## Python

## Существует много причин распространения Python, в частности:

##  Простота использования. Для тех, кто новичок в кодировании и программировании, Python может стать отличным первым шагом. Ему относительно легко научиться, это отличный способ начать развивать свои знания по программированию.

##  Простой синтаксис.

## Python относительно легко читать и понимать, поскольку его синтаксис больше похож на обычный английский язык

## Что такое Python, как язык программирования

## В целом Python можно описать одним предложением:

## Высокоуровневый, интерпретируемый, объектно-ориентированный, императивный, строго типизированный язык общего назначения, который имеет динамическую типизацию.

## Высокоуровневый

## Языки программирования делятся на высокоуровневые и низкоуровневые. Низкоуровневые языки — языки, близкие к машинному коду или его конструкциям (например, байт-кодам). Классикой таких языков являются C, Assembler, Forth.

## Высокоуровневые языки — соответственно разрабатываются для удобства использования и скорости написания программы. В них применяются определённые абстракции — структуры данных, набор вспомогательных функций и так далее. Это такие языки как Python, JS, PHP, Go,

## Интерпретируемый

## Языки делятся на интерпретируемые(Python, JS, PHP, R, Ruby) и компилируемые (С, С++, Pascal). В первом случае программа выполняется специальной программой —  интерпретатором, во втором программа сначала преобразуется понятные компьютеру исполняемые файлы.

## Объектно-ориентированный

## Все языки также разделяются  на процедурные, функциональные и объектно-ориентированные в зависимости от того, с помощью каких конструкций создаётся программа и как происходит её выполнение.

## В объектно-ориентированных языках основа это классы и экземпляры классов это равносильно типу и объекту этого типа. Выполнение условных задач или же просто работа программы строится на взаимодействии различных классов.

## Python хоть и является объектно-ориентированным языком но также

## поддерживает и процедурное программирование это значит, что программу можно написать без единого класса.

## В основе функциональных языков лежит отличная от предыдущих вычислительная система, называемая лямбда-исчисление, которая, тем не менее, эквивалентна машине Тьюринга, о чём доказана соответствующая теорема

## Императивный

## Языки программирования также могут быть разделены на импертивные и декларативные. В императивном языке программист будет указывать последовательность команд для выполнения (это все языки

## программирования которые мы считаем «языками программирования» )

## Декларативные же языки в свою очередь ожидают от нас описания результата, который мы хотим получить в ходе выполнения запроса. Яркий пример декларативного языка это SQL (Stucted Query Language) или же структурированный язык запросов. Именно в нём мы описываем конкретный результат выполнения программы а не последовательность команд. Декларативными языками еще называют HTML, CSS, SVG, VRML, SQL, lex/VACC.

## Для понимания определений императивного и декларативного языка зададим задачу: построить дом.

## Императивный: Пойти в магазин —> Купить материалы —> Подготовить площадку —> Произвести строительство —> Заселиться.

## Декларативный: Я хочу новый дом.

## Строго (сильно) типизированный

## Что касается типизации. В сильно типизированном языке интерпретатор, при выполнении команд, не станет неявно приводить типы в отличии от слабо типизированных языков, в котором приведения типа могут происходить неявно.

## Общего назначения

## Языки делятся на языки общего назначения (Python, Java, Kotlin, C, Ruby) и специализированные (например предметно-ориентированные языки или DSL)

## DSL(предметно-ориентированный язык) — компьютерный язык, специализированный для конкретной области применения (в противоположность языку общего назначения, применимому к широкому спектру областей и не учитывающему особенности конкретных сфер знаний). Построение такого языка и/или его структура данных отражают специфику решаемых с его помощью задач. Является ключевым понятием языково-ориентированного программирования.  Примеры таких языков Perl, SQL, HTML, Haskell, Verilog, AutoLisp

## Динамическая типизация

## Динамическая типизация предполагает, что в процессе выполнения команды переменная может содержать объекты различных типов. То есть мы объявляем переменную не указывая явно, какой тип данных в ней будет содержаться, и в процессе выполнения программы в одной переменной может побывать как текст так и число, а может и булево значение.

## Статическая типизация предполагает, что при задании переменной сразу указывается тип данных, который она может содержать.

## Отличительные особенности языка Python

## Интроспекция

## В программировании это это способность программы получать различную информацию об объектах во время работы программы. Если конкретно, то вы можете узнать какому классу принадлежит объект, с каким типом данных вы работаете в данный момент (помните про динамическую типизацию?) или список атрибутов и методов, доступных для указанного объекта

## Примеры полезных функций интроспекции в Python:

## dir();

## type();

## hasattr();

## id();

## isinstance().

## Мультиплатформенность

## Нет, ну Java тоже, да. Но Python не отстает, он идёт, что называется, в коробке с Linux и SQL Server, интерпретатор Python можно легко поставить на Windows, не забыли про macOS, и iOS. Да вообще везде работает, в 8 раз медленней чем Swift на iOS, но работает.

## Встроенные паттерны проектирования

## В Python существуют свои встроенные паттерны проектирования, лучшим примером для понимания служат Декоратор и Итератор.

## Декоратор позволяет расширять функциональность без наследования. Чтобы разобраться в паттерне Итератор, следует просто изучить механизм работы итераторов и генераторов языка (о них я писал здесь)

## Обширный набор стандартных библиотек

## При использовании Core Python вы получаете «в коробке» бополезных стандартных библиотек. Примером может служить библиотека для работы с БД SQLite, а также json, math, re, random, csv, html и много других.

## Читаемость языка

## Форматирование текста в Python является частью его синтаксиса и да, это те самые отступы (пробелы, табуляция кому как нравится), которые позволяют повышать читаемость кода.

## Что можно сделать на Python новичку?

## Пять идей простых проектов на Python

## Генератор случайных чисел

## Отгадай число

## Генератор текстов

## Текстовый квест

## «Угадай слово» («Виселица»)

# Зачем разработчики используют Python.

# Python является одним из самых популярных языков общего назначения. Это значит, что на нём можно писать практически всё — от веб-приложений до серьезных низкоуровневых системных штук. Так уж получилось, что у меня есть несколько студентов, род деятельности которых практически никак не связан с программированием, но им захотелось изучать Python. Например, чтобы автоматизировать какие-то ежедневные рутинные задачи, или чтобы лучше понимать исследования, которые проводит аналитический отдел в их компании. Порог вхождения не очень высок, что и позволяет добиться значимых результатов в довольно короткие сроки, а видеть результаты своего труда всегда приятно. Несмотря на то, что язык существует довольно давно, он активно развивается, вбирая в себя всё новые возможности и щеголяя гигантским набором модулей расширения. Кроме того, довольно большое количество разных курсов, начиная от низкоуровневой работы с сетью и кончая Data Science, переехали с других языков на Python в новых версиях. Это можно заметить, например, по проектам на Coursera.

# Пусть даже Python и не является явным лидером в каких-то конкретных областях, обычно он достаточно хорош, чтобы решать задачи практически в любой предметной области. И на горизонте навскидку я не могу назвать других языков, которые были бы настолько универсальны.

# Python в настоящее время переживает пик своей популярности, на нём разрабатывают во множестве компаний, на конференции и митапы по данному языку собирается сотни разработчиков. Всё это делает язык «пайтон» не только востребованным, но и простым в изучении. В связи с этим язык «пайтон» стоит изучать.

#  Вообще, перспективы языка Python отличные, в данный момент он является вторым лучшим языком для любой задачи. «Пайтон» активно развивается, о чем свидетельствует сокращение времени релизов новых версий с восемнадцати до девяти месяцев.

# Это примерно столько, сколько нужно времени для изучения базовых основ Python. Хотя все люди разные и для освоения одних и тех же навыков им нужно разное время. А самое главное — на рынке нет единых требований к джуниору. В связи с этим срок обучения может колебаться от 3 месяцев до года.

# Заключение

# Python — красивый, лаконичный и многофункциональный язык. Он применяется в совершенно не сочетающихся между собой сферах деятельности, способен в одиночку создавать проекты и не смотреть на возможности конкурентных языков.

# Частой проблемой, встречающейся среди разработчиков-любителей является поверхностность в изучении Python. Обучающие материалы, которые вы встречаете в интернете перефразируются сотни раз и скрывают важные, полезные черты и особенности языков программирования.

# Очень важно восполнять белые пятна в своих знаниях не только в начале своего обучения но и на протяжении всей своей карьеры иначе завтра фраза «дай дорогу молодым» станет для вас актуальной, и вы окажетесь за бортом.