Text Analysis for Social Sciences in Python
(MSc Economics)

WINTER SEMESTER 21/22

HUYEN NGUYEN

HTTP://HUYENTTNGUYEN.COM

Who are you & Why are you here?

Name tag in every session – write in large, bolded letters please ☺

Slack Introduction time!

[SLACK] Group matching, code bug, research project discussions:

https://join.slack.com/t/unihamburgworld/shared_invite/zt-wkqywq13-3OQMM~S~gShngJ~VcTP2~Q

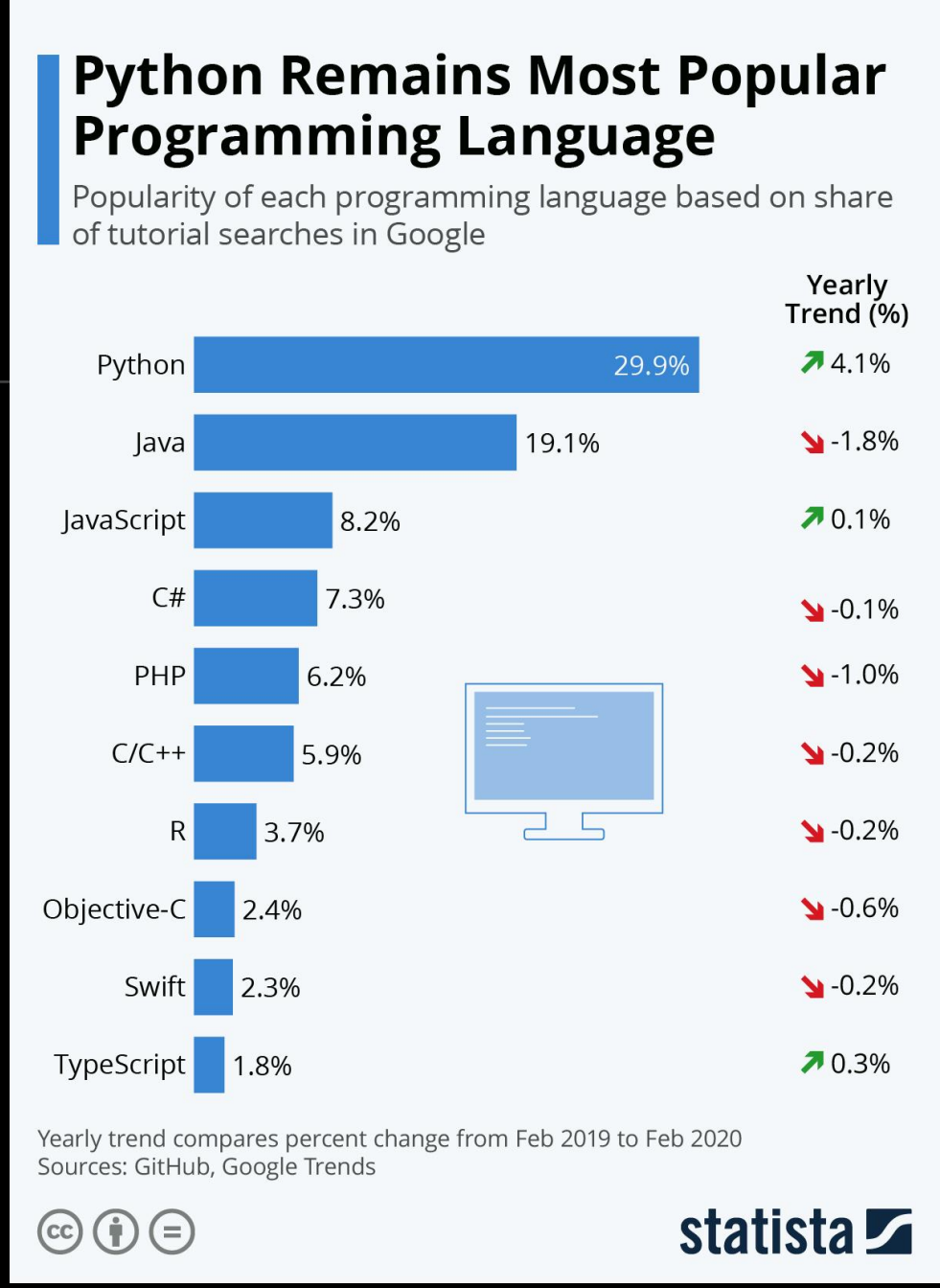
=> Join and introduce yourself, if you haven't done so☺

“The ability to take data - to be able to understand it, to process it, to extract value from it, to visualize it, to communicate it is going to be a hugely important skill in the next decades, not only at the professional level but even at the educational level for elementary school kids, for high school kids, for college kids. Because now we really do have essentially free and ubiquitous data. So the complimentary scarce factor is the ability to understand that data and extract value from it.”

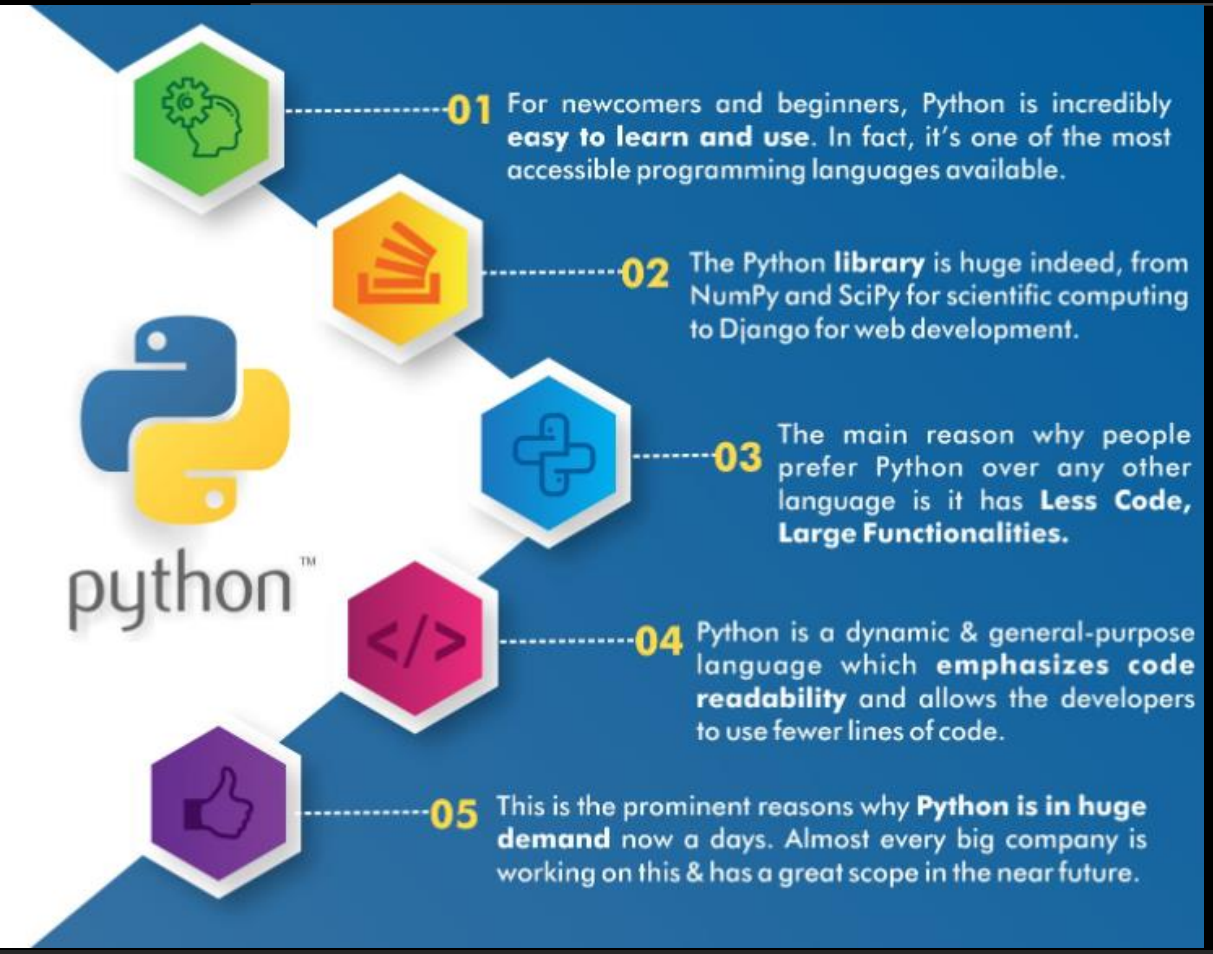
Hal Varian (chief economist, Google)

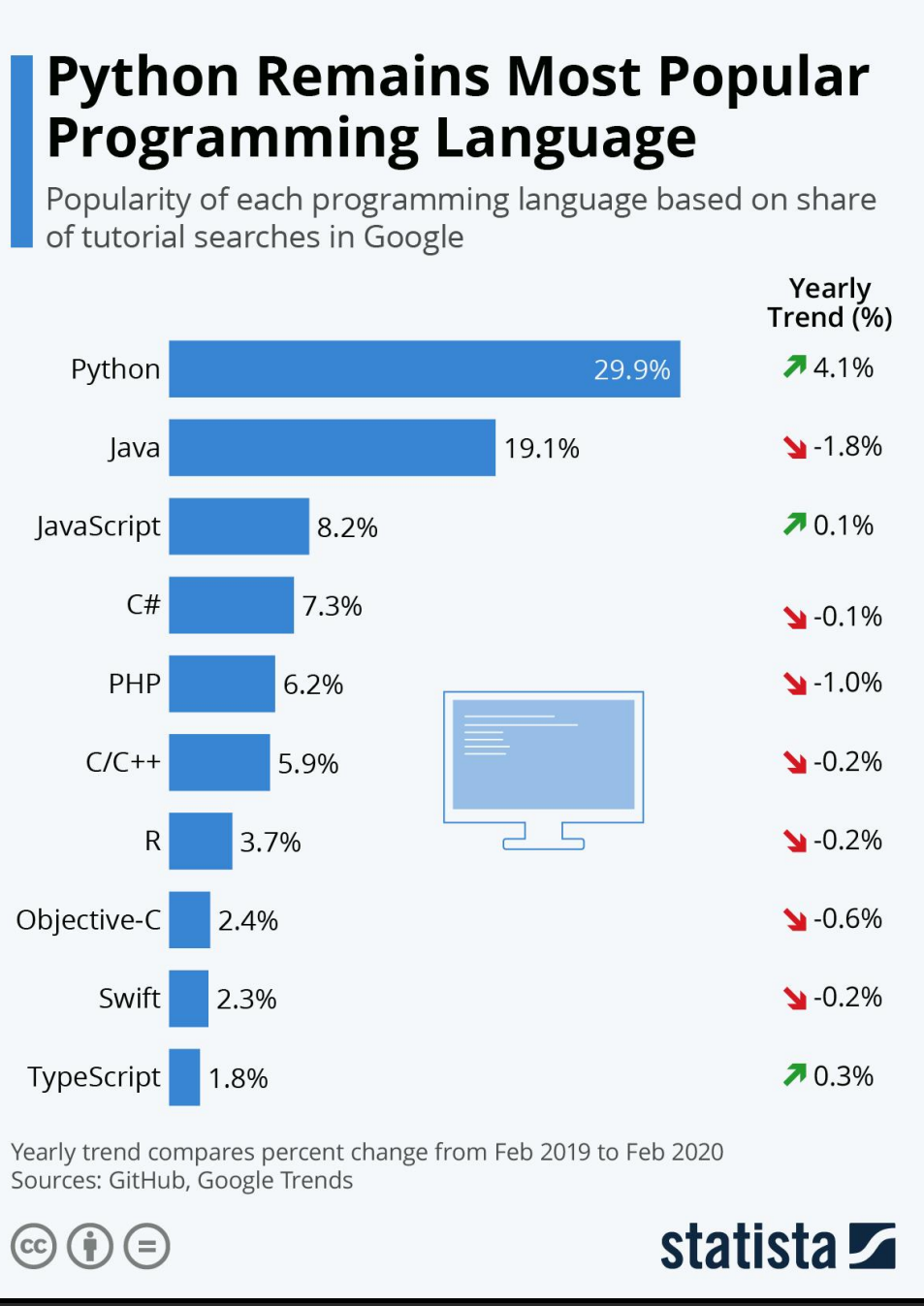
Why Python?

Why Python?


Python Remains Most Popular Programming Language
Popularity of each programming language based on share of tutorial searches in Google
Yearly Trend (%)
Python 29.9% 4.1%
Java 19.1% -1.8%
JavaScript 8.2% 0.1%
C# 7.3% -0.1%
PHP 6.2% -1.0%
C/C++ 5.9% -0.2%
R 3.7% -0.2%
Objective-C 2.4% -0.6%
Swift 2.3% -0.2%
TypeScript 1.8% 0.3%
Yearly trend compares percent change from Feb 2019 to Feb 2020
Sources: GitHub, Google Trends
statista

Why Python?


python™
01 For newcomers and beginners, Python is incredibly **easy to learn and use**. In fact, it's one of the most accessible programming languages available.
02 The Python **library** is huge indeed, from NumPy and SciPy for scientific computing to Django for web development.
03 The main reason why people prefer Python over any other language is it has **Less Code, Large Functionalities.**
04 Python is a dynamic & general-purpose language which **emphasizes code readability** and allows the developers to use fewer lines of code.
05 This is the prominent reasons why **Python is in huge demand** now a days. Almost every big company is working on this & has a great scope in the near future.


Python Remains Most Popular Programming Language
Popularity of each programming language based on share of tutorial searches in Google
Yearly Trend (%)
Python 29.9% 4.1%
Java 19.1% -1.8%
JavaScript 8.2% 0.1%
C# 7.3% -0.1%
PHP 6.2% -1.0%
C/C++ 5.9% -0.2%
R 3.7% -0.2%
Objective-C 2.4% -0.6%
Swift 2.3% -0.2%
TypeScript 1.8% 0.3%
Yearly trend compares percent change from Feb 2019 to Feb 2020
Sources: GitHub, Google Trends
statista

Why


order
data
algorithms
user
developed
Features
abstraction
human
field
order
developed
knowledge
data
order
techniques
language
developed
order
field
erent
loss
visual
within
data
novel
abstraction
Analysis
leverage
patterns
Text
leverage
information
patterns
Mining
leverage
v dimensions
erent
user
uman
developed
veloped
language
another
method
leverage
abstraction
field
erent
order
human
loss
data
knowledge
visual
linguistics
order
problem
problem

?

Course Objectives

- Comprehensive overview of contemporary approaches in using Python to analyze text as data, and how it is applied in social science policy questions.
- Familiarity with statistical and practical issues around textual data.
- Ability to fit, interpret and apply the basic classes of text cleaning and analysis techniques to independent research projects.

Course Format

- Wednesdays 08.15 – 10.45 @ WiWi 1005
  - Exception! [W8] Wed 01.12 no class – Zoom group consultation appointments (15'/group) for project proposals (slot sign-up in [W5])
  - Class starts @ 8.15 AM sharp.
  - Active preparation before and during class is required.
  - Attendance of 11/14 sessions is required.
- 45' lecture → 10' break → 45' practice (individual) → 5' break → 45' practice (in teams, Week 4 onwards)

Week-by-week course overview*

W1: [Getting started]: installation & platforms, Python basics (data types, variables, strings, list vs. dict, functions)

W2: Python: loops, statements, logical operators, file handling

[Wednesday 27.10:Group matching]

W3: Python: packages for text analysis (numpy, panda, nltk, sciket-learn)

W4: State-of-the-art NLP technique overview & research design

W5: Obtaining and preprocessing corpora

W6: Supervised method – tf-idf & Bag-of-word approach (e.g. Sentiment analysis)

W7: Supervised method – Dictionary-Based approach

[Wednesday 24.11] Midterm proposal pitch & research design report

Week-by-week course overview*

W1: [Getting started]: installation & platforms, Python basics (data types, variables, strings, list vs. dict, functions)

W2: Python: loops, statements, logical operators, file handling

[Wednesday 27.10:Group matching]

W3: Python: packages for text analysis (numpy, panda, nltk, sciket-learn)

W4: State-of-the-art NLP technique overview & research design

W5: Obtaining and preprocessing corpora

W6: Supervised method – tf-idf & Bag-of-word approach (e.g. Sentiment analysis)

W7: Supervised method – Dictionary-Based approach

[Wednesday 24.11] Midterm proposal pitch & research design report

W8: Consultation of mid-term project pitch recordings ***(online)***

W9: Unsupervised Method – Space, tf-idf and cosine similarity

W10: Unsupervised Method - Topic Models

W11: Machine Learning - Classification

W12: Machine Learning - Dimensionality Reduction & Feature Selection

W13: Causal inference – Text as Treatment

W14: Causal inference – Text as Outcome

W_final: Oral presentation of group research projects

Course channels

[SLACK] Group matching, code bug, research project discussions:

https://join.slack.com/t/unihamburgworld/shared_invite/zt-wkqywq13-3OQMM~S~gShngJ~VcTP2~Q

[GITHUB] Slides, syllabuses, reading articles, supplementary materials, exercise sessions:

https://github.com/httn21uhh/Text-Analysis-for-Social-Sciences-in-Python

=> Check now if it works for you!

Course Grading - overview

Task	Due date	Percentage
Mid-term group task: 1) pitch recording (5'/team) 2) research proposal (max 1 page)	Wednesday 24.11.2021 @ 23.59 CEST	30%
Final oral presentation (20-25'/team) + slides	To be confirmed (early 02.2022)	50%
Replicable code package	3 days before your presentation	20%

Teams of two to three students per group (Deadline: Wednesday 27.10)

Written contribution of an individual about his/her/their own contribution and members of his/her/their own team in a confidential form is required.

Course Grading – class participation

Active participation (Q&A, collaboration with classmates) throughout all class sessions are expected.

Students cannot miss more than 2/14 in-class sessions.

Course Grading – research project

Midterm research project proposal (30%) consists of two components:

i) Research pitch recording (5'/team) AND

ii) Research proposal summary (max 1 page/500 words)

Data resources?

- Self - sourced
- Ask instructor for ideas and sources
- Example corpora & projects:
  - mhttps://digitalhumanities.berkeley.edu/projectsple Corpora
  - U.S. Congressional Record, https://data.stanford.edu/congress_text
  - CourtListener, https://github.com/idc9/law-net
  - Chris Bail's list of digital data sets

=> SEE DETAILS IN THE Research_project.pdf file on Github!

Course grading – final presentation (50%) & replicable code package (20%)

=> SEE DETAILS IN THE Research_project.pdf file on Github!

House rules Q&A

What can I e-mail the instructor about?

Can I invite someone who is not part of the course to join Slack and share with him the materials?

Where can I find course materials?

Can I work alone for the research group project?

Where can I upload the recordings? The midterm research design?

Where can I find more information about the research project details?

How do I get course announcement?

Can I attend and get credits for the course if I miss more than 2 in-person sessions?

Installations

https://docs.google.com/document/d/1UkCytHT4ZF-rDoh_buH6xb9mLz4GcGjT1qIu-pEWThI/edit

(Anaconda => Jupyter Notebook, Python & more)

StackOverflow = Your best debug buddy & 24/7 instructor

https://stackoverflow.com/

EXERCISE PART: Python basics

- Open your Jupyter Notebook OR Google Colab
- Follow closely the illustrated examples and replicate yourself as the session proceeds.
- Raise your hands to ask questions at any point, including when you think things go too fast/slow for you.