**IT-Колледж “Сириус”**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**ДОКЛАД**

по дисциплине “**Введение в специальность**”

на тему “**Python**”

Выполнил:  
Студент группы

1.9.7.2

Малыгин Дмитрий Андреевич

Принял:

Старший преподаватель  
Тенигин Альберт Андреевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

IT-Колледж “Сириус”  
2022

Оглавление

[**Введение**](#_heading=h.m8cioz64tlxl) **2**

[**Python**](#_heading=h.2qhanzkvn0o0) **3**

[**Как и где используется**](#_heading=h.fa4z80xgk9i7) **3**

[**История создания**](#_heading=h.6f8ocp3qbisp) **4**

[**Концепция и философия языка**](#_heading=h.s7psfyr29kre) **5**

[**Синтаксис**](#_heading=h.9uqm3j686cmv) **6**

[**Отличие Python от других языков программирования**](#_heading=h.ih8jr7fmqi20) **8**

[**Популярность Python**](#_heading=h.iz0f1cr36fcy) **9**

# Введение

Программирование – то, благодаря чему, мы все сейчас можем пользоваться телефонами, компьютерами и другими техническими устройствами. На данный момент существует огромное множество различных языков программирования. Их насчитывается более 8 тысяч. Из них всего малая часть нашла свое применение и некоторые используются по сей день. К языкам, использующимся в программировании или использовавшимися ранее относятся: Java, JavaScript, C, C++, C#, PHP, Python и др. Каждый из них по-своему уникален. Один из самых популярных языков программирования - Python.

# Python

Python - это высокоуровневый язык программирования. Он является полностью объектно-ориентированным. Все элементы в коде - объекты. Он разрабатывался с упором на то, чтобы быть максимально читабельным и доступным к пониманию. Это делает написание кода на Python очень простым, а разработку программного обеспечения на Python — очень быстрой.

# Как и где используется

Python часто используется для разработки серверной части веб-сайта или приложения — частей, которые пользователь не видит. Роль Python в веб-разработке может включать отправку данных на серверы и с серверов, обработку данных и взаимодействие с базами данных, маршрутизацию URL-адресов и обеспечение безопасности. Python предлагает несколько фреймворков для веб-разработки. Обычно используемые включают Django и Flask.

Такие библиотеки, как NumPy, SciPy и Matplotlib, позволяют эффективно использовать Python в научных вычислениях, а специализированные библиотеки, такие как Biopython и Astropy, обеспечивают специфичные для предметной области функции.

Широко распространено мнение, что Python — лучший язык программирования для искусственного интеллекта (ИИ) из-за его простого синтаксиса и быстрого изучения. Что касается машинного обучения, платформа обмена идеями «На пути к науке о данных» сообщает нам, что 57% специалистов по данным и разработчиков машинного обучения используют Python, а 33% отдают ему приоритет в разработке.

Хотя Python не является самым популярным языком программирования, используемым при разработке игр, он по-прежнему невероятно популярен, особенно для более простых игр. Скорость, с которой можно создавать игры на Python, благодаря его простоте, означает, что он также является великолепным вариантом для создания прототипов и разработки идей в игровой индустрии, обеспечивая большую гибкость и более быстрые процессы, чем другие альтернативы.

Python был успешно встроен во многие программные продукты в качестве языка сценариев, в том числе в программное обеспечение для метода конечных элементов, такое как Abaqus, средства параметрического 3D-моделирования, такие как FreeCAD, пакеты 3D-анимации, такие как 3ds Max, Blender, Cinema 4D. Отладчик GNU использует Python в качестве красивого принтера для отображения сложных структур, таких как контейнеры C++. Он также использовался в нескольких видеоиграх и был принят в качестве одного из трех доступных языков программирования в Google App Engine, два других - Java и Go.

Многие операционные системы включают Python в качестве стандартного компонента. Он поставляется с большинством дистрибутивов Linux, т.е является встроенным. Многие дистрибутивы Linux используют установщики, написанные на Python, например, Ubuntu использует установщик Ubiquity.

Python широко используется в индустрии информационной безопасности, в том числе при разработке эксплойтов.

LibreOffice включает Python и намерен заменить Java на Python. Его Python Scripting Provider является основной функцией, начиная с версии 4.0 от 7 февраля 2013 года. Python также используется в Google, Youtube, Yahoo.

Python предназначен не только для программистов и специалистов по данным. Изучение Python может открыть новые возможности для представителей профессий с меньшим объемом данных, таких как журналисты, владельцы малого бизнеса или маркетологи в социальных сетях. Python также может позволить простым людям упростить определенные задачи в своей жизни. Вот лишь несколько задач, которые можно автоматизировать с помощью Python:

* Следить за фондовым рынком или ценами на криптовалюту
* Отправить себе текстовое напоминание о необходимости брать с собой зонт в любое время, когда идет дождь.
* Обновить список покупок в продуктовых магазинах
* Переименовать большие пакеты файлов
* Преобразовать текстовые файлы в электронные таблицы
* Случайное распределение обязанностей по дому между членами семьи
* Автоматическое заполнение онлайн-форм

# История создания

Python разработан человеком по имени Гвидо Ван Россум. В 1980 году он предложил идею по созданию скриптового языка для расширения существующей и развивающейся на тот момент операционной системой Amoeba.

Amoeba - открытая микроядерная ОС, разработанная Эндрю Таненбаумом и его группой. Ее цель - создание системы, которая предоставляла бы пользователю несколько компьютеров как одну рабочую станцию.

В 1989 году началась разработка реализации языка. Гвидо Ван Россум, участвовавший в создании языка ABC, позаимствовал из этого языка наработки для создания своего. Python с самого начала планировался как язык ООП. Название языка появилось от телешоу 1970-х годов “Летающий цирк Монти Пайтона”.

Первая версия Python - Python 3000, которая вышла в 2008 году. В ней были устранены многие недостатки экспериментальных версий языка. При этом вид и структура языка почти полностью сохранилась.

# Концепция и философия языка

Python — мультипарадигменный язык программирования. Объектно-ориентированное программирование и структурное программирование полностью поддерживаются, и многие из их функций поддерживают функциональное программирование и аспектно-ориентированное программирование (включая метапрограммирование и метаобъекты). Многие другие парадигмы поддерживаются через расширения, включая проектирование по контракту и логическое программирование.

Python использует динамическую типизацию и комбинацию подсчета ссылок и сборщика мусора с обнаружением циклов для управления памятью. Он использует динамическое разрешение имен (позднее связывание), которое связывает имена методов и переменных во время выполнения программы.

Его дизайн предлагает некоторую поддержку функционального программирования в традициях Лиспа. Он имеет функции фильтра, карты и сокращения; списки, словари, наборы и генераторные выражения. В стандартной библиотеке есть два модуля (itertools и functools), которые реализуют функциональные инструменты, заимствованные из Haskell и Standard ML.

Основные принципы языка:

* Чем красивее, тем лучше
* Чем точнее, тем лучше
* Чем проще, тем лучше
* Чем читабельнее, тем лучше

# Синтаксис

Python изначально разрабатывался как язык для обучения и простота использования и чистый синтаксис привели к тому, что его приняли как новички, так и эксперты. Чистота синтаксиса Python привела к тому, что некоторые назвали его «исполняемым псевдокодом». Намного проще читать и понимать скрипт Python, чем читать аналогичный скрипт, написанный, скажем, на C.

Python должен быть легко читаемым языком. Его форматирование визуально не загромождено и часто использует английские ключевые слова там, где другие языки используют знаки препинания. В отличие от многих других языков, он не использует фигурные скобки для разделения блоков, а точки с запятой после операторов разрешены, но редко используются. В нем меньше синтаксических исключений и особых случаев, чем в C или Pascal.

В Python, как и в любом другом языке программирования существует много операторов. Оператор - простыми словами, просто команда или набор команд. Самые базовые:

* print - оператор вывода информации
* input - оператор ввода
* def - оператор определения функции
* if-else-elif - оператор условий
* while - оператор цикла
* for - оператор цикла
* class - оператор определения класса
* return - оператор возврата данных после выполнения функции
* import - оператор загрузки библиотек

Этот список - самая малая часть всех, используемых операторов программистами. Но уже с этими операторами можно написать простую программу.

Также, в языке есть и другие различные операторы. Операторы бывают разные: арифметические, логические и так далее. Они нужны для выполнения определенного действия над информацией, которая вводится пользователем или возвращается другой функцией.

Виды операторов:

* Арифметические операторы - используются для выполнения арифметических вычислений (+/-/:/\*/^)
* Логические операторы - используются для проверки того, является ли выражение истинным или ложным. (True/False)
* Операторы присваивания - присваивает значение переменной. (=)
* Операторы сравнения - используются для сравнения каких-либо чисел и переменных между собой. (>/</=)
* Операторы принадлежности - проверяют, является ли значение членом списка, словаря и т.п
* Операторы тождественности - проверяют, имеют ли два операнда одинаковые идентификаторы. (is/is not)
* Битовые операторы - используются для выполнения операций над значениями и переменными. Это специальные символы, которые выполняют арифметические и логические вычисления. Например, вместо оператора AND можно использовать &, вместо OR - |.

Одна из особенностей синтаксиса языка - разделение блоков кода с помощью отступов.В Python нет начальных/конечных скобок, как в Pascal, или фигурных скобок, как в C. Это позволяет сократить количество строк и символов в программе и преображает вид кода. Однако, неправильные отступы могут нарушить работоспособность кода. Для привыкших выделять начало и конец блоков такое поведение может показаться неинтуитивным и неудобным.

# Отличие Python от других языков программирования

Python часто сравнивают с другими интерпретируемыми языками, такими как Java, JavaScript, Perl, Tcl или Smalltalk. Сравнение с C++, Common Lisp и Scheme также может быть поучительным. Эти сравнения концентрируются только на языковых проблемах. На практике выбор языка программирования часто диктуется другими реальными ограничениями, такими как стоимость, доступность, обучение и предварительные инвестиции или даже эмоциональная привязанность. Поскольку эти аспекты очень изменчивы, кажется пустой тратой времени рассматривать их слишком много для этого сравнения.

Обычно ожидается, что программы на Python будут работать медленнее, чем программы на Java, но их разработка также займет гораздо меньше времени. Программы на Python обычно в 3-5 раз короче, чем эквивалентные программы на Java. Это различие может быть связано со встроенными высокоуровневыми типами данных Python и его динамической типизацией. Например, программист Python не тратит время на объявление типов аргументов или переменных, а мощные полиморфные типы списков и словарей Python, для которых богатая синтаксическая поддержка встроена прямо в язык, находят применение почти в каждой программе Python.

Подмножество Python, основанное на объектах, примерно эквивалентно JavaScript. Подобно JavaScript (и в отличие от Java), Python поддерживает стиль программирования, в котором используются простые функции и переменные без участия в определениях классов. Однако для JavaScript это все, что есть. Python, с другой стороны, поддерживает написание гораздо более крупных программ и лучшее повторное использование кода благодаря настоящему объектно-ориентированному стилю программирования, в котором классы и наследование играют важную роль.

Python и Perl происходят из одного и того же источника (скрипты Unix, которые давно переросли) и обладают многими схожими функциями, но имеют разную философию. Perl уделяет особое внимание поддержке общих задач, ориентированных на приложения, например. благодаря встроенным регулярным выражениям, функциям сканирования файлов и создания отчетов. Python делает упор на поддержку общих методологий программирования, таких как проектирование структуры данных и объектно-ориентированное программирование, и поощряет программистов писать читаемый (и, следовательно, удобный для сопровождения) код, предоставляя элегантную, но не слишком загадочную нотацию. Как следствие, Python приближается к Perl, но редко превосходит его в исходной области применения; однако Python имеет применение далеко за пределами ниши Perl.

Как и Python, Tcl можно использовать как язык расширения приложений, а также как самостоятельный язык программирования. Однако Tcl, который традиционно хранит все данные в виде строк, плохо работает со структурами данных и выполняет типичный код гораздо медленнее, чем Python. В Tcl также отсутствуют функции, необходимые для написания больших программ, такие как модульные пространства имен. Таким образом, в то время как «типичное» большое приложение, использующее Tcl, обычно содержит расширения Tcl, написанные на C или C++, специфичные для этого приложения, эквивалентное приложение Python часто может быть написано на «чистом Python». Конечно, разработка на чистом Python намного быстрее, чем писать и отлаживать компонент на C или C++. Говорят, что единственное достоинство Tcl — набор инструментов Tk. Python принял интерфейс к Tk в качестве своей стандартной библиотеки компонентов графического интерфейса.

Возможно, самая большая разница между Python и Smalltalk заключается в более «популярном» синтаксисе Python, что дает ему преимущество в обучении программистов. Как и в Smalltalk, Python имеет динамическую типизацию и привязку, и все в Python является объектом. Однако Python отличает встроенные типы объектов от определяемых пользователем классов и в настоящее время не допускает наследование от встроенных типов. Стандартная библиотека типов данных коллекции Smalltalk более совершенна, в то время как библиотека Python имеет больше возможностей для работы с реалиями Интернета и WWW, такими как электронная почта, HTML и FTP.

Почти все, что сказано о Java, применимо и к C++, только в большей степени: там, где код Python обычно в 3-5 раз короче, чем эквивалентный код Java, он часто в 5-10 раз короче, чем эквивалентный код C++ Неподтвержденные данные говорят о том, что один программист на Python может за два месяца сделать то, что два программиста на C++ не могут сделать за год. Python отлично подходит как связующий язык, используемый для объединения компонентов, написанных на C++.

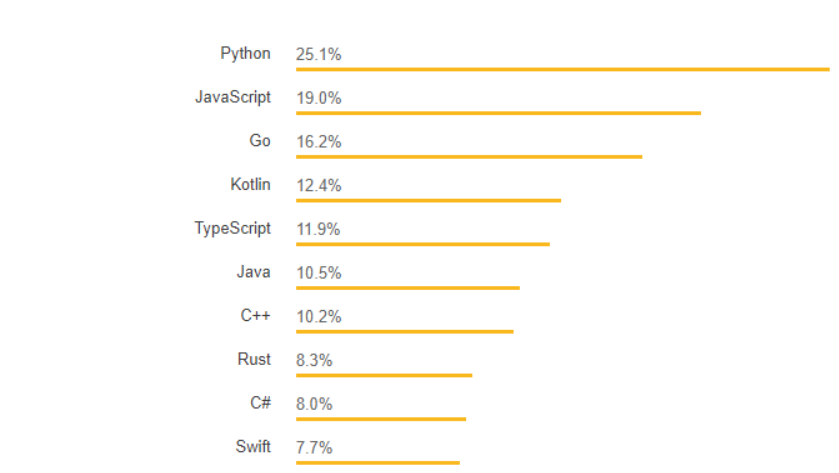
# Популярность Python

Хотя обычно то, что популярно, не всегда является лучшим, в случае с языками программирования популярность окупается.

Благодаря популярности Python вы, скорее всего, найдете готовое решение любой проблемы, с которой вы можете столкнуться. Сообщество энтузиастов Python сильное, и они неустанно работают над улучшением языка каждый день.

У Python также есть ряд корпоративных спонсоров, которые способствуют дальнейшей популяризации языка. Среди них такие технологические гиганты, как Google, который сам использует Python.

Список наиболее любимых пользователями языков программирования на 2022 г.:



Причины, по которым Python так популярен:

* **Легкость изучения -** Python очень легко выучить, потому что большая часть команд означает то же, что и в английском языке. Любой человек, хоть как-то знающий английский, сможет понять что написано в коде Python
* **Гибкость** - Python довольно универсален. Он может использоваться для разработки практически любых видов программ.
* **Отзывчивое, поддерживающее общество** - Все программисты зависят друг от друга. Нет такого человека, который не нуждался бы в поддержке. Сообщество Python, как раз-таки может поддержать и подсказать что и как делать.
* **Веб-разработка** - Python является хорошим выбором для разработки разных видов программ. Помимо программ, на Python можно писать backend сайтов. Хороший frontend на текущий момент вряд ли получится сделать на Python, ибо этот язык не предназначен для frontend’а.
* **Хорошо подходит для аналитики** - Python могут использовать не только программисты, но и журналисты, бизнесмены и т.п. Благодаря этому языку можно изучать, например, какой товар является наиболее продаваемым и наоборот. Можно составлять список покупок или дел на день.
* **Быстрота языка** - По скорости работы Python часто уступает другим языкам программирования, однако по скорости всей разработки Python является лучшим. В этом плане он непревзойден.
* **Python - академический язык** - его преподают в школах, колледжах и т.п. Это связано с быстротой и надежностью языка

# Языки программирования, на которые повлиял Python

Дизайн и философия Python повлияли на многие другие языки программирования:

* Boo использует отступы, аналогичный синтаксис и аналогичную объектную модель.
* Cobra использует отступы и аналогичный синтаксис, а в документе «Благодарности» Python стоит на первом месте среди языков, повлиявших на него.
* CoffeeScript, язык программирования, который кросс-компилируется в JavaScript, имеет синтаксис, вдохновленный Python.
* ECMAScript/JavaScript позаимствовал итераторы и генераторы у Python.
* GDScript, язык сценариев, очень похожий на Python, встроенный в игровой движок Godot.
* Go разработан для «скорости работы на таком динамическом языке, как Python» и использует тот же синтаксис для нарезки массивов.
* Groovy был мотивирован желанием перенести философию дизайна Python на Java.
* Julia была разработана так, чтобы ее можно было «также использовать для общего программирования, как и Python».
* Nim использует отступы и аналогичный синтаксис.
* Создатель Ruby, Юкихиро Мацумото, сказал: «Я хотел, чтобы язык сценариев был более мощным, чем Perl, и более объектно-ориентированным, чем Python. Вот почему я решил разработать свой собственный язык».
* Swift, язык программирования, разработанный Apple, имеет синтаксис, вдохновленный Python.

Методы разработки Python также были скопированы другими языками. Например, практика требования документа, описывающего обоснование и проблемы, связанные с изменением языка (в Python, PEP), также используется в Tcl, Erlang и Swift.

# Разработка

Разработка Python осуществляется в основном с помощью процесса Python Enhancement Proposal (PEP), основного механизма для предложения основных новых функций, сбора отзывов сообщества по проблемам и документирования проектных решений Python. Стиль кодирования Python описан в PEP 8. Выдающиеся PEP рассматриваются и комментируются сообществом Python и руководящим советом.

Улучшение языка соответствует разработке эталонной реализации CPython. Список рассылки python-dev является основным форумом для разработки языка. Конкретные проблемы изначально обсуждались в системе отслеживания ошибок Roundup, размещенной на сайте фонда. В 2022 году все вопросы и обсуждения были перенесены на GitHub. Первоначально разработка велась в собственном репозитории исходного кода под управлением Mercurial, пока Python не переместился на GitHub в январе 2017 года.

Публичные выпуски CPython бывают трех типов, отличающихся тем, какая часть номера версии увеличивается:

* Версии, несовместимые с предыдущими версиями, в которых ожидается, что код сломается и его необходимо будет портировать вручную. Увеличивается первая часть номера версии. Эти выпуски случаются нечасто — версия 3.0 была выпущена через 8 лет после версии 2.0. По словам Гвидо ван Россума, версия 4.0 вряд ли когда-либо появится.
* Основные или «функциональные» выпуски в значительной степени совместимы с предыдущей версией, но содержат новые функции. Увеличивается вторая часть номера версии. Ожидается, что начиная с Python 3.9 эти выпуски будут выпускаться ежегодно. Каждая основная версия поддерживается исправлениями ошибок в течение нескольких лет после ее выпуска.
* Выпуски с исправлением ошибок, которые не вводят новых функций, происходят примерно каждые 3 месяца и производятся, когда с момента последнего выпуска было исправлено достаточное количество ошибок. В этих выпусках также исправлены уязвимости безопасности. Увеличивается третья и последняя часть номера версии.

Многие альфа-, бета-версии и версии-кандидаты также выпускаются в качестве предварительных версий и для тестирования перед окончательными выпусками. Хотя для каждого релиза есть примерное расписание, они часто задерживаются, если код не готов. Команда разработчиков Python отслеживает состояние кода, запуская большой набор модульных тестов во время разработки.

Крупнейшая научная конференция по Python — PyCon. Существуют также специальные программы наставничества по Python, такие как Pyladi.

# Заключение

Python - очень гибкий и довольно простой язык программирования. Его можно использовать не только программистам, но и другим людям для какого-либо анализа данных и, в целом, для тех кто использует какую-то вычислительную технику.

Разработка на Python может быть в несколько раз быстрее, чем на других языках программирования, так как код занимает намного меньше места. Он намного компактнее, чем тот же Java или C++.

Python может освоить любой желающий человек. Если не совсем углубленно, так хотя бы необходимую базу. Поэтому Python является одним из самых популярных языков программирования.

# Приложения

Так выглядит код, на примере, написанного мной, простого калькулятора:

# Список использованной литературы:

1. Марк Лутц “Изучаем Python”
2. Amoeba OS <https://ru.wikipedia.org/wiki/Amoeba_(%D0%BE%D0%BF%D0%B5%D1%80%D0%B0%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0)>
3. Информация про Python <https://ru.wikipedia.org/wiki/Python>
4. Топ языков программирования PYPL <https://pypl.github.io/PYPL.html>
5. Операторы <https://pythonru.com/osnovy/operatory-python>