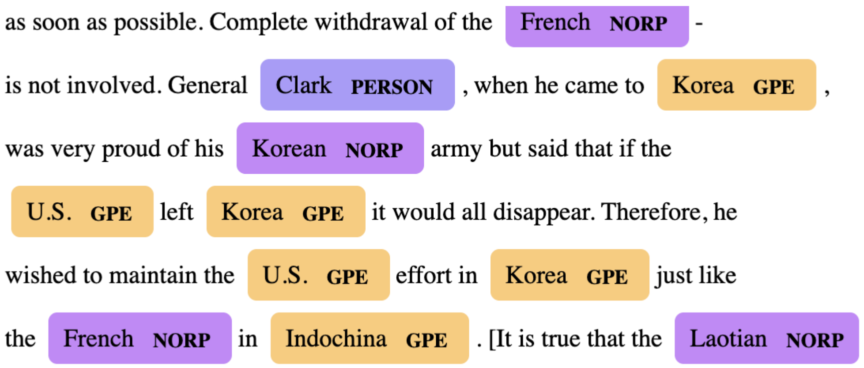
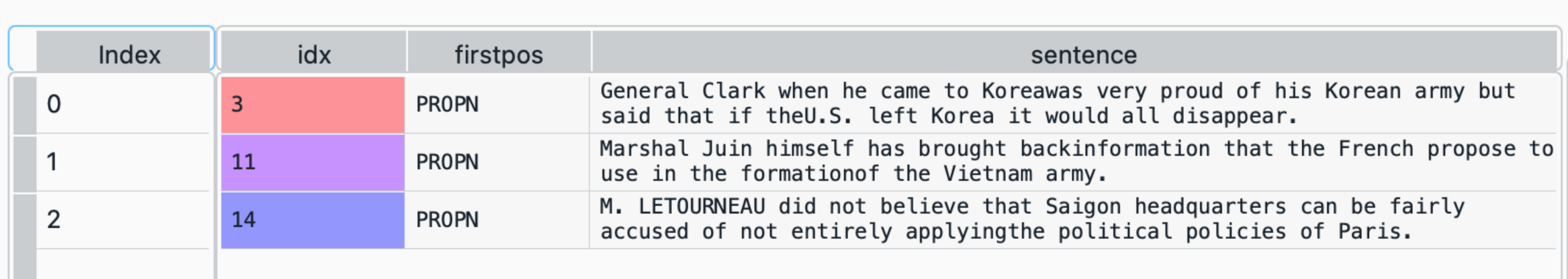
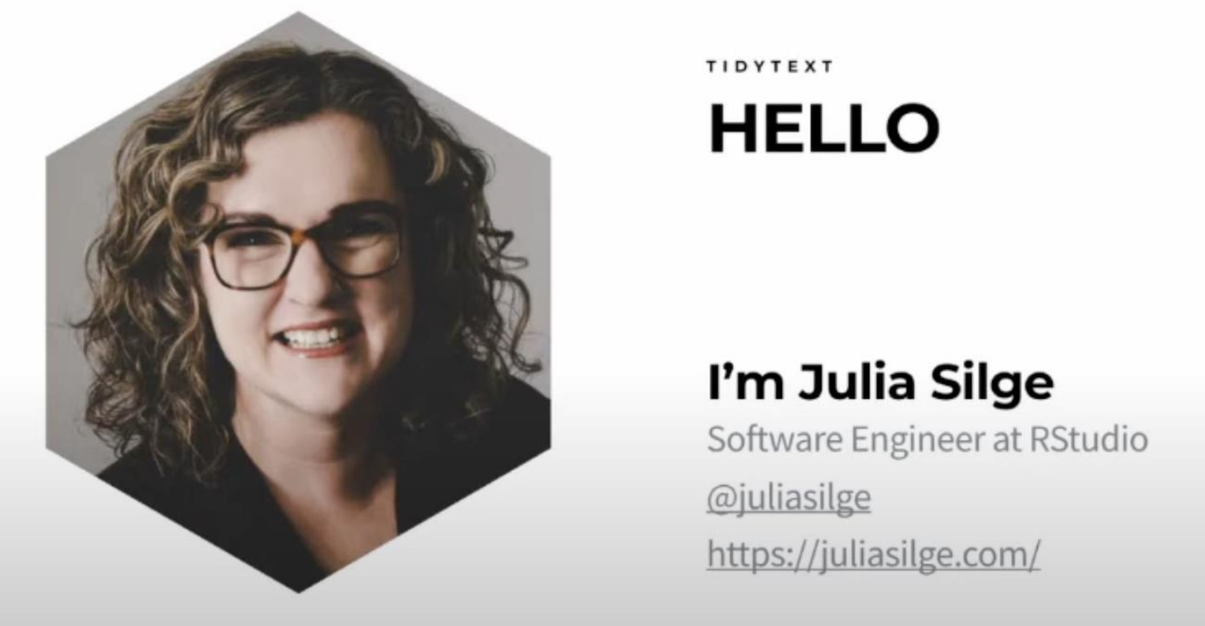
1. Github på CV'et
2. Python test
3. Pentagon Papers øvelser


as soon as possible. Complete withdrawal of the French NORP -
is not involved. General Clark PERSON , when he came to Korea GPE ,
was very proud of his Korean NORP army but said that if the
U.S. GPE left Korea GPE it would all disappear. Therefore, he
wished to maintain the U.S. GPE effort in Korea GPE just like
the French NORP in Indochina GPE . [It is true that the Laotian NORP



Index	idx	firstpos	sentence
0	3	PROPN	General Clark when he came to Koreawas very proud of his Korean army but said that if theU.S. left Korea it would all disappear.
1	11	PROPN	Marshal Juin himself has brought backinformation that the French propose to use in the formationof the Vietnam army.
2	14	PROPN	M. LETOURNEAU did not believe that Saigon headquarters can be fairly accused of not entirely applyingthe political policies of Paris.



TIDYTEXT

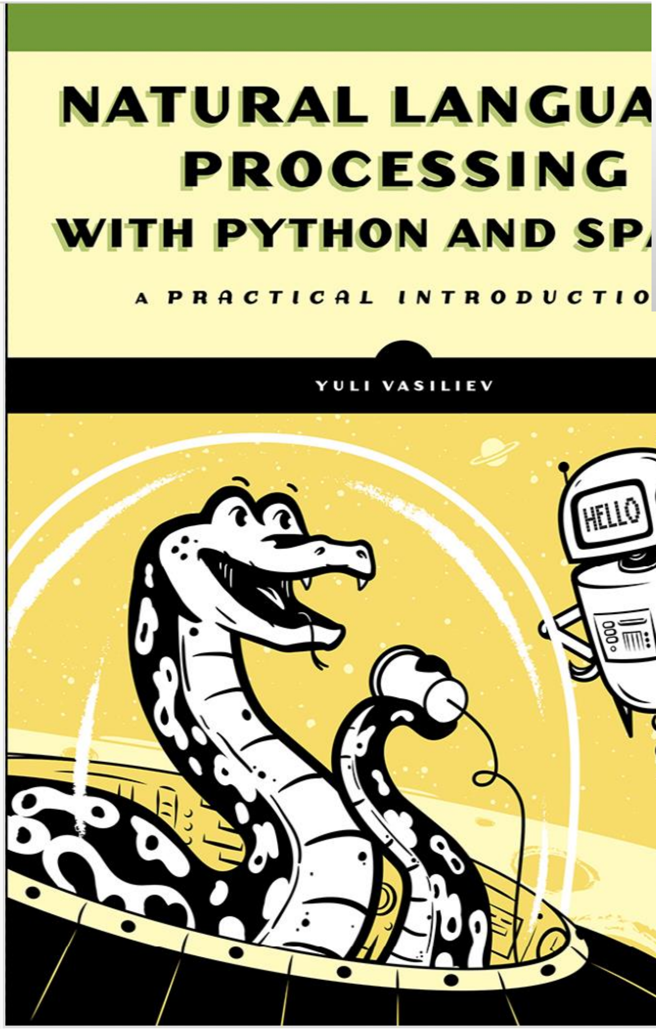
HELLO

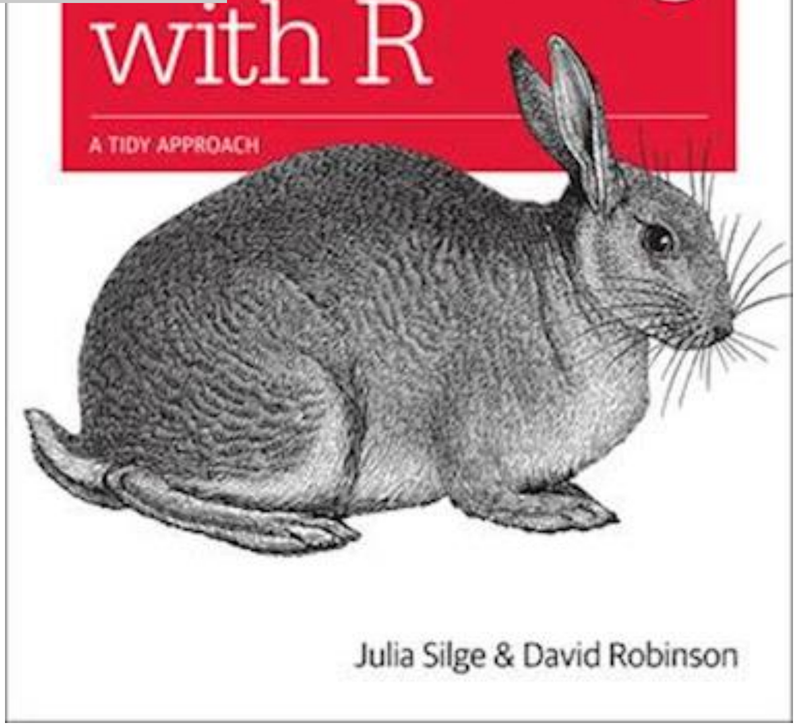
I'm Julia Silge

Software Engineer at RStudio

@juliasilge

https://juliasilge.com/

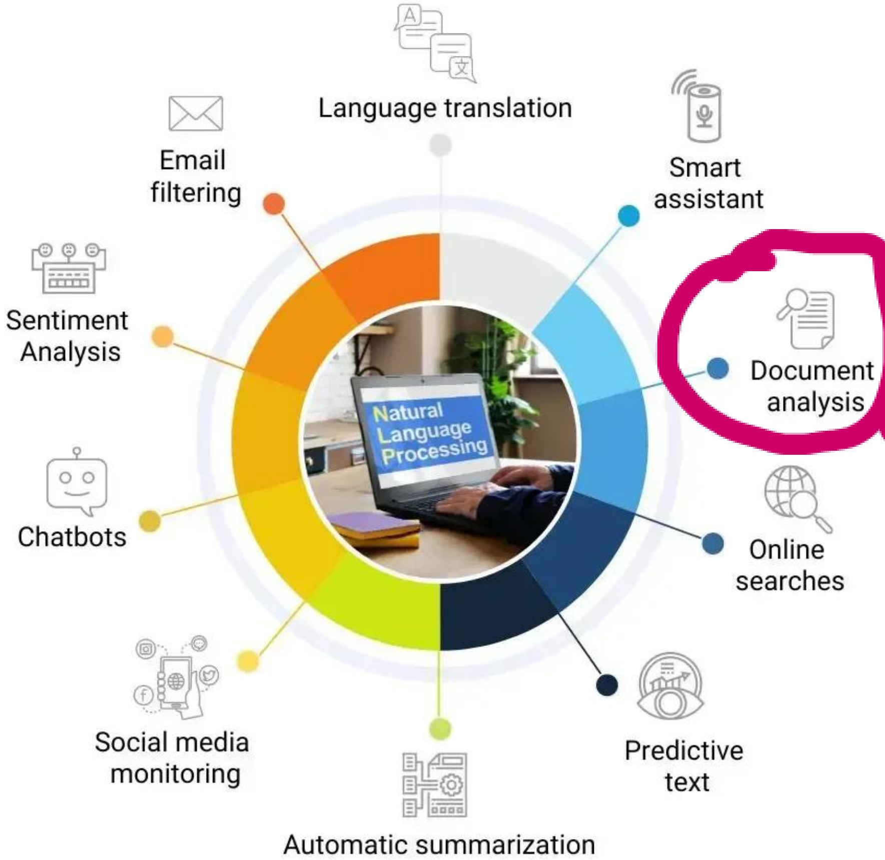




Planen

SILVAN

jem & fix
LAVPRIS BYGGEMARKED


Natural Language Processing

Language translation

Email filtering

Smart assistant

Sentiment Analysis

Document analysis

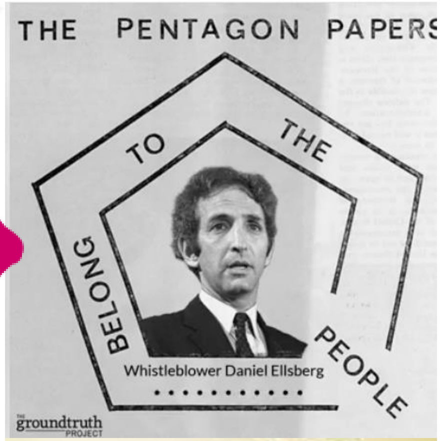
Chatbots

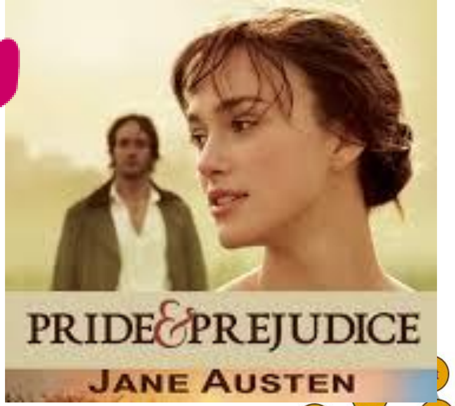
Online searches

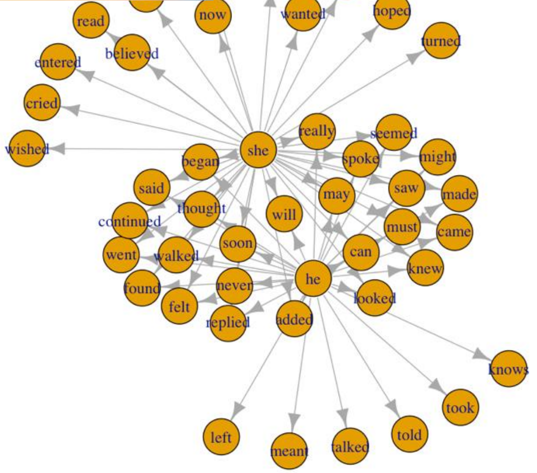
Social media monitoring

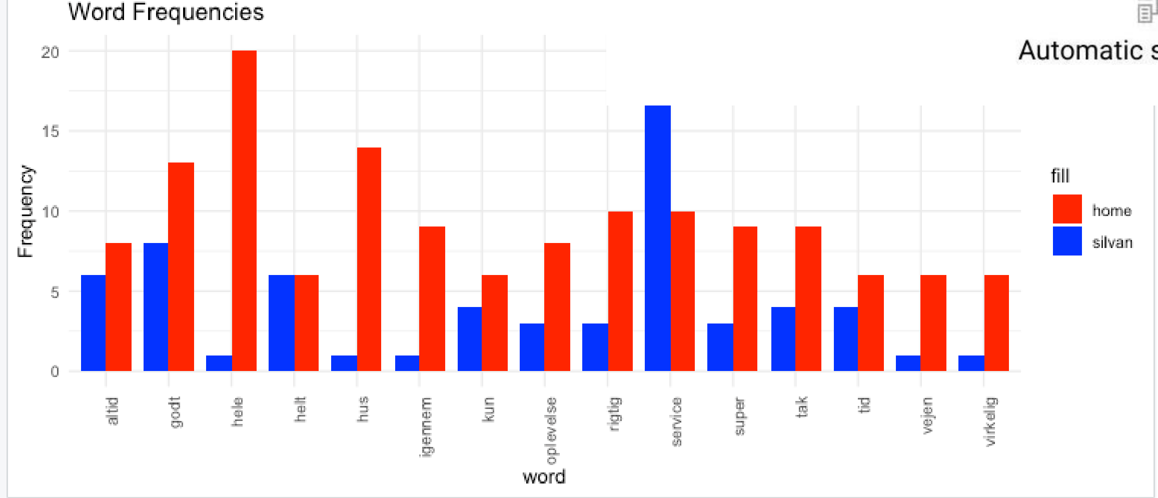
Predictive text

Automatic summarization

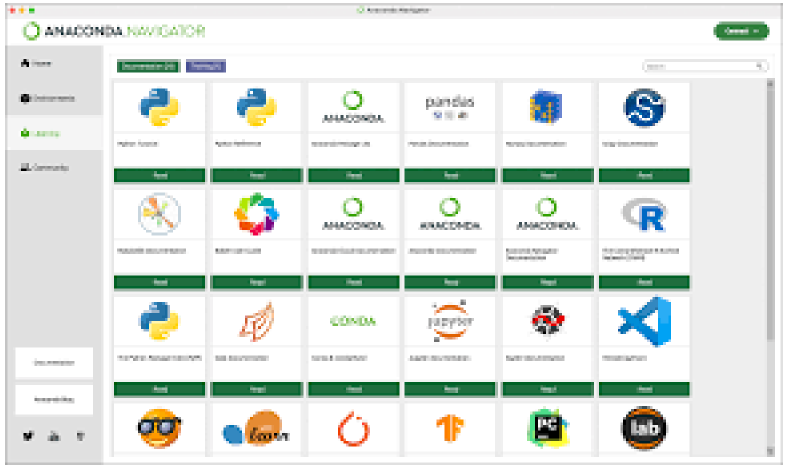


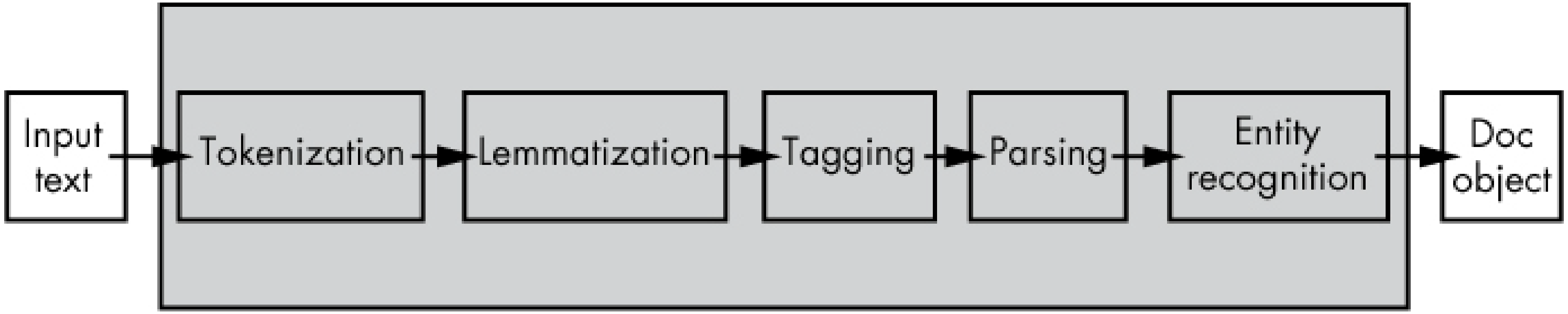






Værktøj





Pipeline

Input text → Tokenization → Lemmatization → Tagging → Parsing → Entity recognition → Doc object

Advarsel

Hvad skete der med Biblioteket i Alexandria?

©Getty Images

Det Store Bibliotek i Alexandria var en af de mest ambitiøse intellektuelle bedrifter i menneskets historie – et sted hvor viden ingen grænser havde, og jagten på visdom blev betragtet som hellig. Oprettet i det 3. århundrede f.v.t. i den blomstrende by Alexandria i Egypten, var det langt mere end blot et bibliotek. Det var et centrum for læring, et kulturelt samlingspunkt og et fyrtårn for oplysning, der tiltrak datidens største tænkere, videnskabsfolk og filosoffer fra hele den kendte verden.

En vision om fremtiden

©Getty Images

Med udgangspunkt i Alexander den Stores drøm om en verden forenet gennem viden, forestillede Ptolemaios I og senere Ptolemaios II sig et storslået sted, hvor al menneskelig viden kunne samles, bevares og udvikles af tidens skarpeste hjerner.

Bøger ved kajen

©Getty Images

Under Ptolemaios III blev det lov, at alle skibe, der lagde til i Alexandrias havn, skulle aflevere eventuelle bøger. Skrivere lavede kopier, originalerne blev beholdt, og kopierne røg tilbage ombord. Sådan voksede samlingen konstant.

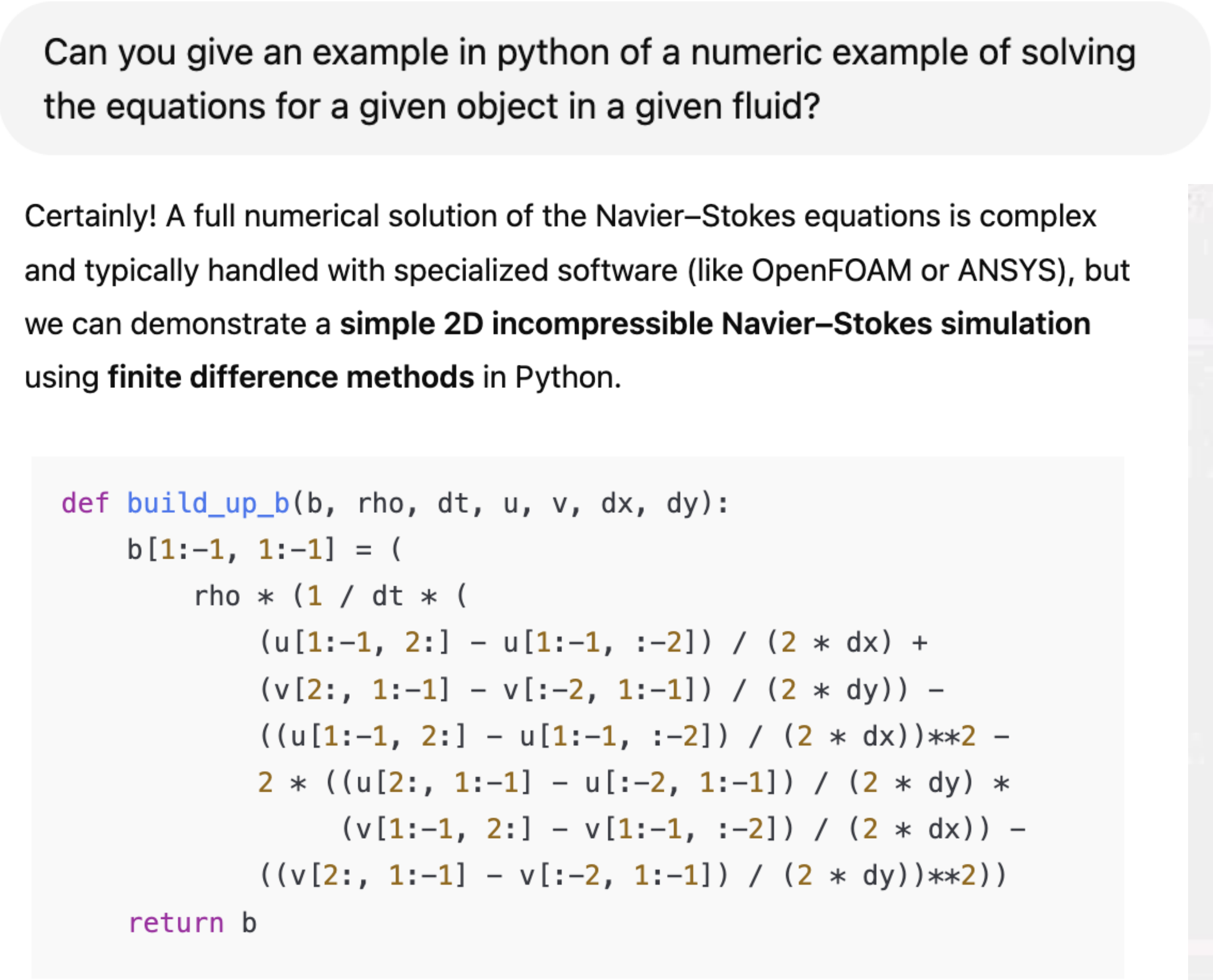
Musernes hellige hjem

©Public Domain

Biblioteket var en del af Mouseion – opkaldt efter muserne, de ni gudinder for kunst og viden. Med op mod en halv million skriftruller blev det et magnetisk centrum for lærde fra hele den antikke verden.

Mette Mortensen
PREPPERENS HÅNDBOG
NATURAL LANGUAGE PROCESSING WITH PYTHON AND SPACY
A PRACTICAL INTRODUCTION
YULI VASILIEV
O'REILLY
Text Mining with R
A TIDY APPROACH
Julia Silge & David Robinson
whiskas
FIRST AID KIT
DIN13164

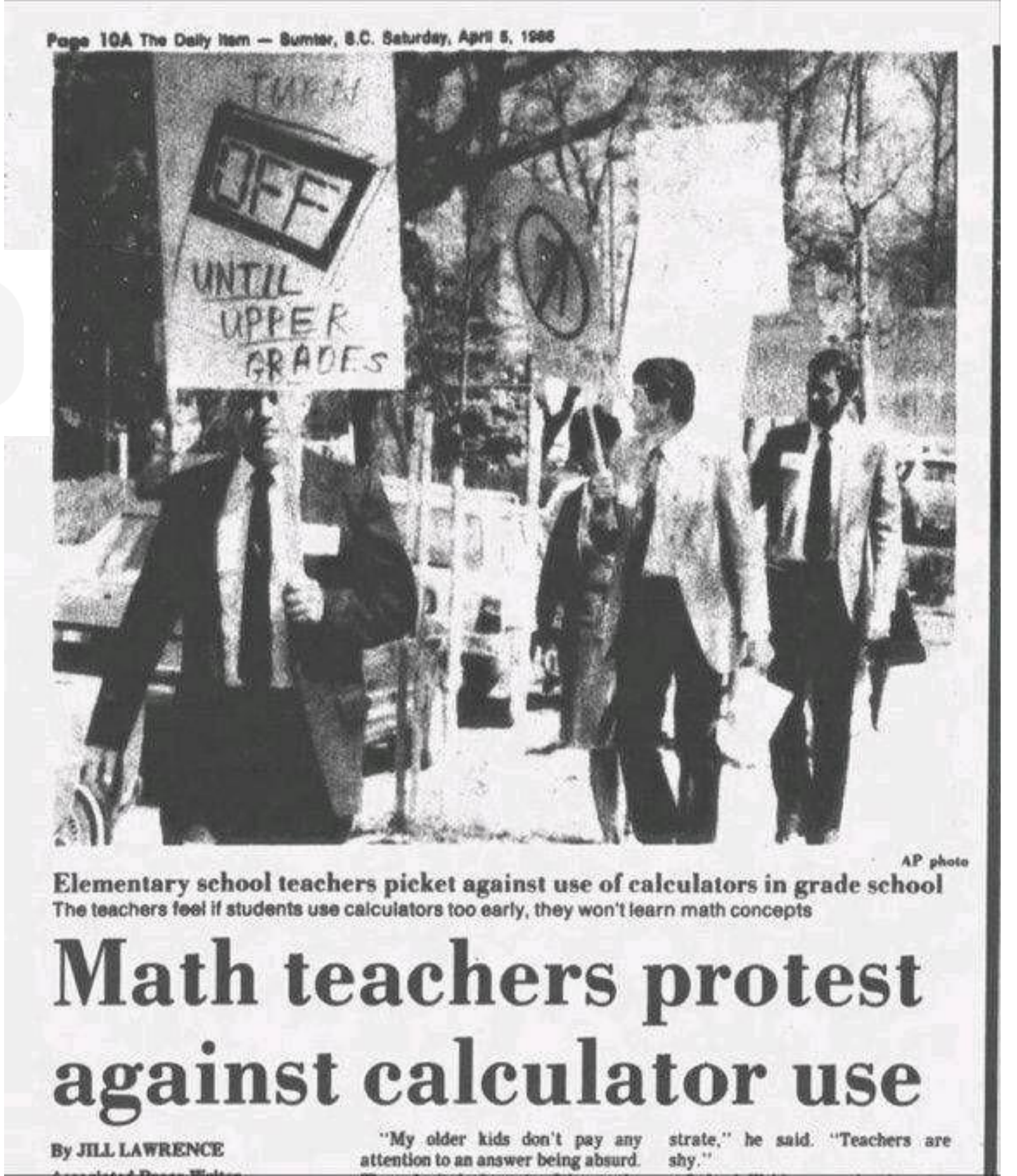
Er ChatGPT for datascience hvad lommeregneren var for anvendt matematik?



Can you give an example in python of a numeric example of solving the equations for a given object in a given fluid?

Certainly! A full numerical solution of the Navier–Stokes equations is complex and typically handled with specialized software (like OpenFOAM or ANSYS), but we can demonstrate a **simple 2D incompressible Navier–Stokes simulation** using **finite difference methods** in Python.

```python
def build_up_b(b, rho, dt, u, v, dx, dy):
    b[1:-1, 1:-1] = (
        rho * (1 / dt * (
            (u[1:-1, 2:] - u[1:-1, :-2]) / (2 * dx) +
            (v[2:, 1:-1] - v[:-2, 1:-1]) / (2 * dy)) -
            ((u[1:-1, 2:] - u[1:-1, :-2]) / (2 * dx))**2 -
            2 * ((u[2:, 1:-1] - u[:-2, 1:-1]) / (2 * dy) *
                 (v[1:-1, 2:] - v[1:-1, :-2]) / (2 * dx)) -
            ((v[2:, 1:-1] - v[:-2, 1:-1]) / (2 * dy))**2))
    return b
```


TURN
OFF
UNTIL
UPPER
GRADES

Page 10A The Daily Item — Sumter, S.C. Saturday, April 5, 1986

AP photo

Elementary school teachers picket against use of calculators in grade school
The teachers feel if students use calculators too early, they won't learn math concepts

Math teachers protest against calculator use

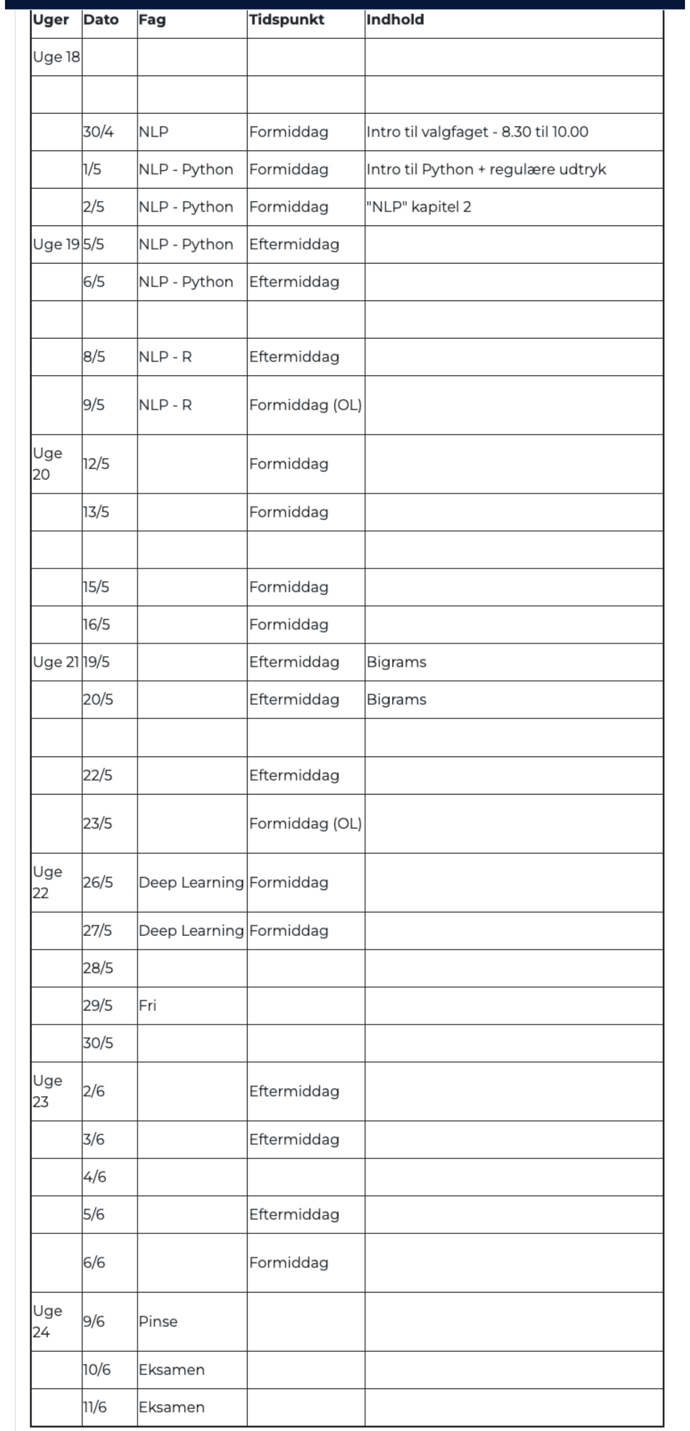
By JILL LAWRENCE

"My older kids don't pay any attention to an answer being absurd.

strate," he said. "Teachers are shy."

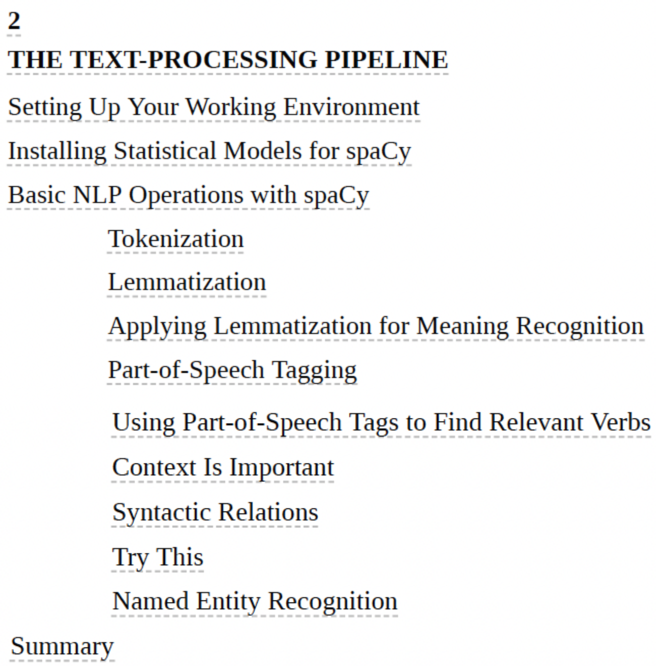
Planen

Tid

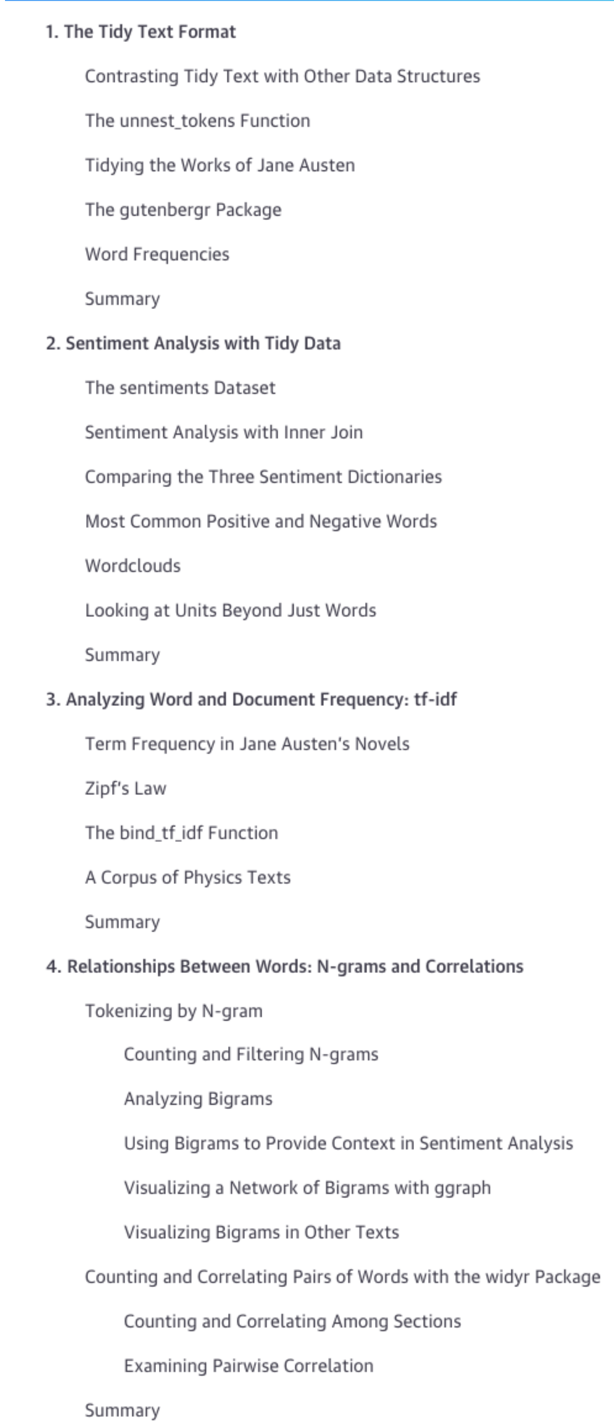


Uger	Dato	Fag	Tidspunkt	Indhold
Uge 18				
	30/4	NLP	Formiddag	Intro til valgfaget - 8.30 til 10.00
	1/5	NLP - Python	Formiddag	Intro til Python + regulære udtryk
	2/5	NLP - Python	Formiddag	"NLP" kapitel 2
Uge 19	5/5	NLP - Python	Eftermiddag	
	6/5	NLP - Python	Eftermiddag	
	8/5	NLP - R	Eftermiddag	
	9/5	NLP - R	Formiddag (OL)	
Uge 20	12/5		Formiddag	
	13/5		Formiddag	
	15/5		Formiddag	
	16/5		Formiddag	
Uge 21	19/5		Eftermiddag	Bigrams
	20/5		Eftermiddag	Bigrams
	22/5		Eftermiddag	
	23/5		Formiddag (OL)	
Uge 22	26/5	Deep Learning	Formiddag	
	27/5	Deep Learning	Formiddag	
	28/5			
	29/5	Fri		
	30/5			
Uge 23	2/6		Eftermiddag	
	3/6		Eftermiddag	
	4/6			
	5/6		Eftermiddag	
	6/6		Formiddag	
Uge 24	9/6	Pinse		
	10/6	Eksamen		
	11/6	Eksamen		

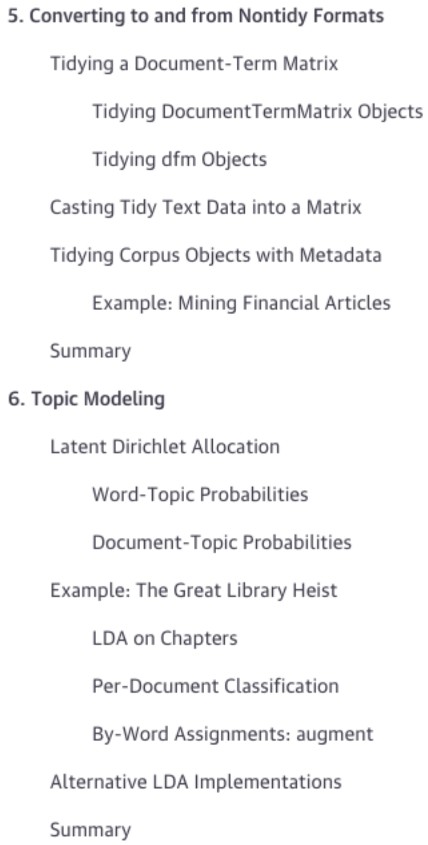
Planen Indhold



2
THE TEXT-PROCESSING PIPELINE
- Setting Up Your Working Environment
- Installing Statistical Models for spaCy
- Basic NLP Operations with spaCy
  - Tokenization
  - Lemmatization
  - Applying Lemmatization for Meaning Recognition
  - Part-of-Speech Tagging
  - Using Part-of-Speech Tags to Find Relevant Verbs
  - Context Is Important
  - Syntactic Relations
  - Try This
  - Named Entity Recognition
- Summary



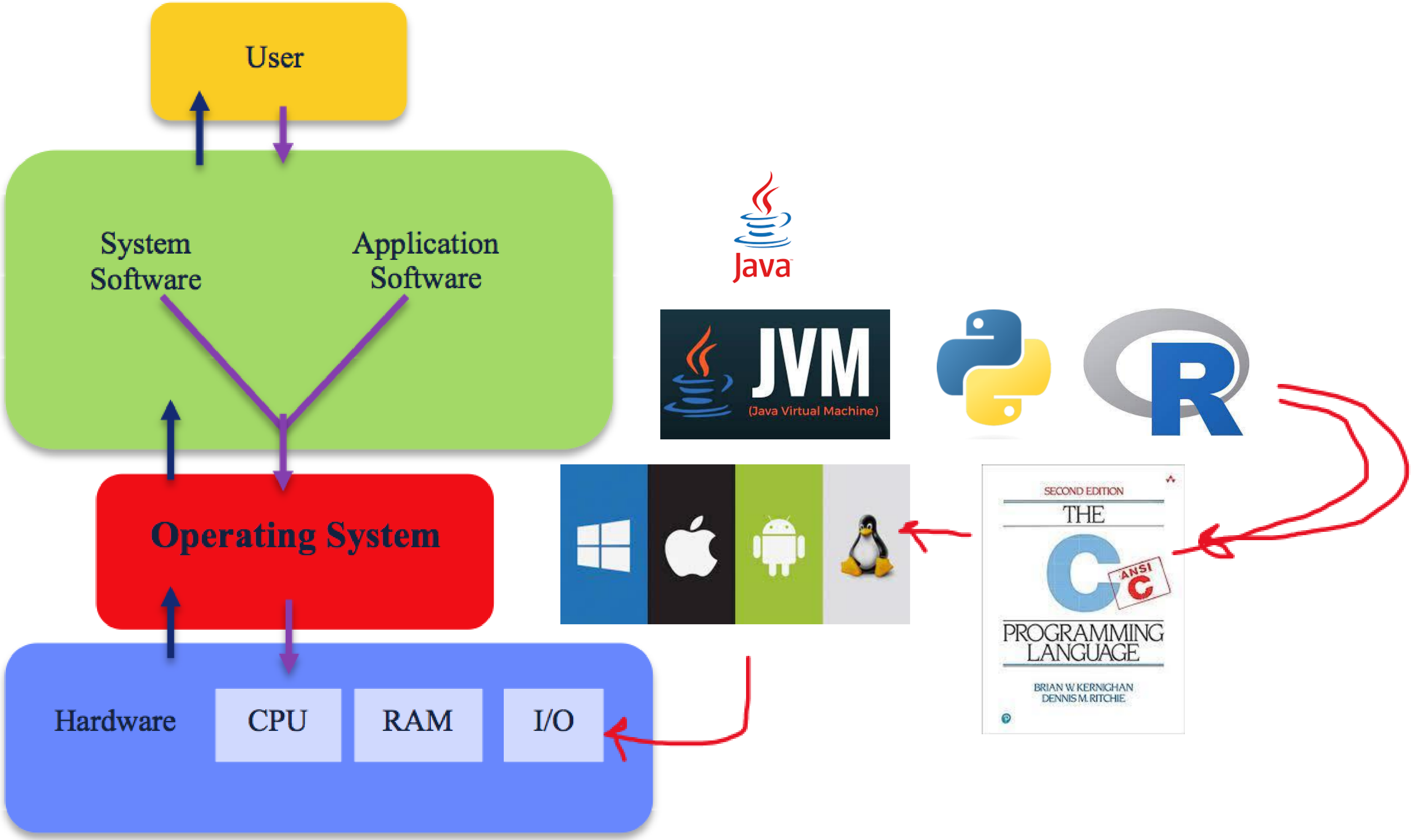
1. The Tidy Text Format
   - Contrasting Tidy Text with Other Data Structures
   - The unnest_tokens Function
   - Tidying the Works of Jane Austen
   - The gutenbergr Package
   - Word Frequencies
   - Summary
2. Sentiment Analysis with Tidy Data
   - The sentiments Dataset
   - Sentiment Analysis with Inner Join
   - Comparing the Three Sentiment Dictionaries
   - Most Common Positive and Negative Words
   - Wordclouds
   - Looking at Units Beyond Just Words
   - Summary
3. Analyzing Word and Document Frequency: tf-idf
   - Term Frequency in Jane Austen's Novels
   - Zipf's Law
   - The bind_tf_idf Function
   - A Corpus of Physics Texts
   - Summary
4. Relationships Between Words: N-grams and Correlations
   - Tokenizing by N-gram
     - Counting and Filtering N-grams
     - Analyzing Bigrams
     - Using Bigrams to Provide Context in Sentiment Analysis
     - Visualizing a Network of Bigrams with ggraph
     - Visualizing Bigrams in Other Texts
   - Counting and Correlating Pairs of Words with the widyr Package
     - Counting and Correlating Among Sections
     - Examining Pairwise Correlation
   - Summary



5. Converting to and from Nontidy Formats
   - Tidying a Document-Term Matrix
     - Tidying DocumentTermMatrix Objects
     - Tidying dfm Objects
   - Casting Tidy Text Data into a Matrix
   - Tidying Corpus Objects with Metadata
     - Example: Mining Financial Articles
   - Summary
6. Topic Modeling
   - Latent Dirichlet Allocation
     - Word-Topic Probabilities
     - Document-Topic Probabilities
   - Example: The Great Library Heist
     - LDA on Chapters
     - Per-Document Classification
     - By-Word Assignments: augment
   - Alternative LDA Implementations
   - Summary

Python

https://www.w3schools.com/python/python_quiz.asp



User

System Software

Application Software

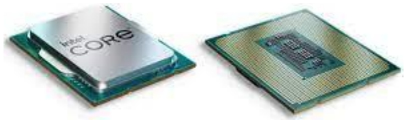
Operating System

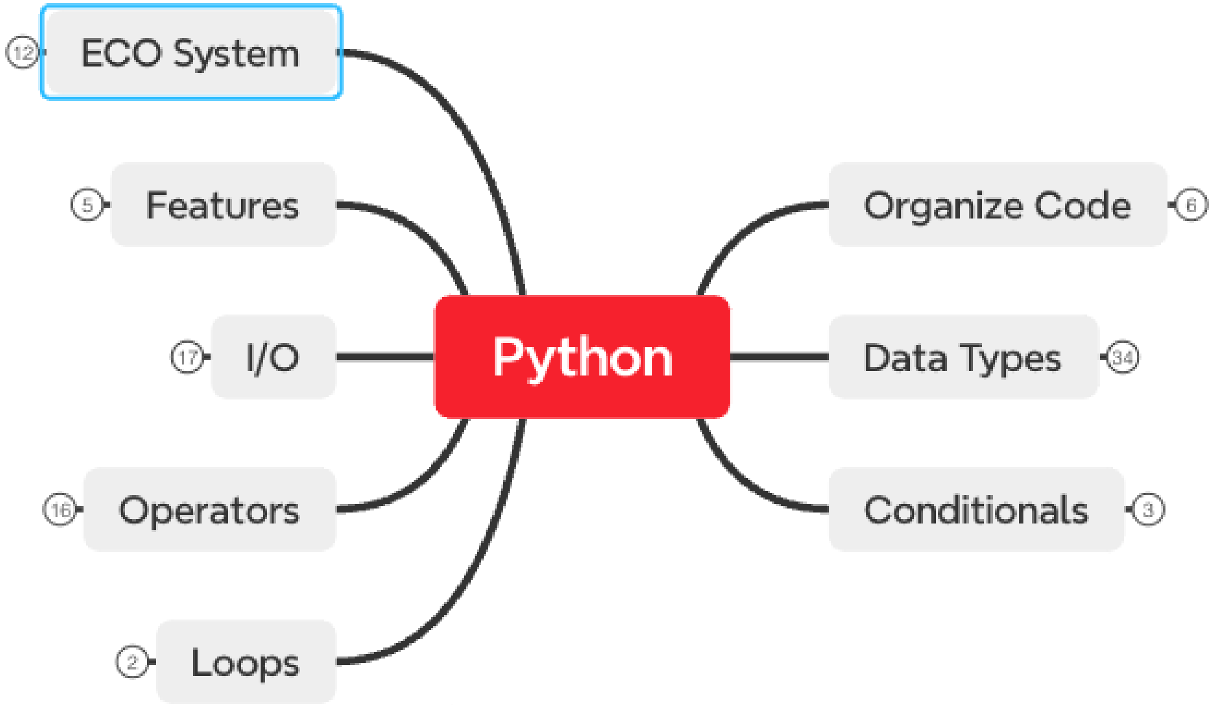
Hardware

CPU

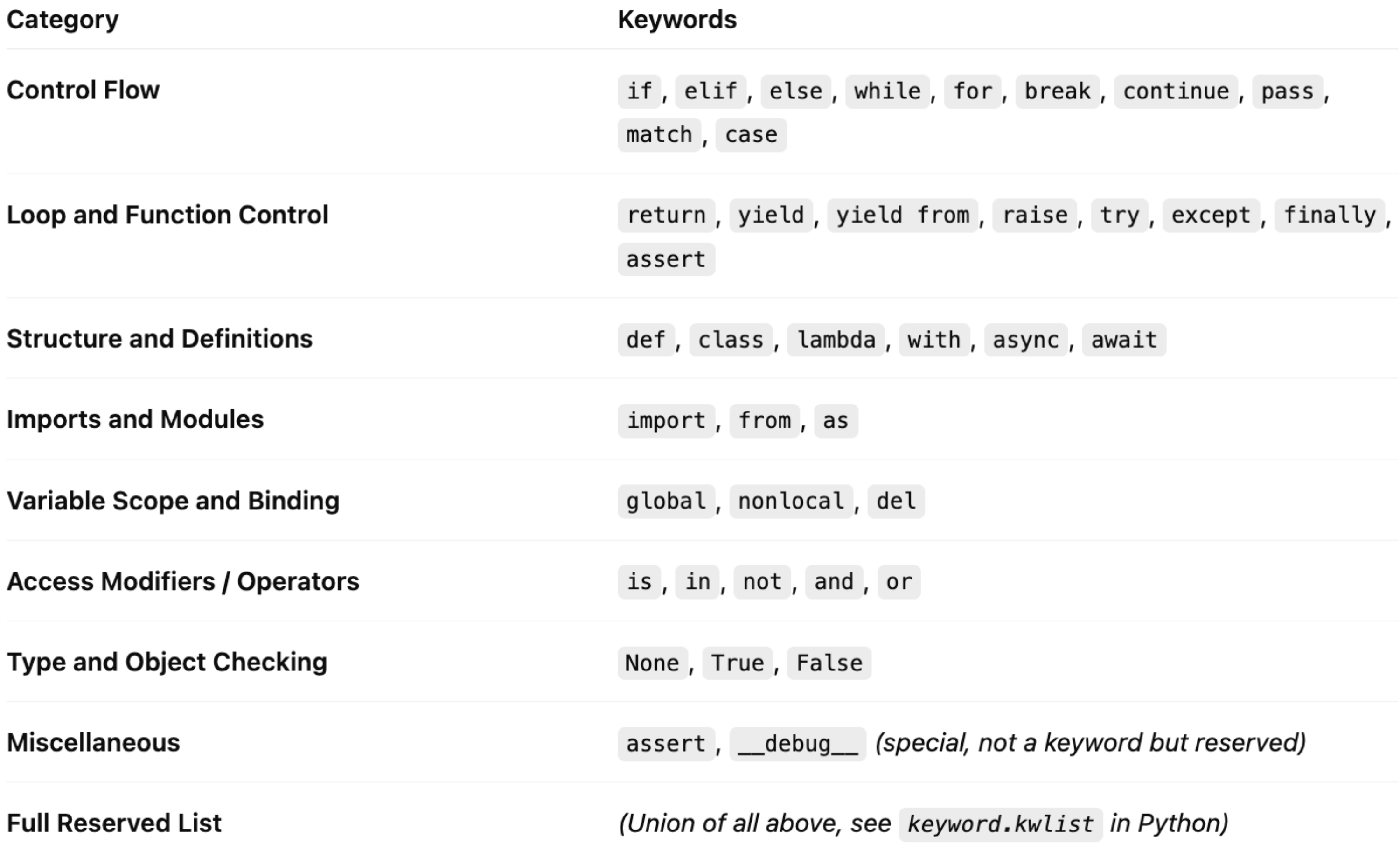
RAM

I/O

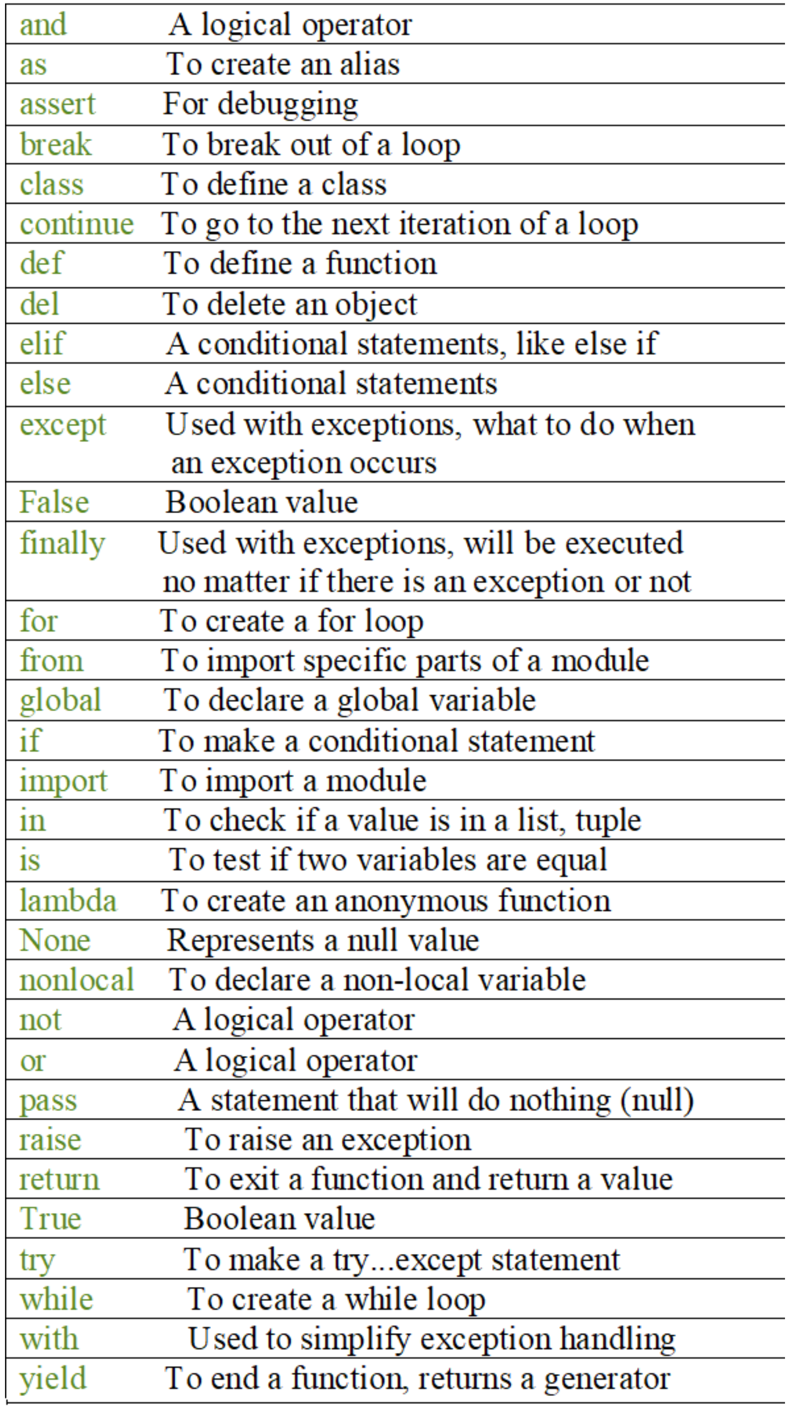




- Python
  - ⑫ ECO System
  - ⑤ Features
  - ⑰ I/O
  - ⑯ Operators
  - ② Loops
  - Organize Code ⑥
  - Data Types (34)
  - Conditionals ③

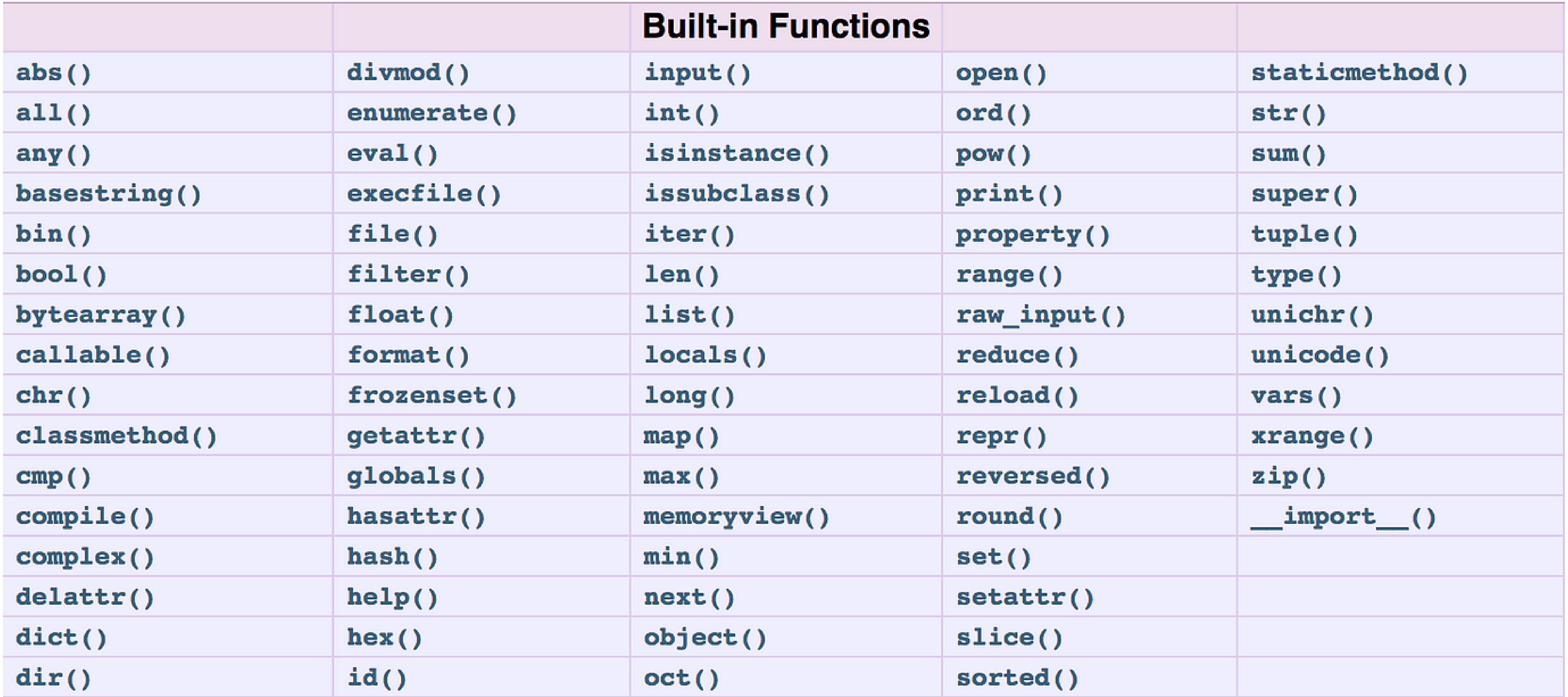


Category	Keywords
Control Flow	`if`, `elif`, `else`, `while`, `for`, `break`, `continue`, `pass`, `match`, `case`
Loop and Function Control	`return`, `yield`, `yield from`, `raise`, `try`, `except`, `finally`, `assert`
Structure and Definitions	`def`, `class`, `lambda`, `with`, `async`, `await`
Imports and Modules	`import`, `from`, `as`
Variable Scope and Binding	`global`, `nonlocal`, `del`
Access Modifiers / Operators	`is`, `in`, `not`, `and`, `or`
Type and Object Checking	`None`, `True`, `False`
Miscellaneous	`assert`, `__debug__` *(special, not a keyword but reserved)*
Full Reserved List	*(Union of all above, see* `keyword.kwlist` *in Python)*



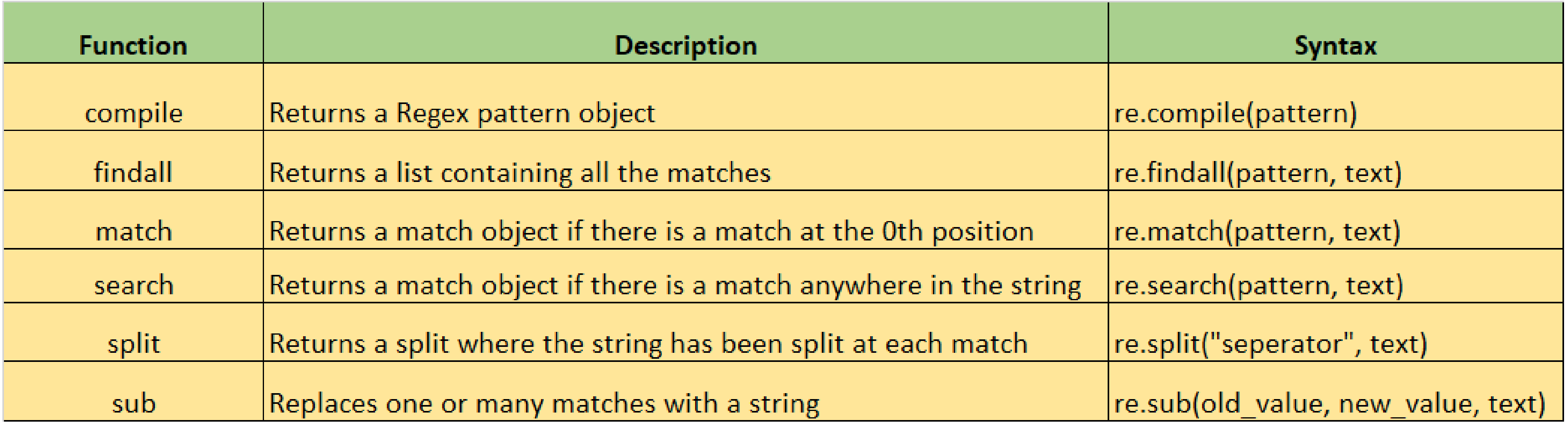
and	A logical operator
as	To create an alias
assert	For debugging
break	To break out of a loop
class	To define a class
continue	To go to the next iteration of a loop
def	To define a function
del	To delete an object
elif	A conditional statements, like else if
else	A conditional statements
except	Used with exceptions, what to do when an exception occurs
False	Boolean value
finally	Used with exceptions, will be executed no matter if there is an exception or not
for	To create a for loop
from	To import specific parts of a module
global	To declare a global variable
if	To make a conditional statement
import	To import a module
in	To check if a value is in a list, tuple
is	To test if two variables are equal
lambda	To create an anonymous function
None	Represents a null value
nonlocal	To declare a non-local variable
not	A logical operator
or	A logical operator
pass	A statement that will do nothing (null)
raise	To raise an exception
return	To exit a function and return a value
True	Boolean value
try	To make a try...except statement
while	To create a while loop
with	Used to simplify exception handling
yield	To end a function, returns a generator

Python funktioner

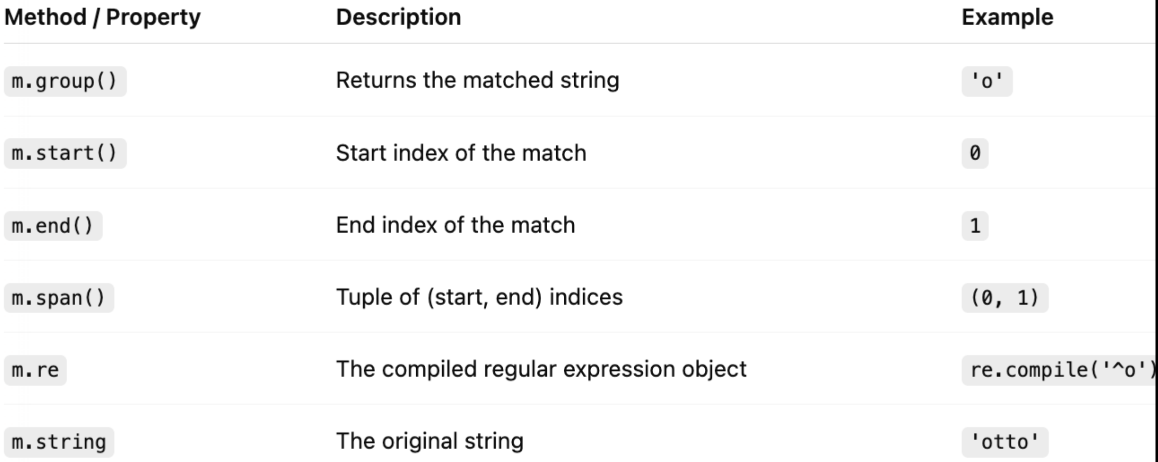


		Built-in Functions		
`abs()`	`divmod()`	`input()`	`open()`	`staticmethod()`
`all()`	`enumerate()`	`int()`	`ord()`	`str()`
`any()`	`eval()`	`isinstance()`	`pow()`	`sum()`
`basestring()`	`execfile()`	`issubclass()`	`print()`	`super()`
`bin()`	`file()`	`iter()`	`property()`	`tuple()`
`bool()`	`filter()`	`len()`	`range()`	`type()`
`bytearray()`	`float()`	`list()`	`raw_input()`	`unichr()`
`callable()`	`format()`	`locals()`	`reduce()`	`unicode()`
`chr()`	`frozenset()`	`long()`	`reload()`	`vars()`
`classmethod()`	`getattr()`	`map()`	`repr()`	`xrange()`
`cmp()`	`globals()`	`max()`	`reversed()`	`zip()`
`compile()`	`hasattr()`	`memoryview()`	`round()`	`__import__()`
`complex()`	`hash()`	`min()`	`set()`	
`delattr()`	`help()`	`next()`	`setattr()`	
`dict()`	`hex()`	`object()`	`slice()`	
`dir()`	`id()`	`oct()`	`sorted()`	

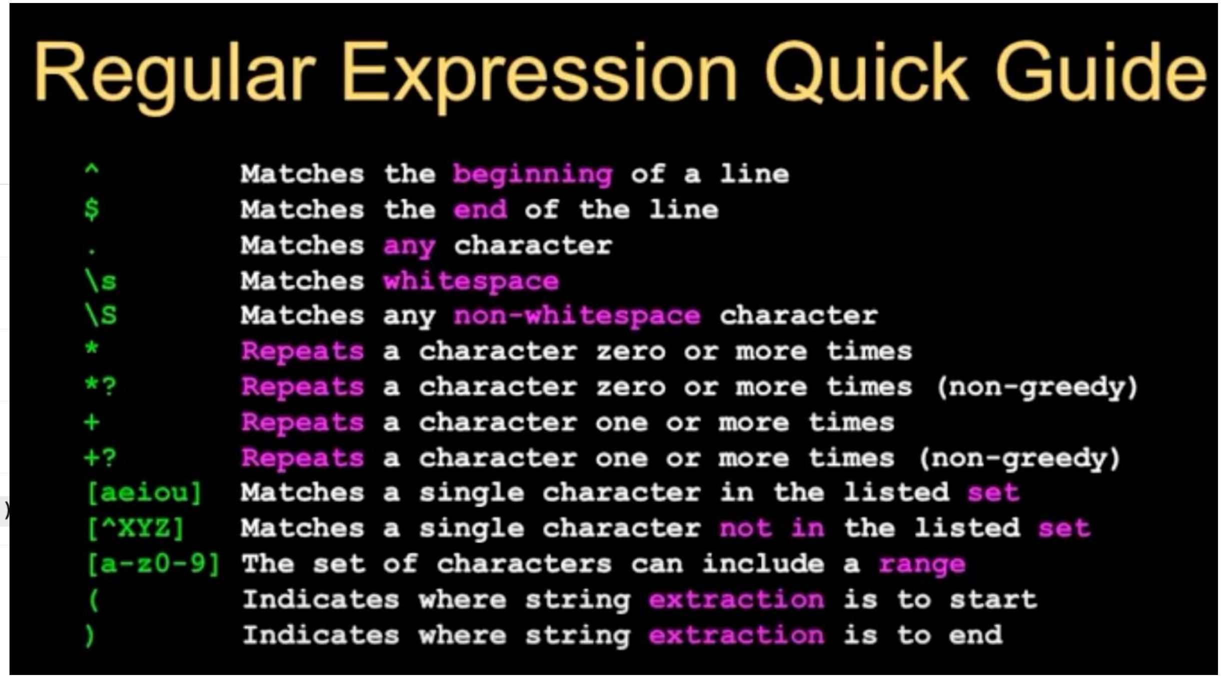
Regex



Function	Description	Syntax
compile	Returns a Regex pattern object	re.compile(pattern)
findall	Returns a list containing all the matches	re.findall(pattern, text)
match	Returns a match object if there is a match at the 0th position	re.match(pattern, text)
search	Returns a match object if there is a match anywhere in the string	re.search(pattern, text)
split	Returns a split where the string has been split at each match	re.split("seperator", text)
sub	Replaces one or many matches with a string	re.sub(old_value, new_value, text)

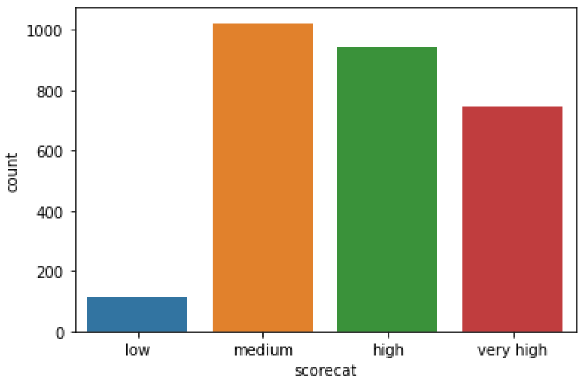


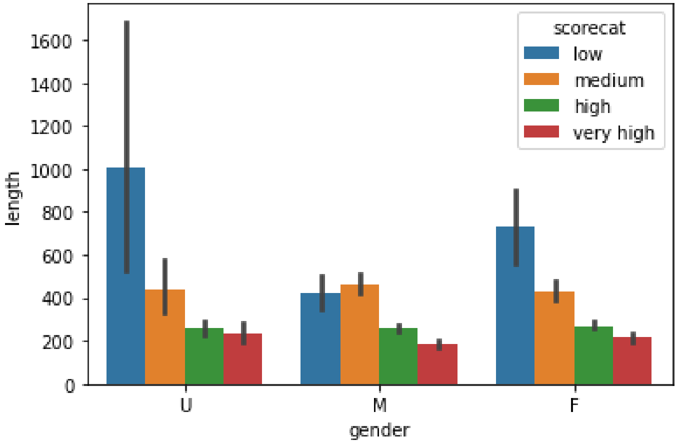
Method / Property	Description	Example
`m.group()`	Returns the matched string	`'o'`
`m.start()`	Start index of the match	`0`
`m.end()`	End index of the match	`1`
`m.span()`	Tuple of (start, end) indices	`(0, 1)`
`m.re`	The compiled regular expression object	`re.compile('^o')`
`m.string`	The original string	`'otto'`

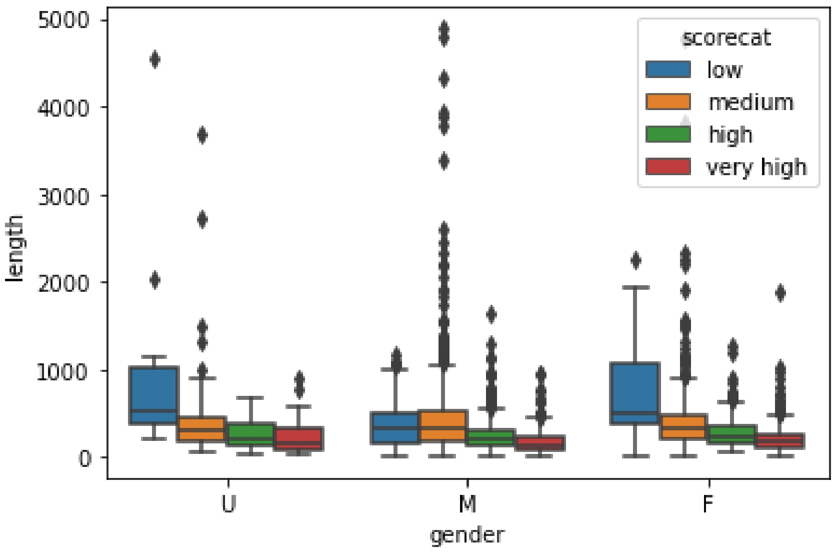


Regular Expression Quick Guide

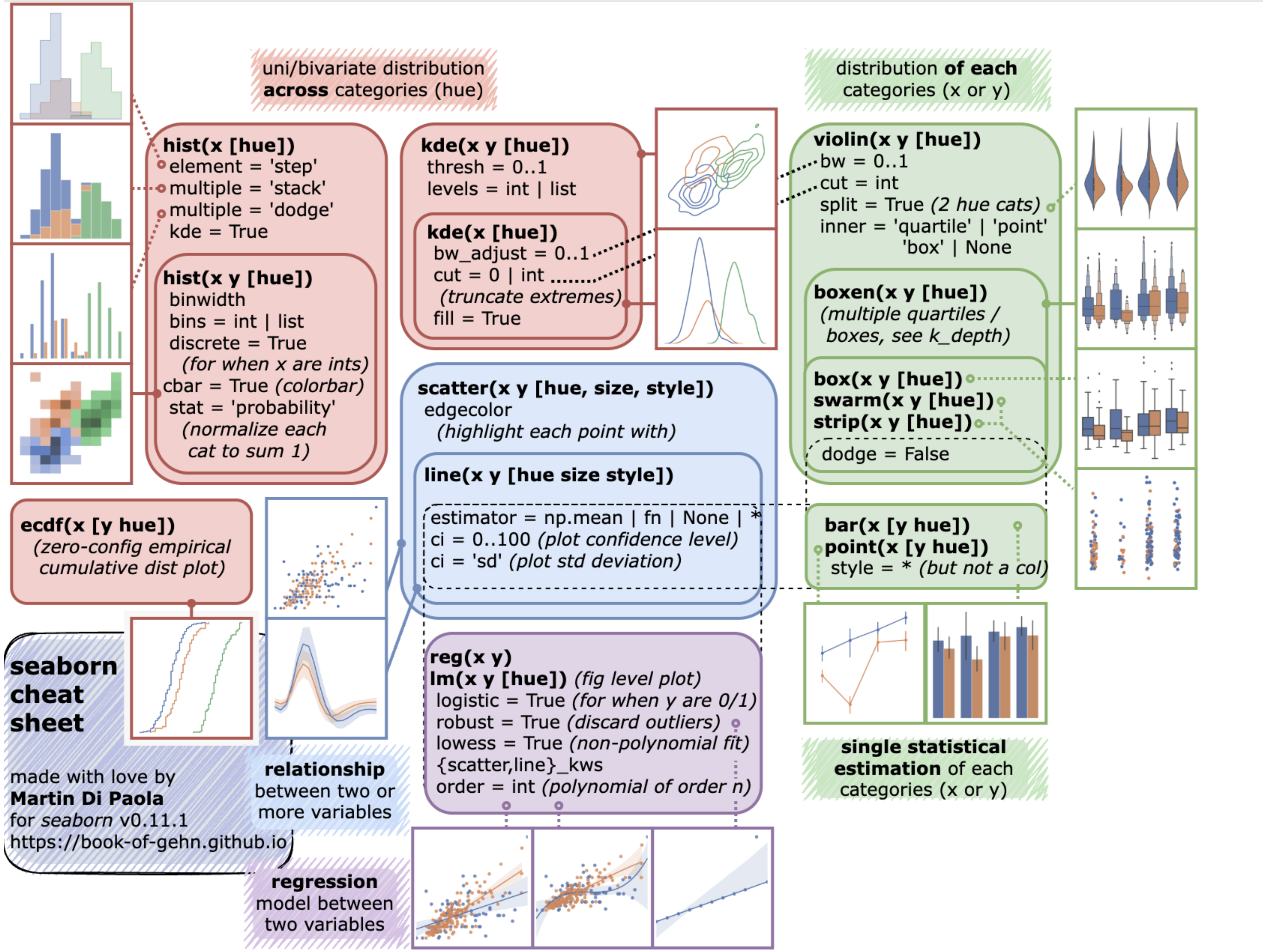
`^`	Matches the beginning of a line
`$`	Matches the end of the line
`.`	Matches any character
`\s`	Matches whitespace
`\S`	Matches any non-whitespace character
`*`	Repeats a character zero or more times
`*?`	Repeats a character zero or more times (non-greedy)
`+`	Repeats a character one or more times
`+?`	Repeats a character one or more times (non-greedy)
`[aeiou]`	Matches a single character in the listed set
`[^XYZ]`	Matches a single character not in the listed set
`[a-z0-9]`	The set of characters can include a range
`(`	Indicates where string extraction is to start
`)`	Indicates where string extraction is to end


count
1000
800
600
400
200
0
low
medium
high
very high
scorecat


length
1600
1400
1200
1000
800
600
400
200
0
U
M
F
gender
scorecat
low
medium
high
very high


length
5000
4000
3000
2000
1000
0
U
M
F
gender
scorecat
low
medium
high
very high

Seaborn



seaborn cheat sheet

made with love by
Martin Di Paola
for *seaborn* v0.11.1
https://book-of-gehn.github.io

uni/bivariate distribution
across categories (hue)

hist(x [hue])
element = 'step'
multiple = 'stack'
multiple = 'dodge'
kde = True

hist(x y [hue])
binwidth
bins = int | list
discrete = True
(for when x are ints)
cbar = True *(colorbar)*
stat = 'probability'
(normalize each cat to sum 1)

kde(x y [hue])
thresh = 0..1
levels = int | list

kde(x [hue])
bw_adjust = 0..1
cut = 0 | int
(truncate extremes)
fill = True

ecdf(x [y hue])
(zero-config empirical cumulative dist plot)

relationship
between two or
more variables

scatter(x y [hue, size, style])
edgecolor
(highlight each point with)

line(x y [hue size style])
estimator = np.mean | fn | None | *
ci = 0..100 *(plot confidence level)*
ci = 'sd' *(plot std deviation)*

regression
model between
two variables

reg(x y)
lm(x y [hue]) *(fig level plot)*
logistic = True *(for when y are 0/1)*
robust = True *(discard outliers)*
lowess = True *(non-polynomial fit)*
{scatter,line}_kws
order = int *(polynomial of order n)*

distribution **of each**
categories (x or y)

violin(x y [hue])
bw = 0..1
cut = int
split = True *(2 hue cats)*
inner = 'quartile' | 'point'
'box' | None

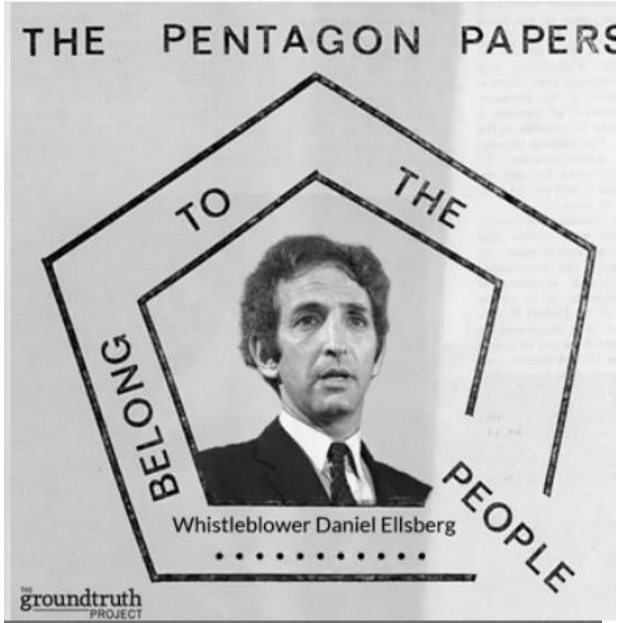
boxen(x y [hue])
(multiple quartiles / boxes, see k_depth)

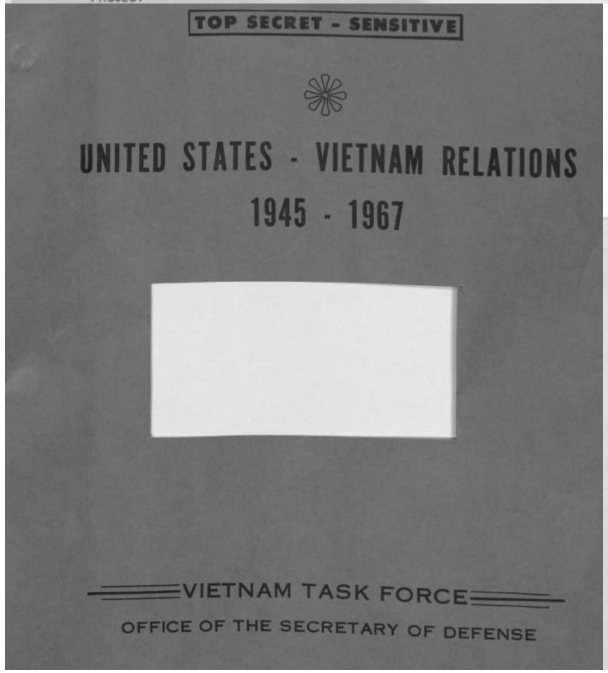
box(x y [hue])
swarm(x y [hue])
strip(x y [hue])
dodge = False

bar(x [y hue])
point(x [y hue])
style = * *(but not a col)*

single statistical
estimation of each
categories (x or y)

Projektet




TOP SECRET - SENSITIVE

UNITED STATES - VIETNAM RELATIONS
1945 - 1967

VIETNAM TASK FORCE
OFFICE OF THE SECRETARY OF DEFENSE

- https://www.archives.gov/research/pentagon-papers
- Download en pdf
  - cd Downloads
  - mkdir tmp
  - cd tmp
  - Flyt pdf’en til tmp-folderen

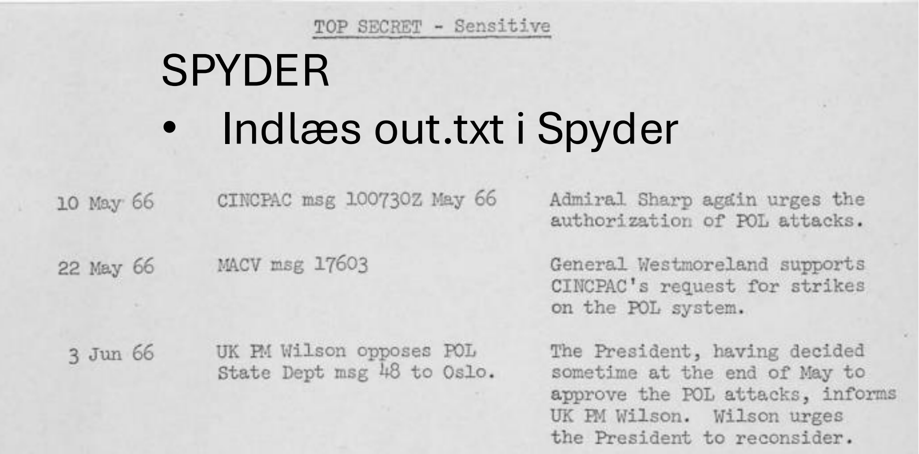
TERMINALEN

Installer poppler, imagemagic og tesseract

- pdfseparate Pentagon-Papers-Part-IV-C-7-b.pdf page-%d.pdf
- ls *pdf | while read x; do convert -density 300 $x $x.png; done
- ls *png | while read x; do tesseract $x $x; done
- LØS SORTERINGSPROBLEMET vha Python
- ls *txt | while read x; do cat $x >> out.txt; done

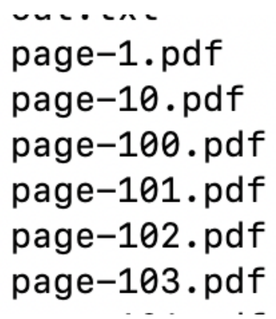
SPYDER

- Indlæs out.txt i Spyder


TOP SECRET - Sensitive

10 May 66	CINCPAC msg 100730Z May 66	Admiral Sharp again urges the authorization of POL attacks.
22 May 66	MACV msg 17603	General Westmoreland supports CINCPAC's request for strikes on the POL system.
3 Jun 66	UK PM Wilson opposes POL State Dept msg 48 to Oslo.	The President, having decided sometime at the end of May to approve the POL attacks, informs UK PM Wilson. Wilson urges the President to reconsider.

LØS SORTERINGSPROBLEMET vha Python



```
page-1.pdf
page-10.pdf
page-100.pdf
page-101.pdf
page-102.pdf
page-103.pdf
```

1. Lav en liste med filnavne i folderen med separerede *pdf-filer
2. Loop igennem listen og
    1. Identificér filer med kun ét tal og erstat med 00 (page-1 til page-001)
    2. Gør det samme for filer med to tal og erstat med 0 (page-10 til page-010)
3. Modificer trin 2 så du får en tekststreng pr filnavn på følgende form: ”mv page-1.pdf page-001.pdf”
4. Udskriv listen til en fil.

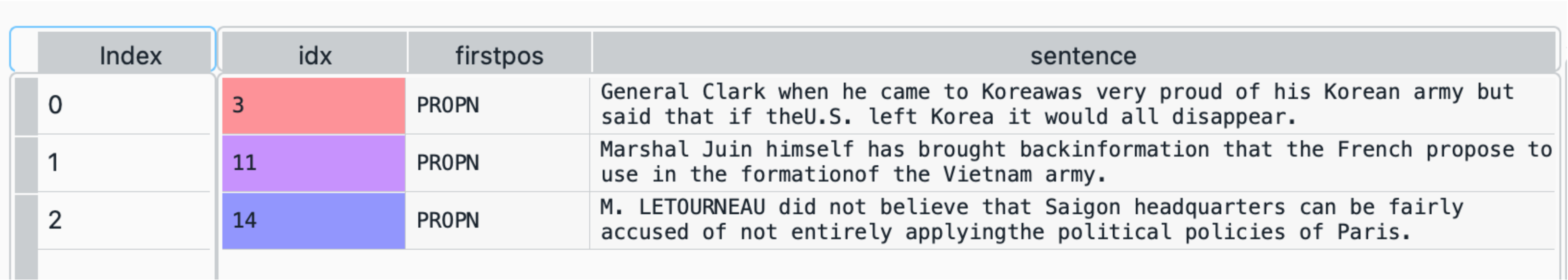
Projektet Organiseres

1. Trelloboards
    1. Inviteres én for hver gruppe
2. PentagonPapers
    1. På trello skal hver gruppe byde ind på et eller flere dokumenter
    2. Hver gruppe laver en kolonne med
        1. size,dates(liste),nltk-score,nounsfreq,personer,OE
        2. Upload til deres git-branch
    3. Hver gruppe undersøger deres tekst for hvad der skal graves efter

3. Github
    1. Branchworkflow

Projektet øvelser

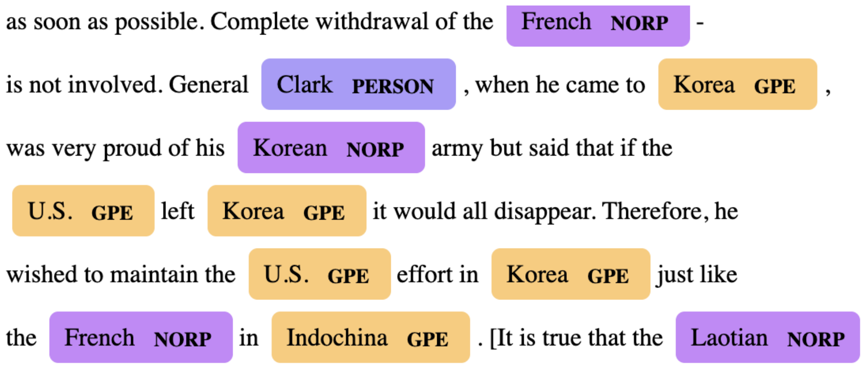
1. Subsetting på første ordklasse i sætningerne
    1. Udvælg én række og lav et doc-objekt
    2. Print alle tokens text og pos-attributter
    3. Lav en liste af tokens
    4. Lav en liste sætninger i dit doc-object og print antal ord i hver
    5. Samme som 4 men kun de sætninger der begynder med et navn
    6. Lav en dataframe som vist nedenfor (bogen side 34)



Index	idx	firstpos	sentence
0	3	PROPN	General Clark when he came to Koreawas very proud of his Korean army but said that if theU.S. left Korea it would all disappear.
1	11	PROPN	Marshal Juin himself has brought backinformation that the French propose to use in the formationof the Vietnam army.
2	14	PROPN	M. LETOURNEAU did not believe that Saigon headquarters can be fairly accused of not entirely applyingthe political policies of Paris.

Projektet øvelser

1. Named Entities
    1. Find en tilfældig række af din pp-dataframe hvor ‘size’ ligger i miderste kvartil
    2. Lav en doc ud af teksten
    3. Lav en kolonne ”dates” med lister af alle datoer i hver tekst
        1. vha ent.label_ == "DATE”
        2. Skriv din egen reg-ex så du får to kolonner: year og date
    4. Lav en visualisering af din test-text som vist nedenfor



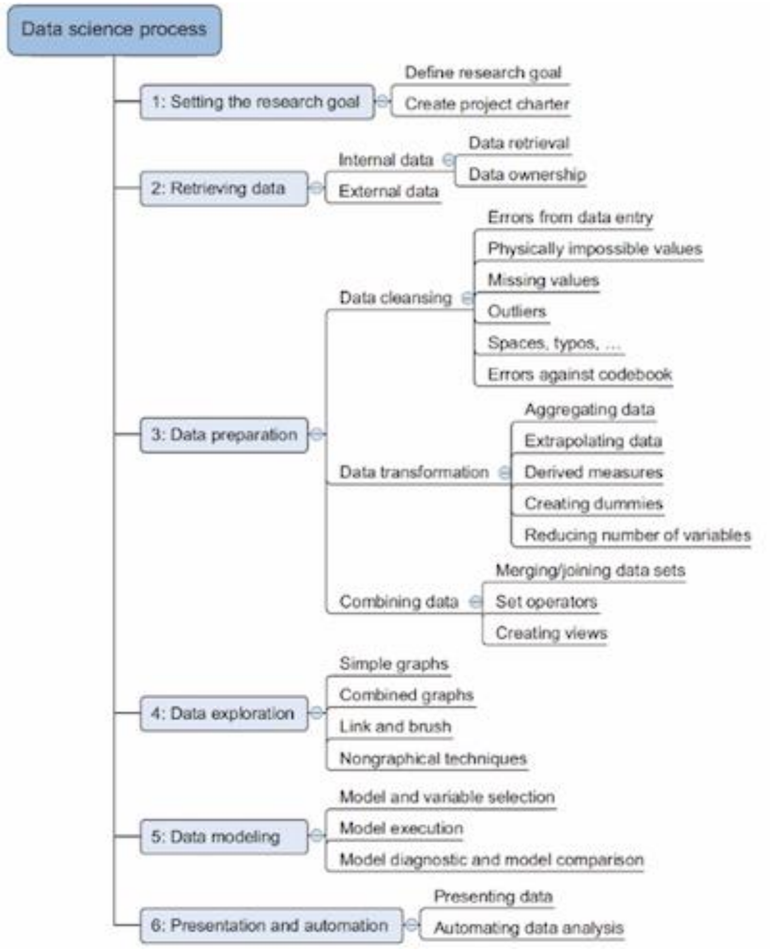
as soon as possible. Complete withdrawal of the French **NORP** -
is not involved. General Clark **PERSON** , when he came to Korea **GPE** ,
was very proud of his Korean **NORP** army but said that if the
U.S. **GPE** left Korea **GPE** it would all disappear. Therefore, he
wished to maintain the U.S. **GPE** effort in Korea **GPE** just like
the French **NORP** in Indochina **GPE** . [It is true that the Laotian **NORP**

Projektet og bogen

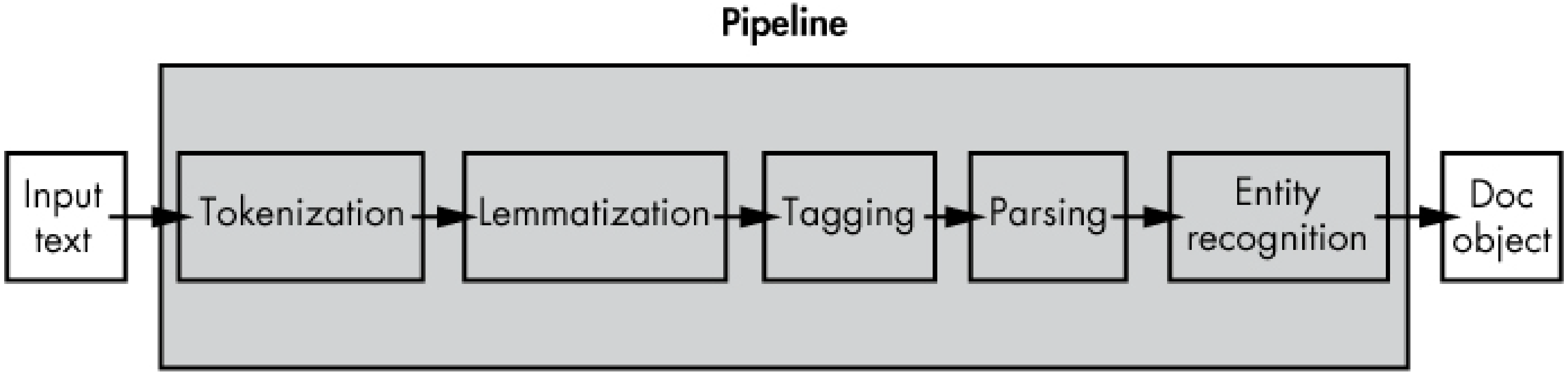
Suppose you're developing an application for processing financial reports that must extract necessary pieces of information from long, boring texts. In practice, financial reports can be quite large, but all you really need are the figures. In particular, you're interested in phrases that refer to an amount of money and start with a currency symbol. For example, your script should pick out the phrase "$1.5 million" from the previous sample sentence, but not "2017".

Projektet slutprodukt

- Research Goal fra Digital Humanities
  - Find selv på
  - Kontakt NN (det Kgl. Bibliotek, DataTeamet på Inf)
- Retrieve fra https://www.archives.gov/research/pentagon-papers
  - BS4
  - Split, convert, OCR and load into Python
  - Validate OCR
- Prepare
  - Clean data
  - Transform
    - Derive (sentiment, size)
- Explore
  - Simple viz
  - Spacy viz



NLP



Pipeline

Input text

Tokenization

Lemmatization

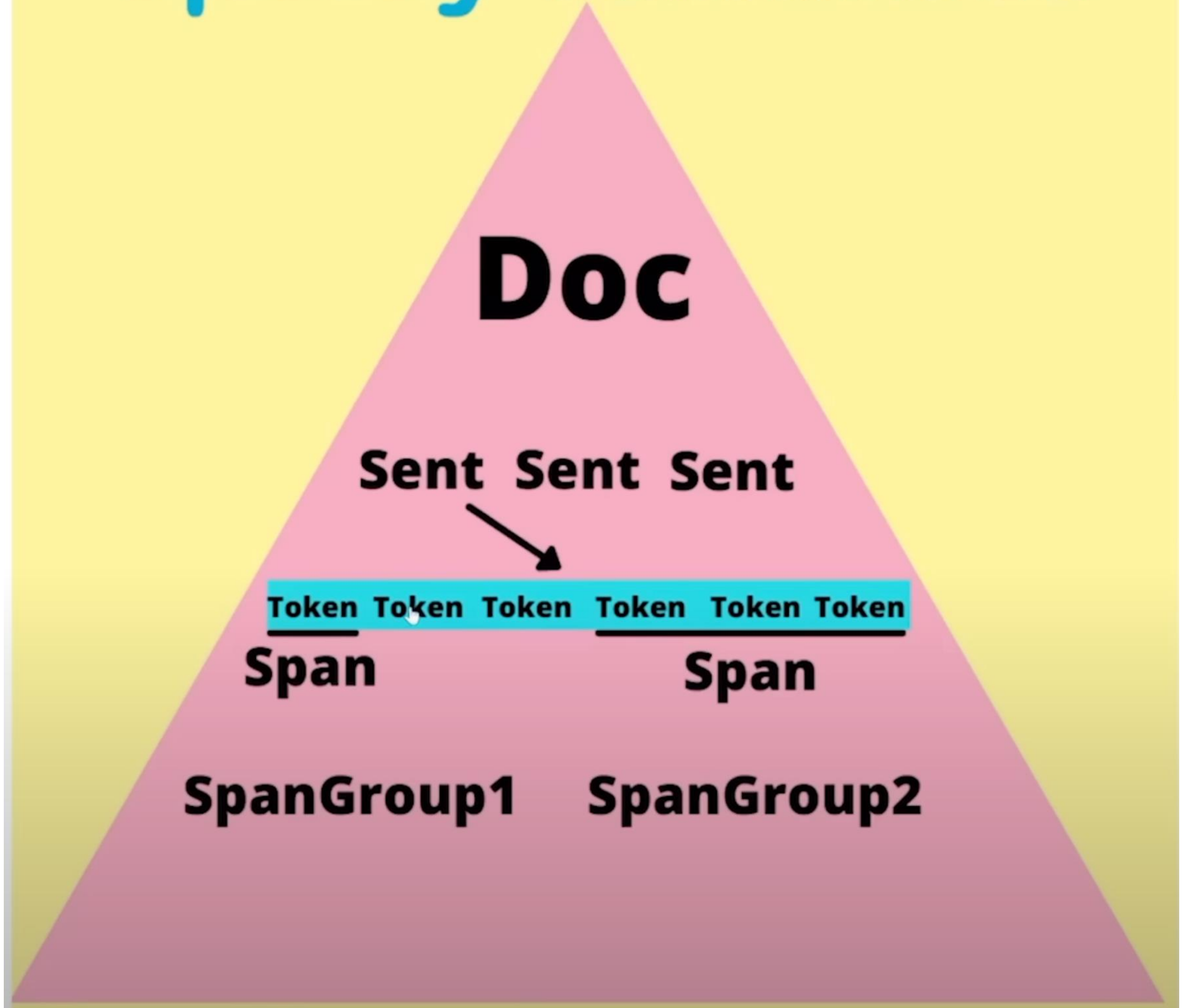
Tagging

Parsing

Entity recognition

Doc object

spaCy Containers



Doc

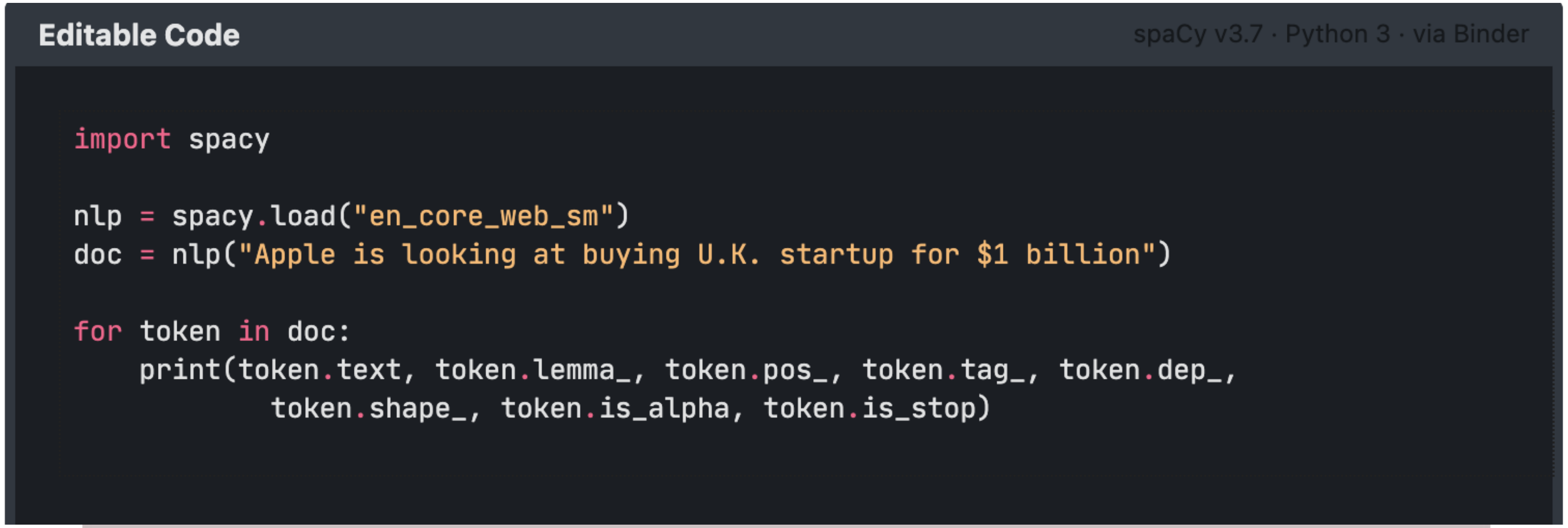
Sent Sent Sent

Token Token Token Token Token Token

Span

Span

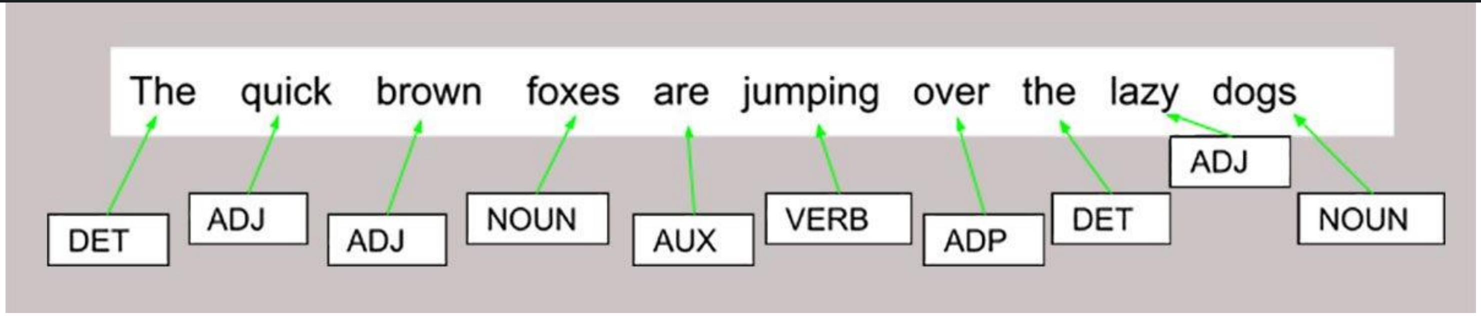
SpanGroup1 SpanGroup2

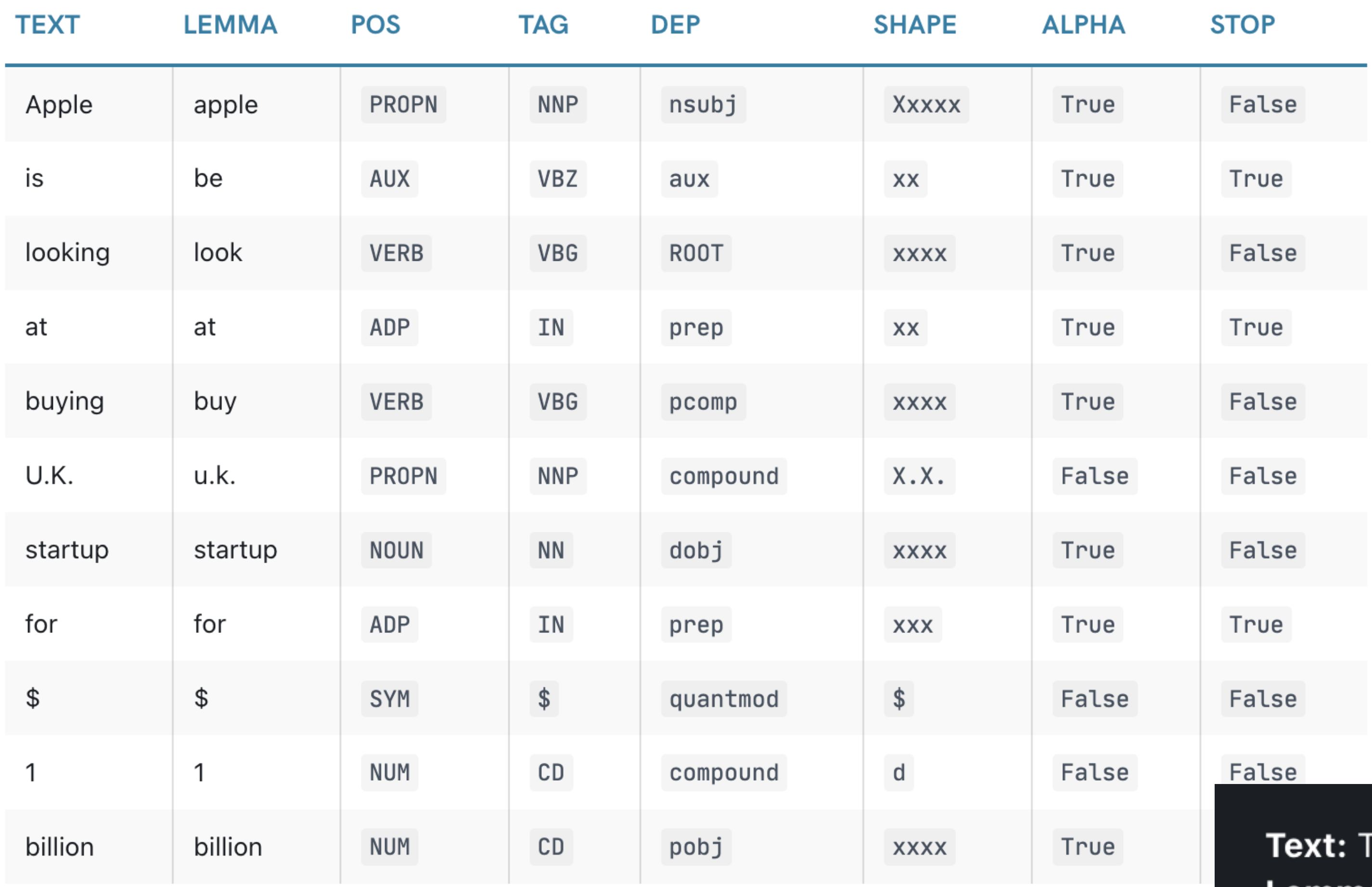


```python
import spacy

nlp = spacy.load("en_core_web_sm")
doc = nlp("Apple is looking at buying U.K. startup for $1 billion")

for token in doc:
    print(token.text, token.lemma_, token.pos_, token.tag_, token.dep_,
            token.shape_, token.is_alpha, token.is_stop)
```

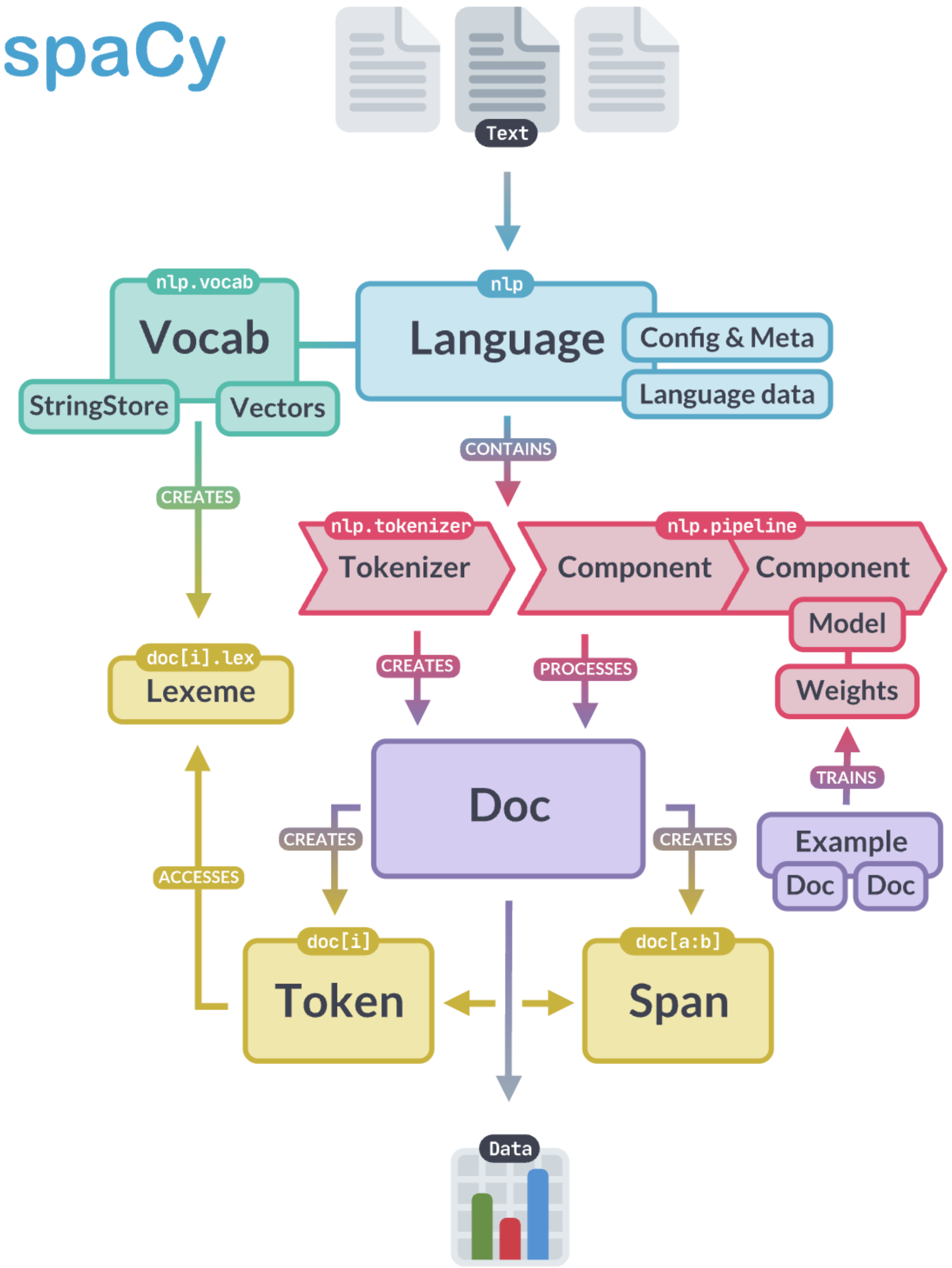




TEXT	LEMMA	POS	TAG	DEP	SHAPE	ALPHA	STOP
Apple	apple	`PROPN`	`NNP`	`nsubj`	`Xxxxx`	`True`	`False`
is	be	`AUX`	`VBZ`	`aux`	`xx`	`True`	`True`
looking	look	`VERB`	`VBG`	`ROOT`	`xxxx`	`True`	`False`
at	at	`ADP`	`IN`	`prep`	`xx`	`True`	`True`
buying	buy	`VERB`	`VBG`	`pcomp`	`xxxx`	`True`	`False`
U.K.	u.k.	`PROPN`	`NNP`	`compound`	`X.X.`	`False`	`False`
startup	startup	`NOUN`	`NN`	`dobj`	`xxxx`	`True`	`False`
for	for	`ADP`	`IN`	`prep`	`xxx`	`True`	`True`
$	$	`SYM`	`$`	`quantmod`	`$`	`False`	`False`
1	1	`NUM`	`CD`	`compound`	`d`	`False`	`False`
billion	billion	`NUM`	`CD`	`pobj`	`xxxx`	`True`	

Text: The original word text.
Lemma: The base form of the word.
POS: The simple UPOS part-of-speech tag.
Tag: The detailed part-of-speech tag.
Dep: Syntactic dependency, i.e. the relation between tokens.
Shape: The word shape – capitalization, punctuation, digits.
is alpha: Is the token an alpha character?
is stop: Is the token part of a stop list, i.e. the most common words of the language?

spaCy



Text

nlp.vocab

Vocab

StringStore

Vectors

nlp

Language

Config & Meta

Language data

CONTAINS

CREATES

nlp.tokenizer

Tokenizer

nlp.pipeline

Component

Component

Model

Weights

doc[i].lex

Lexeme

CREATES

PROCESSES

TRAINS

Doc

Example

Doc

Doc

ACCESSES

CREATES

CREATES

doc[i]

Token

doc[a:b]

Span

Data

The Book

- BASIC NLP OPERATIONS WITH SPACY
  - Tokenization
  - Lemmatization
    - Applying Lemmatization for Meaning Recognition
  - Part-of-Speech Tagging
    - Find Relevant Verbs
  - Syntactic Relations
    - for sent in doc.sents:print([w.text for w in sent if w.dep_ == 'ROOT' or w.dep_ == 'pobj’])
  - Named Entity Recognition
- SPACY’S CONTAINER OBJECTS (Token, Span, and Doc)
  - Doc-container
    - Syntactic Children
    - doc.sents - for sent in doc.sents
    - doc.noun_chunks - for chunk in doc.noun_chunks
    - Span object - a slice from a Doc object
  - Pipeline Components
    - Disabling Pipeline Components
    - Customizing the Pipeline Components

- EXTRACTING AND USING LINGUISTIC FEATURES

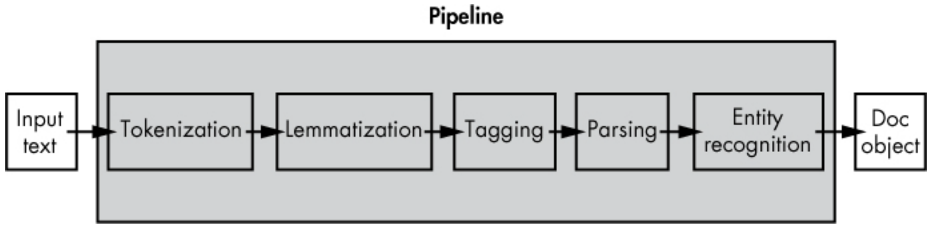
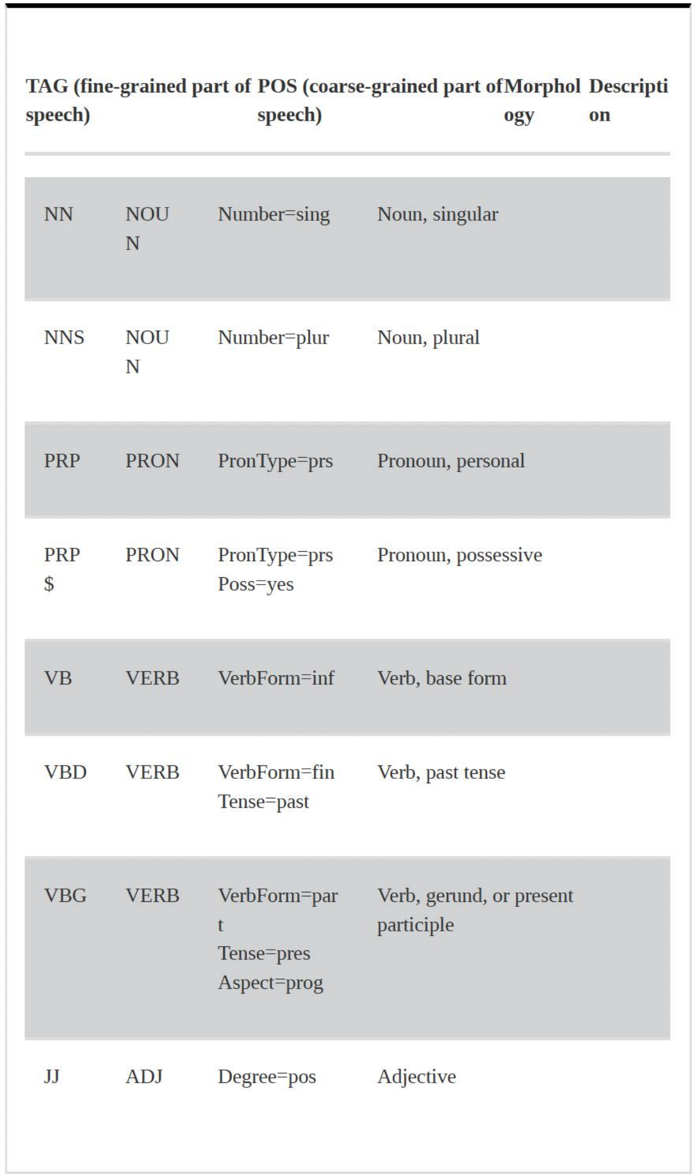


Table 2-1: Some Common spaCy Part-of-Speech Tags



TAG (fine-grained part of speech)	POS (coarse-grained part of speech)	Morphology	Description
NN	NOUN	Number=sing	Noun, singular
NNS	NOUN	Number=plur	Noun, plural
PRP	PRON	PronType=prs	Pronoun, personal

PRP$	PRON	PronType=prs
		Poss=yes
			Pronoun, possessive

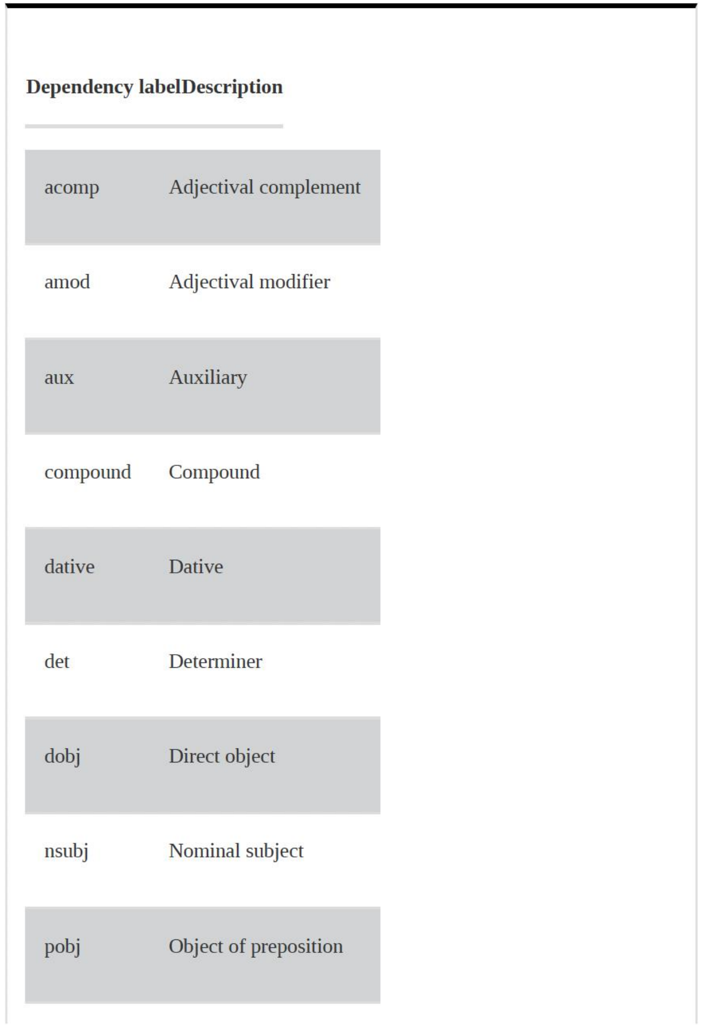
VB	VERB	VerbForm=inf	Verb, base form

VBD	VERB	VerbForm=fin
		Tense=past
			Verb, past tense

VBG	VERB	VerbForm=part
		Tense=pres
		Aspect=prog
			Verb, gerund, or present participle

JJ	ADJ	Degree=pos	Adjective

Table 2-2: Some Common Dependency Labels



Dependency label	Description
acomp	Adjectival complement
amod	Adjectival modifier
aux	Auxiliary
compound	Compound
dative	Dative
det	Determiner
dobj	Direct object
nsubj	Nominal subject
pobj	Object of preposition