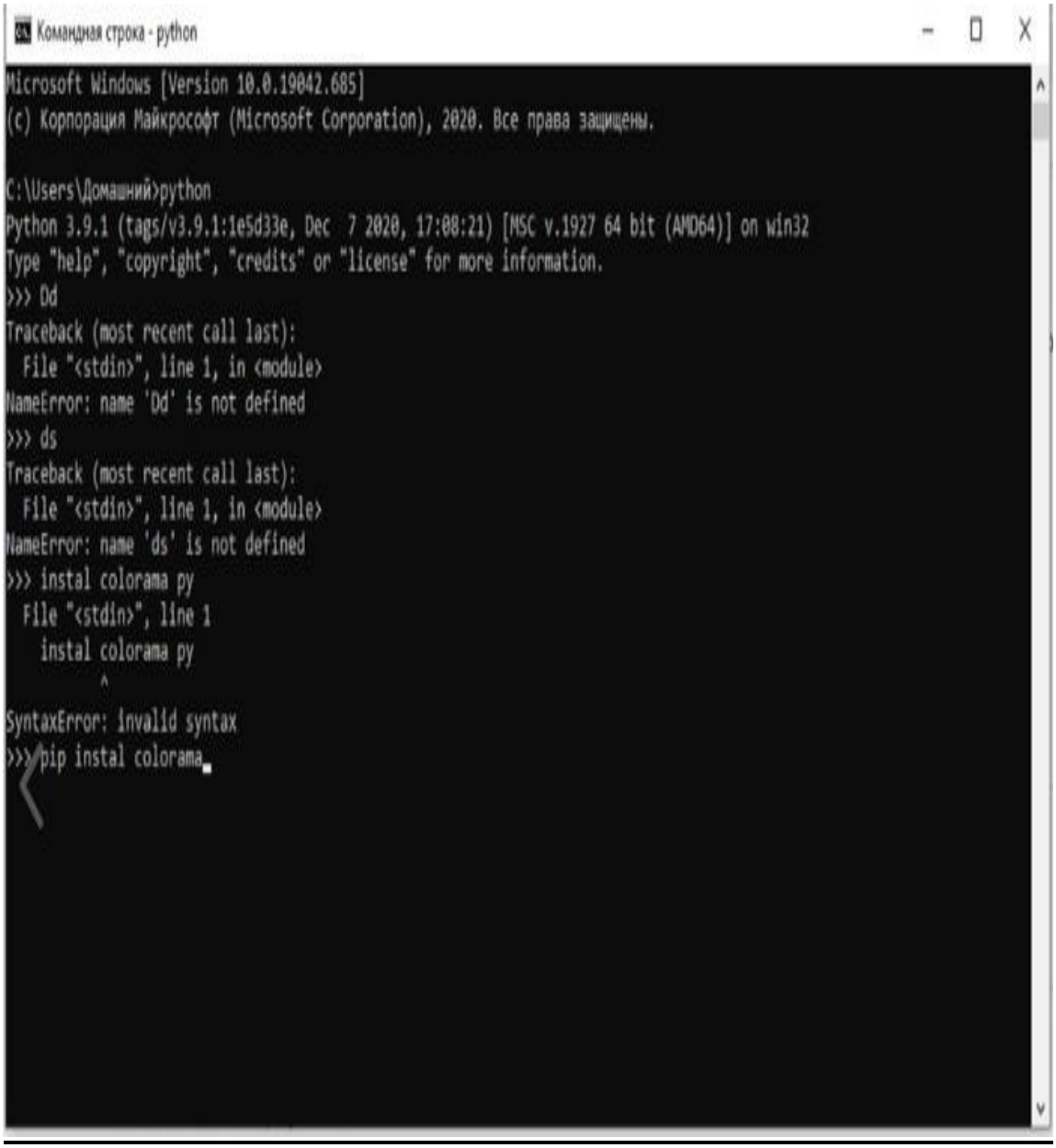
Feb 2021	Feb 2020	Change	Programming Language	Ratings	Change
1	2	▲	C	16.34%	-0.43%
2	1	▼	Java	11.29%	-6.07%
3	3		Python	10.86%	+1.52%
4	4		C++	6.88%	+0.71%
5	5		C#	4.44%	-1.48%
6	6		Visual Basic	4.33%	-1.53%
7	7		JavaScript	2.27%	+0.21%
8	8		PHP	1.75%	-0.27%
9	9		SQL	1.72%	+0.20%
10	12	▲	Assembly language	1.65%	+0.54%
11	13	▲	R	1.56%	+0.55%
12	26	▲▲	Groovy	1.50%	+1.08%

ПРИЛОЖЕНИЕ 3

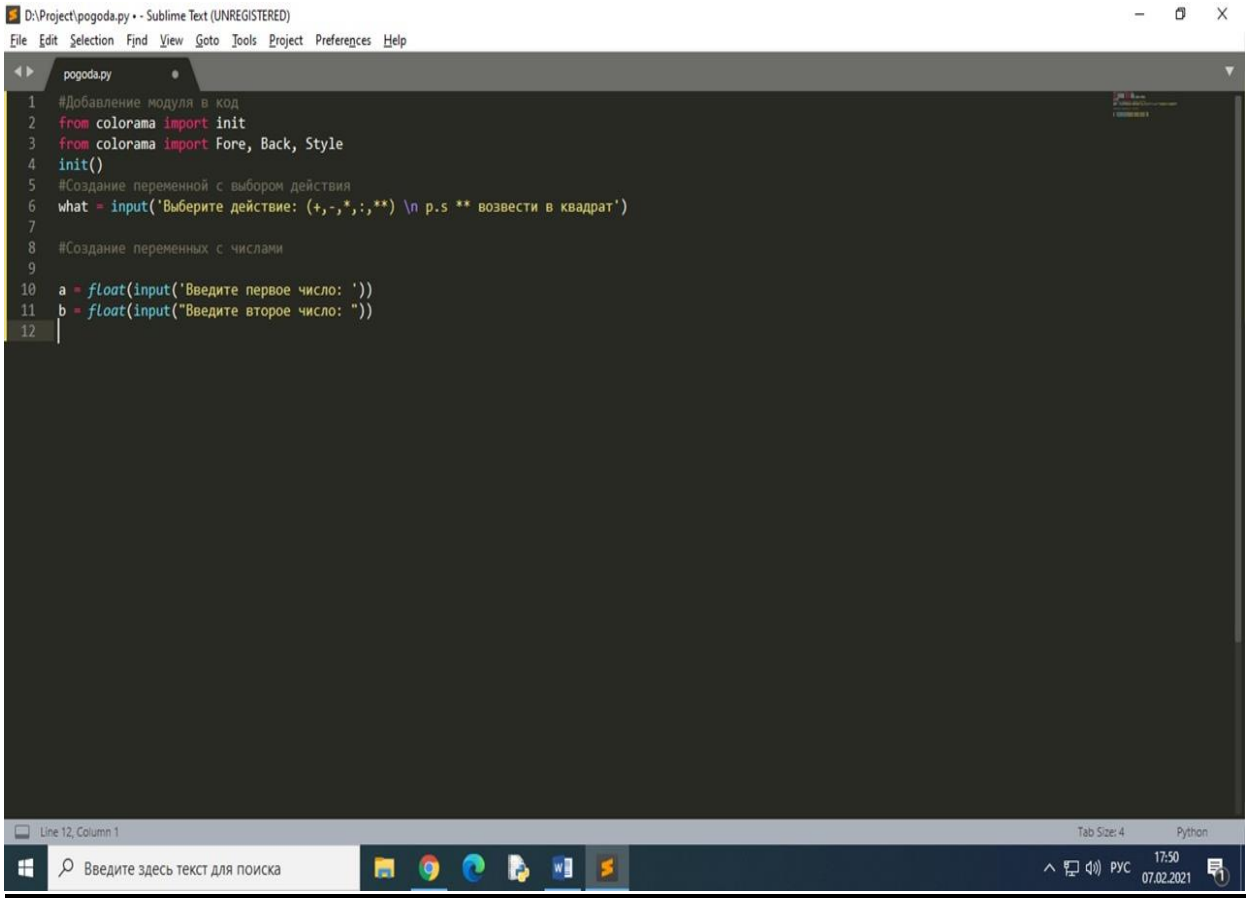
Worldwide, Feb 2021 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	30.06 %	+0.3 %
2		Java	16.88 %	-1.7 %
3		JavaScript	8.43 %	+0.4 %
4		C#	6.69 %	-0.6 %
5	↑	C/C++	6.5 %	+0.5 %
6	↓	PHP	6.19 %	-0.1 %
7		R	3.82 %	+0.0 %
8		Objective-C	3.66 %	+1.2 %
9		Swift	2.05 %	-0.3 %
10		TypeScript	1.87 %	+0.0 %
11		Matlab	1.76 %	-0.0 %

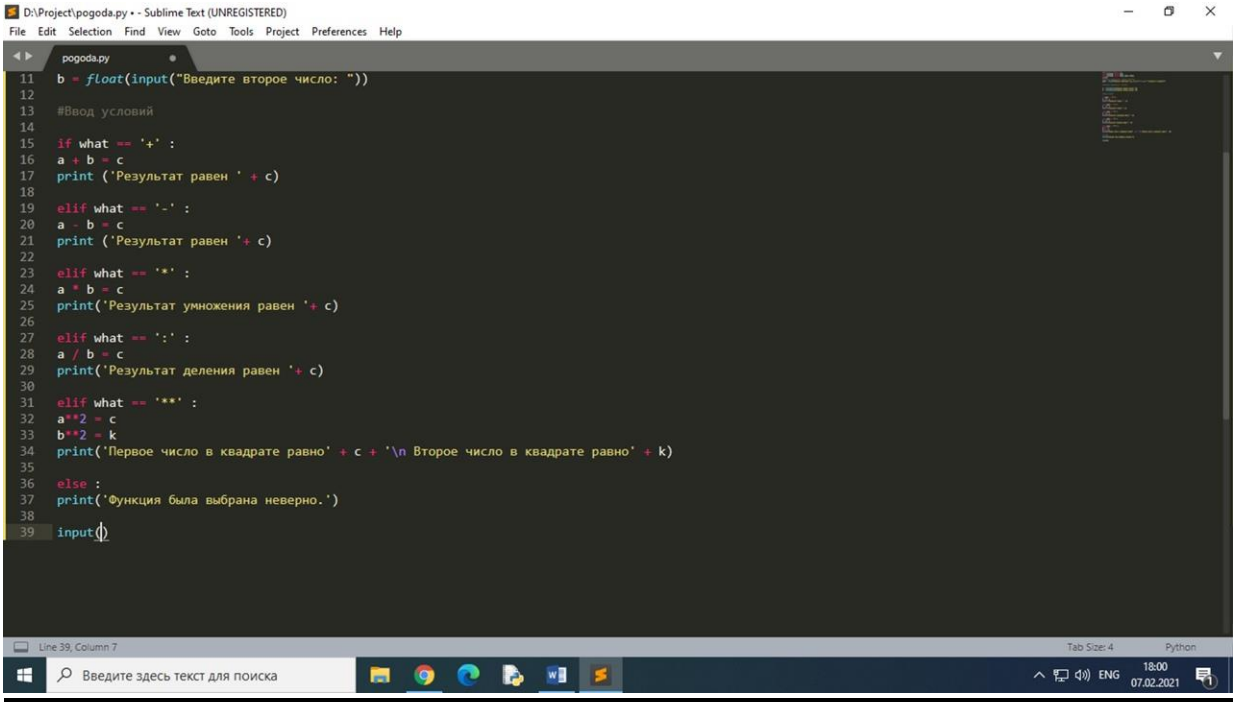
ПРИЛОЖЕНИЕ 4



ПРИЛОЖЕНИЕ 5



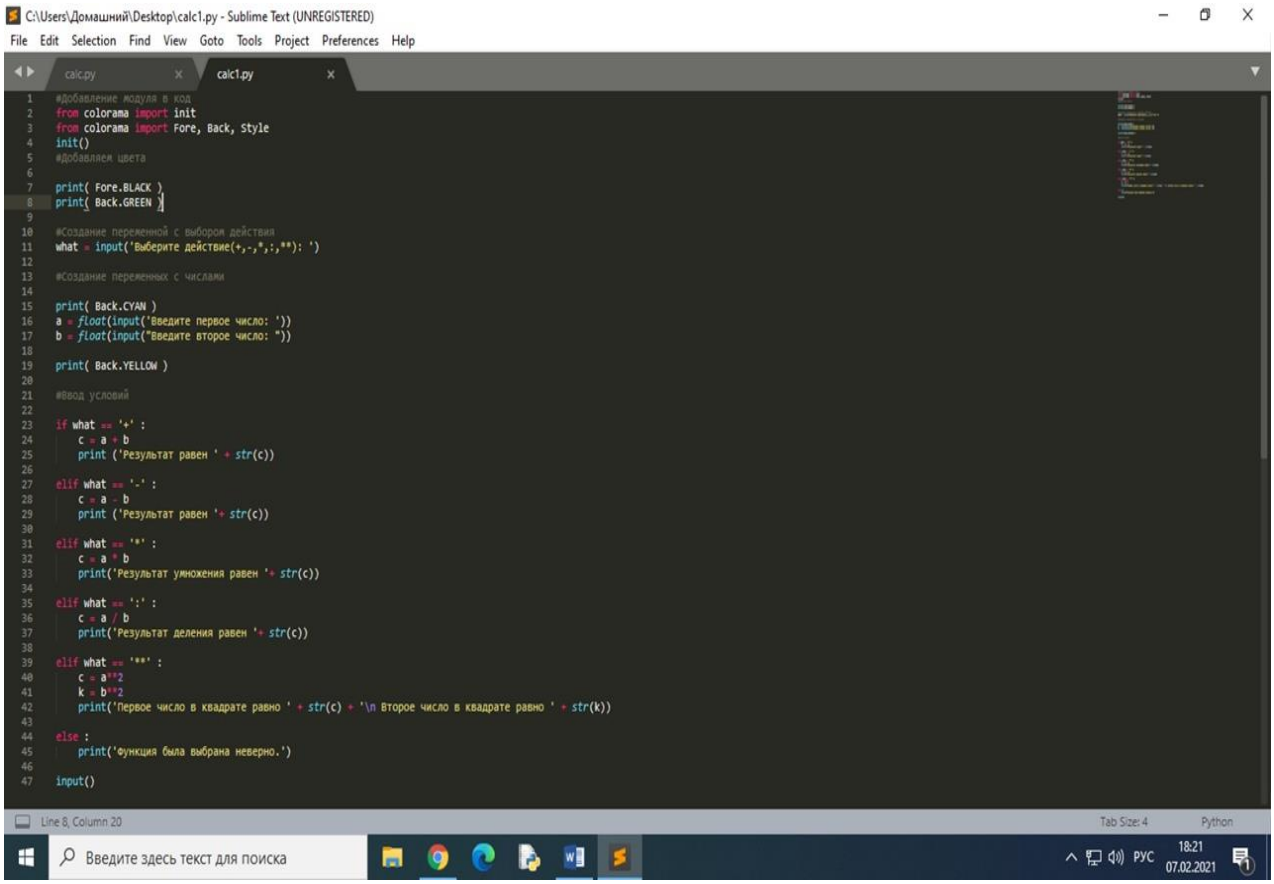
ПРИЛОЖЕНИЕ 6



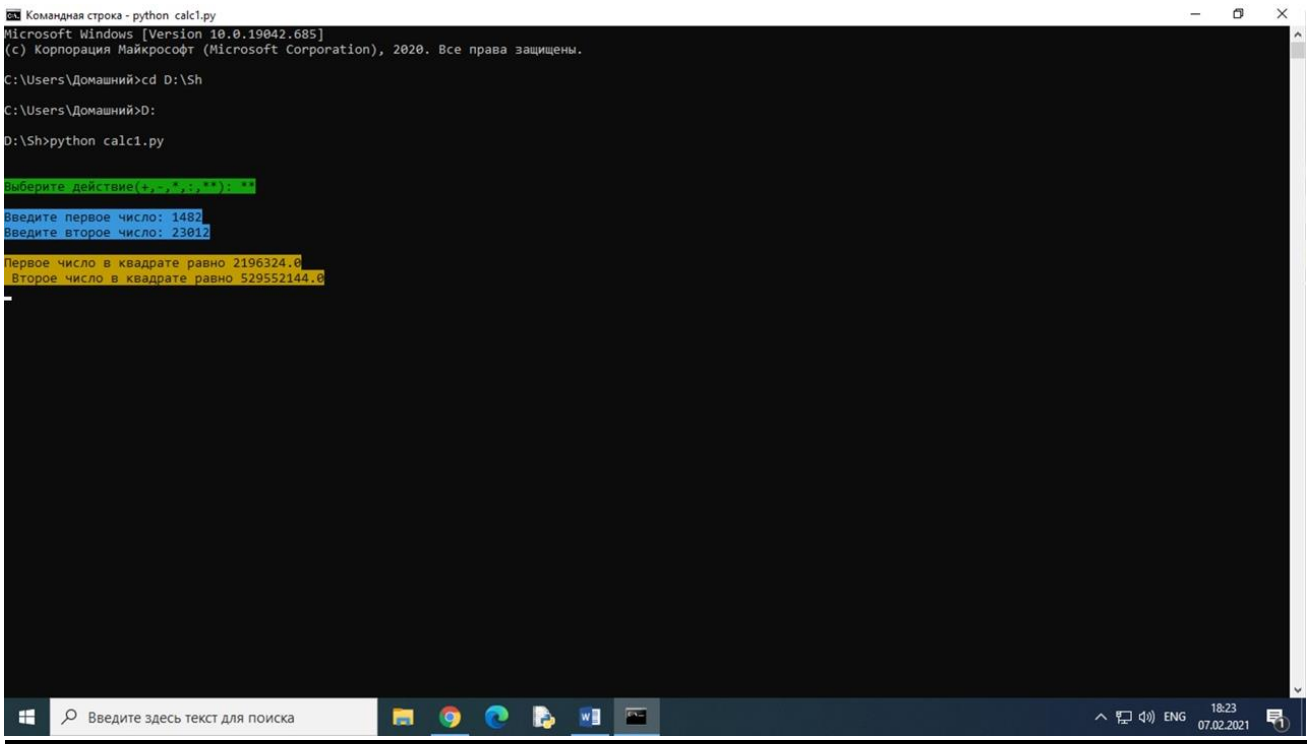
ПРИЛОЖЕНИЕ 7

```
4   init()
5   #Добавляем цвета
6
7   print( Fore.BLACK )
8   print( Back.GREEN )
9
10  #Создание переменной с выбором действия
11  what = input('Выберите действие(+,-,*,:,**): ')
12
13  #Создание переменных с числами
14
15  print( Back.CYAN )
16  a = float(input('Введите первое число: '))
17  b = float(input("Введите второе число: "))
18
19  print( Back.YELLOW )
```

ПРИЛОЖЕНИЕ 8



ПРИЛОЖЕНИЕ 9



СПИСОК ЛИТЕРАТУРЫ

1) История языка программирования Python — Википедия (wikipedia.org)

2) Python. История создания | Робототехника | Яндекс Дзен (yandex.ru)
3) https://pythontutor.ru/
4) https://pypi.org/project/colorama/
5) https://www.tiobe.com/tiobe-index/
6) https://pypl.github.io/PYPL.html