

Основы Python для работы с научными данными

к.ф.-м.н.
Ямалетдинов Руслан Дамирович
ИНХ СО РАН

Цели и задачи курса

Знакомство обучающихся с языком программирования Python, для работы с научными данными

Базовые возможности языка

Математические и научные библиотеки

Некоторые подходы вычислительной математики

Наработка опыта автоматизации рутинных задач

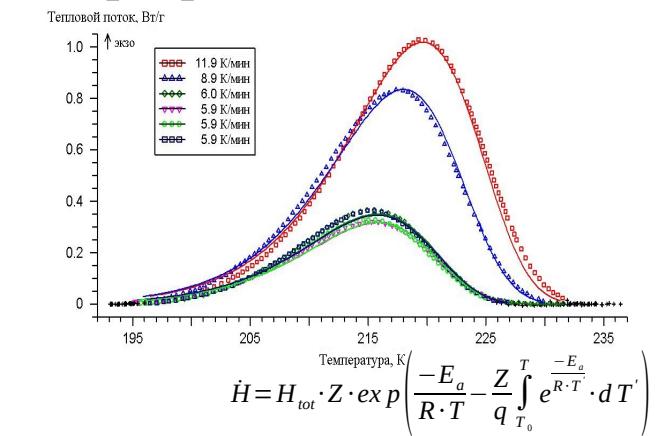
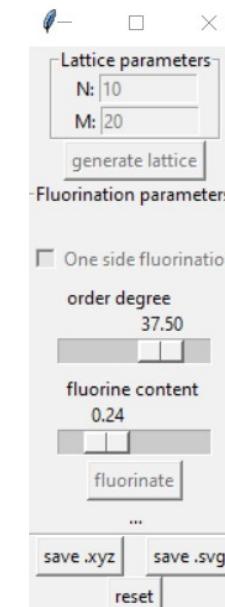
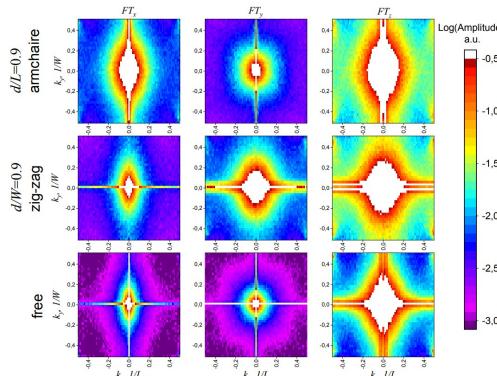
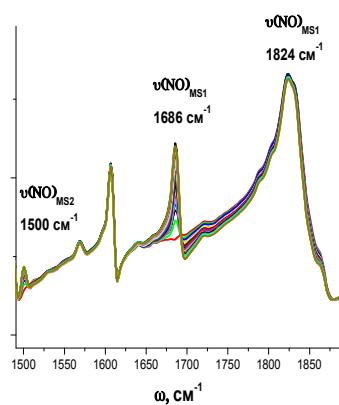
Работа с компьютерной документацией

Содержание курса

- Введение в курс и основную терминологию. Основы написания программного кода. Изучение базового функционала и особенностей языка Python.
- Базовые понятия о представлении компьютерного числа. Работа и оценка алгоритмов вычислительной математики.
- Работа с математическими конструктами с помощью базовых и специализированных средств языка Python.
- Способы реализации и использования продвинутых алгоритмов вычислительной математики реализованных в специализированных научных библиотеках языка Python.
- Основные методы и специфика анализа экспериментальных данных.
- Построение графиков и визуализация результатов средствами специализированных библиотек языка python.

Зачем это все надо?

- Рутинные задачи
- Сложная обработка
- Много всяких других штук



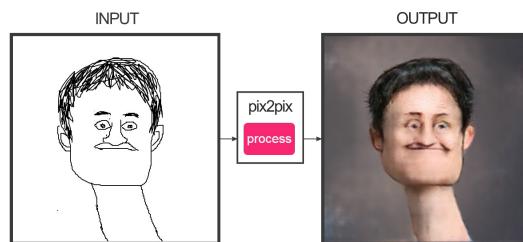
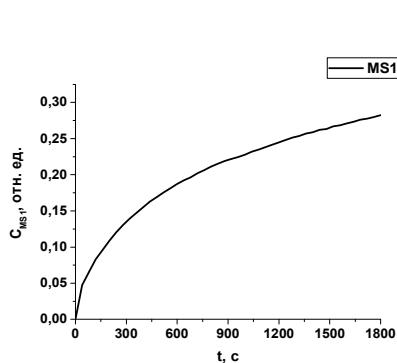
Основы Python для работы с научными данными

Курс Основы Python для работы с научными данными ставит своей целью знакомство студентов с языком программирования Python, для работы с научными данными.

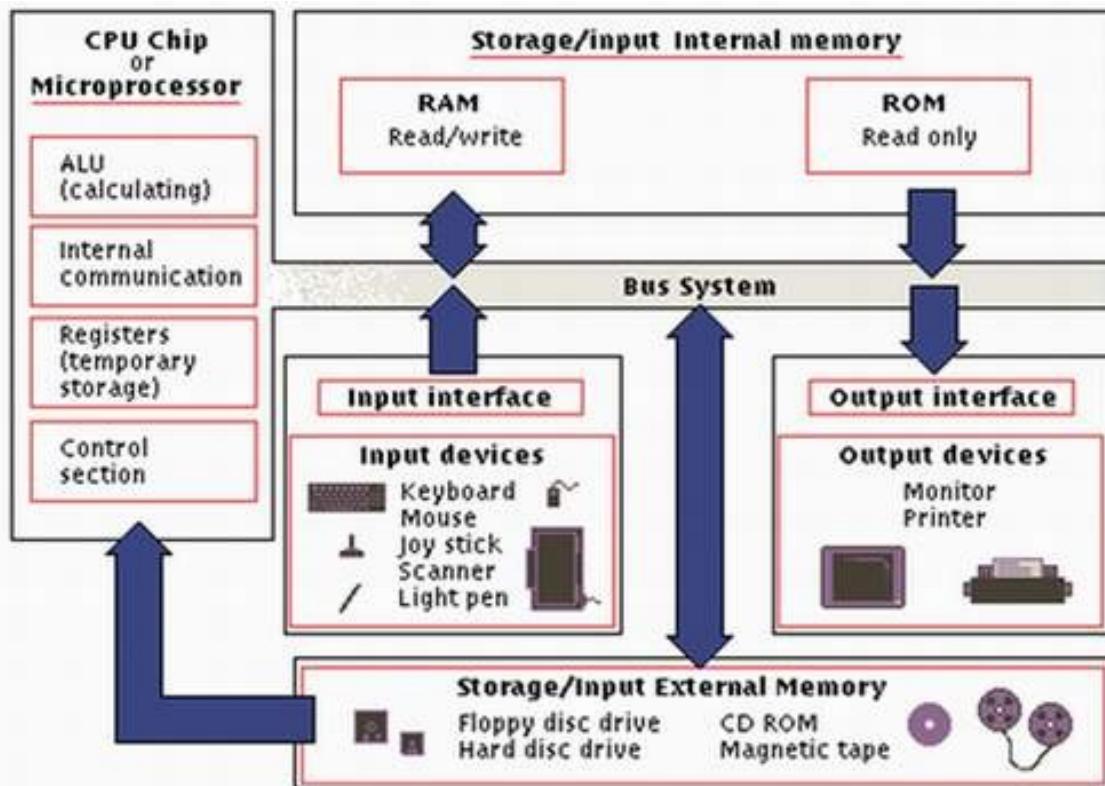
В рамках курса аспиранты будут ознакомлены с некоторыми конструкциями языка Python. Получат базовый обзор математических и научных библиотек, таких как NumPy и SciPy. Узнают некоторые подходы вычислительной математики. В качестве учебных и тренировочных примеров предполагается использование реальных или приближенных к ним задач.

Основными задачами курса являются наработка опыта программирования для алгоритмизации рутинных компьютерных задач связанных с обработкой экспериментальных данных и простейшим компьютерным моделированием, знакомство с методами вычислительной математики, математическими и научными библиотеками, получение навыка работы с программной и компьютерной документацией.

- 1. Введение. Установка интерпретатора Python
 - 1.1. О языке Python
 - 1.2. Установка интерпретатора Python
 - 1.2.1. Установка Anaconda на свой компьютер
 - 1.2.2. Установка стандартных средств для python разработки
- 2. Основы Синтаксиса Python
 - 2.1. Форматирование блоков кода
 - 2.2. Ввод и вывод в консоль
 - 2.3. Ключевые слова и встроенные функции в Python
 - 2.4. Основные операторы и типы
 - 2.4.1. Обзор Типов данных
 - 2.4.2. Операторы сравнения
 - 2.4.3. Условные операторы

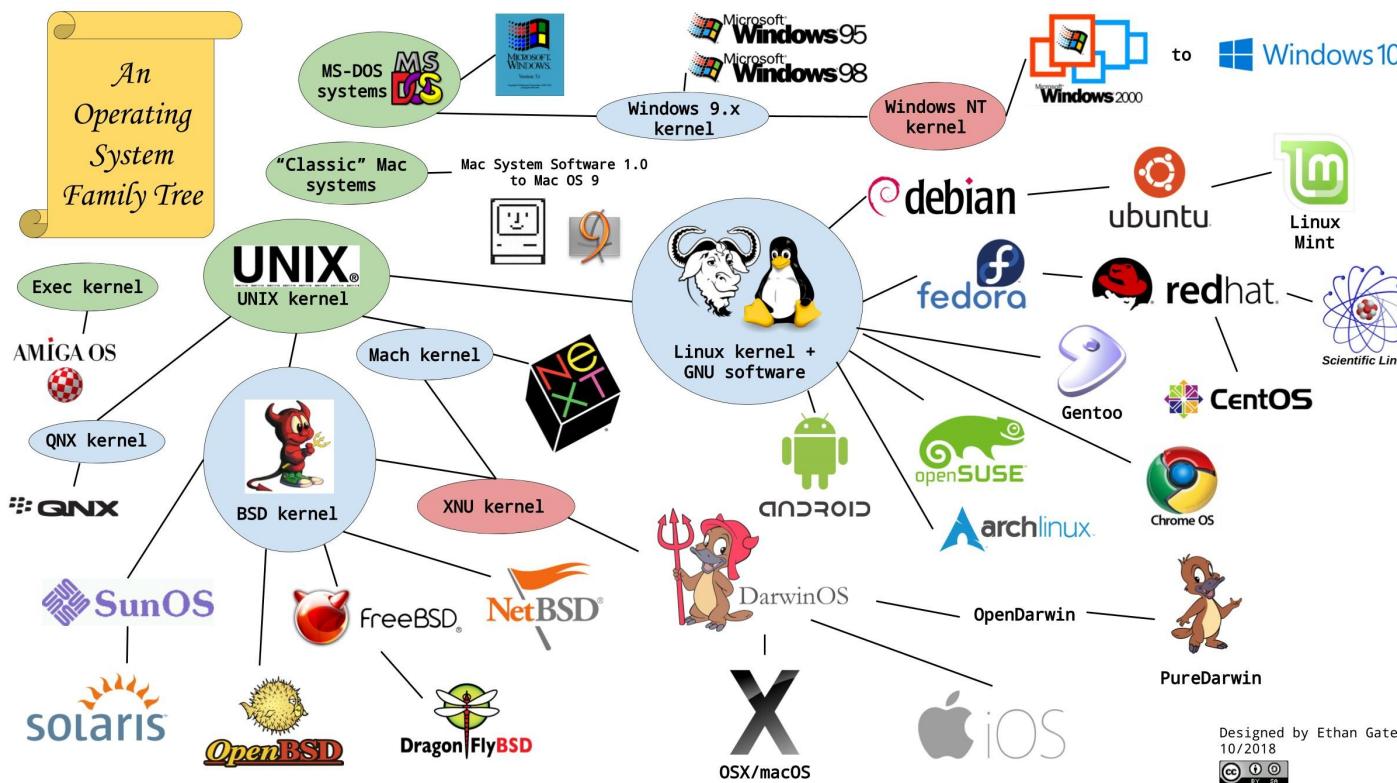
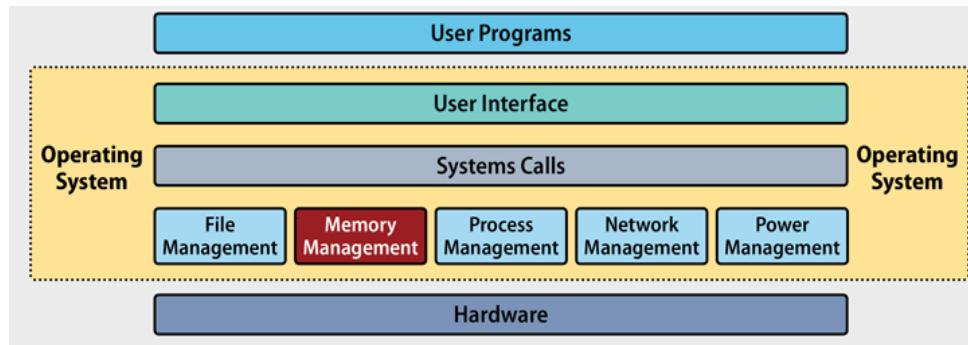


Как работает компьютер

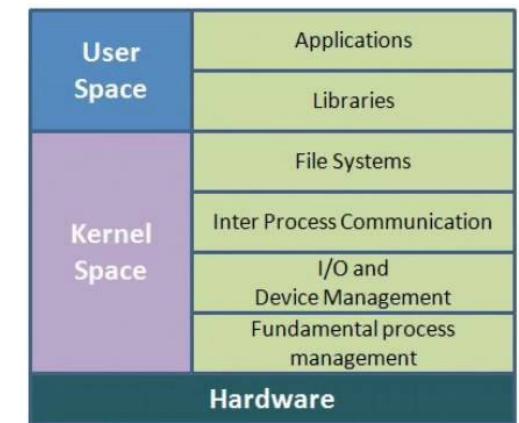


Normal instructions				Compressed	
00000	SUB	10000	CMP	000	SUB
00001	AND	10001	TEST	001	AND
00010	ADD	10010	LW	010	ADD
00011	OR	10011	SW	011	CMP
00100	XOR	10100	LH	100	LW
00101	LSR	10101	SH	101	SW
00110	LSL	10110	LB	110	LDI
00111	ASR	10111	SB	111	MOV
01000	BREV	11000	LDI		Reserved for FPU
01001	LDILO	11001			
01010	MPYUHI			11010	FPADD
01011	MPYSHI		Special Insn	11011	FPSUB
01100	MPY	11100	BREAK	11100	FPMPY
01101	MOV	11101	LOCK	11101	FPDIV
01110	DIVU	11110	SIM	11110	FPI2F
01111	DIVS	11111	NOOP	11111	FPF2I

Как работает операционная система



Monolithic Kernel



ЯЗЫКИ ПРОГРАММИРОВАНИЯ

Virtual Reality and Programming Hierarchy

Scripting/Interpreted Languages

Perl, Python, Shell, Java

Average humans

High/Middle Level Languages

C, C++

(What Most Malware Is Written In)

Gifted humans

Assembly Language

Intel X86, etc.

(First Layer of Human Readable Code)

Intuitive "Old Soul"

Machine Code

Hexadecimal representations of Binary Code Read
By The Operating System

Chiu-fang Kao
Liezi Book 8 Part 9

Binary code

Binary code read by hardware
Not Human Readable

Mental
Illness

→ 0804:9165	e8 00 00 00 00	call 0x0804916a
0804:916a	5d	pop ebp
0804:916b	83 ed 05	sub ebp, 5
0804:916e	31 c9	xor ecx, ecx
0804:9170	31 c0	xor eax, eax
0804:9172	51	push ecx
0804:9173	8a 94 0d 99 00 00 00	mov dl, byte ptr [ebp+ecx+153]
0804:917a	84 d2	test dl, dl
0804:917c	74 08	jmp 0x08049186
0804:917e	f2 0f 38 f0 c2	crc32 rax, dl
0804:9183	41	inc ecx
0804:9184	eb ed	jmp 0x08049173
0804:9186	59	pop ecx
0804:9187	39 84 8d 3d 00 00 00	cmp dword ptr [ebp+ecx*4+61], eax
0804:918e	75 0c	jmp 0x0804919c
0804:9190	41	inc ecx
0804:9191	80 f9 17	cmp cl, 23
0804:9194	75 da	jmp 0x08049170

Language Rank	Types	Spectrum Ranking
1. Python	🌐💻📱	100.0
2. C++	💻📱	99.7
3. Java	🌐💻📱	97.5
4. C	💻📱	96.7
5. C#	🌐💻📱	89.4
6. PHP	🌐	84.9
7. R	💻	82.9
8. JavaScript	🌐📱	82.6
9. Go	🌐💻	76.4
10. Assembly	💻	74.1

Компиляция — перевод программы с языка среднего и высокого уровня в исполняемый файл

C/C++ - исполняется на устройстве.
Компиляция под каждый тип процессора

Java etc. - исполняется в виртуальной машине с стандартным набором команд

Сложность алгоритмов

- Constant Time: $O(1)$
- Linear Time: $O(n)$
- Logarithmic Time: $O(\log n)$
- Quadratic Time: $O(n^2)$

$$5! = 5 * 4 * 3 * 2 * 1$$

```
factorial(n):  
    if n is 0:  
        return 1  
    return n * factorial(n-1)
```

$$T(0) = 1$$

одно сравнение

$$T(n) = T(n-1) + 3$$

1 сравнение, 1 умножение,
1 вычитание + $T(n-1)$

$$T(n) = 3n + T(0)$$

Сложность — $O(n)$

Интерпретируемые языки программирования

Compiled Language

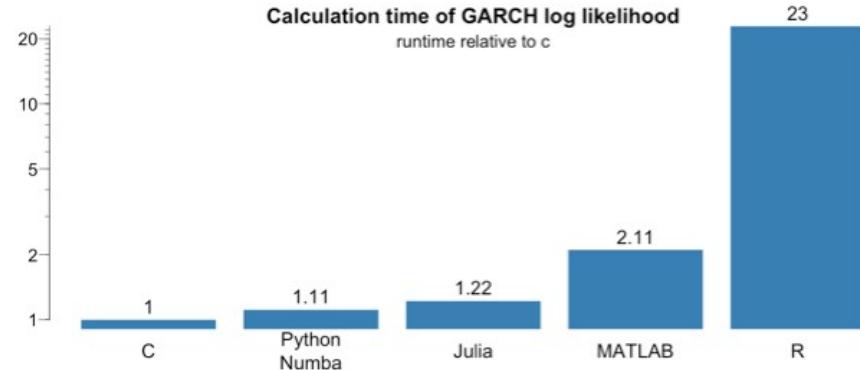
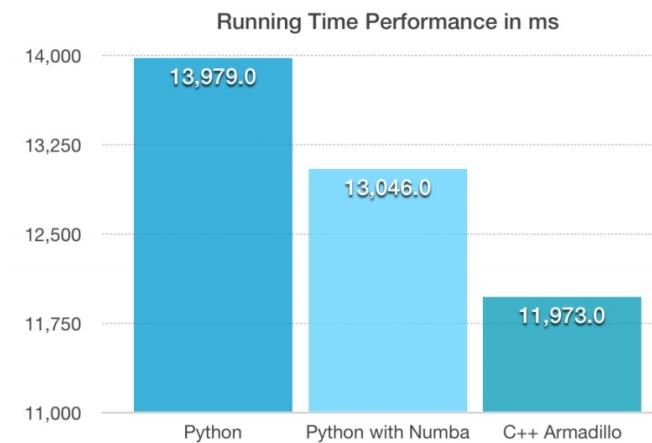
VS

Interpreted Language

Comparison Chart

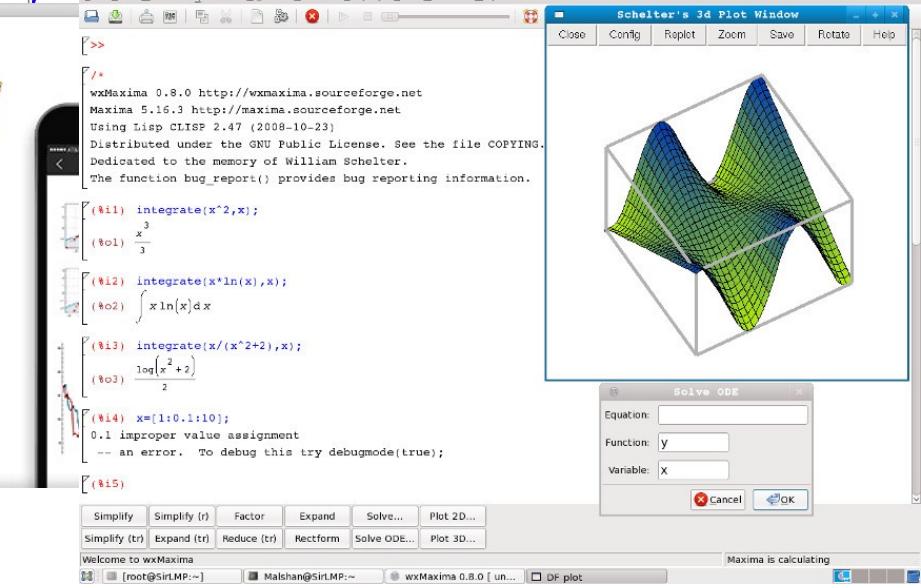
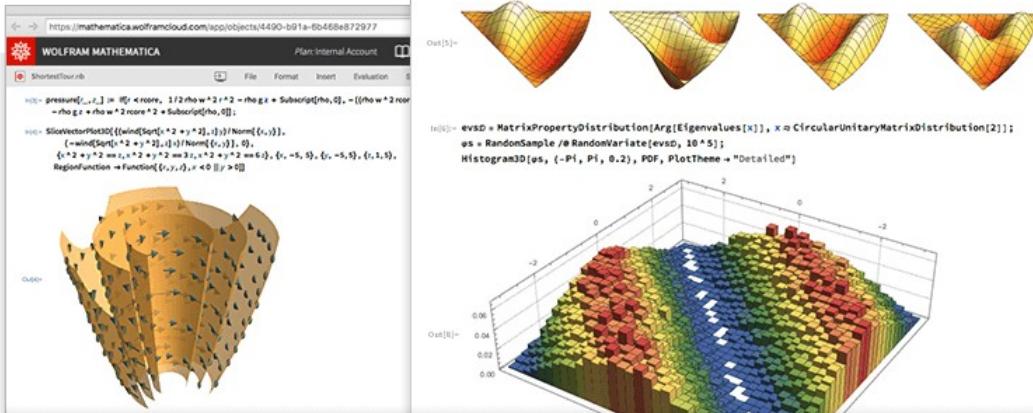
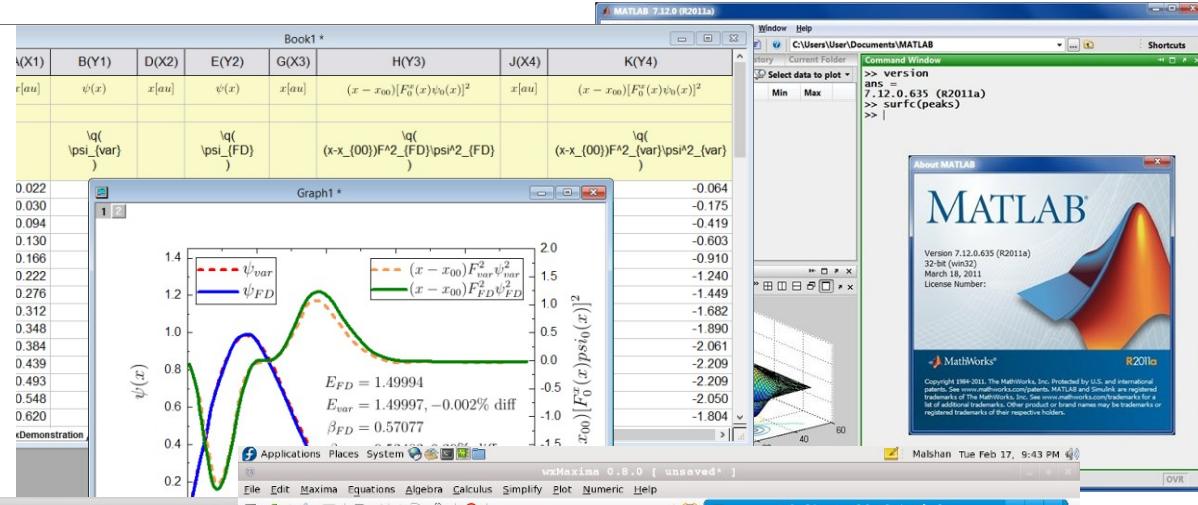
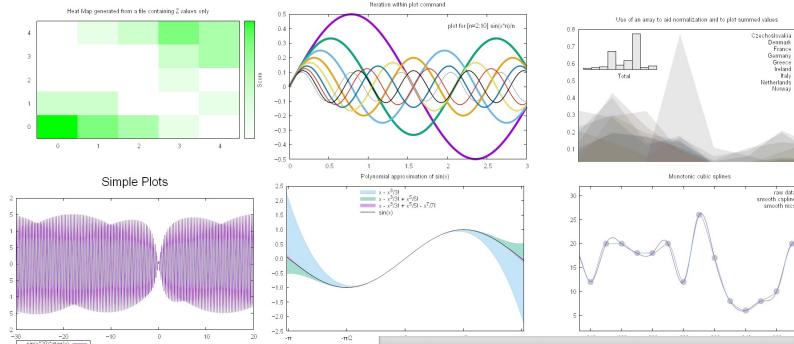
Compiled Language	Interpreted Language
The code of compiled languages can be executed directly by the computer's CPU.	A program written in an interpreted language is not compiled, it is interpreted.
The source code must be transformed into machine readable instructions prior to execution.	It does not compile the source code into machine language prior to running the program.
Compiled programs run faster than interpreted programs.	Interpreted programs can be modified while the program is running.
Delivers better performance.	Delivers relatively slower performance.
C, Fortran, and COBOL are languages used to produce compiled programs.	Java and C# are compiled into bytecode, the virtual interpreted language.

Difference
Between.net



Программные системы и встроенные языки

Гнуплот

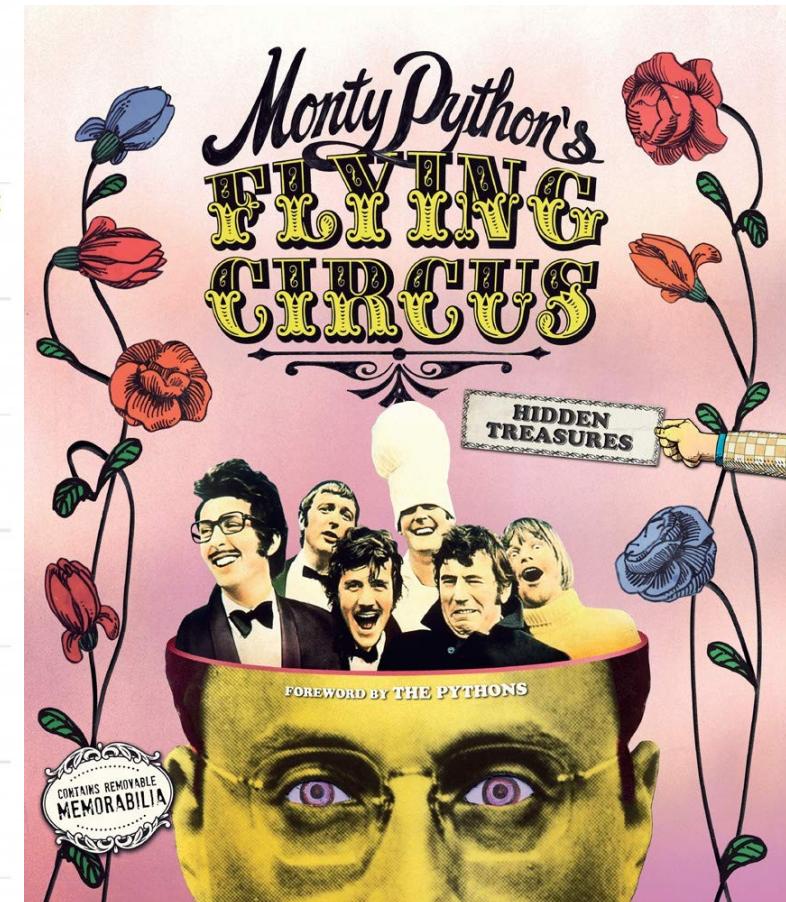
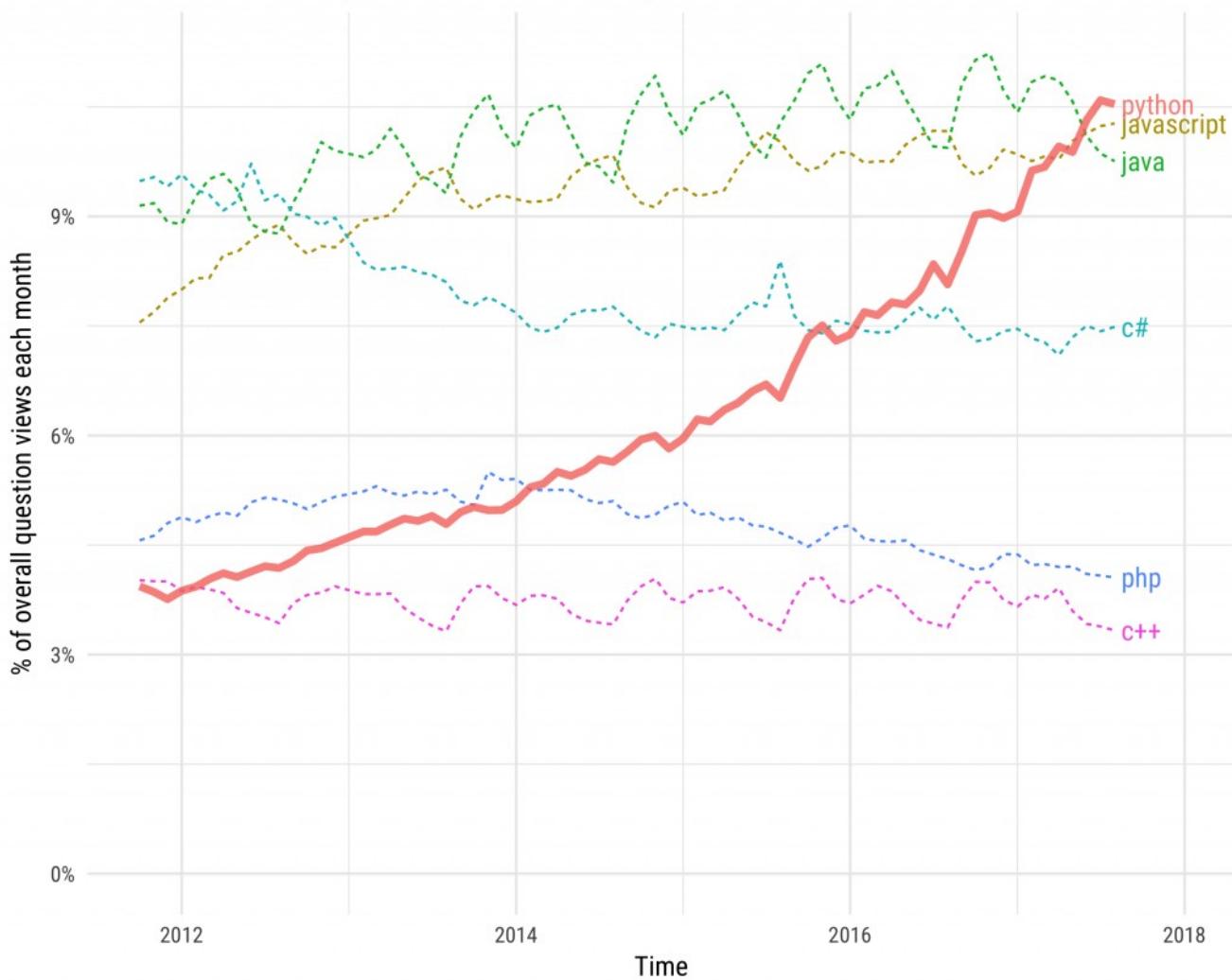


Зачем Python

Полноценный язык программирования. Расчеты, обработка, графики, любые иные задачи (docx, html, ssh, etc.)

Growth of major programming languages

Based on Stack Overflow question views in World Bank high-income countries



История плюсы и минусы Python

1991

Guido Van Rossum published the first version of Python code

2000

Python 2.0 was introduced.

2019

Python 3.7.3 was released.

1980

Python was conceptualized in the late 1980s.

1994

Python version 1.0 was released

2008

Python 3.0 was released

Zen of Python

Beautiful is better than ugly.
(Красивое лучше, чем уродливое.)

Explicit is better than implicit.
(Явное лучше, чем неявное.)

Simple is better than complex.
(Простое лучше, чем сложное.)

Complex is better than complicated.
(Сложное лучше, чем запутанное.)

Flat is better than nested.
(Плоское лучше, чем вложенное.)

Sparse is better than dense.
(Разреженное лучше, чем плотное.)

Readability counts.
(Читаемость имеет значение.)

Advantages and Disadvantages of Python



Advantages



- Easy-to-learn and use
- Improved productivity
- Interpreted language
- Open-source
- Easily portable
- Massive libraries
- Easy-to-integrate



Disadvantages

- Low speed
- Inefficient memory consumption
- Weak in mobile devices
- Difficult to access database
- Prone to cause runtime errors



Как ЭТО заставить работать

- Интерпретатор — выполняет скрипты
- Консоль — интерактивное выполнение команд (ввод — вывод)
- Рекомендуется использовать [Anaconda](#)
- Spyder, Jupyter Notebook
- Notepad++

Основные ресурсы

- yamarus.github.io/PyForSci -
задачи и информация по курсу
- python.org/doc/ - документация
python
- scipy.org/docs.html -
документация SciPy