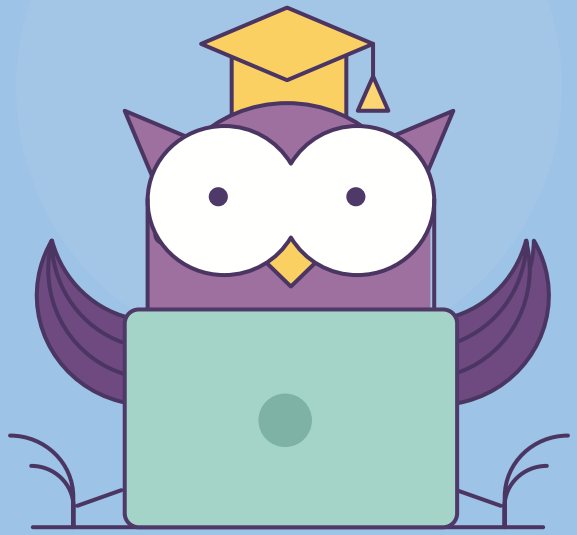
OTUS

Дальнейшее развитие. Hard skills + Soft skills

Козырь Артемий Сергеевич

Не забыть включить запись

Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте + если все хорошо
Ставьте - если есть проблемы

Правила вебинара

Активно участвуем

Задаем вопрос в чат или голосом

slack

Off-topic обсуждаем в Slack #канал группы или #general

Вопросы вижу в чате, могу ответить не сразу

План занятия

Твердые навыки

Мягкие навыки

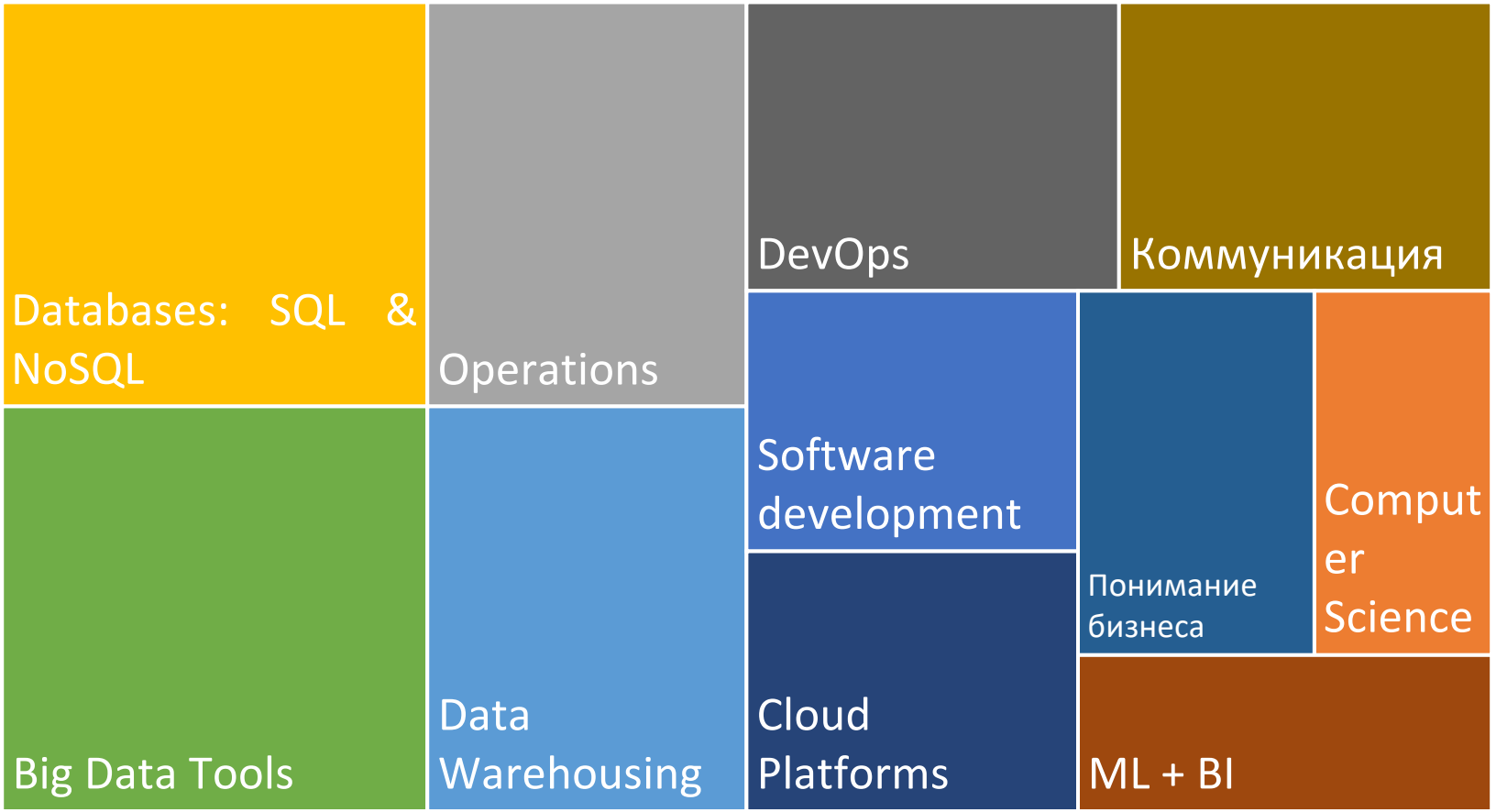
Что умеет хороший DE – Hard

- Выбрать правильные инструменты для задачи
- Fine Tuning & Benchmarking: эксперименты с настройками
- Диагностика: Know your data
- Идентифицировать узкие места / bottlenecks
- Автоматизировать всё

Развитие твердых навыков

- Развить навык с нуля
- Где и как искать ответы на вопросы
- Повысить производительность труда
- Выбор инструментов для различных задач
- Лучшие практики и опыт компаний

Карта навыков



Правило 20 часов

Правило 20 часов

- Декомпозировать навык.
  - Definition of Done. Что означает обладать навыком.
  - Что я умею / могу делать в результате.
- Практика. Концентрация. Без отвлечений.
- Заниматься как минимум 20 часов.
- Достаточный уровень чтобы исправлять себя.

Решать практические задачи

- Для себя
- Помощь кому-либо
- Pro bono

Быстрый старт – Quick start

Docs » Quick Start View page source

Quick Start

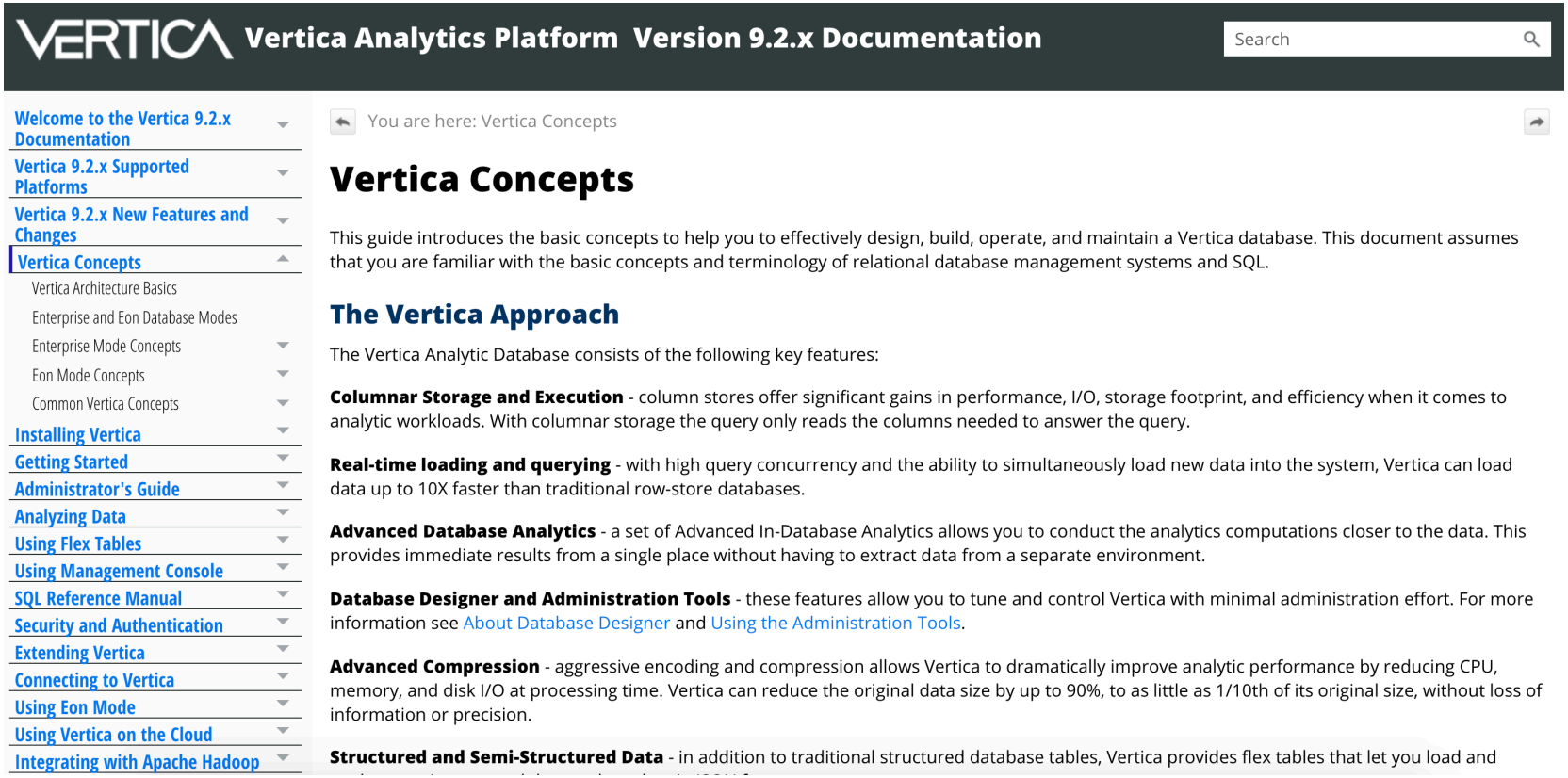
The installation is quick and straightforward.

```
# airflow needs a home, ~/airflow is the default,
# but you can lay foundation somewhere else if you prefer
# (optional)
export AIRFLOW_HOME=~/airflow

# install from pypi using pip
pip install apache-airflow

# initialize the database
airflow initdb

# start the web server, default port is 8080
airflow webserver -p 8080

# start the scheduler
airflow scheduler

# visit localhost:8080 in the browser and enable the example dag in the home page
```

Где искать ответы на вопросы?

- Официальная документация
- Сообщество
- Ментор
- Команда

Официальная документация



VERTICA Vertica Analytics Platform Version 9.2.x Documentation

Search

- Welcome to the Vertica 9.2.x Documentation
- Vertica 9.2.x Supported Platforms
- Vertica 9.2.x New Features and Changes
- Vertica Concepts
  - Vertica Architecture Basics
  - Enterprise and Eon Database Modes
  - Enterprise Mode Concepts
  - Eon Mode Concepts
  - Common Vertica Concepts
- Installing Vertica
- Getting Started
- Administrator's Guide
- Analyzing Data
- Using Flex Tables
- Using Management Console
- SQL Reference Manual
- Security and Authentication
- Extending Vertica
- Connecting to Vertica
- Using Eon Mode
- Using Vertica on the Cloud
- Integrating with Apache Hadoop

You are here: Vertica Concepts

Vertica Concepts

This guide introduces the basic concepts to help you to effectively design, build, operate, and maintain a Vertica database. This document assumes that you are familiar with the basic concepts and terminology of relational database management systems and SQL.

The Vertica Approach

The Vertica Analytic Database consists of the following key features:

Columnar Storage and Execution - column stores offer significant gains in performance, I/O, storage footprint, and efficiency when it comes to analytic workloads. With columnar storage the query only reads the columns needed to answer the query.

Real-time loading and querying - with high query concurrency and the ability to simultaneously load new data into the system, Vertica can load data up to 10X faster than traditional row-store databases.

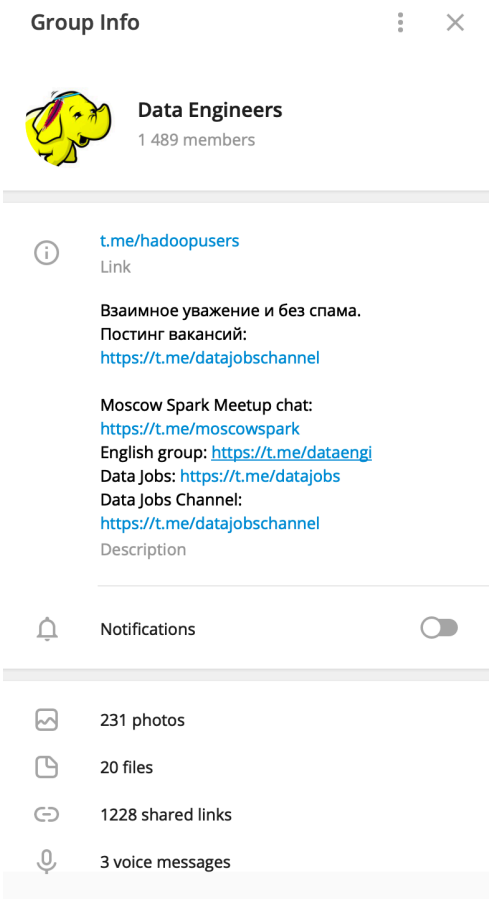
Advanced Database Analytics - a set of Advanced In-Database Analytics allows you to conduct the analytics computations closer to the data. This provides immediate results from a single place without having to extract data from a separate environment.

Database Designer and Administration Tools - these features allow you to tune and control Vertica with minimal administration effort. For more information see About Database Designer and Using the Administration Tools.

Advanced Compression - aggressive encoding and compression allows Vertica to dramatically improve analytic performance by reducing CPU, memory, and disk I/O at processing time. Vertica can reduce the original data size by up to 90%, to as little as 1/10th of its original size, without loss of information or precision.

Structured and Semi-Structured Data - in addition to traditional structured database tables, Vertica provides flex tables that let you load and

Сообщество


Group Info
Data Engineers
1 489 members
t.me/hadoopusers
Link
Взаимное уважение и без спама.
Постинг вакансий:
https://t.me/datajobschannel
Moscow Spark Meetup chat:
https://t.me/moscowspark
English group: https://t.me/dataengi
Data Jobs: https://t.me/datajobs
Data Jobs Channel:
https://t.me/datajobschannel
Description
Notifications
231 photos
20 files
1228 shared links
3 voice messages

Ментор - консультация

Message More...

Kent Graziano · 1st

Chief Technical Evangelist and Strategic Advisor at Snowflake, Data Vault Master, Author, Speaker, Trainer, Mentor

Spring, Texas · 500+ connections · Contact info

Snowflake

University of Denver

Улучшить производительность

- Hotkeys
- Cheat sheets
- Примеры кода
- Расширенный поиск

Hotkeys – быстро на раз-два

- Keyboard shortcuts for Terminal on Mac
- 20 Terminal shortcuts developers need to know
- IntelliJ IDEA Keyboard Shortcuts
- Sublime Text Keyboard Shortcuts

IntelliJ IDEA Keyboard Shortcuts

IntelliJ IDEA

DEFAULT KEYMAP

Remember these Shortcuts

Smart code completion	^⇧ Space
Search everywhere	Double ⇧
Show intention actions and quick-fixes	⌘↩
Generate code	⌘N, ^↩
Parameter info	⌘P
Extend selection	⌥⌘↑
Shrink selection	⌥⌘↓
Recent files popup	⌘E
Rename	⇧F6

General

Open corresponding tool window	⌘0 ... ⌘9
Save all	⌘S
Synchronize	⌘⌥Y
Toggle maximizing editor	⌘⇧F12
Inspect current file with current profile	⌥⇧I
Quick switch current scheme	^§, ^`
Open Settings dialog	⌘,
Open Project Structure dialog	⌘;
Find Action	⌘⇧A

Debugging

Step over / into	F8 / F7
Smart step into / Step out	⇧F7 / ⇧F8
Run to cursor	⌥F9
Evaluate expression	⌥F8
Resume program	⌘⌥R
Toggle breakpoint	⌘F8
View breakpoints	⌘⇧F8

Search / Replace

Search everywhere	Double ⇧
Find	⌘F
Find next / previous	⌘G / ⌘⇧G
Replace	⌘R
Find in path	⌘⇧F
Replace in path	⌘⇧R
Select next occurrence	^G
Select all occurrences	^⌘G
Unselect occurrence	^G

—Productivity Boosters

Editing

Basic code completion	^ Space
Smart code completion	^⇧ Space
Complete statement	⌘⇧↩
Parameter info (within method call arguments)	⌘P
Quick documentation lookup	^J,
External Doc	⇧F1
Brief Info	⌘+ mouse
Show descriptions of error at caret	⌘F1
Generate code...	⌘N, ^↩
Override methods	^O
Implement methods	^I
Surroundwith...	⌘⌥T
Comment / uncomment with line comment	⌘/
Comment / uncomment with block comment	⌘⌥/
Extend selection	⌥↑
Shrink selection	⌥↓
Context info	^⇧Q
Show intention actions and quick-fixes	⌥↩
Reformat code	⌘⌥L
Optimize imports	^⌥O
Auto-indent line(s)	^⌥I
Indent / unindent selected lines	→I / ⇧→I
Cut current line to clipboard	⌘X
Copy current line to clipboard	⌘C
Paste from clipboard	⌘V
Paste from recent buffers...	⌘⇧V
Duplicate current line	⌘D
Delete line at caret	⌘⌫
Smart line join	^⇧J
Smart line split	⌘↩
Start new line	⇧↩
Toggle case for word at caret or selected block	⌘⇧U
Select till code block end / start	⌘⇧] / ⌘⇧[
Delete to word end	⌥⌦
Delete to word start	⌥⌫
Expand / collapse code block	⌘+ / ⌘-
Expand all	⌘⇧+
Collapse all	⌘⇧-
Close active editor tab	⌘W

Refactoring

Copy	F5
Move	F6
Safe Delete	⌘Delete
Rename	⇧F6
Refactor this	^T
Change Signature	⌘F6
Inline	⌘⌥N
Extract Method	⌘⌥M
Extract Variable	⌘⌥V
Extract Field	⌘⌥F
Extract Constant	⌘⌥C
Extract Parameter	⌘⌥P

Navigation

Go to class	⌘O
Go to file	⌘⇧O
Go to symbol	⌘⌥O
Go to next / previous editor tab	^←/^→
Go back to previous tool window	F12
Go to editor (from tool window)	⎋
Hide active or last active window	⇧⎋
Go to line	⌘L
Recent files popup	⌘E
Navigate back / forward	⌘⌥←/⌘⌥→
Navigate to last edit location	⌘⇧⌫
Select current file or symbol in any view	⌥F1
Go to declaration	⌘B, ⌘Click
Go to implementation(s)	⌘⌥B
Open quick definition lookup	⌥Space, ⌘Y
Go to type declaration	^⇧B
Go to super-method / super-class	⌘U
Go to previous / next method	^↑/^↓
Move to code block end / start	⌘] / ⌘[
File structure popup	⌘F12
Type hierarchy	^H
Method hierarchy	⌘⇧H
Call hierarchy	^⌥H
Next / previous highlighted error	F2 / ⇧F2
Edit source / View source	F4 / ⌘↓
Show navigation bar	⌥Home
Toggle bookmark	F3
Toggle bookmark with mnemonic	⌥F3
Go to numbered bookmark	^0 ... ^9
Show bookmarks	⌘F3

Compile and Run

Make project	⌘F9
Compile selected file, package or module	⌘⇧F9
Select configuration and run / debug	^⌥R / D
Run / Debug	^R / D
Run context configuration from editor	^⇧R, ^⇧D

Usage Search

Find usages / Find usages in file	⌥F7 / ⌘F7
Highlight usages in file	⌘⇧F7
Show usages	⌘⌥F7

VCS / Local History

Commit project to VCS	⌘K
Update project from VCS	⌘T
Push commits	⌘⇧K
'VCS' quick popup	^V

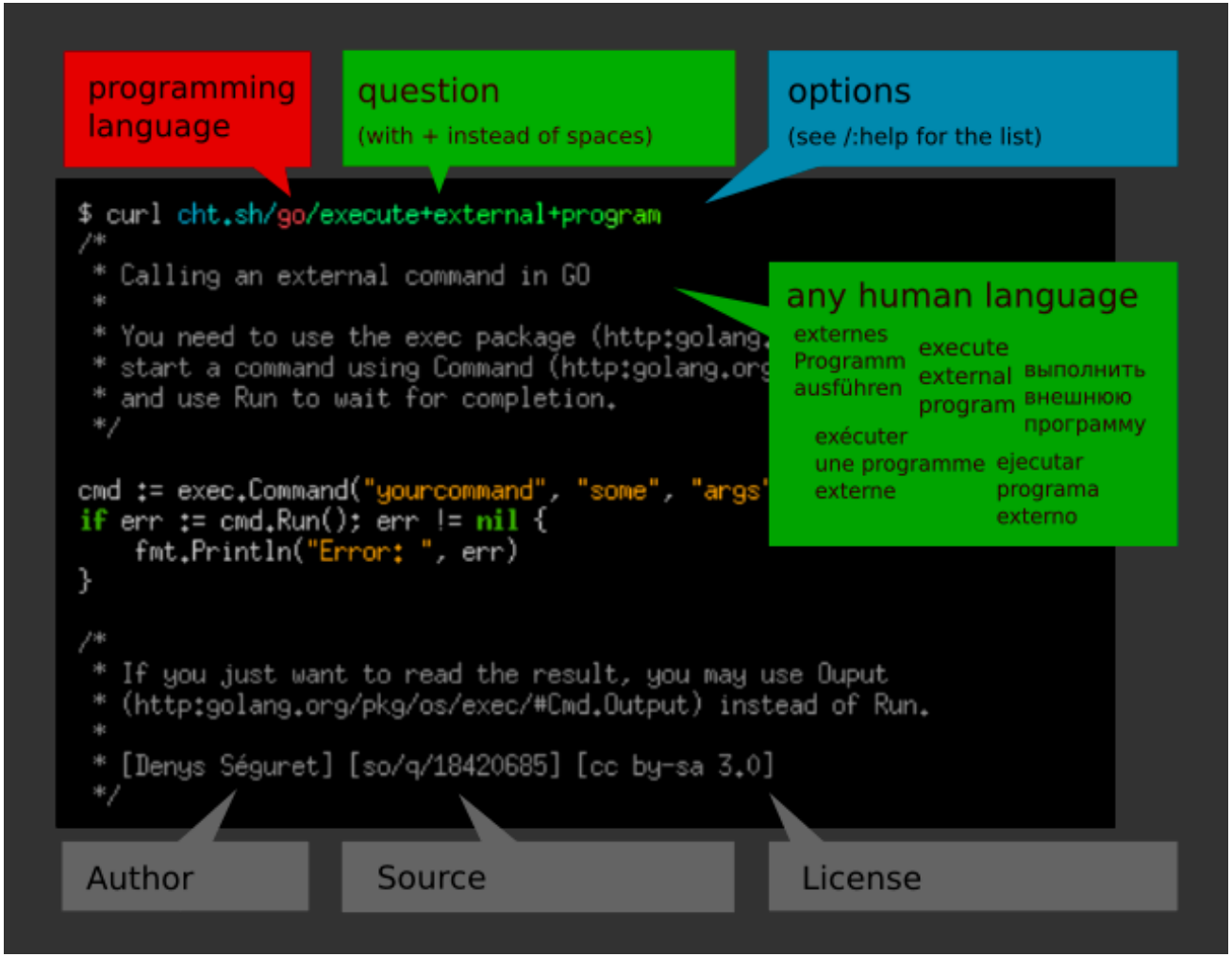
Live Templates

Surround with Live Template	⌘⌥J
Insert Live Template	⌘J

Cheat Sheets

- The only cheat sheet you need cheat.sh
- GitHub Cheat Sheet + Git Cheat Sheet
- PySpark Cheat Sheet
- Docker Cheat Sheet
- Hive cheat sheet

cheat.sh


programming language
question
(with + instead of spaces)
options
(see /:help for the list)
$ curl cht.sh/go/execute+external+program
/*
* Calling an external command in GO
*
* You need to use the exec package (http:golang.
* start a command using Command (http:golang.org
* and use Run to wait for completion.
*/
cmd := exec.Command("yourcommand", "some", "args"
if err := cmd.Run(); err != nil {
fmt.Println("Error: ", err)
}
/*
* If you just want to read the result, you may use Ouput
* (http:golang.org/pkg/os/exec/#Cmd.Output) instead of Run.
*
* [Denys Séguret] [so/q/18420685] [cc by-sa 3.0]
*/
any human language
externes Programm ausführen
execute external program
выполнить внешнюю программу
exécuter une programme externe
ejecutar programa externo
Author
Source
License

PySpark Cheat Sheet

Python For Data Science *Cheat Sheet*

PySpark - SQL Basics

Learn Python for data science Interactively at www.DataCamp.com

PySpark & Spark SQL

Spark SQL is Apache Spark's module for working with structured data.

Initializing SparkSession

A SparkSession can be used create DataFrame, register DataFrame as tables, execute SQL over tables, cache tables, and read parquet files.

```
>>> from pyspark.sql import SparkSession
>>> spark = SparkSession \
        .builder \
        .appName("Python Spark SQL basic example") \
        .config("spark.some.config.option", "some-value") \
        .getOrCreate()
```

Creating DataFrames

From RDDs

```
>>> from pyspark.sql.types import *
```

Infer Schema

```
>>> sc = spark.sparkContext
>>> lines = sc.textFile("people.txt")
>>> parts = lines.map(lambda l: l.split(","))
>>> people = parts.map(lambda p: Row(name=p[0],age=int(p[1])))
>>> peopledf = spark.createDataFrame(people)
```

Specify Schema

```
>>> people = parts.map(lambda p: Row(name=p[0],
                                     age=int(p[1].strip())))
>>> schemaString = "name age"
>>> fields = [StructField(field_name, StringType(), True) for
field_name in schemaString.split()]
>>> schema = StructType(fields)
>>> spark.createDataFrame(people, schema).show()
  +--------+---+
  |    name|age|
  +--------+---+
  |    Mine| 28|
  |   Filip| 29|
  |Jonathan| 30|
  +--------+---+
```

From Spark Data Sources

JSON

```
>>> df = spark.read.json("customer.json")
>>> df.show()
+--------------------+---+---------+--------+--------------------+
|             address|age|firstName |lastName|         phoneNumber|
+--------------------+---+---------+--------+--------------------+
|[New York,10021,N...| 25|     John|   Smith|[[212 555-1234,ho...|
|[New York,10021,N...| 21|     Jane|     Doe|[[322 888-1234,ho...|
```

Duplicate Values

```
>>> df = df.dropDuplicates()
```

Queries

```
>>> from pyspark.sql import functions as F
```

Select

```
>>> df.select("firstName").show()
>>> df.select("firstName","lastName") \
      .show()
>>> df.select("firstName",
              "age",
              explode("phoneNumber") \
              .alias("contactInfo")) \
      .select("contactInfo.type",
              "firstName",
              "age") \
      .show()
```

Show all entries in `firstName` column

Show all entries in `firstName`, `age` and `type`

```
>>> df.select(df["firstName"],df["age"]+ 1)
      .show()
>>> df.select(df['age'] > 24).show()
```

Show all entries in `firstName` and `age`, add `1` to the entries of `age`
Show all entries where `age` >24

When

```
>>> df.select("firstName",
              F.when(df.age > 30, 1) \
              .otherwise(0)) \
      .show()
>>> df[df.firstName.isin("Jane","Boris")]
                .collect()
```

Show `firstName` and 0 or 1 depending on `age` >30

Show `firstName` if in the given options

Like

```
>>> df.select("firstName",
              df.lastName.like("Smith")) \
      .show()
```

Show `firstName`, and `lastName` is `TRUE` if `lastName` is like `Smith`

Startswith - Endswith

```
>>> df.select("firstName",
              df.lastName \
                .startswith("Sm")) \
      .show()
>>> df.select(df.lastName.endswith("th")) \
      .show()
```

Show `firstName`, and `TRUE` if `lastName` starts with `Sm`

Show last names ending in `th`

Substring

```
>>> df.select(df.firstName.substr(1, 3) \
                       .alias("name")) \
      .collect()
```

Return substrings of `firstName`

Between

```
>>> df.select(df.age.between(22, 24)) \
      .show()
```

Show `age`: values are `TRUE` if between `22` and `24`

Add, Update & Remove Columns

Adding Columns

```
>>> df = df.withColumn('city',df.address.city) \
           .withColumn('postalCode',df.address.postalCode) \
           .withColumn('state',df.address.state) \
           .withColumn('streetAddress',df.address.streetAddress) \
           .withColumn('telePhoneNumber',
                       explode(df.phoneNumber.number)) \
           .withColumn('telePhoneType',
                       explode(df.phoneNumber.type))
```

Updating Columns

GroupBy

```
>>> df.groupBy("age")\
      .count() \
      .show()
```

Group by `age`, count the members in the groups

Filter

```
>>> df.filter(df["age"]>24).show()
```

Filter entries of `age`, only keep those records of which the values are `>24`

Sort

```
>>> peopledf.sort(peopledf.age.desc()).collect()
>>> df.sort("age", ascending=False).collect()
>>> df.orderBy(["age","city"],ascending=[0,1])\
      .collect()
```

Missing & Replacing Values

```
>>> df.na.fill(50).show()
>>> df.na.drop().show()
>>> df.na \
      .replace(10, 20) \
      .show()
```

Replace null values
Return new `df` omitting rows with null values
Return new `df` replacing one value with another

Repartitioning

```
>>> df.repartition(10)\
      .rdd \
      .getNumPartitions()
>>> df.coalesce(1).rdd.getNumPartitions()
```

`df` with 10 partitions

`df` with 1 partition

Running SQL Queries Programmatically

Registering DataFrames as Views

```
>>> peopledf.createGlobalTempView("people")
>>> df.createTempView("customer")
>>> df.createOrReplaceTempView("customer")
```

Query Views

```
>>> df5 = spark.sql("SELECT * FROM customer").show()
>>> peopledf2 = spark.sql("SELECT * FROM global_temp.people")\
                 .show()
```

Output

Data Structures

```
>>> rdd1 = df.rdd
>>> df.toJSON().first()
>>> df.toPandas()
```

Convert `df` into an RDD
Convert `df` into a RDD of string
Return the contents of `df` as Pandas `DataFrame`

Примеры кода

- Cookbooks (O’Reilly)
- Github + расширенный поиск
- ProgramCreek

Cookbooks – oreilly.com

Python Cookbook, 3rd Edition

★★★★★ 47 REVIEWS

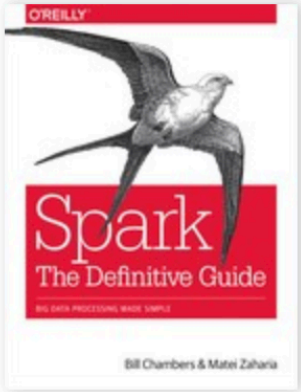
by David Beazley, Brian K. Jones

Publisher: O'Reilly Media, Inc.

Release Date: May 2013

ISBN: 9781449340377

Topic: Python



Spark: The Definitive Guide

★★★★½ 31 REVIEWS

by Matei Zaharia, Bill Chambers

Publisher: O'Reilly Media, Inc.

Release Date: February 2018

ISBN: 9781491912218

Topic: Spark

GitHub search code

- About searching on GitHub
- Searching code
- Understanding the search syntax

Search by the file contents or file path

With the `in` qualifier you can restrict your search to the contents of the source code file, the file path, or both. When you omit this qualifier, only the file contents are searched.

Qualifier	Example
`in:file`	**octocat in:file** matches code where "octocat" appears in the file contents.
`in:path`	**octocat in:path** matches code where "octocat" appears in the file path.
	octocat in:file,path matches code where "octocat" appears in the file contents or the file path.

ProgramCreek - Spark

Class Names that contain "spark".

1. org.apache.spark.SparkConf	Used in 998 projects.
2. org.apache.spark.SparkContext	Used in 975 projects.
3. org.apache.spark.rdd.RDD	Used in 720 projects.
4. org.apache.spark.sql.SQLContext	Used in 453 projects.
5. org.apache.spark.sql.DataFrame	Used in 443 projects.
6. org.apache.spark.sql.Row	Used in 419 projects.
7. org.apache.spark.sql.SparkSession	Used in 398 projects.
8. org.apache.spark.streaming.StreamingContext	Used in 308 projects.
9. org.apache.spark.storage.StorageLevel	Used in 304 projects.
10. org.apache.spark.sql.types.StructType	Used in 301 projects.

org.apache.spark.SparkConf

org.apache.spark.SparkConf Scala Examples

The following code examples show how to use org.apache.spark.SparkConf. These examples are extracted from open source projects. You can vote up the examples you like and your votes will be used in our system to product more good examples.

+ Save this class

Example 1

Project: *SparkApps* Author: *malli3131* File: *Histogram.scala* *View Source Project* (license) 14 votes

```
import org.apache.spark.SparkConf
import org.apache.spark.SparkContext
import org.apache.spark.rdd.RDD

object Histogram{
        def main(args:Array[String]){
                val conf:SparkConf = new SparkConf().setAppName("Histogram").setMaster("local")
                val sc:SparkContext = new SparkContext(conf)
                val dataset1:RDD[String] =
 sc.textFile("/home/hadoop/spark/scala/mllib/core/data1")
                val dataset2:RDD[String] =
 sc.textFile("/home/hadoop/spark/scala/mllib/core/data2");
                val subRDD:RDD[String] = dataset1.subtract(dataset2)
                val keyValueRDD:RDD[(String, String)] = subRDD.map(line => (line.split(",")(1),
 line.split(",")(0)))
                val hist = keyValueRDD.countByKey
                for((k,v) <- hist){
                        println(k + "===>" + v)
                }
        }
}
```

Python logging.getLogger()

Python logging.getLogger() Examples

The following are code examples for showing how to use *logging.getLogger()*. They are extracted from open source Python projects. You can vote up the examples you like or vote down the ones you don't like. You can also save this page to your account.

+ Save to library

Example 1

Project: *kas* Author: *siemens* File: *kas.py* (license) *View Source Project* 17 votes

```
def create_logger():
    """
        Setup the logging environment
    """
    log = logging.getLogger()  # root logger
    log.setLevel(logging.INFO)
    format_str = '%(asctime)s - %(levelname)-8s - %(message)s'
    date_format = '%Y-%m-%d %H:%M:%S'
    if HAVE_COLORLOG and os.isatty(2):
        cformat = '%(log_color)s' + format_str
        colors = {'DEBUG': 'reset',
                  'INFO': 'reset',
                  'WARNING': 'bold_yellow',
                  'ERROR': 'bold_red',
                  'CRITICAL': 'bold_red'}
        formatter = colorlog.ColoredFormatter(cformat, date_format,
                                              log_colors=colors)
    else:
        formatter = logging.Formatter(format_str, date_format)
    stream_handler = logging.StreamHandler()
    stream_handler.setFormatter(formatter)
    log.addHandler(stream_handler)
    return logging.getLogger(__name__)
```

Выбирать инструменты и решения

- Использовать уже имеющийся опыт
- Кто-то уже мог решать эту задачу
- Правильно подбирать инструменты
- Взаимодействие между подсистемами

Опыт компаний и доклады

- HighLoad Channel
- DataWorks Summit
- Databricks
- Spark Summit

План занятия

Твердые навыки

Мягкие навыки

Что умеет хороший DE – Soft

- Презентовать себя, свои навыки и опыт
- Работать в команде. Быть частью команды
- Получать и давать обратную связь
- Задавать вопросы и получать ответы
- Понять что от него требуется

Развитие мягких навыков

- Резюме (CV)
- Профиль в LinkedIn
- Собеседования и интервью
- Карьерное развитие
- Эмоциональный интеллект
- Делиться опытом и достижениями
- Участие в конференциях и сообществах

Резюме - CV

- 1 страница (!)
- Результаты и достижения, а не обязанности
- Релевантные знания и навыки
- Участие в конференциях, проекты, статьи
- Орфография и форматирование
- .pdf под печать

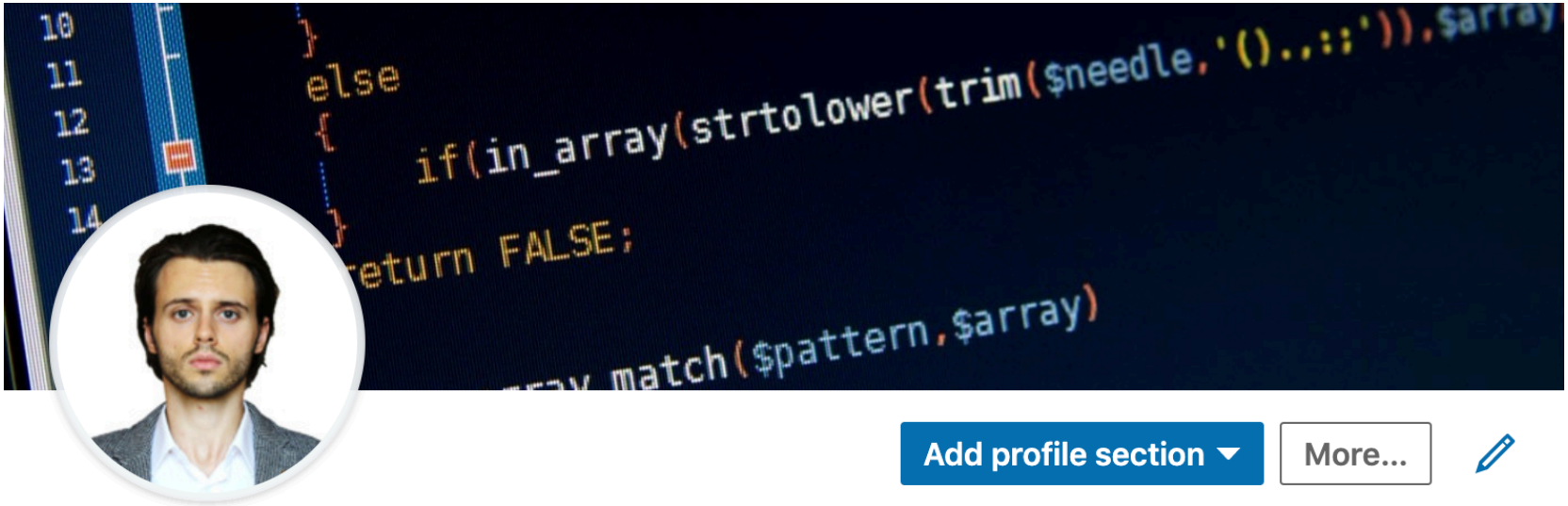
Резюме – содержание

- Контакты
- Опыт
- Навыки / Умения
- Образование
- * Проекты, курсы
- * Саммари

Резюме – шаблоны

- Software Engineering resume samples
- cvmkr.com
- cv-template.com

Личная страничка в LinkedIn



Add profile section ▾ More...

Artemiy Kozyr

Data Engineer DWH / Big Data / BI

Moscow, Russian Federation · 500+ connections · Contact info

SIBUR

Higher School of Economics

Личная страничка в LinkedIn

- Социальная сеть для профессионалов
- Networking & Personal branding
- Хантинг, свежие вакансии, новости компаний
- Возможность общаться и задавать вопросы напрямую кому угодно

LinkedIn – Цели?

- Выдача в поиске по ключевым словам
  - Используем Keywords
- Выделяться в результатах поиска
  - Заполняем Headline, Summary, Skills
- Сигнализировать о профессионализме и мотивации
  - Опыт, результаты, проекты, рекомендации, группы

Собеседования - интервью

- Готовьтесь. Изучите компанию, проекты, коллег.
- Прогнозируйте вопросы. Имейте ответ.
- Тренируйтесь. Разыгрывайте по ролям.
- Будьте готовы ответить за каждое предложение в резюме.
- Техническая часть интервью. Задание.

Карьерное развитие

- Моя роль, ответственность, ожидания сторон
- Попадать в KPI
- Стратегия Quick Win
- Правило консалтинга:
  - *Help others get what they want, and you'll get what you want*
  - Translation: Do Your Job + 1/2 the Job of the Person Above

3 ключевые вещи в развитии

- Помогать коллегам получать желаемое
  - Команда, клиенты, менеджмент
- Делать то что должно
  - А не только то, о чем просят
- Быть востребованным
  - Качественный продукт / сервис
  - Спрос на услуги от клиентов

Эмоциональный интеллект

Эмоциональный интеллект

- Дэниел Гоулман: Эмоциональный интеллект. Почему он может значить больше, чем IQ
- ЭМОЦИОНАЛЬНЫЙ ИНТЕЛЛЕКТ и Как его прокачать

Делиться опытом и достижениями

- Мой вклад в развитие сообщества
- Структурировать проделанную работу для себя
- Находить идеи для улучшения, развития
- Формат: статья, репозиторий, доклад, вебинар

[Teradata] Boosting Retail Banking operations with improved JOIN performance and QUALIFY clause

[TERADATA] Unneccessary IO elimination – View Optimization Showcase

Делиться опытом - Платформы

- LinkedIn
- Github
- Medium
- Личный блог

Участие в конференциях

- Взглянуть на задачи под другим углом
- Знания, опыт, идеи, мысли
- Новые знакомства, единомышленники
- Тренды в технологиях, развитие
- Оценка своего положения относительно рынка
- *Выступить со своим кейсом (contribution)*

Ближайшие конференции

- Big Data Conference (13 сентября)
- BIG DATA & AI CONFERENCE (19-20 сентября)
- Big Data Days 2019 (8-10 октября)
- Highload++ (7-8 ноября)
- Devternity (6-7 декабря)
- Куча митапов
- …

Домашнее задание*

1. CV:
   - Подготовить CV
   - Рецензировать CV
2. LinkedIn:
   - Заполнить профиль
   - Добавить коллег
   - Рекомендация, skills endorsement
3. Интервью:
   - Подготовить вопросы и ответы
   - Разбиться на пары и практиковаться

Рефлексия

- Что вам запомнилось больше всего?
- Пройти опрос

Ваши вопросы?