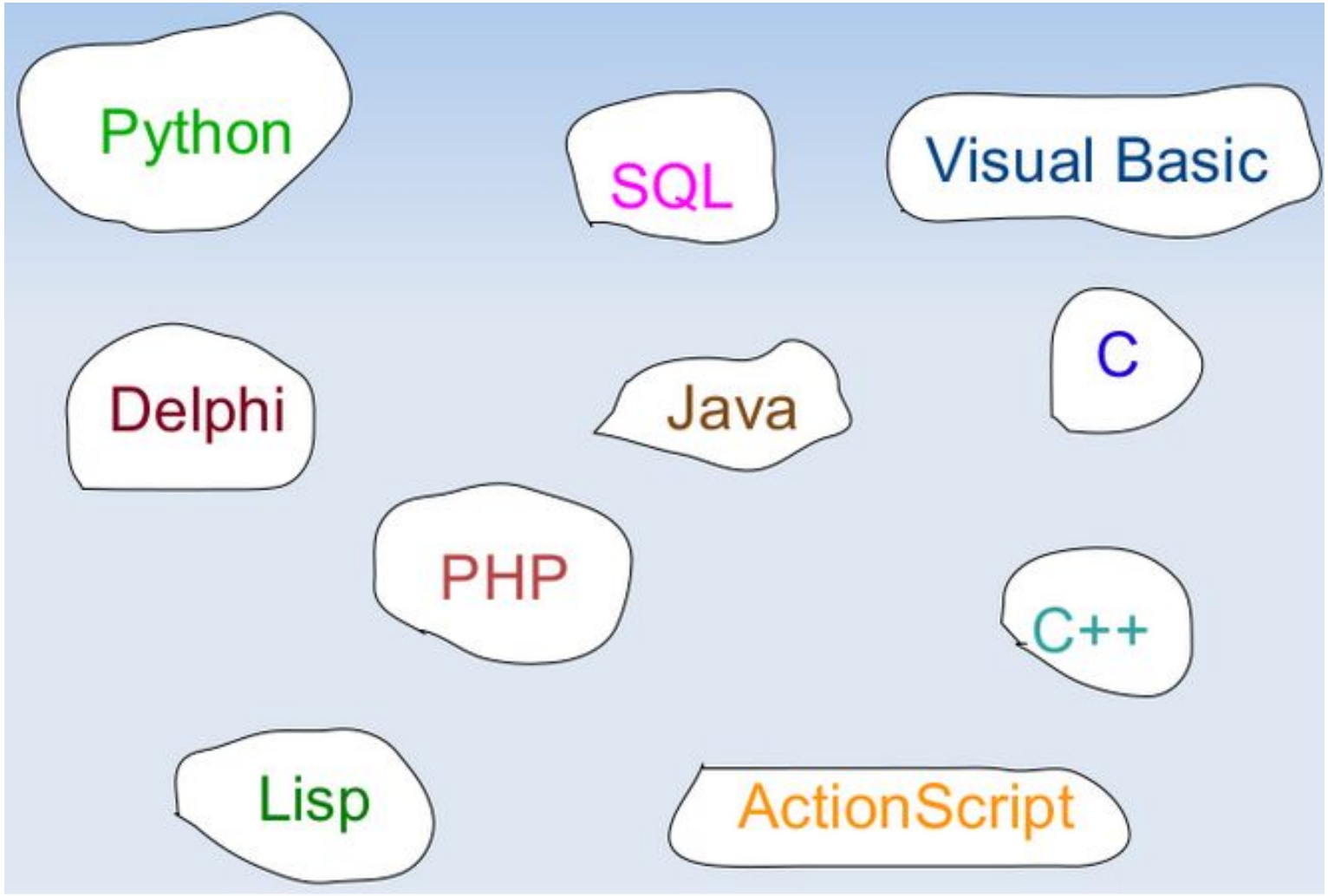
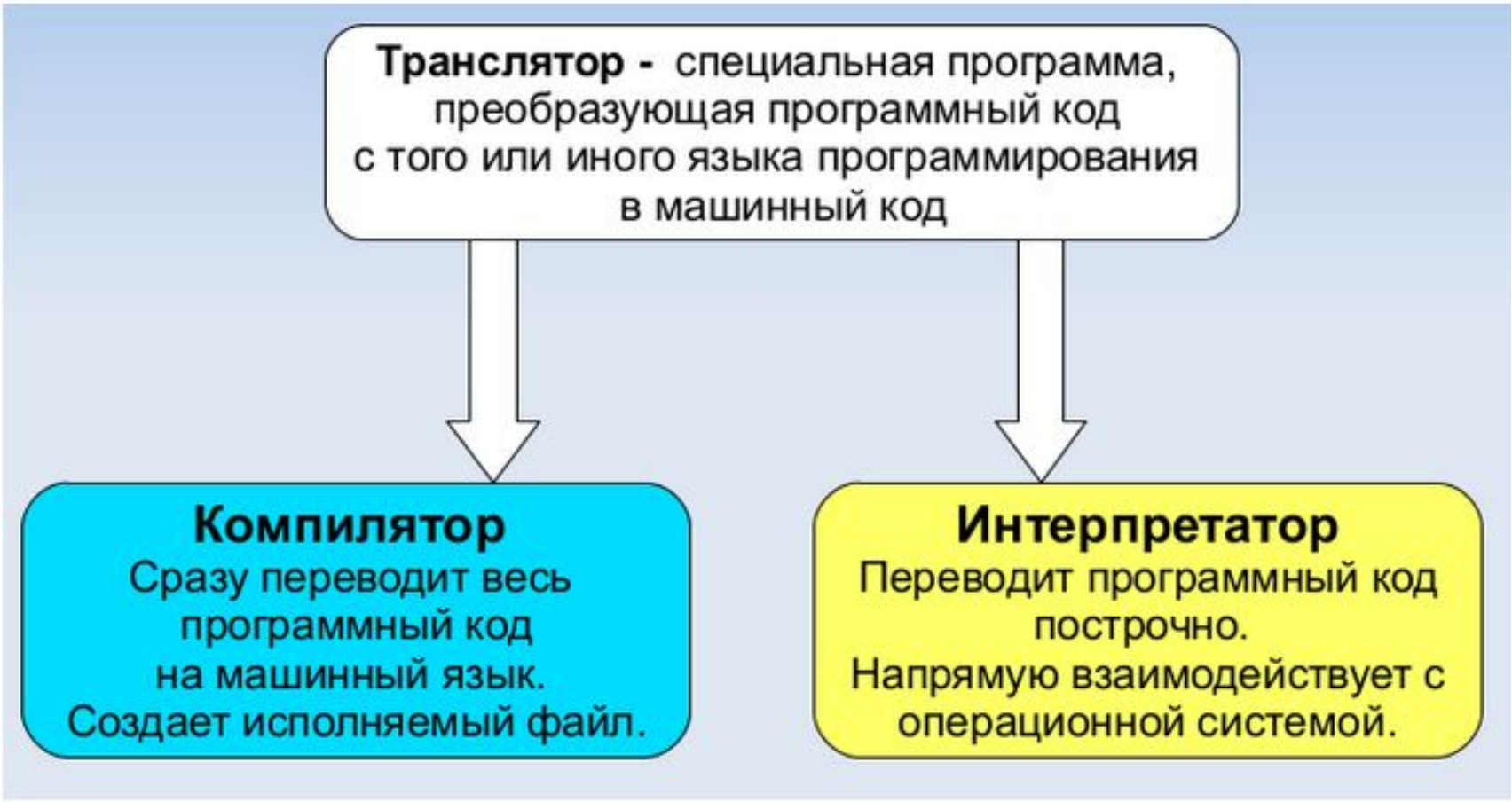
Python


Машинные языки
Ассемблеры
Языки высокого уровня
Объектно-ориентированные языки


Python
SQL
Visual Basic
C
Delphi
Java
PHP
C++
Lisp
ActionScript


Транслятор - специальная программа, преобразующая программный код с того или иного языка программирования в машинный код
Компилятор
Сразу переводит весь программный код на машинный язык.
Создает исполняемый файл.
Интерпретатор
Переводит программный код построчно.
Напрямую взаимодействует с операционной системой.

Python - высокоуровневый язык программирования общего назначения с акцентом на производительность разработчика и читаемость кода. Синтаксис ядра Python минималистичен.

Python поддерживает несколько парадигм программирования, в том числе структурное, объектно-ориентированное, функциональное, императивное и аспектно-ориентированное. Основные архитектурные черты — динамическая типизация, автоматическое управление памятью, полная интроспекция, механизм обработки исключений, поддержка многопоточных вычислений.

Почему Python?

- Прост в изучении
- Большое количество модулей
- Простые конструкции
- …

Python 2.x vs 3.x

- В третьей версии улучшена стандартная библиотека и добавлены новые функции.
- Много библиотек не стабильно работают на версии 3.

- На данный момент 2.7.5 vs 3.3.2

Краткая историческая справка

Язык программирования Python был создан к 1991 году голландцем Гвидо ван Россумом.

Свое имя – Пайтон (или Питон) – получил от названия телесериала, а не пресмыкающегося.

После того, как Россум разработал язык, он выложил его в Интернет, где сообщество программистов присоединилось к его улучшению.

В каких областях применяется Python?

Web-разработка

В этой области Python, пожалуй, используется больше всего. Веб-фреймворк Django продолжает набирать обороты, пополняя армию своих фанатов. Многие начинающие программисты даже думают, что Python больше нигде не используется. Но на Python написаны многие другие веб-фреймворки: Pylons, TurboGears, CherryPy, Flask, Pyramid и другие. С более полным списком можно ознакомится здесь.
Есть и CMS на базе Django, она так и называется DjangoCMS.
Очень часто на Python пишут и парсеры сайтов. Обычно для этого используют Requests, aiohttp, BeautifulSoup, html5lib.
Есть и более высокоуровневые инструменты для парсинга сайтов: Scrapy, Grab.

Системное администрирование

Python - это отличный язык для автоматизации работы системного администратора. Он установлен по умолчанию на все Linux-сервера. Он простой, понятный. Код на Python легко читается. Некоторые любят Perl, я тоже его люблю за удобную работу с регулярными выражениями, но я ненавижу Perl за его синтаксис. Bash удобен для относительно небольших и средних скриптов, но Python мощнее и в некоторых случаях позволяет писать намного меньше кода.

Встроенные системы (embedded systems)

Очень часто Python используется для программирования встроенных систем. Самый известный проект, который использует Python - это Raspberry Pi. Но он не единственный:
Embedded Python
Raspberry Pi
Python Embedded Tools
The Owl Embedded Python System

Разработка прикладного ПО, в том числе игр

Python часто используется как вспомогательный язык при разработке прикладного программного обеспечения. Примеры я уже приводил выше, не буду повторяться.

Научные исследования

Физики и математики очень любят Python за его простоту. Кроме того для Python существует огромное количество библиотек, облегчающих жизнь ученому. Например:

1. **SciPy** — это открытая библиотека высококачественных научных инструментов для языка программирования Python. SciPy содержит модули для оптимизации, интегрирования, специальных функций, обработки сигналов, обработки изображений, генетических алгоритмов, решения обыкновенных дифференциальных уравнений и других задач, обычно решаемых в науке и при инженерной разработке.
2. **Matplotlib** — библиотека на языке программирования Python для визуализации данных двумерной (2D) графикой (3D графика также поддерживается). Получаемые изображения могут быть использованы в качестве иллюстраций в публикациях.
3. **NumPy** — это расширение языка Python, добавляющее поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых математических функций для операций с этими массивами.

Синтаксис

Во первых стоит отметить интересную особенность Python. Он не содержит операторных скобок (begin..end в pascal или {..}в Си), вместо этого **блоки выделяются отступами**: пробелами или табуляцией, а вход в блок из операторов осуществляется двоеточием. Однострочные комментарии начинаются со знака фунта «#», многострочные — начинаются и заканчиваются тремя двойными кавычками «"""».

Чтобы присвоить значение пременной используется знак «=», а для сравнения — «==». Для увеличения значения переменной, или добавления к строке используется оператор «+=», а для уменьшения — «-=». Все эти операции могут взаимодействовать с большинством типов, в том числе со строками. Например

```
>>> myvar = 3
>>> myvar += 2
>>> myvar -= 1
"""«Это многострочный комментарий
Строки заключенные в три двойные кавычки игнорируются»"'
>>> mystring = «Hello»
>>> mystring += " world."
>>> print mystring
Hello world.
# Следующая строка меняет
значения переменных местами. (Всего одна строка!)
>>> myvar, mystring = mystring, myvar
```

Структуры данных

Python содержит такие структуры данных как **списки (lists), кортежи (tuples) и словари (dictionaries)**. Списки — похожи на одномерные массивы (но вы можете использовать Список включающий списки — многомерный массив), кортежи — неизменяемые списки, словари — тоже списки, но индексы могут быть любого типа, а не только числовыми. "Массивы" в Python могут содержать данные любого типа, то есть в одном массиве может могут находиться числовые, строковые и другие типы данных. Массивы начинаются с индекса 0, а последний элемент можно получить по индексу -1 Вы можете присваивать переменным функции и использовать их соответственно.

Строки

Строки в Python **обособляются кавычками двойными «"» или одинарными «'»**. Внутри двойных ковычек могут присутствовать одинарные или наоборот. К примеру строка «Он сказал 'привет'!» будет выведена на экран как «Он сказал 'привет'!». Если нужно использовать строку из несколько строчек, то эту строку надо начинать и заканчивать тремя двойными кавычками «"""». Вы можете подставить в шаблон строки элементы из кортежа или словаря. Знак процента «%» между строкой и кортежем, заменяет в строке символы «%s» на элемент кортежа. Словари позволяют вставлять в строку элемент под заданным индексом. Для этого надо использовать в строке конструкцию «%(индекс)s». В этом случае вместо «%(индекс)s» будет подставлено значение словаря под заданным индексом.

Преимущества Python

- Скорость выполнения программ написанных на Python очень высока. Это связанно с тем, что основные библиотеки Python
  написаны на C++ и выполнение задач занимает меньше времени, чем на других языках высокого уровня.

- В связи с этим вы можете писать свои собственные модули для Python на C или C++

- В стандартных библиотеках Python вы можете найти средства для работы с электронной почтой, протоколами
  Интернета, FTP, HTTP, базами данных, и пр.

- Скрипты, написанные при помощи Python выполняются на большинстве современных ОС. Такая переносимость обеспечивает Python применение в самых различных областях.

- Python подходит для любых решений в области программирования, будь то офисные программы, вэб-приложения, GUI-приложения и т.д.

- Над разработкой Python трудились тысячи энтузиастов со всего мира. Поддержкой современных технологий в стандартных библиотеках мы можем быть обязаны именно тому, что Python был открыт для всех желающих.

Синтаксис

```
>>> print "Hello, Python!"
```

```
>>> print ("Hello, Python!")
```

```
if (a == 1 and b == 2 and
    c == 3 and d == 4):   # Не забываем про двоеточие
      print('spam' * 3)
```

```
if x > y: print(x)
```

```
a = 1; b = 2; print(a, b)
```

Идентификаторы в Python

Идентификаторы в **Python** — это имена, используемые для определения («*идентификации*») переменных, функций, классов, модулей и других объектов. Идентификатор начинается с букв `A-Z` или `a-z`, либо знака подчеркивания (_), после чего следуют ноль или больше букв (*совет — никогда не создавайте свою собственную переменную с именем «_», т.к. это имя зарезервировано самими интерпретатором), знаков подчеркивания или цифр от 0 до 9.

В идентификаторах **Python** не используются знаки `@`, `$` и `%`. Так же — **Python** чувствителен к регистру символов, т.е. `Manpower` и `manpower` являются двумя различными именами (*идентификаторами*).

Вот основные правила именования идентификаторов в **Python**:

- Имена классов начинаются с заглавной буквы, а все остальные идентификаторы — со строчных;
- Если идентификатор начинается со знака подчёркивания (_) — то он является приватным;
- Starting an identifier with two leading underscores indicates a strongly private identifier.
- If the identifier also ends with two trailing underscores, the identifier is a language-defined special name.

```python
A = 3.14
print(type(A)) # float
A = 'Hello'
print(type(A)) # str
a = b = c = 0
a += 1     # a = a + 1
c = 5//2   # int
d = 5/2    # float
b = c**2   # b = c^2 (степень)
a, b = b, a # обмен значениями a=b, b=a
```