

Урок №1

# Вводная лекция

на которой расскажут про курс, что такое система  
информационного поиска, булевый поиск.

27 сентября 2023 года

Антон Кухтичев



## Основан на Introduction to Information Retrieval

<http://nlp.stanford.edu/IR-book/>

(доступен русский перевод)

**Много слайдов тоже взяты оттуда, авторы:**

Pandu Nayak, Prabhakar Raghavan

Hinrich Schutze, Christina Lioma

До 2021 года курс в МАИ читал Андрей Калинин



# Основные тезисы



- Цель курса — разработать поисковую систему;
- Разработка — индивидуальная, без команд;
- $\leq 12$  лекций;
- $\leq 12$  лабораторных работы;
- 1 курсовая работа;
- 1 экзамен;

Ссылка на презентации/задания: <https://github.com/toshunster/MAI-IR>



- В качестве языка программирования для всех основных компонент поисковой системы может быть выбран С или С++;
- Для обвязки, выкачки, может быть выбран любой интерпретируемый язык программирования (Python, Perl, Shell, ... ) и дополнительные утилиты (curl, wget, ... )
- Оценка 3 ставится если задача выполнена для корпуса размером в 30-50 тысяч документов
- Оценка 5 ставится при количестве документов больше 1 миллиона (при условии, что они не помещаются все в оперативную память используемого компьютера)
- Кодировка файлов ввода-вывода должна быть единой для всех лабораторных работ, UTF-8.



- Кухтичев Антон Алексеевич,
- Окончил МАИ в 2012 году,
- Работаю в Mail.Ru Group с 2011 года;
- Преподаю:
  - в МАИ с 2013 года принимаю лабораторные работы по курсу «Дискретный анализ»,
  - в 2018-2020 гг. в МФТИ вёл лекции по курсу «Backend разработка» (от Mail.Ru Group),
  - в 2019-2020 гг. в МИФИ вёл лекции по курсу «Backend разработка» и «Углублённый Python» (от Mail.Ru Group),
  - с 2020 года в МГУ веду лекции по курсу «Углублённый C++» (от Mail.Ru Group) и «Углублённый Python»,
  - с 2020 года в МАИ веду лекции по курсу «Программная инженерия».

---

## Содержание занятия

1. Информационный поиск
2. Булевский поиск
3. Домашнее задание;



# Информационный поиск

# Информационный поиск



Поиск информации (обычно содержащейся в документах) бесструктурной природы (обычно, текстовой), удовлетворяющей информационным нуждам пользователя в больших массивах данных (обычно в компьютерных хранилищах).



# Примеры поисковых систем

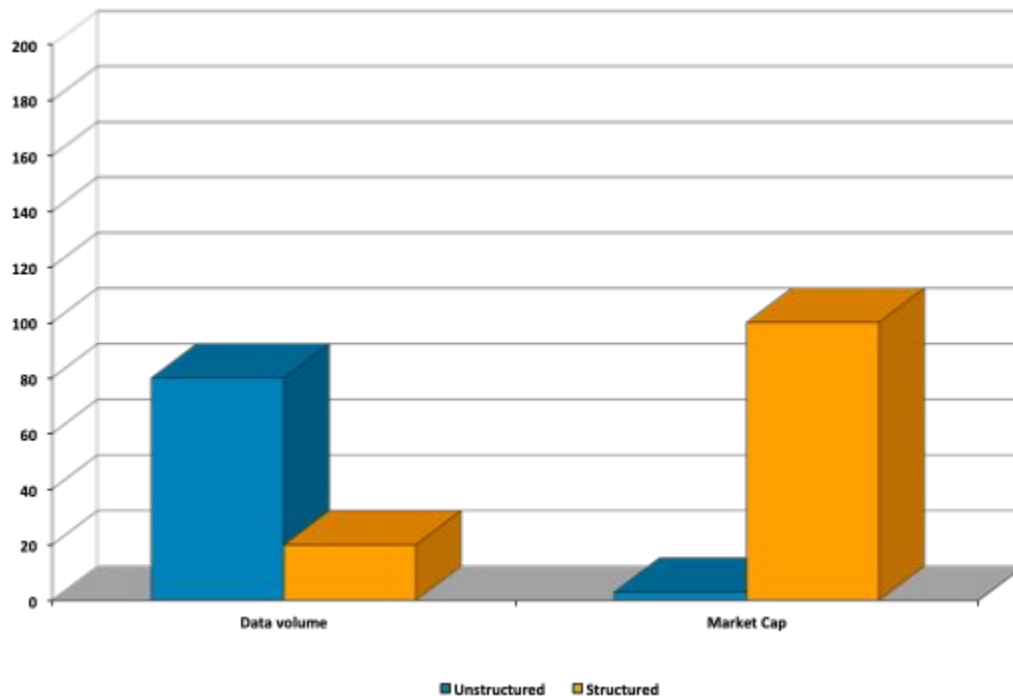


- grep,
- find,
- запросы в багтрекере:
  - поиск по патентам
  - поиск по тендерам
- новости и блоги,
- поиск по микроблогам (например, twitter или instagram),
- поиск по изображениям, по видео, по музыке,
- веб-поиск.

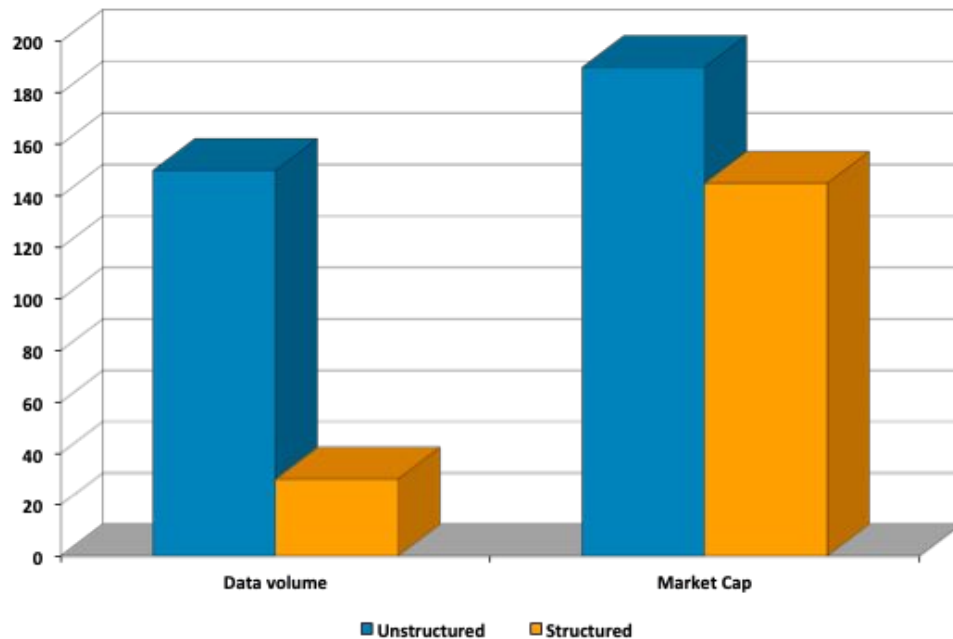


- Машинный перевод
- Извлечение мнений
- Распознавание речи
- Синтез речи
- Организация диалога с пользователем

# Текстовый поиск против SQL, 1996



# Текстовый поиск против SQL, 2006



# Информационный поиск как...



- **Наука**

- Математические модели
- Статистика, теория вероятностей
- Лингвистика

- **Практическая задача**

- Сложность в объёме
- Принцип KISS
  - Keep It Simple, Stupid
- Много нюансов и неочевидных, но простых решений

# Предположения



## Корпус:

- фиксированный набор документов

## Цель:

- найти документы, релевантные **информационным потребностям пользователя**
- помогающие ему решить свою **задачу**.

# Модель поиска

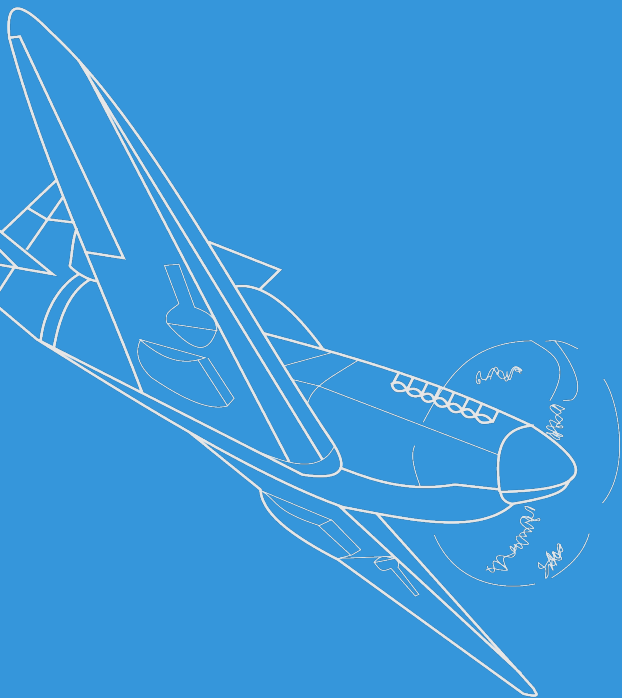


# С какими сложностями сталкиваемся



- лингвистика: что написано в тексте?
- запросы: что хотел пользователь?
- статистика;
- машинное обучение;
- большие объёмы данных;
- большие нагрузки;
- пользовательский интерфейс;
- форматы данных.





# Булевский поиск

# Принципы булевого поиска



**Запросы** – булевские выражения, предикаты

- Brutus AND Caesar AND NOT Calpurnia

Возвращаем документы, удовлетворяющие предикату



## Antony and Cleopatra, Act III, Scene ii

*Agrippa* [Aside to DOMITIUS ENOBARBUS]: Why, Enobarbus,  
When Antony found Julius *Caesar* dead,  
He cried almost to roaring; and he wept  
When at Philippi he found *Brutus* slain.

## Hamlet, Act III, Scene ii

*Lord Polonius*: I did enact Julius *Caesar* I was killed i' the  
Capitol; *Brutus* killed me.



# Google, Яндекс, Поиск Mail.ru – булевские?



**Запрос**  $[w_1 \ w_2 \ \dots \ w_n]$  интерпретируется как

$w_1 \text{ AND } w_2 \text{ AND } \dots \text{ AND } w_n$

**Но можно получить документ без  $w_i$ :**

- Ссылки;
- Вариант  $w_i$  (морфология, опечатка, синоним);
- Длинный запрос;
- Булев поиск вернёт мало документов.

**Ранжированный поиск**

# Как реализовать булев поиск?



## grep

- найти строки с ***Brutus AND Caesar***,
- удалить ***Calpurnia***.

## Почему плохо?

- медленно;
- построчно, нужно подокументно;
- сложно для NOT Calpurnia;
- плохо расширяется;
- не отменяет ранжирование.

# Матрица «термин-документ»



	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
Antony	1	1	0	0	0	1
Brutus	1	1	0	1	0	0
Caesar	1	1	0	1	1	1
Calpurnia	0	1	0	0	0	0
Cleopatra	1	0	0	0	0	0
mercy	1	0	1	1	1	1
worser	1	0	1	1	1	0

*Brutus AND Caesar BUT NOT  
Calpurnia*

В пьесе Julius Caesar  
встречается слово Calpurnia

# Битовые операции



- Для каждого термина – вектор из 0 и 1
- Извлекаем столбцы для *Brutus*, *Caesar* и инвертируем столбец для *Calpurnia*
- $110100 \text{ AND } 110111 \text{ AND } 101111 = 100100.$
- А какие есть проблемы с таким подходом к поиску?

# Построить матрицу нельзя



- $N=10^6$  документов по 1000 токенов
- 6 байтов на токен (8-9 для русского языка), включая пробелы и пунктуацию
- Размер корпуса  $6 \cdot 10^9 = 6$  ГиБ
- $M=500\,000$  разных терминов
- $500\,000 \cdot 10^6 =$  полтриллиона нулей и единиц
- Но единиц – не более миллиарда



# Кстати, о количестве терминов



- Телефонные справочники
- md5-хеши файлов
- Udaff.com
- Сайт с диссертациями

# Обратный индекс



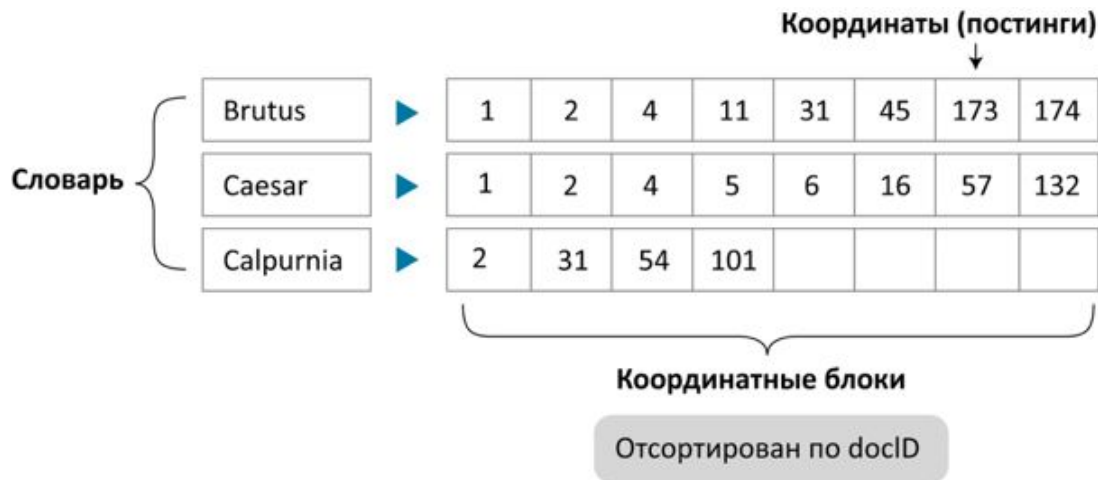
- Для каждого термина  $t$  будем хранить список документов, где он встречается
  - Каждый документ представлен docID
- Как хранить? Как изменять?

Brutus	▶	1	1	4	11	31	45	173	174
Caesar	▶	1	2	4	5	6	16	57	132
Calpurnia	▶	31	54	101					

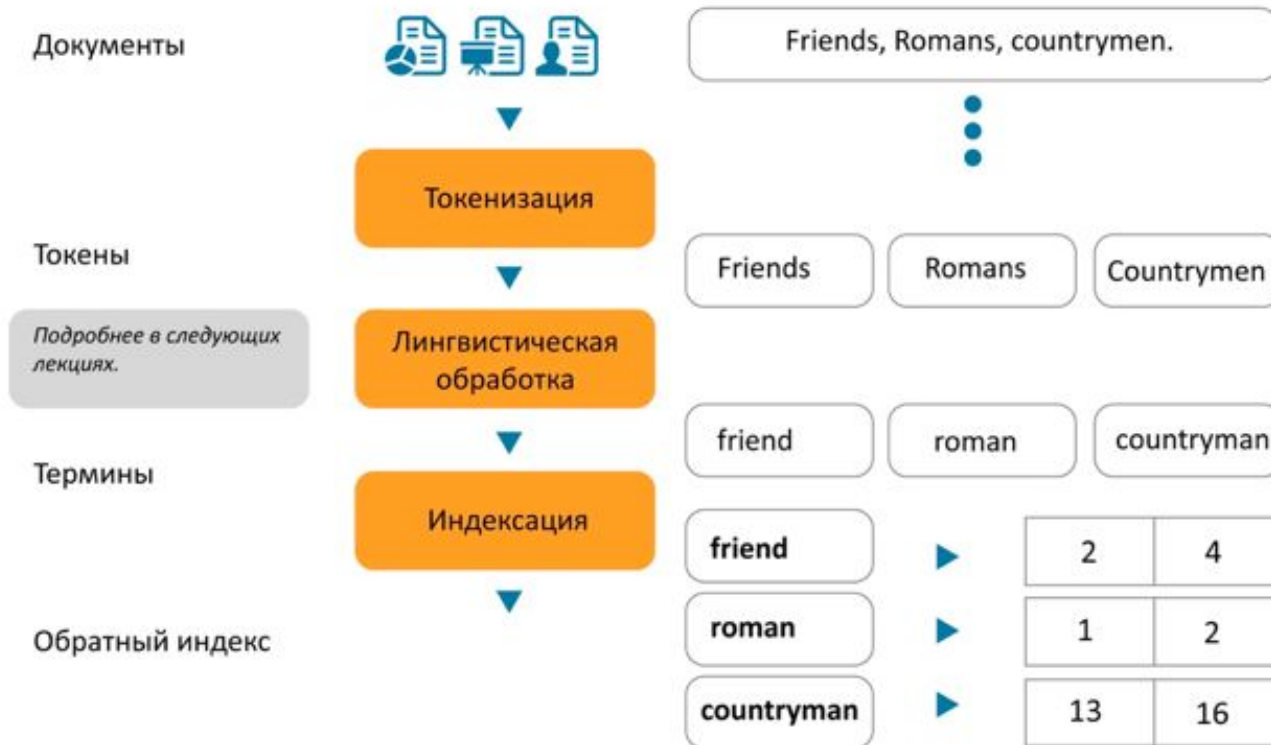
# Словарь и координатные блоки



- Массивы переменной длины
- Списки (или списки массивов)
- Всегда компромисс



# Создание обратного индекса



# Разбиение на токены



Последовательность пар (термин, DocID)

Doc 1

I did enact Julius  
Caesar I was killed  
i' the Capitol;  
Brutus killed me.

Doc 2

So let it be with  
Caesar. The noble  
Brutus hath told you  
Caesar was  
ambitious



Term	docID
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

# Сортировка



## Сортируем по терминам

Потом по docID



Основа индексации!

Term	docID	Term	docID
I	1	ambitious	2
did	1	be	2
enact	1	brutus	1
julius	1	brutus	2
caesar	1	capitol	1
I	1	caesar	1
was	1	caesar	2
killed	1	caesar	2
i'	1	did	1
the	1	enact	1
capitol	1	hath	1
brutus	1	I	1
killed	1	I	1
me	1	i'	1
so	2	it	2
let	2	julius	1
it	2	killed	1
be	2	killed	1
with	2	let	2
caesar	2	me	1
the	2	noble	2
noble	2	so	2
brutus	2	the	1
hath	2	the	2
told	2	told	2
you	2	you	2
caesar	2	was	1
was	2	was	2
ambitious	2	with	2

# Словари и координатные блоки



- Вхождения термина в один документ сливаются
- Делим на словарь и координатные блоки
- Запоминаем частотную информацию



Про необходимость  
частот позже

Term	docID	term	doc. freq.	→	postings lists
ambitious	2	ambitious	1	→	2
be	2	be	1	→	2
brutus	1	brutus	2	→	1 → 2
brutus	2	capitol	1	→	1
capitol	1	caesar	2	→	1 → 2
caesar	1	did	1	→	1
caesar	2	enact	1	→	1
caesar	2	hath	1	→	2
did	1	i	1	→	1
enact	1	i'	1	→	1
hath	1	it	1	→	2
i	1	julius	1	→	1
i	1	killed	1	→	1
i'	1	let	1	→	2
it	2	me	1	→	1
julius	1	noble	1	→	2
killed	1	so	1	→	2
killed	1	the	2	→	1 → 2
let	1	told	1	→	2
me	2	you	1	→	2
noble	1	was	2	→	1 → 2
so	2	with	1	→	2
the	1				
the	2				
told	2				
you	2				
was	1				
was	2				
with	2				

# Затраты на хранение



Термины  
и счётчики

Списки docID

Указатели

term	doc. freq.	→	postings lists
ambitious	1	→	2
be	1	→	2
brutus	2	→	1 → 2
capitol	1	→	1
caesar	2	→	1 → 2
did	1	→	1
enact	1	→	1
hath	1	→	2
i	1	→	1
i'	1	→	1
it	1	→	2
julius	1	→	1
killed	1	→	1
let	1	→	2
me	1	→	1
noble	1	→	2
so	1	→	2
the	2	→	1 → 2
told	1	→	2
you	1	→	2
was	2	→	1 → 2
with	1	→	2

Позже:

- Эффективная индексация
- Сколько нужно места

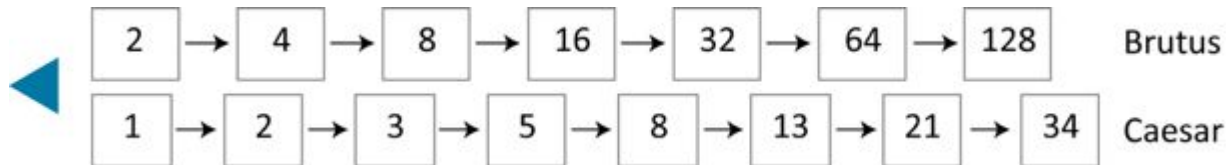


# Выполнение запросов: AND



## Запрос: *Brutus AND Caesar*

- Найдём **Brutus** в словаре;
  - Получим координатные блоки.
- Найдём **Caesar** в словаре;
  - Получим координатные блоки.
- Пересечём координатные блоки:

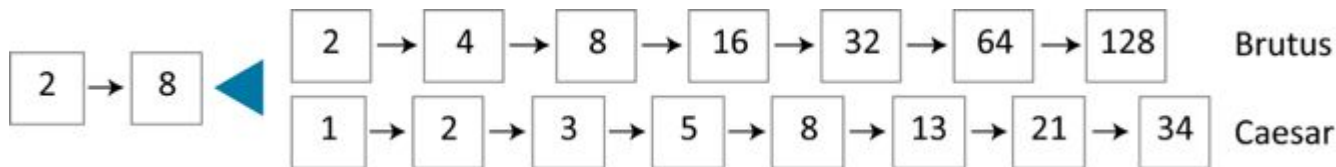


# Пересечение



Будем идти по спискам параллельно друг другу

- Если размеры списков  $m$  and  $n$ , потребуется  $O(m+n)$  действий
- **Важно:** сортировка по docID.



# Пересечение координатных блоков



```
INTERSECT( $p_1, p_2$ )
1   $answer \leftarrow \langle \rangle$ 
2  while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$ 
3  do if  $\text{docID}(p_1) = \text{docID}(p_2)$ 
4      then  $\text{ADD}(answer, \text{docID}(p_1))$ 
5           $p_1 \leftarrow \text{next}(p_1)$ 
6           $p_2 \leftarrow \text{next}(p_2)$ 
7      else if  $\text{docID}(p_1) < \text{docID}(p_2)$ 
8          then  $p_1 \leftarrow \text{next}(p_1)$ 
9          else  $p_2 \leftarrow \text{next}(p_2)$ 
10 return  $answer$ 
```

# Точное совпадение



Булев поиск прост для реализации и понимания пользователем

- Документ – множество слов
- Условие либо выполняется, либо нет

Основная модель в течение 3 десятилетий

Множество поисковых систем до сих пор булевские

# Ещё запросы – как их выполнять?



- *Brutus AND NOT Caesar*
- *Brutus OR NOT Caesar*
- *(Brutus OR Caesar) AND  
NOT (Antony OR Cleopatra)*
- Будут ли эти запросы выполняться линейно?
  - Вообще – линейно относительно чего?
- Можно представить в виде дерева!

# Оптимизация выполнения запросов



- Запрос из N операторов AND
- Brutus AND Calpurnia AND Caesar
- В каком порядке выполнять запрос?

Brutus	▶	2	4	8	16	32	64	128	
Caesar	▶	1	2	3	5	8	16	21	34
Calpurnia	▶	13	16						

# Пример оптимизации запроса



Выполняем в порядке возрастания частот:

- От самой маленькой к большим
- Запрос выполняется как ***(Calpurnia AND Brutus) AND Caesar.***

Brutus	▶	2	4	8	16	32	64	128	
Caesar	▶	1	2	3	5	8	16	21	34
Calpurnia	▶	13	16						



**Запрос:** (madding OR crowd) AND (ignoble OR strife)

- Получаем частоты для всех терминов
- Оцениваем «частоты» каждого OR суммой терминов
- Обрабатываем в порядке возрастания частот.





1. Необходимо подготовить корпус документов, который будет использован при выполнении остальных лабораторных работ;
  - а. Ознакомиться с ним, изучить его характеристики. Из чего состоит текст? Есть ли дополнительная мета-информация? Если разметка текста, какая она?
  - б. Разбить на документы.
  - с. Выделить текст.
2. Подготовить отчёт

Введение в информационный поиск  
I Маннинг Кристофер Д., Шютце  
Хайнрих

---

Рекомендуемая  
литература

Для саморазвития (опционально)  
Чтобы не набирать двумя  
пальчиками



Спасибо за  
внимание!

**Антон Кухтичев**



[a.kukhtichev@mail.ru](mailto:a.kukhtichev@mail.ru)



[@toshunster](https://t.me/toshunster)