BI 前的悲哀

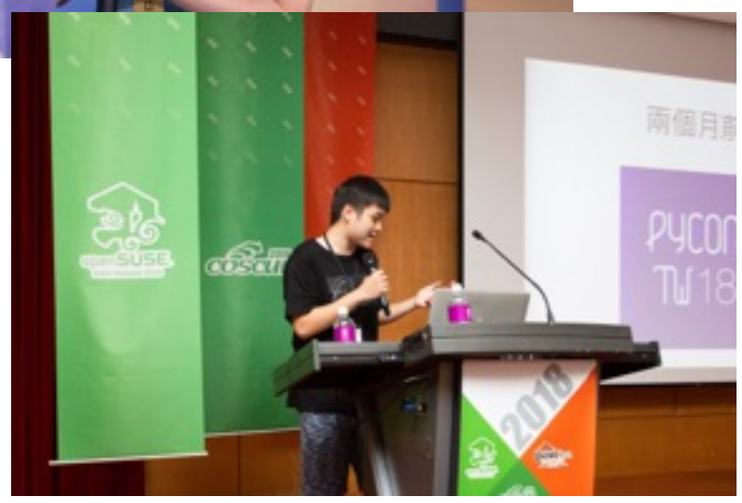
- 那些User所謂的**Bi**g(?) data

The Sadness before **B**usiness **I**ntelligence:
When User says **Bi**g data…

Bingroom

About Me

1. **Inspired** by PyCon



2. Got **donated** in PyCon

3. **Contribute** to PyCon

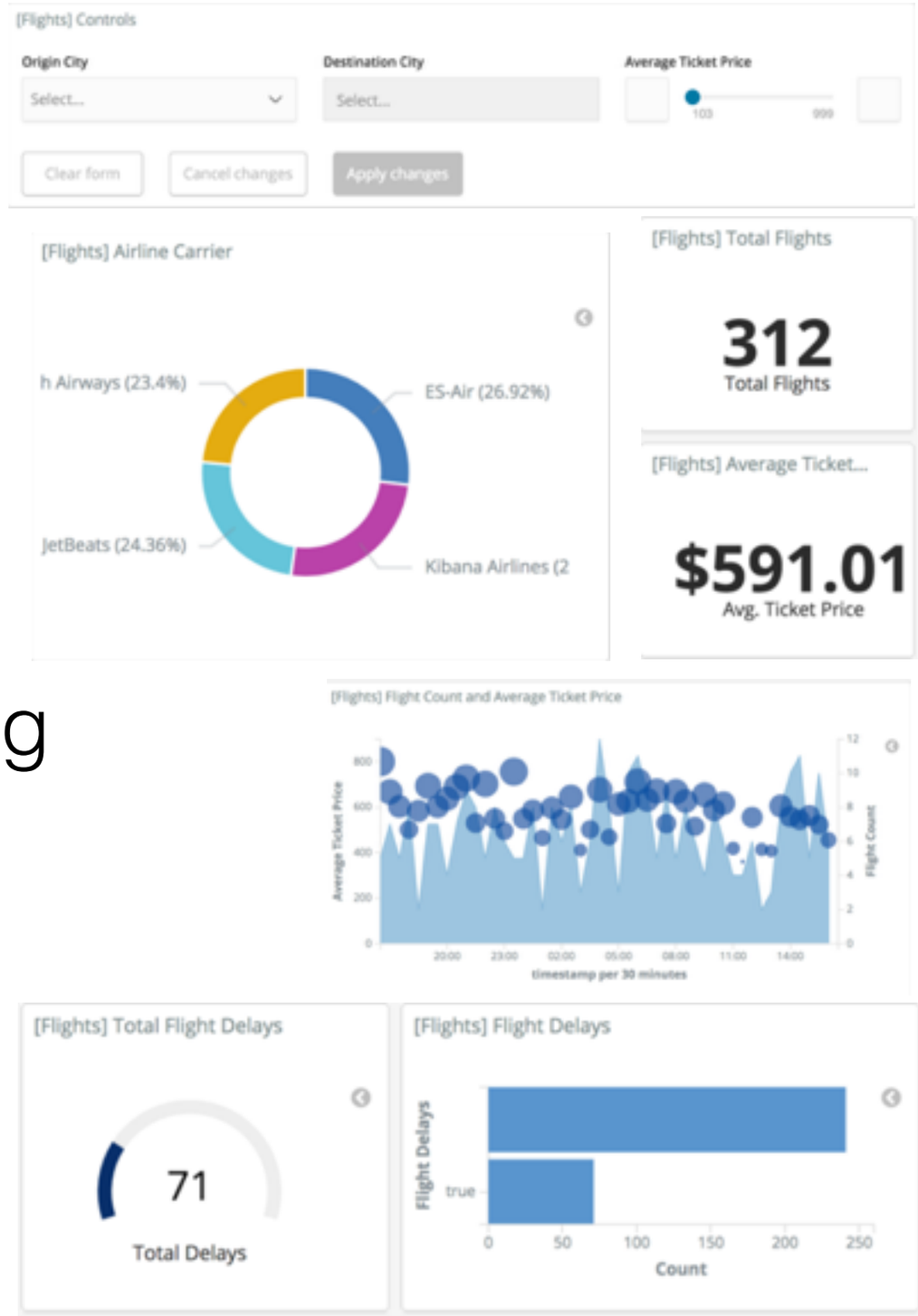
PyCon TW 19

Outline

- BI
- ETL
- Data parser
  - Common document file extensions
  - Software for developers
- Case sharing

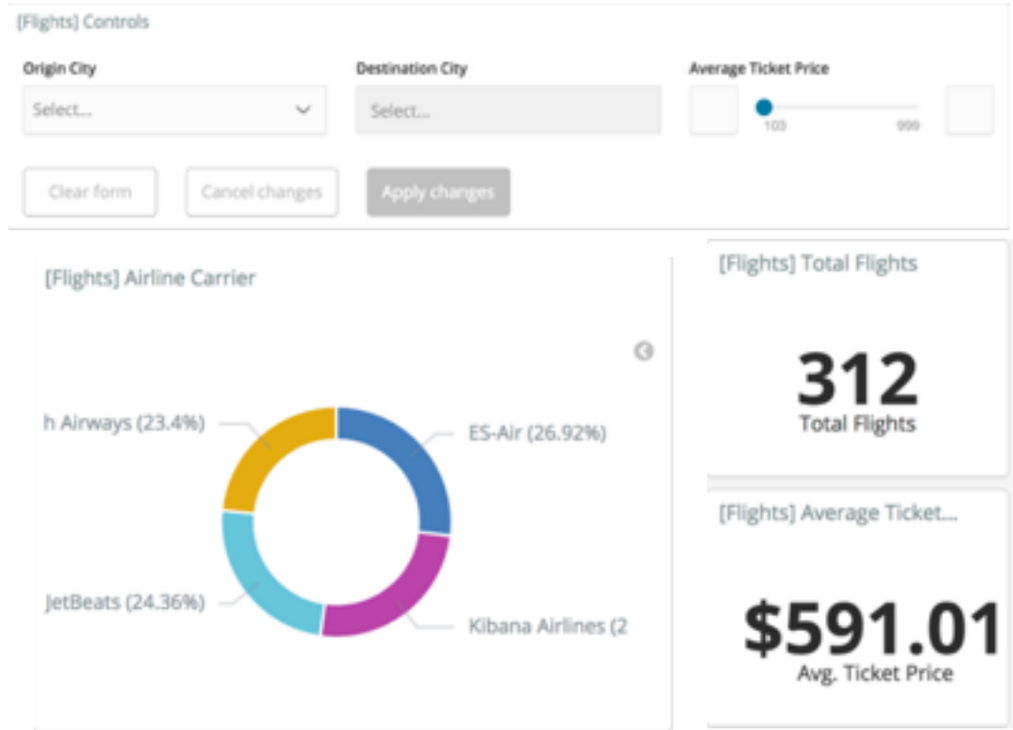
Business Intelligence

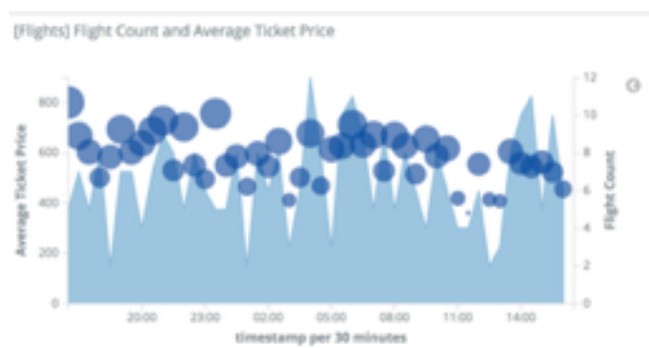
- Know business
- Decision making
- Real-time navigation & monitoring
- Make things easier & sharable

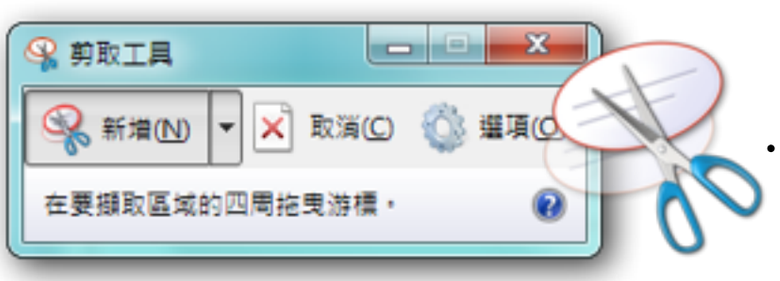


Business Intelligence

- Know business
- Decision making
- Real-time navigation & monitoring
- Make things easier & **sharable**

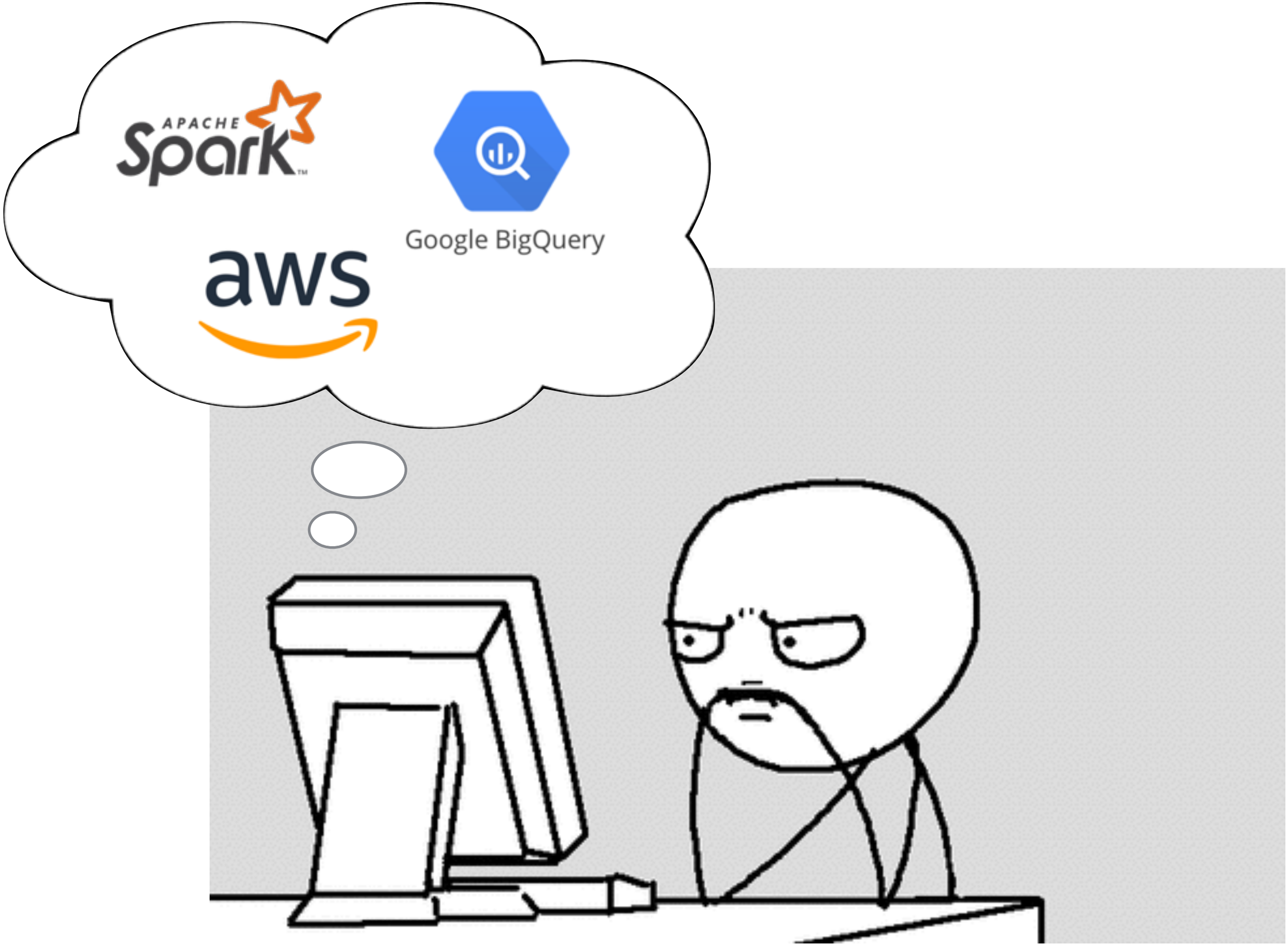






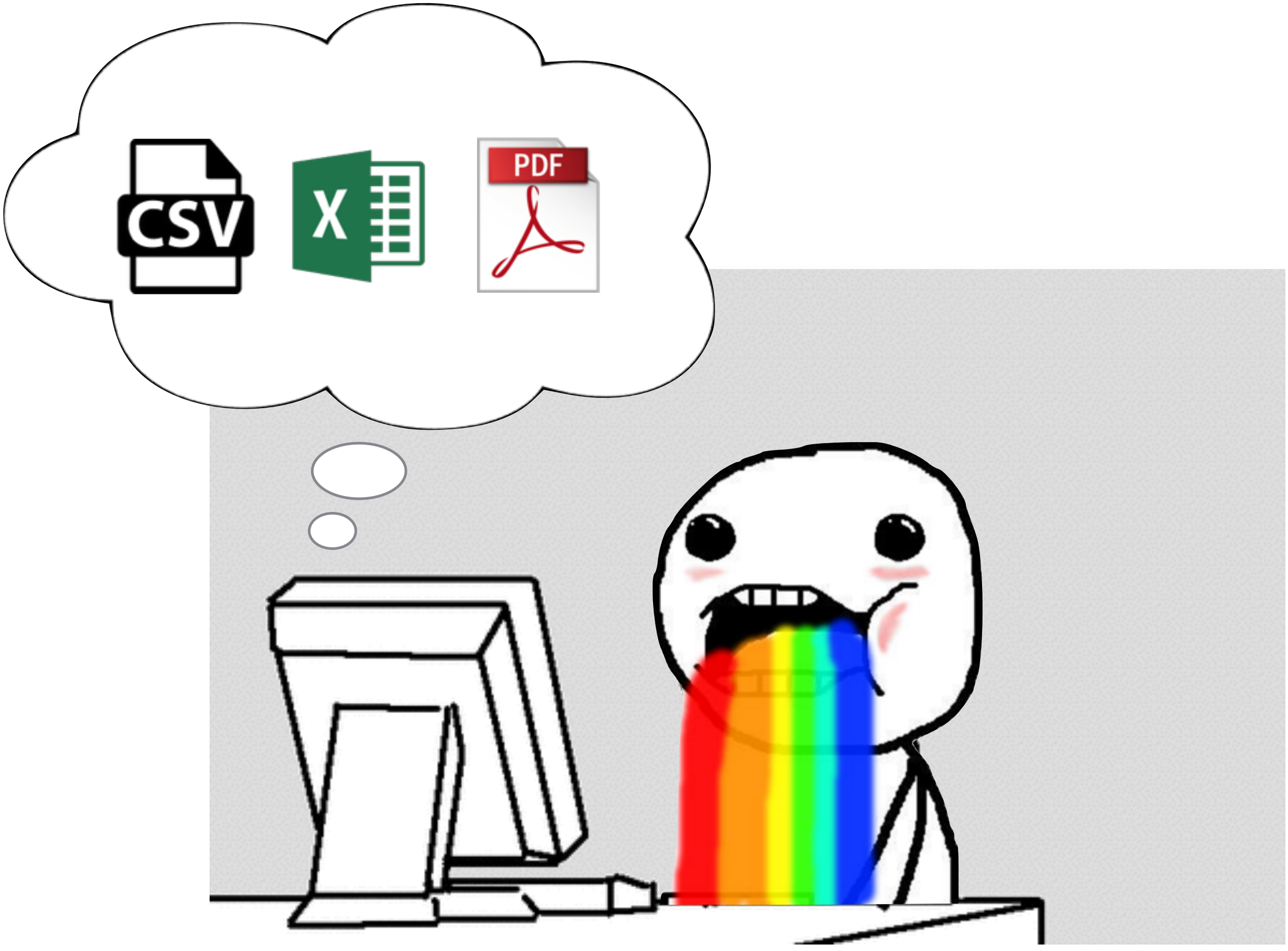
Before the Fall

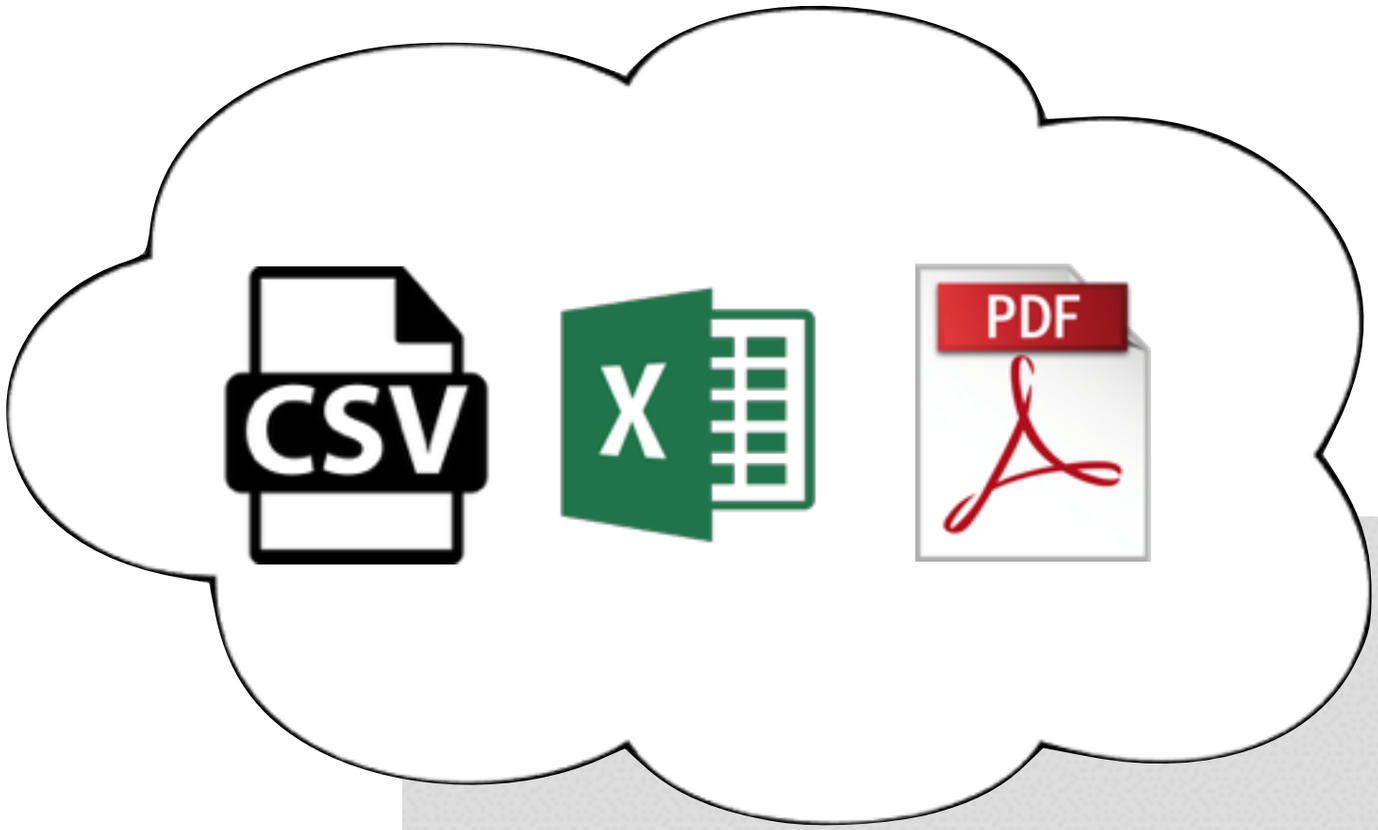
As a Data Engineer


APACHE
Spark
Google BigQuery
aws

THREE

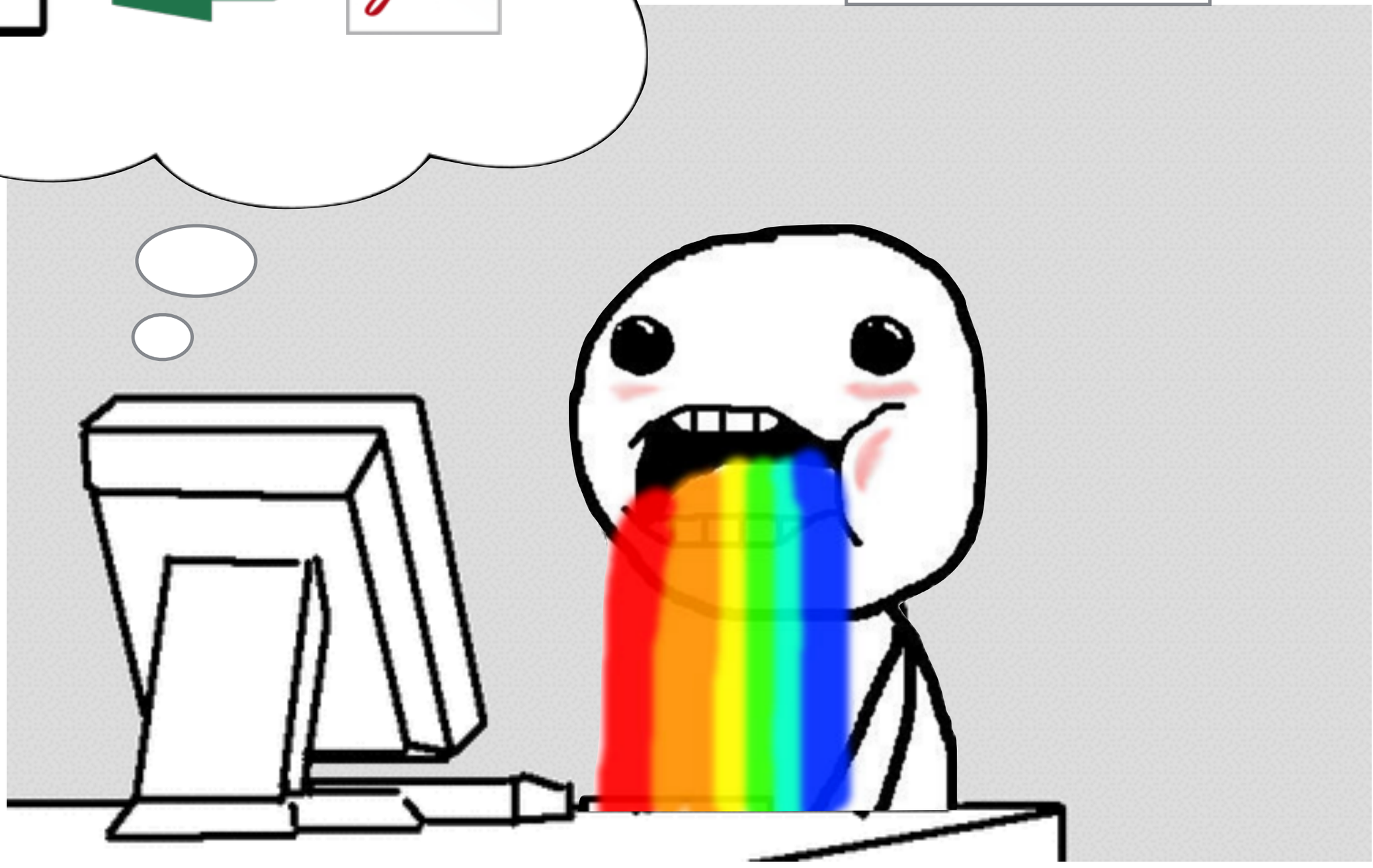
WEEKS LATER

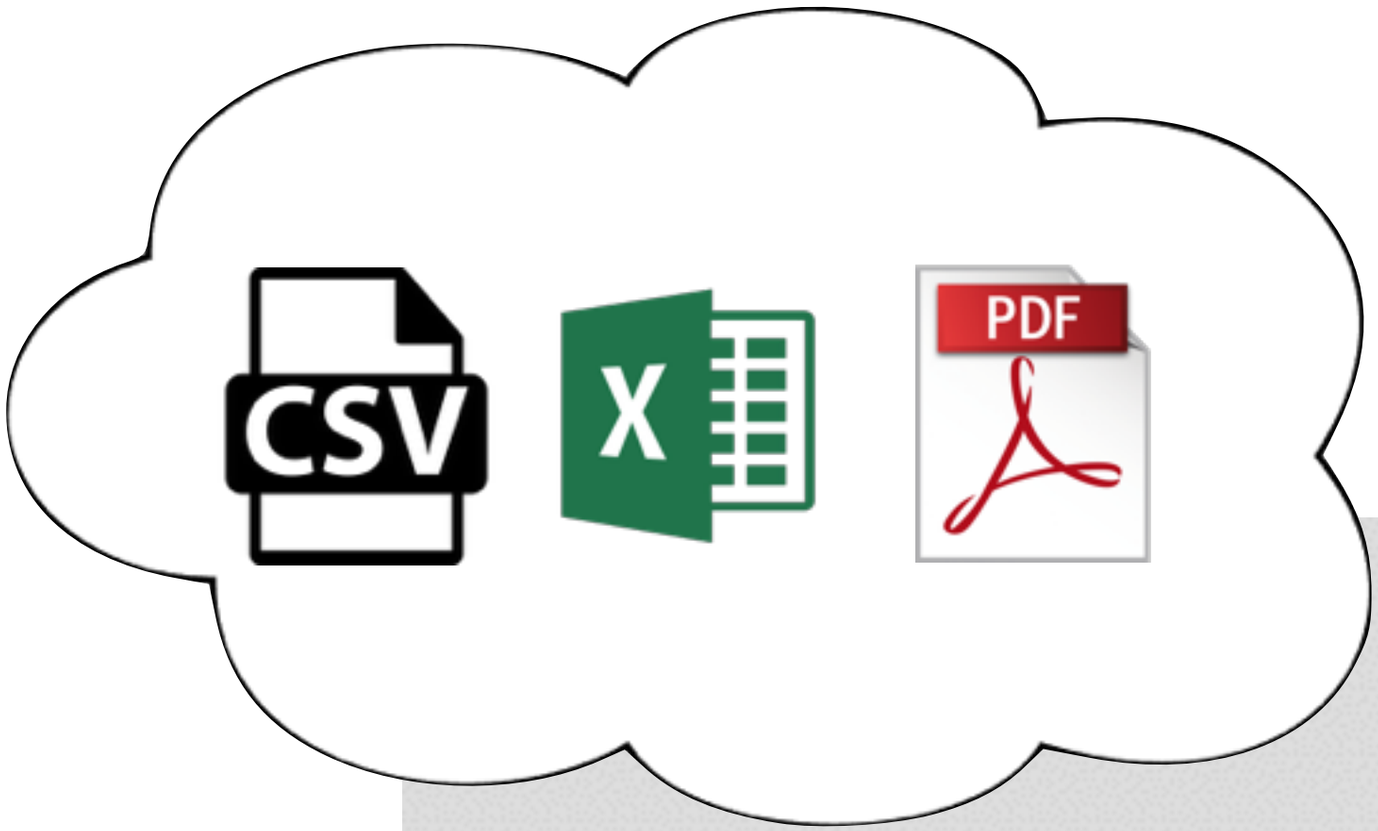

CSV
X
PDF



Big Data:

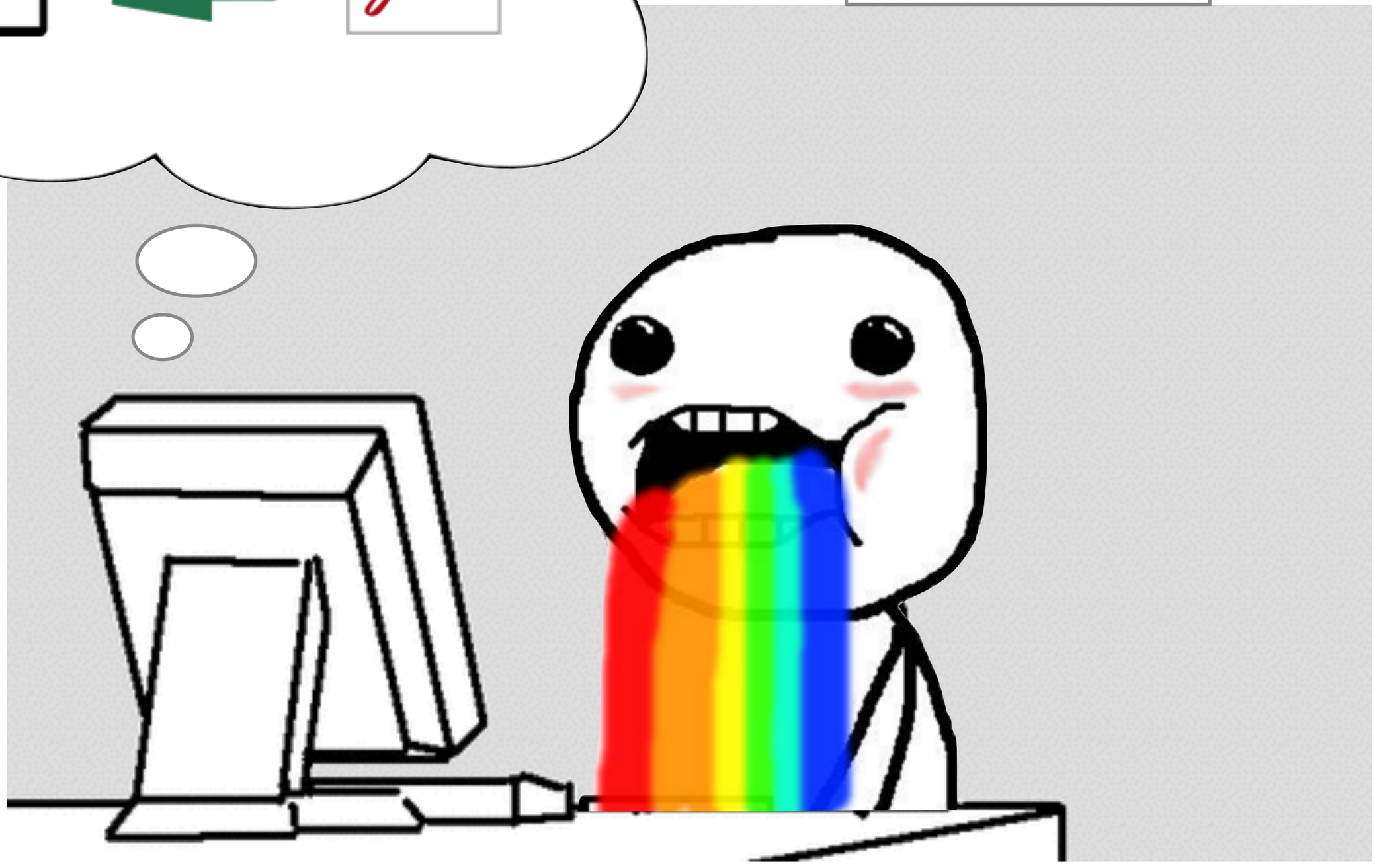
1. Volume
2. Variety
3. Velocity





Big(**?**) Data:

1. ~~Volume~~
2. Variety
3. Velocity



Extract-**T**ransform-**L**oad

Extract

- Load DB tables
- Read CSV files
- Web crawlers
- import requests

Extract-**T**ransform-**L**oad

Extract

- Load DB tables
- Read CSV files
- Web crawlers
- import requests

Transform

- Data clean
  - ✖ incomplete
  - ✖ inaccurate
  - ✖ inrrelevant
- Data transform
  - ✓ re-index
  - ✓ aggregation
  - ✓ transpose

Extract-**T**ransform-**L**oad

Extract

- Load DB tables
- Read CSV files
- Web crawlers
- import requests

Transform

- Data clean
  - ✖ incomplete
  - ✖ inaccurate
  - ✖ inrrelevant
- Data transform
  - ✓ re-index
  - ✓ aggregation
  - ✓ transpose

Load

- Data warehouse
- SQL / NoSQL
- from sqlalchemy import *

Extract-**T**ransform-**L**oad

Extract

- Load DB tables
- Read CSV files
- Web crawlers
- import requests

Transform

- Data clean
  - ✗ incomplete
  - ✗ inaccurate
  - ✗ inrrelevant
- Data transform
  - ✓ re-index
  - ✓ aggregation
  - ✓ transpose

Load

- Data warehouse
- SQL / NoSQL
- from sqlalchemy import *


精緻の假牙
大[redacted]子
享受一杯夏日芒果微醺
愛文芒果冰沙
SUMMER TRAVEL
foodpanda
DaYung

Data Transform (Preprocessing)

Don’t reinvent the wheel!

A short recap…

Bingroom，迷途中的一個Python小書僮，由於之前Crawler寫太多，現在都在寫Parser還債。

Data Parser

Excel

- Similar to CSV - Lots of Python modules
- For Windows user: **win32com**
- Read only: **xlrd**
- Read & Write: **openpyxl**
- Scientific computing: **pandas**

maybe **xlwings**
for VBA lover?

Excel

Compatibility

	Win	Mac	Py2	Py3	xlsx	安装
win32com	✓	✗	✓	✓	✓	pip
xlwings	✓	✓	✓	✓	✓	pip
xlsxwriter	✓	✓	✓	✓	✓	pip
DataNitro	✓	✗	✓	✓	✓	安装包
pandas	✓	✓	✓	✓	✓	pip
openpyxl	✓	✓	✓	✓	✓	pip
xlutils	✓	✓	✓	✓	✗	pip

Task

	打开文档	新建文档	修改文档	保存文档
win32com	✓	✓	✓	✓
xlwings	✓	✓	✓	✓
xlsxwriter	✗	✓	✗	✓
pandas	✓	✗	✓	✓
openpyxl	✓	✓	✓	✓
xlutils	✓	✓	✓	✓

Efficiency

	写入用时(s)	读取用时(s)
win32com	3	2
xlwings	5	1.5
xlsxwriter	14	
DataNitro	21	
pandas	22	12
openpyxl	32	17
xlutils		11

https://zhuanlan.zhihu.com/p/23998083

Excel
(In Practice)

- Background color of cell

Account	Region	2018 Jan	2018 Feb
PyCon TW	TW	0	0
PyCon JP	JP	0	1

```
from openpyxl import load_workbook
wb = load_workbook('sample.xlsx')
ws = wb['工作表1']
cell_A5 = ws['A5'].fill.start_color.index[2:]
cell_C5 = ws['C5'].fill.start_color.index[2:]
cell_A5, cell_C5
```

```
('FFC000', '66FF66')
```

#FFC000

#66FF66

Excel

(In Practice)

- Transpose

Excel

(In Practice)

- Transpose

  - Take apart df.columns
    => fixed_cols and scan_cols

  - Fixed: keep value and repeat

  - Scan: get value by key
    then append to row

fixed_cols **scan_cols**

excel_df

	Account	Region	2018-01	2018-02	2018-03	2018-04	2018-05
0	PyCon TW	TW	0	0	0	10	11
1	PyCon JP	JP	0	1	0	3	0

(Repeat) (key) (value)

	Account	Region	Date	Commit
0	PyCon TW	TW	2018-01	0
1	PyCon JP	JP	2018-01	0
2	PyCon TW	TW	2018-02	0
3	PyCon JP	JP	2018-02	1
4	PyCon TW	TW	2018-03	0
5	PyCon JP	JP	2018-03	0
6	PyCon TW	TW	2018-04	10
7	PyCon JP	JP	2018-04	3
8	PyCon TW	TW	2018-05	11

Excel
(In Practice)

- Transpose
  - pandas.melt()


```
df = pd.melt(excel_df, id_vars=['Account', 'Region'], var_name='Date', value_name='Commit')
```

	Account	Region	Date	Commit
0	PyCon TW	TW	2018-01	0
1	PyCon JP	JP	2018-01	0
2	PyCon TW	TW	2018-02	0
3	PyCon JP	JP	2018-02	1

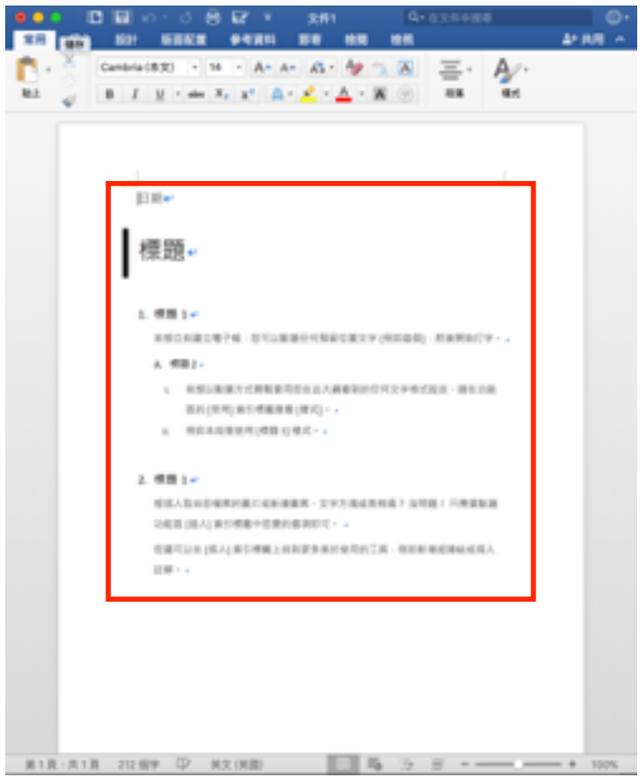
https://pandas.pydata.org/pandas-docs/version/0.23.4/generated/pandas.melt.html

Excel

(In Practice)

I just reinvented a wheel!

Word

- Before talking about 
- Let's extract 

first (or **text**, precisely)

Text Extraction

- On Linux: **libreoffice**

```
libreoffice --headless --convert-to "txt:Text (encoded):UTF8" mydocument.doc
```

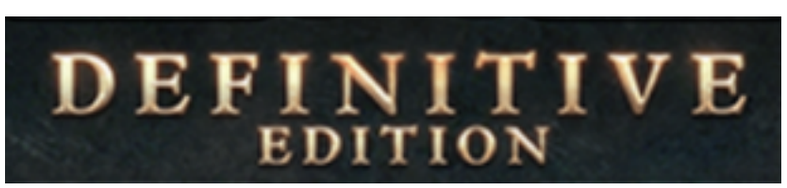
https://ask.libreoffice.org/en/question/2641/convert-to-command-line-parameter/

- Cons:
  - High memory usage in batch processing
  - No root, no apt-get

Text Extraction

- **textract**: all-mighty text extractor
- textract is not tesseract and tesseract in textract

```
text = textract.process(
    'path/to/norwegian.pdf',
    method='tesseract',
    language='nor',
)
```

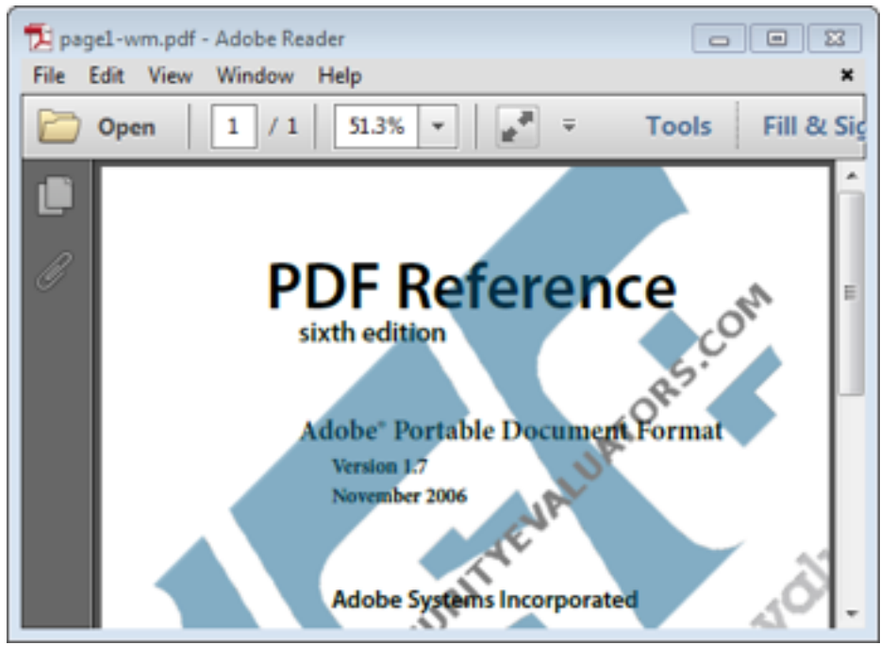


Currently supporting

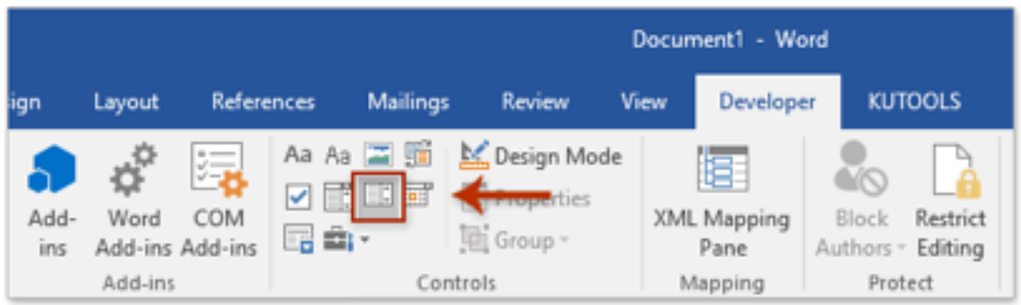
- `.csv` via python builtins
- `.doc` via antiword
- `.docx` via python-docx2txt
- `.eml` via python builtins
- `.epub` via ebooklib
- `.gif` via tesseract-ocr
- `.jpg` and `.jpeg` via tesseract-ocr
- `.json` via python builtins
- `.html` and `.htm` via beautifulsoup4
- `.mp3` via sox, SpeechRecognition, and pocketsphinx
- `.msg` via msg-extractor
- `.odt` via python builtins
- `.ogg` via sox, SpeechRecognition, and pocketsphinx
- `.pdf` via pdftotext (default) or pdfminer.six
- `.png` via tesseract-ocr
- `.pptx` via python-pptx
- `.ps` via ps2text
- `.rtf` via unrtf
- `.tiff` and `.tif` via tesseract-ocr
- `.txt` via python builtins
- `.wav` via SpeechRecognition and pocketsphinx
- `.xlsx` via xlrd
- `.xls` via xlrd

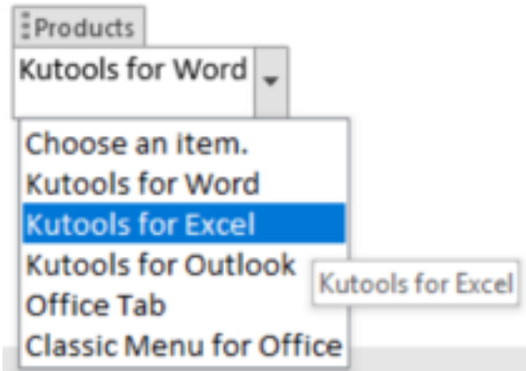
Text Extraction

- Challenge 1: Digital watermarking



- Challenge 2: Develop controls





Word

(In Practice)

- Text extraction

- Table extraction

Table Extraction

(Basic)

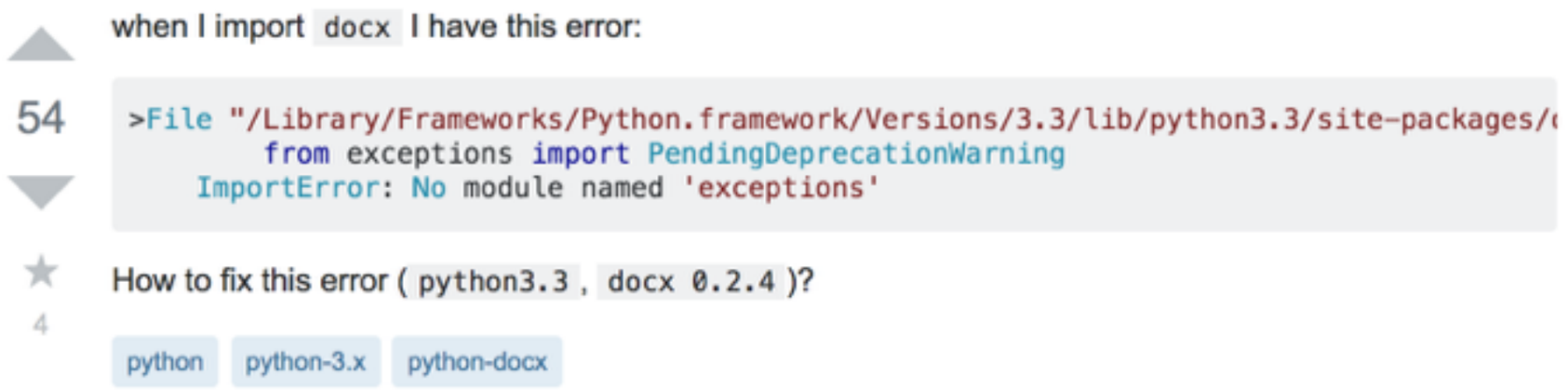
- import python-docx

If you are using python 3x don't do `pip install docx` instead go for

```
pip install python-docx
```

it is compatible with python 3x

official Document: https://pypi.org/project/python-docx/



when I import `docx` I have this error:

```
>File "/Library/Frameworks/Python.framework/Versions/3.3/lib/python3.3/site-packages/
        from exceptions import PendingDeprecationWarning
    ImportError: No module named 'exceptions'
```

How to fix this error (`python3.3` , `docx 0.2.4`)?

54

4

python python-3.x python-docx

Table Extraction

(Basic)

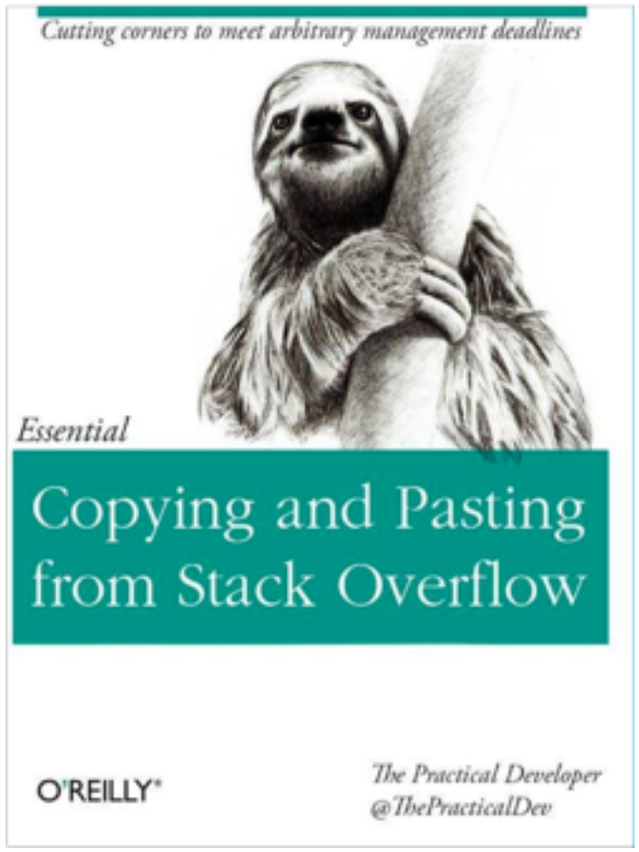
- import python-docx
- Parse **XML**

Result : Beautify XML

```
23                  <w:widowControl w:val="1"/>
24                  <w:tabs>
25                      <w:tab w:val="left" w:pos="9214"/>
26                      <w:tab w:val="left" w:pos="9639"/>
27                  </w:tabs>
28                  <w:spacing w:after="0" w:before="0" w:line="240"
                        w:lineRule="auto"/>
29                  <w:ind w:right="107"/>
30                      <w:contextualSpacing w:val="0"/>
31                      <w:jc w:val="right"/>
32                  </w:pPr>
33                  <w:r w:rsidDel="00000000" w:rsidR="00000000" w
                        :rsidRPr="00000000">
34                      <w:rPr>
35                          <w:rFonts w:ascii="Arial Unicode MS" w:cs
                                ="Arial Unicode MS" w:eastAsia="Arial
                                Unicode MS" w:hAnsi="Arial Unicode MS"/>
36                          <w:b w:val="1"/>
37                          <w:sz w:val="24"/>
38                          <w:szCs w:val="24"/>
39                          <w:vertAlign w:val="baseline"/>
40                          <w:rtl w:val="0"/>
41                      </w:rPr>
42                      <w:t xml:space="preserve">【機密】</w:t>
43                  </w:r>
44                  <w:r w:rsidDel="00000000" w:rsidR="00000000" w
                        :rsidRPr="00000000">
45                      <w:rPr>
46                          <w:rtl w:val="0"/>
47                      </w:rPr>
48                  </w:r>
49              </w:p>
```

Table Extraction

(Medium)



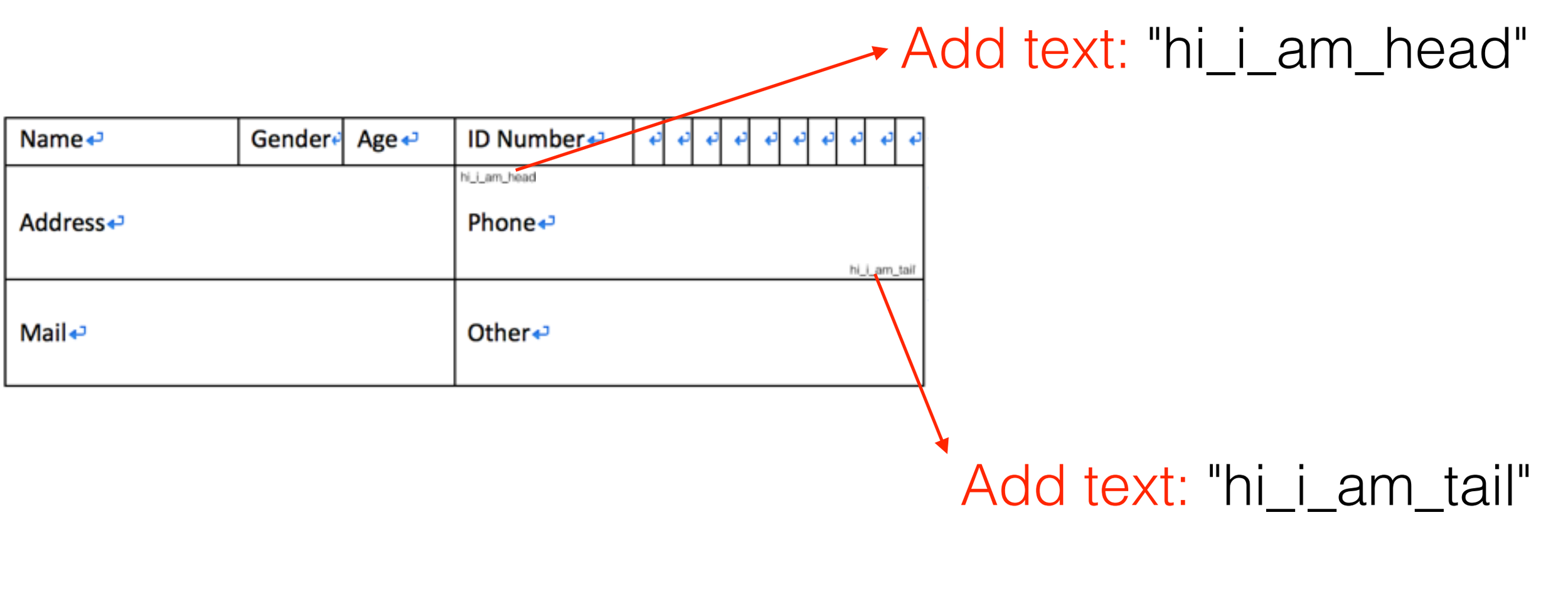
$ INPUT | some scripts from Github | perfect.csv

https://github.com/ivbeg/docx2csv

Table Extraction

(Advanced)

- Add special text as **quotation** marks (" ") in target cell



PDF

AFTERSCHOOL TRAINING TOOLKIT

Tutoring to Enhance Science Skills

Tutoring Two: Learning to Make Data Tables

Sample Data for Data Tables

Use these data to create data tables following the Guidelines for Making a Data Table and Checklist for a Data Table.

Example 1: Pet Survey (GR 2–3)
Ms. Hubert's afterschool students took a survey of the 600 students at Morales Elementary School. Students were asked to select their favorite pet from a list of eight animals. Here are the results.

Lizard 25, Dog 250, Cat 115, Bird 50, Guinea pig 30, Hamster 45, Fish 75, Ferret 10

Example 2: Electromagnets—Increasing Coils (GR 3–5)
The following data were collected using an electromagnet with a 1.5 volt battery, a switch, a piece of #20 insulated wire, and a nail. Three trials were run. *Safety precautions in repeating this experiment include using safety goggles or safety spectacles and avoiding short circuits.*

Number of Coils	Number of Paperclips
5	3, 5, 4
10	7, 8, 6
15	11, 10, 12
20	15, 13, 14

Example 3: pH of Substances (GR 5–10)
The following are pH values of common household substances taken by three different teams using pH probes. *Safety precautions in repeating this experiment include hooded ventilation, chemical-splash safety goggles, gloves, and apron. Do not use bleach, ammonia, or strong acids with children.*

Lemon juice 2.4, 2.0, 2.2; Baking soda (1 Tbsp) in Water (1 cup) 8.4, 8.3, 8.7; Orange juice 3.5, 4.0, 3.4; Battery acid 1.0, 0.7, 0.5; Apples 3.0, 3.2, 3.5; Tomatoes 4.5, 4.2, 4.0; Bottled water 6.7, 7.0, 7.2; Milk of magnesia 10.5, 10.3, 10.6; Liquid hand soap 9.0, 10.0, 9.5; Vinegar 2.2, 2.9, 3.0; Household bleach 12.5, 12.5, 12.7; Milk 6.6, 6.5, 6.4; Household ammonia 11.5, 11.0, 11.5; Lye 13.0, 13.5, 13.4; and Sodium hydroxide 14.0, 14.0, 13.9; Anti-freeze 10.1, 10.9, 9.7; Windex 9.9. 10.2, 9.5; Liquid detergent 10.5, 10.0, 10.3; and Cola 3.0, 2.5, 3.2

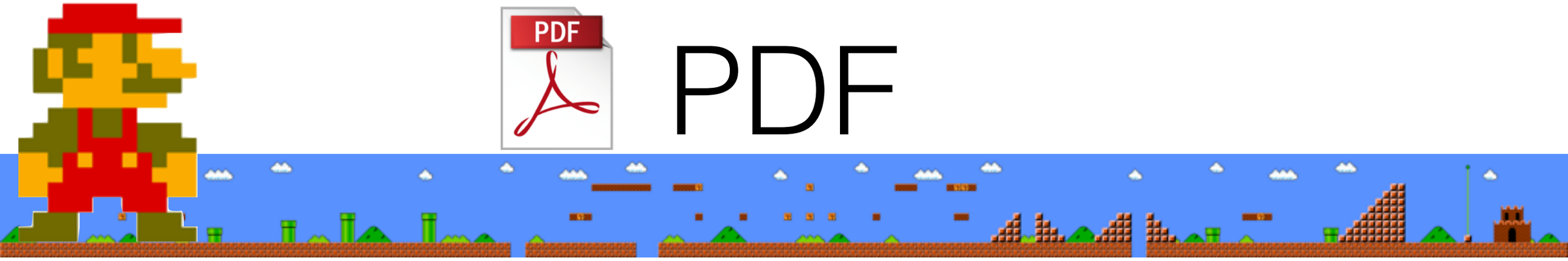
Teaching tip: *The pH scale is from 0 to 14. Have students make two data tables, one with the data as given and one with the pH scale 0 to 14 with the substances' average pH in rank order on the scale (Battery acid at the lower end and Sodium hydroxide at the upper end) or create a pH graphic organizer.*

Example 4: Automobile Land Speed Records (GR 5-10)
In the first recorded automobile race in 1898, Count Gaston de Chasseloup-Laubat of Paris, France, drove 1 kilometer in 57 seconds for an average speed of 39.2 miles per hour (mph) or 63.1 kilometers per hour (kph). In 1904, Henry Ford drove his Ford Arrow across frozen Lake St. Clair, MI, at an average speed of 91.4 mph. Now, the North American Eagle is trying to break a land speed record of 800 mph. The Federation International de L'Automobile (FIA), the world's governing body for motor sport and land speed records, recorded the following land speed records. (Retrieved on February 5, 2006, from http://www.landspeed.com/lsrinfo.asp.)

Speed (mph)	Driver	Car	Engine	Date
407.447	Craig Breedlove	Spirit of America	GE J47	8/5/63
413.199	Tom Green	Wingfoot Express	WE 346	10/2/64
434.22	Art Arfons	Green Monster	GE J79	10/5/64
468.719	Craig Breedlove	Spirit of America	GE J79	10/13/64
526.277	Craig Breedlove	Spirit of America	GE J79	10/15/65
536.712	Art Arfons	Green Monster	GE J79	10/27/65
555.127	Craig Breedlove	Spirit of America, Sonic 1	GE J79	11/2/65
576.553	Art Arfons	Green Monster	GE J79	11/7/65
600.601	Craig Breedlove	Spirit of America, Sonic 1	GE J79	11/15/65
622.407	Gary Gabelich	Blue Flame	Rocket	10/23/70
633.468	Richard Noble	Thrust 2	RR RG 146	10/4/83
763.035	Andy Green	Thrust SSC	RR Spey	10/15/97

Example 5: Distance and Time (GR 8-10)
The following data were collected using a car with a water clock set to release a drop in a unit of time and a meter stick. The car rolled down an inclined plane. Three trials were run. Create a data table with an average distance column and an average velocity column, create an average distance-time graph, and draw the best-fit line or curve. Estimate the car's distance traveled and velocity at six drops of water. Describe the motion of the car. Is it going at a constant speed, accelerating, or decelerating? How do you know?

Time (drops of water)	Distance (cm)
1	10,11,9
2	29, 31, 30
3	59, 58, 61
4	102, 100, 98
5	122, 125, 127



PDF

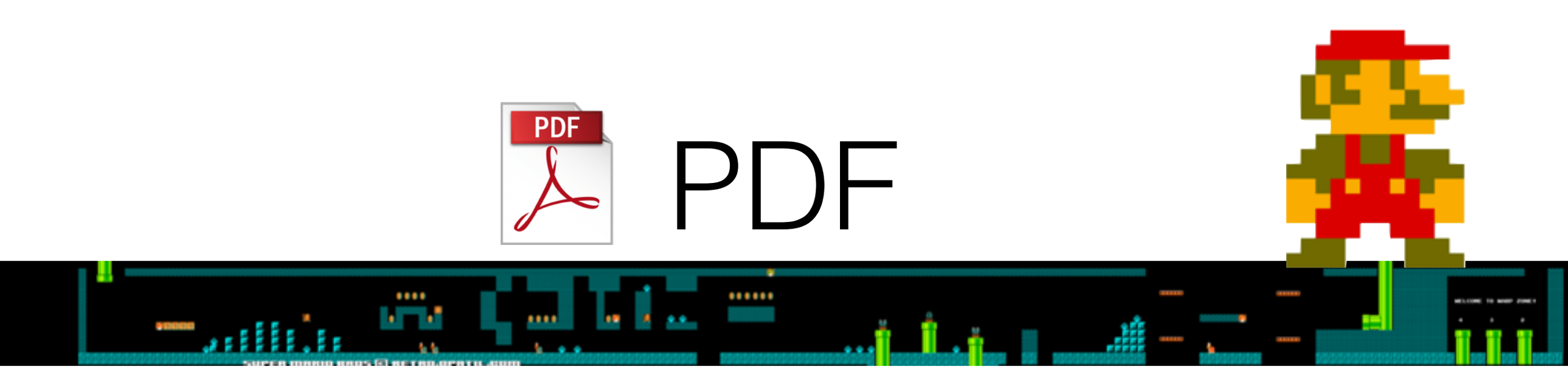
- Zone 1 - **tabula** (pip install tabula-py)

```
from tabula import read_pdf
df_list = read_pdf('sample.pdf', pages='all', multiple_tables=True)
for df in df_list:
    df
```

	0	1
0	Number of Coils	Number of Paperclips
1	5	3, 5, 4
2	10	7, 8, 6
3	15	11, 10, 12
4	20	15, 13, 14

	0	1	2	3	4
0	Speed (mph)	Driver	Car	Engine	Date
1	407.447	Craig Breedlove	Spirit of America	GE J47	8/5/63
2	413.199	Tom Green	Wingfoot Express	WE J46	10/2/64
3	434.22	Art Arfons	Green Monster	GE J79	10/5/64
4	468.719	Craig Breedlove	Spirit of America	GE J79	10/13/64
5	526.277	Craig Breedlove	Spirit of America	GE J79	10/15/65
6	536.712	Art Arfons	Green Monster	GE J79	10/27/65
7	555.127	Craig Breedlove	Spirit of America, Sonic 1	GE J79	11/2/65
8	576.553	Art Arfons	Green Monster	GE J79	11/7/65
9	600.601	Craig Breedlove	Spirit of America, Sonic 1	GE J79	11/15/65
10	622.407	Gary Gabelich	Blue Flame	Rocket	10/23/70
11	633.468	Richard Noble	Thrust 2	RR RG 146	10/4/83
12	763.035	Andy Green	Thrust SSC	RR Spey	10/15/97

https://github.com/tabulapdf/tabula


PDF
PDF

PDF

WELCOME TO WRAP ZONE !

PDF

- Zone 2 - Decrypt password-protected pdf

  - **pyPDF2**

```
from PyPDF2 import PdfFileReader

fp = open(filename)
pdfFile = PdfFileReader(fp)
password = "mypassword"
if pdfFile.isEncrypted:
    try:
        pdfFile.decrypt(password)
        print('File Decrypted (PyPDF2)')
```

  - **QPDF**: qpdf --decrypt input.pdf output.pdf

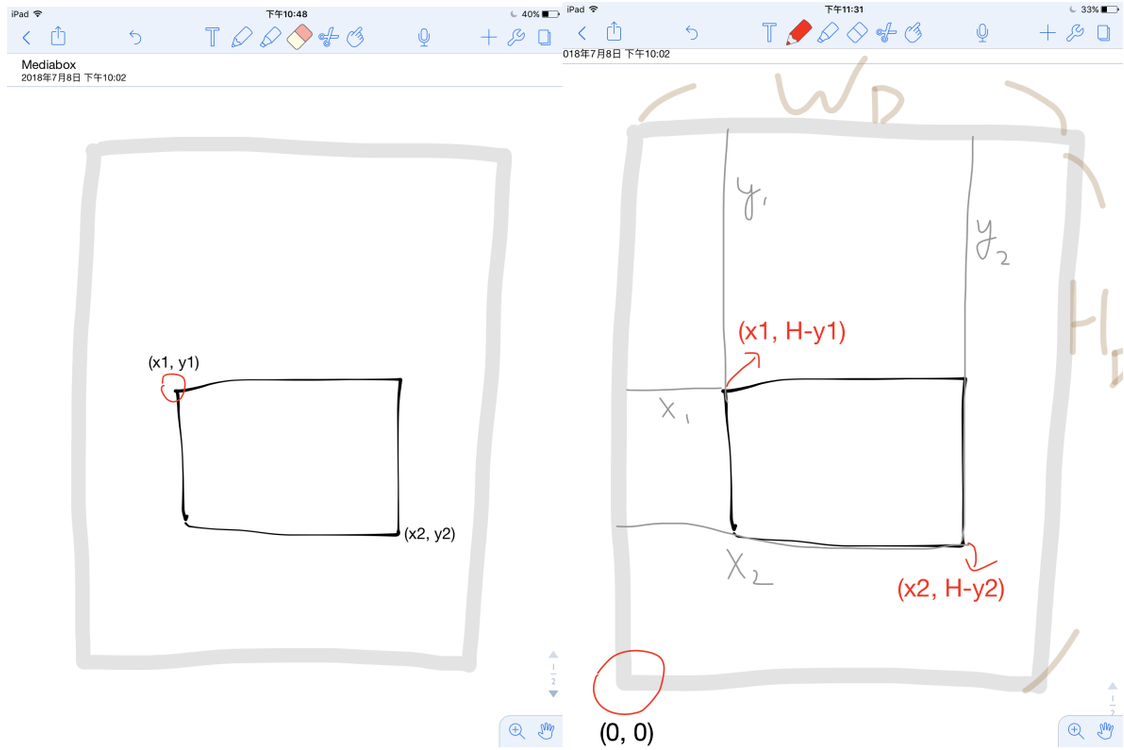
```
    except:
        command = ("cp "+ filename +
            " temp.pdf; qpdf --password='' --decrypt temp.pdf " + filename
            + "; rm temp.pdf")
        os.system(command)
        print('File Decrypted (qpdf)')
        fp = open(filename)
        pdfFile = PdfFileReader(fp)
```

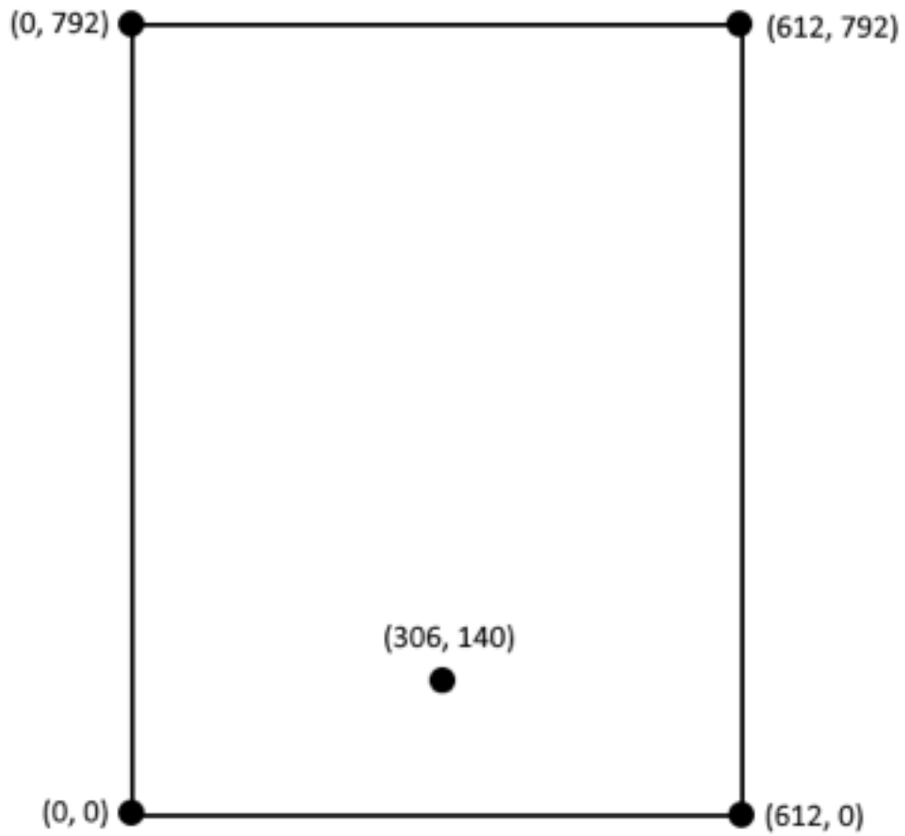
https://stackoverflow.com/questions/38838930/how-to-decode-a-pdf-encrypted-file-from-python

PDF

- Zone 3 - Find table captions

  - 3-1 Find area coordinate





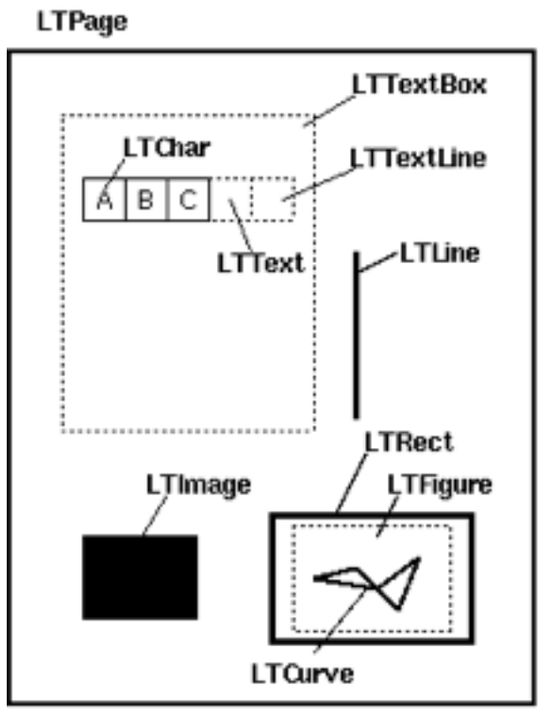

```
W, H = caption.mediaBox.getUpperRight()
left, top, width, height = 73, 167 , 489, 178
x1 = left
y1 = H - top - height
x2 = x1 + width
y2 = y1 + height
```

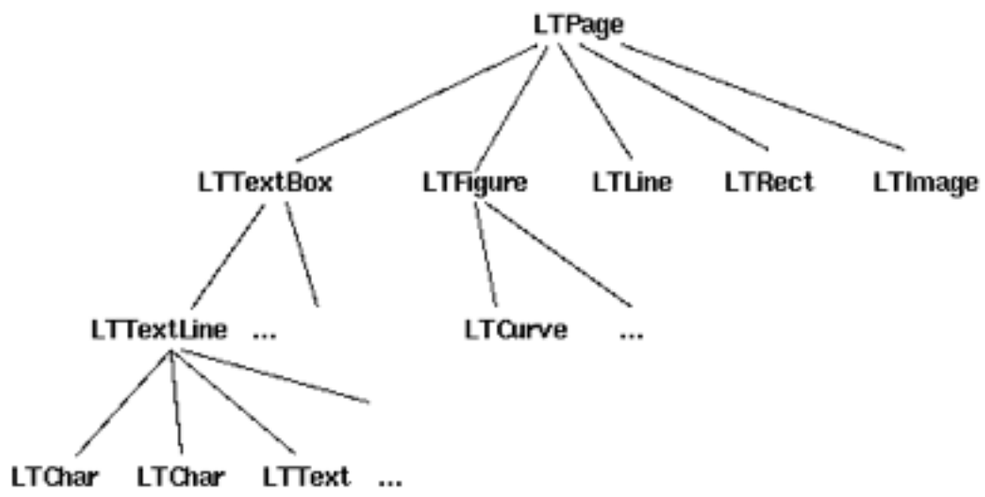
PDF

- Zone 3 - Find table captions

  - 3-1 Find area coordinate

  - 3-2 Tear down elements





PDFMiner: https://github.com/euske/pdfminer

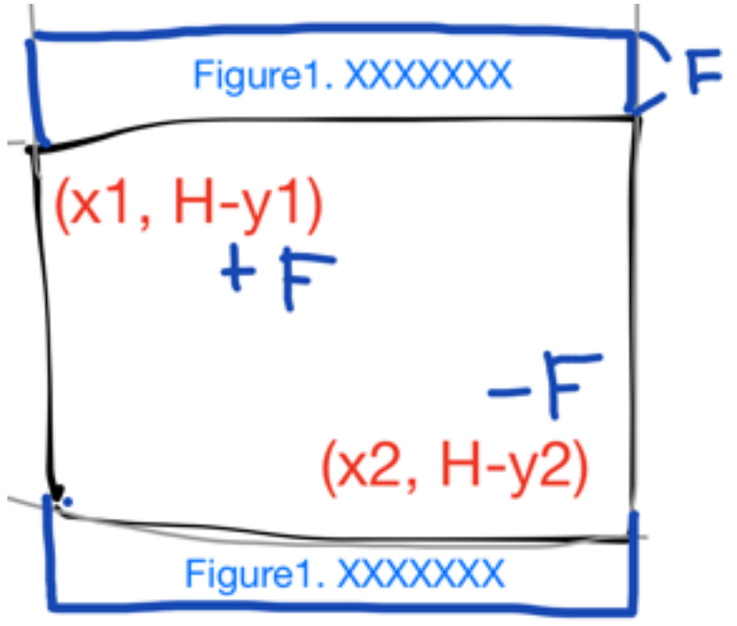
PDF

- Zone 3 - Find table captions

  - 3-1 Find area coordinate

  - 3-2 Tear down elements

  - 3-3 Get captions



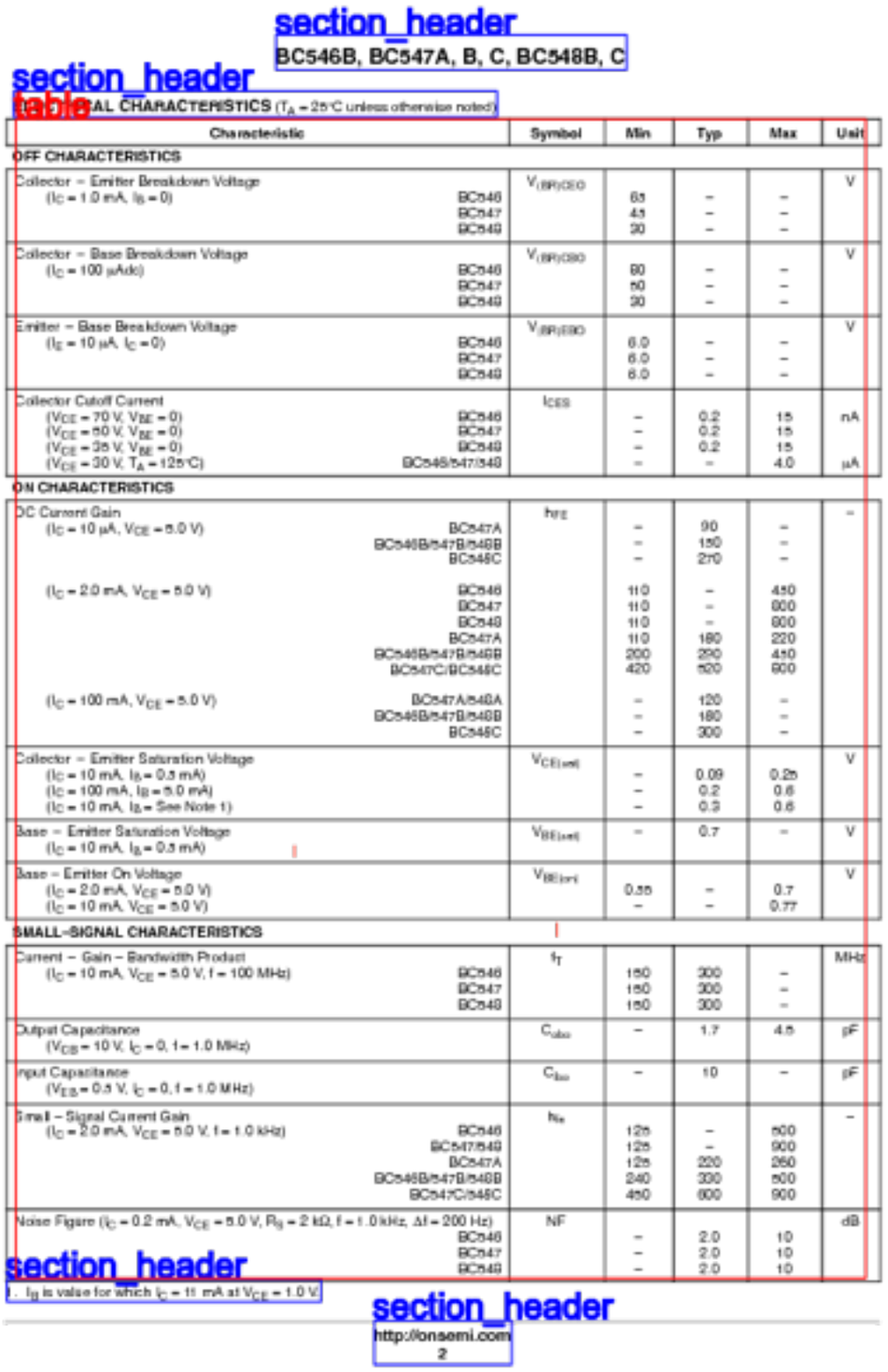
PDF

- Zone 4 - Section tagging

  - Utilize OCR
    https://github.com/WZBSocialScienceCenter/pdftabextract/blob/master/examples/catalogue_30s/catalog_30s_notebook.ipynb

  - ML approach
    https://github.com/HazyResearch/pdftotree

ML Approach to tag sections



ELECTRICAL CHARACTERISTICS (T_A = 25°C unless otherwise noted)

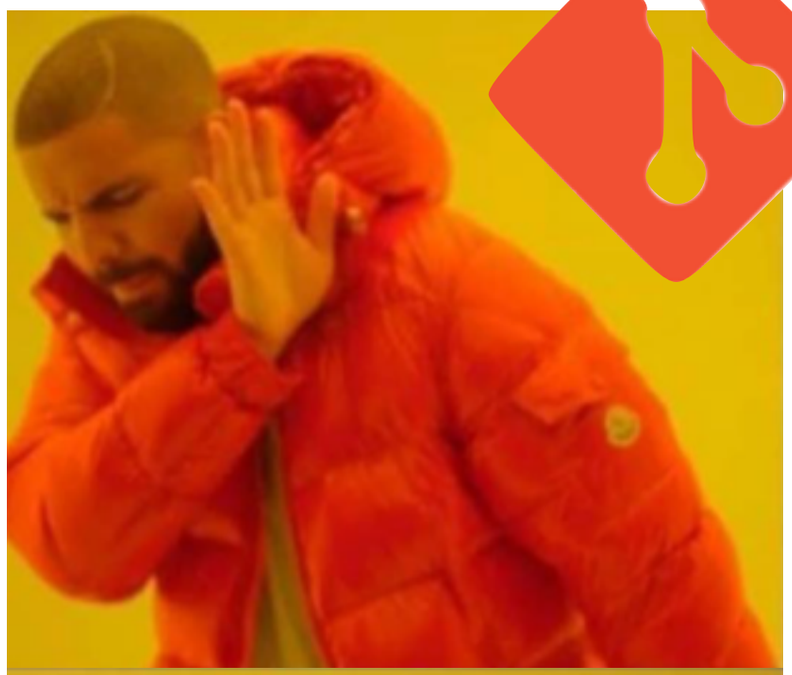
Characteristic		Symbol	Min	Typ	Max	Unit
OFF CHARACTERISTICS						
Collector − Emitter Breakdown Voltage (I_C = 1.0 mA, I_B = 0)	BC546	$V_{(BR)CEO}$	65	–	–	V
	BC547		45	–	–	
	BC548		30	–	–	
Collector − Base Breakdown Voltage (I_C = 100 μAdc)	BC546	$V_{(BR)CBO}$	80	–	–	V
	BC547		50	–	–	
	BC548		30	–	–	
Emitter − Base Breakdown Voltage (I_E = 10 μA, I_C = 0)	BC546	$V_{(BR)EBO}$	6.0	–	–	V
	BC547		6.0	–	–	
	BC548		6.0	–	–	
Collector Cutoff Current		I_{CES}				
(V_{CE} = 70 V, V_{BE} = 0)	BC546		–	0.2	15	nA
(V_{CE} = 50 V, V_{BE} = 0)	BC547		–	0.2	15	
(V_{CE} = 35 V, V_{BE} = 0)	BC548		–	0.2	15	
(V_{CE} = 30 V, T_A = 125°C)	BC546/547/548		–	–	4.0	μA
ON CHARACTERISTICS						
DC Current Gain		h_{FE}				–
(I_C = 10 μA, V_{CE} = 5.0 V)	BC547A		–	90	–	
	BC546B/547B/548B		–	150	–	
	BC546C		–	270	–	
(I_C = 2.0 mA, V_{CE} = 5.0 V)	BC546		110	–	450	
	BC547		110	–	800	
	BC548		110	–	800	
	BC547A		110	180	220	
	BC546B/547B/548B		200	290	450	
	BC547C/546C		420	520	800	
(I_C = 100 mA, V_{CE} = 5.0 V)	BC547A/548A		–	120	–	
	BC546B/547B/548B		–	180	–	
	BC546C		–	300	–	
Collector − Emitter Saturation Voltage		$V_{CE(sat)}$				V
(I_C = 10 mA, I_B = 0.5 mA)			–	0.09	0.25	
(I_C = 100 mA, I_B = 5.0 mA)			–	0.2	0.6	
(I_C = 10 mA, I_B = See Note 1)			–	0.3	0.6	
Base − Emitter Saturation Voltage (I_C = 10 mA, I_B = 0.5 mA)		$V_{BE(sat)}$	–	0.7	–	V
Base − Emitter On Voltage		$V_{BE(on)}$				V
(I_C = 2.0 mA, V_{CE} = 5.0 V)			0.55	–	0.7	
(I_C = 10 mA, V_{CE} = 5.0 V)			–	–	0.77	
SMALL−SIGNAL CHARACTERISTICS						
Current − Gain − Bandwidth Product		f_T				MHz
(I_C = 10 mA, V_{CE} = 5.0 V, f = 100 MHz)	BC546		150	300	–	
	BC547		150	300	–	
	BC548		150	300	–	
Output Capacitance (V_{CB} = 10 V, I_C = 0, f = 1.0 MHz)		C_{obo}	–	1.7	4.5	pF
Input Capacitance (V_{EB} = 0.5 V, I_C = 0, f = 1.0 MHz)		C_{ibo}	–	10	–	pF
Small − Signal Current Gain		h_{fe}				–
(I_C = 2.0 mA, V_{CE} = 5.0 V, f = 1.0 kHz)	BC546		125	–	500	
	BC547/549		125	–	900	
	BC547A		125	220	260	
	BC546B/547B/548B		240	330	500	
	BC547C/546C		450	600	900	
Noise Figure (I_C = 0.2 mA, V_{CE} = 5.0 V, R_S = 2 kΩ, f = 1.0 kHz, Δf = 200 Hz)		NF				dB
	BC546		–	2.0	10	
	BC547		–	2.0	10	
	BC549		–	2.0	10	

1. I_B is value for which I_C = 11 mA at V_{CE} = 1.0 V.

Data Parser

SVN v.s. Git

SVN v.s. Git



GitPython

```python
#!/usr/bin/env python

from git import *

repo = Repo("misctools")
o = repo.remotes.origin
o.pull()

master = repo.head.reference
print master.log()
```

https://gitpython.readthedocs.io/en/stable/tutorial.html

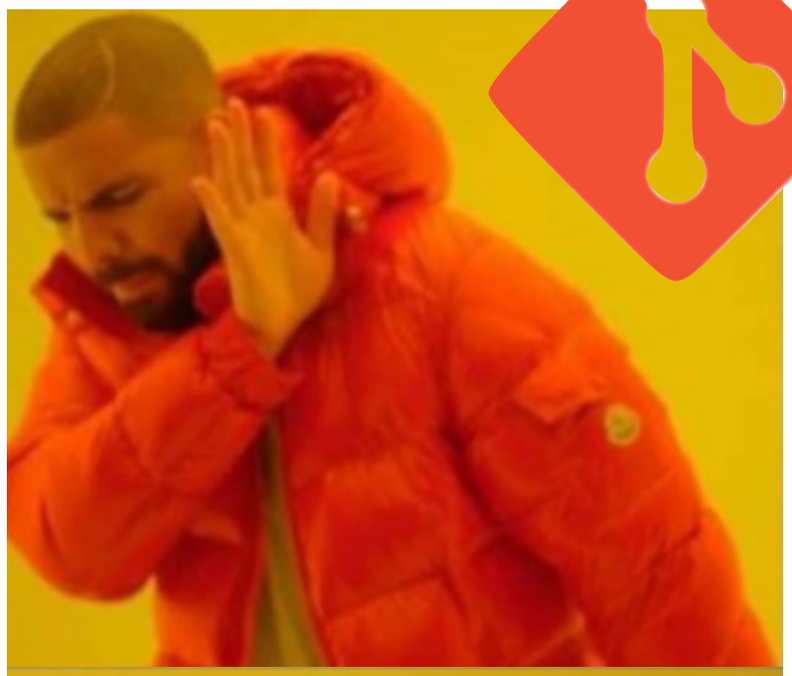
SVN v.s. Git

PySVN

GitPython

```
#!/usr/bin/env python

from git import *

repo = Repo("misctools")
o = repo.remotes.origin
o.pull()

master = repo.head.reference
print master.log()
```

SVN v.s. Git



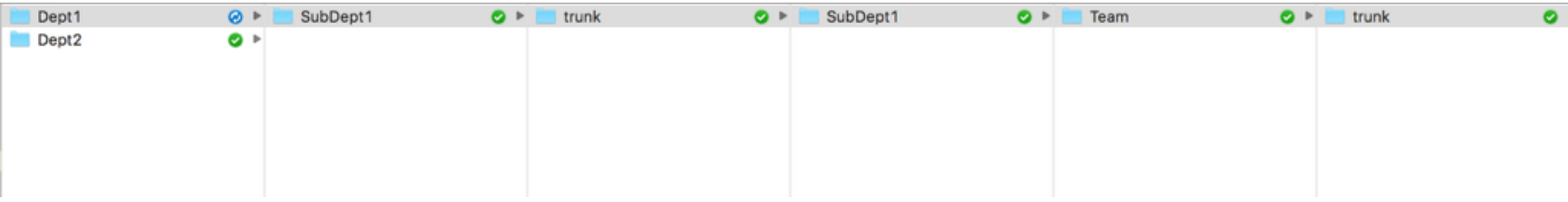
~~PySVN~~

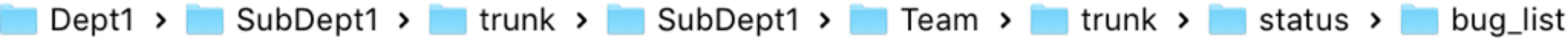
glob

GitPython

```
#!/usr/bin/env python

from git import *

repo = Repo("misctools")
o = repo.remotes.origin
o.pull()

master = repo.head.reference
print master.log()
```

Legacy Repo





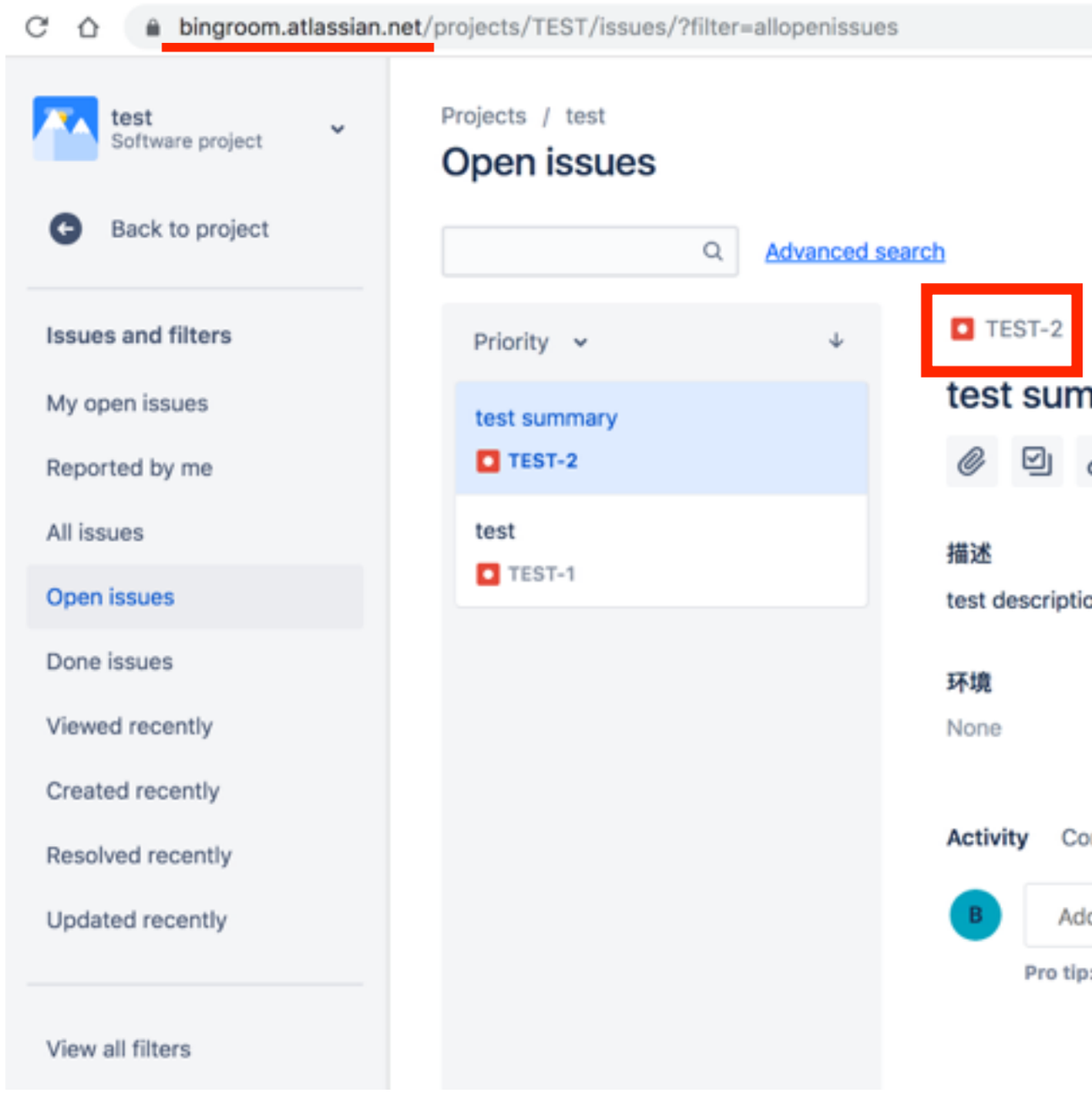
http://hostaddress/svn/Dept1/SubDept1/trunk/SubDept1/Team/trunk/status/bug_list/bug.xls

```
path_exp = "**/Dept1/SubDept1/**/bug_list/**/*.*"

for fp in glob.glob(path_exp, recursive=True):
...
```

JIRA

- pip install **jira**

```python
from jira import JIRA

HOST = 'your JIRA server'
jira = JIRA(HOST, basic_auth=('user', 'token'))
issue = jira.issue('issue key')
```



https://jira.readthedocs.io/en/master/

JIRA

(In Practice)

- Field name mapping

```
field_dict = {d['id']: d['name'] for d in jira.fields()}
field_dict
```

```
{'statuscategorychangedate': 'Status Category Changed',
 'issuetype': 'Issue Type',
 'timespent': 'Time Spent',
 'project': 'Project',
 'fixVersions': 'Fix versions',
 'aggregatetimespent': 'Σ Time Spent',
'statusCategory': 'Status Category',
'resolution': 'Resolution',
'resolutiondate': 'Resolved',
'workratio': 'Work Ratio',
'lastViewed': 'Last Viewed',
'watches': 'Watchers',
'thumbnail': 'Images',
'created': 'Created',
'customfield_10020': 'Sprint',
'customfield_10021': 'Flagged',
'customfield_10022': '[CHART] Date of First Response',
'priority': 'Priority',
'customfield_10023': '[CHART] Time in Status',
'labels': 'Labels',
'customfield_10016': 'Story point estimate',
'customfield_10017': 'Issue color',
'customfield_10018': 'Parent Link',
'customfield_10019': 'Rank',
'timeestimate': 'Remaining Estimate',
'aggregatetimeoriginalestimate': 'Σ Original Estimate',
'versions': 'Affects versions',
```

JIRA

(In Practice)

Priority

test summary

TEST-2

test

TEST-1

TEST-1

test

描述

test description

环境

None

Activity Comments

B 高餅倫 36 minutes ago

gg

Edit · Delete ·

B 高餅倫 4 minutes ago

我也gg

Edit · Delete ·


```
'assignee': None,
'attachment': [],
'comment': [('高餅倫', 'gg'), ('高餅倫', '我也gg')],
'components': [],
'created': '2019-09-17T23:54:37.734+0800',
'creator': {'displayName': '高餅倫', 'emailAddress': 'joekaojoekao@gmail.com'},
'description': 'test description',
'duedate': None,
'environment': None,
'fixVersions': [],
'issuelinks': [],
'labels': [],
'lastViewed': '2019-09-18T00:40:30.881+0800',
'priority': 'Medium',
'progress': {'progress': 0, 'total': 0},
'project': 'test',
'reporter': {'displayName': '高餅倫', 'emailAddress': 'joekaojoekao@gmail.com'},
'resolution': None,
'resolutiondate': None,
'security': None,
'status': 'To Do',
'statuscategorychangedate': '2019-09-17T23:54:38.010+0800',
'subtasks': [],
'summary': 'test',
'timeestimate': None,
'timeoriginalestimate': None,
'timespent': None,
'timetracking': {},
'type': 'Bug',
'updated': '2019-09-18T00:40:36.632+0800',
```

Case 1: Low-cost Data Pipeline

Case 1: Low-cost Data Pipeline

Requirements

- Windows data, Linux backend
- Integration on heterogeneous files
- Error handling, early feedback
- User-friendly

Requirements

- Windows data, Linux backend
- Integration on heterogeneous files
- Error handling, early feedback
- User-friendly

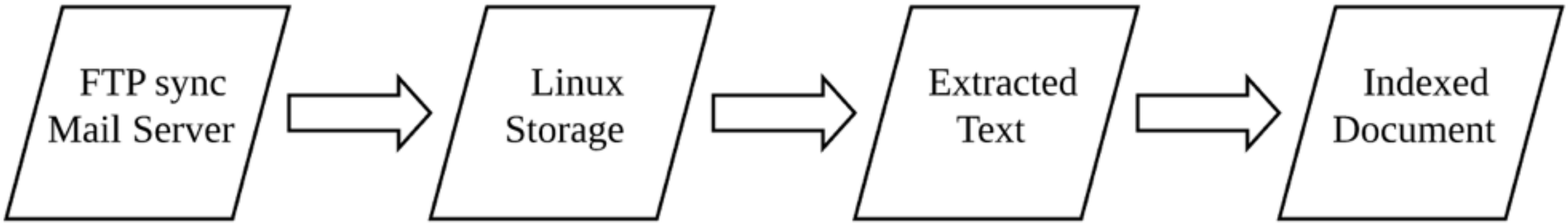
elasticsearch

python™

Pipeline

1. inotifywait

3. Elasticsearch

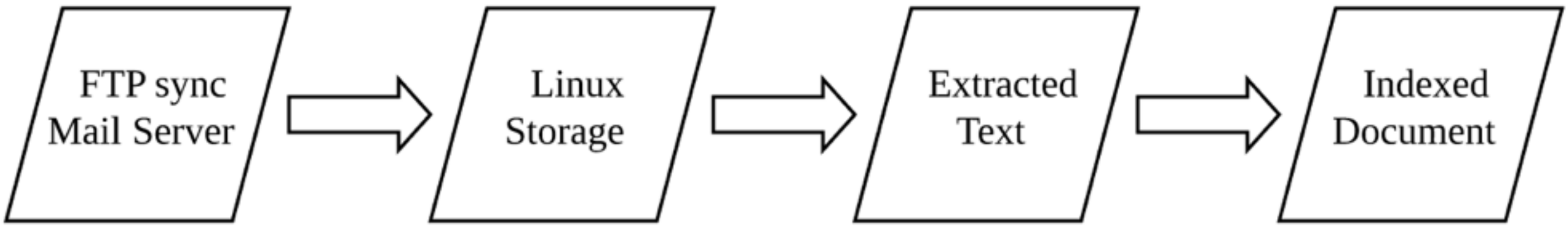


2. Parser with error handling

Pipeline

1. inotifywait

3. Elasticsearch



2. Parser with error handling

(1) inotifywait

You should consider using `inotifywait`, as an example:

```
inotifywait -m /path -e create -e moved_to |
    while read path action file; do
        echo "The file '$file' appeared in directory '$path' via '$action'"
        # do something with the file
    done
```

https://linux.die.net/man/1/inotifywait

(1) inotifywait

You should consider using `inotifywait`, as an example:

close_write

```
inotifywait -m /path -e create -e moved_to |
    while read path action file; do
        echo "The file '$file' appeared in directory '$path' via '$action'"
        # do something with the file
    done
```

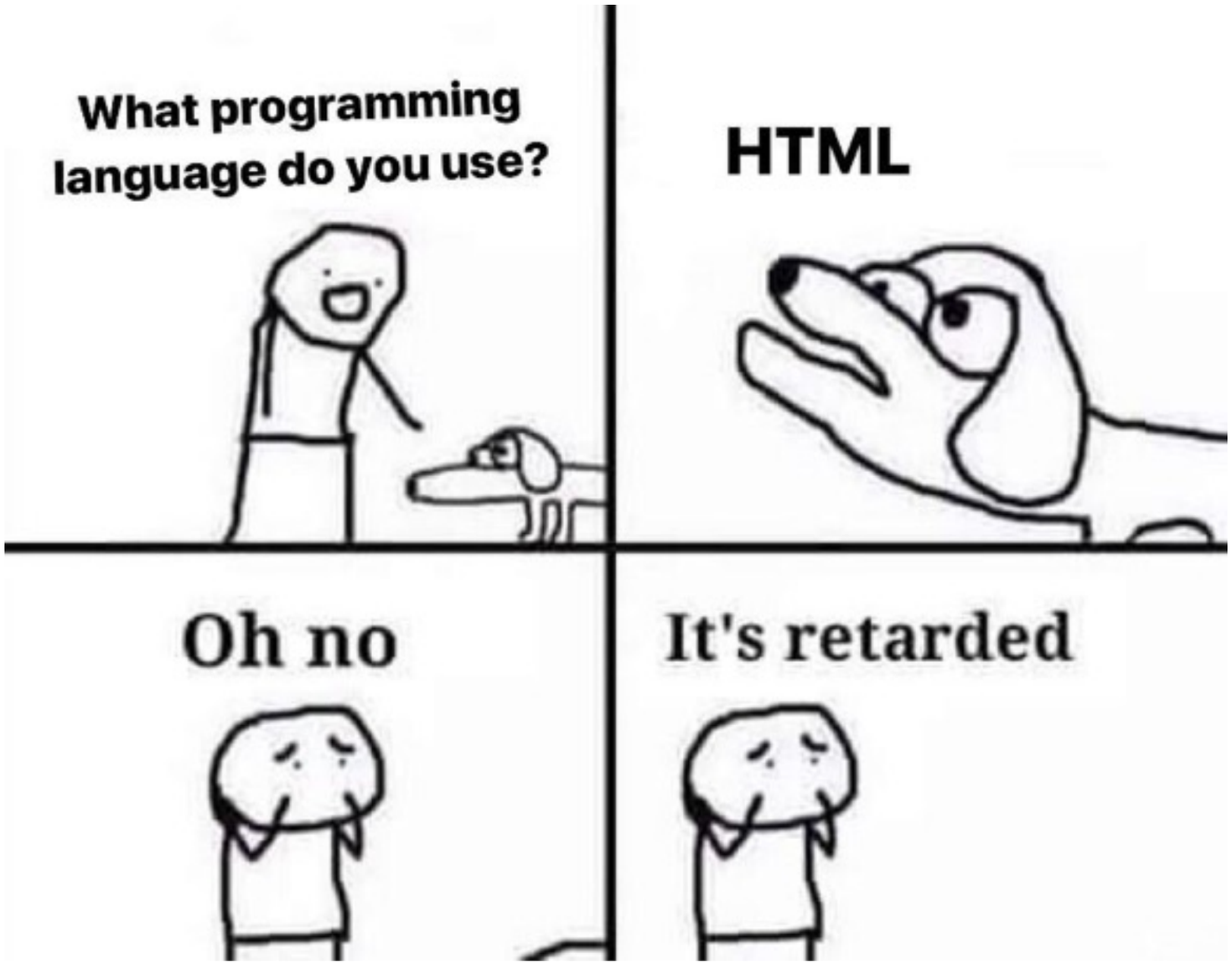
(2) Error handling

- Massive if-else and try-except
- Report locations of bad cells when handling **Excel**

  e.g. [Error] Wrong data type in column ‘est. cost’,
  In Line 2, 5,10.

- Reply in email using **HTML**

(2) Error handling



(2) Error handling

```
import html
from htmltag import HTML, td, tr, th

table = ""
rows = [1, 2, 'x', 4]
for row in rows:
    s = ""
    if row == 'x':
        s += td(str(row), _class="bad-cell")
    else:
        s += td(str(row))

    table += tr(s)

html_str = """
[]
<h3 class="error-msg">Invalid format: </h3>
<table align="center" cellpadding="2" cellspacing="0" width="5%">
    <tr>[]</tr>
    []
</table>""".replace("{", "{{").replace("}", "}}") \
.replace("[", "{").replace("]", "}").format(style, th('index'), table)

print(html.unescape(html_str))
```

```
<style>
table, th, td {
  border: 1px solid black;
}
td.bad-cell {
  width: 100%;
  background-color: #ff0000;
}
</style>
<h3 class="error-msg">Invalid format: </h3>
<table align="center" cellpadding="2" cellspacing="0" width="5%">
    <tr><th>index</th></tr>
    <tr><td>1</td></tr><tr><td>2</td></tr><tr><td class="bad-cell">x</td></tr><tr><td>4</td></tr>
</table>
```

Invalid format:

index
1
2
x
4

htmltag

http://liftoff.github.io/htmltag/

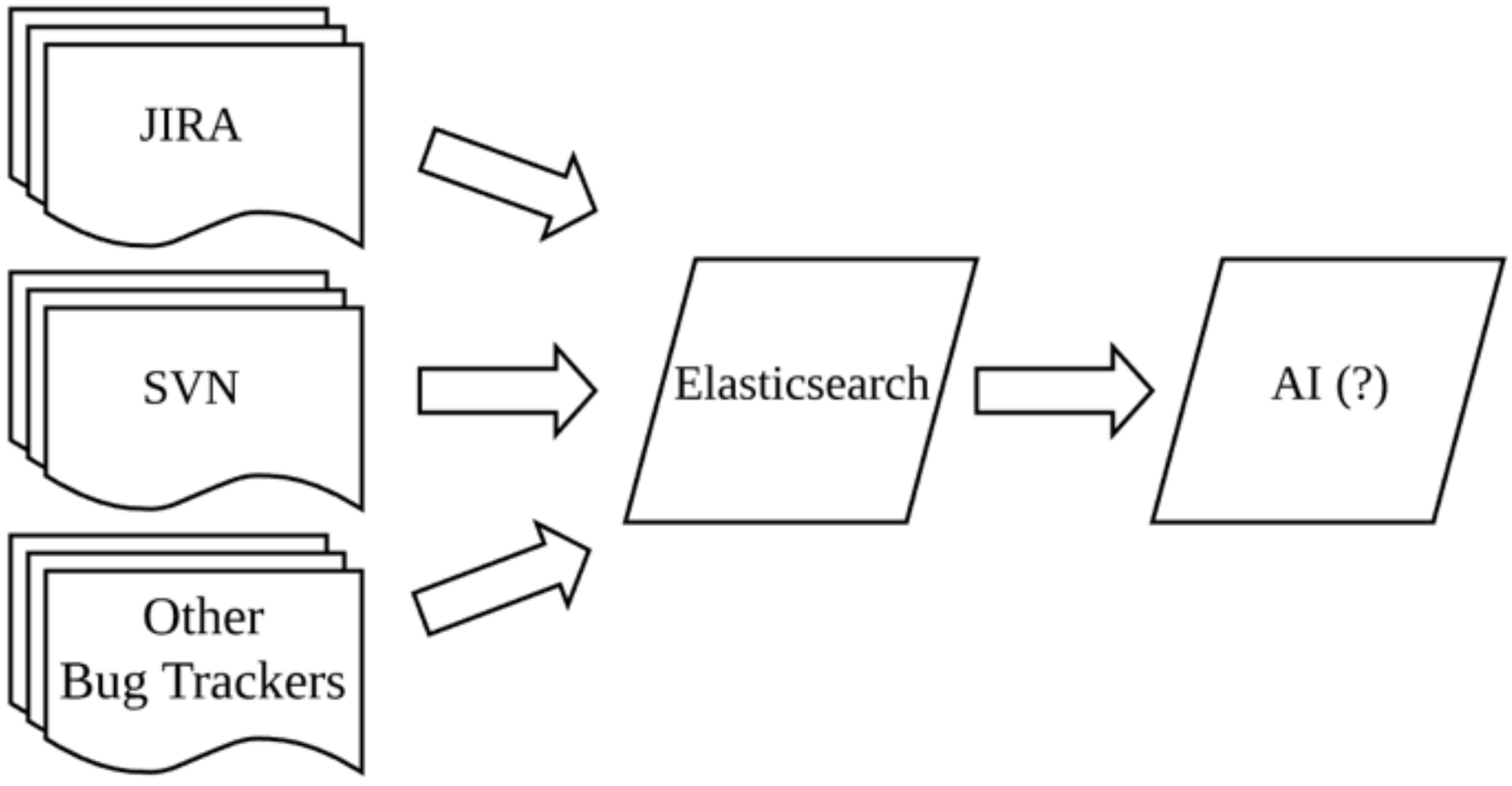
(2) Error handling

Case 2: Bug Search System

Case 2: Bug Search System

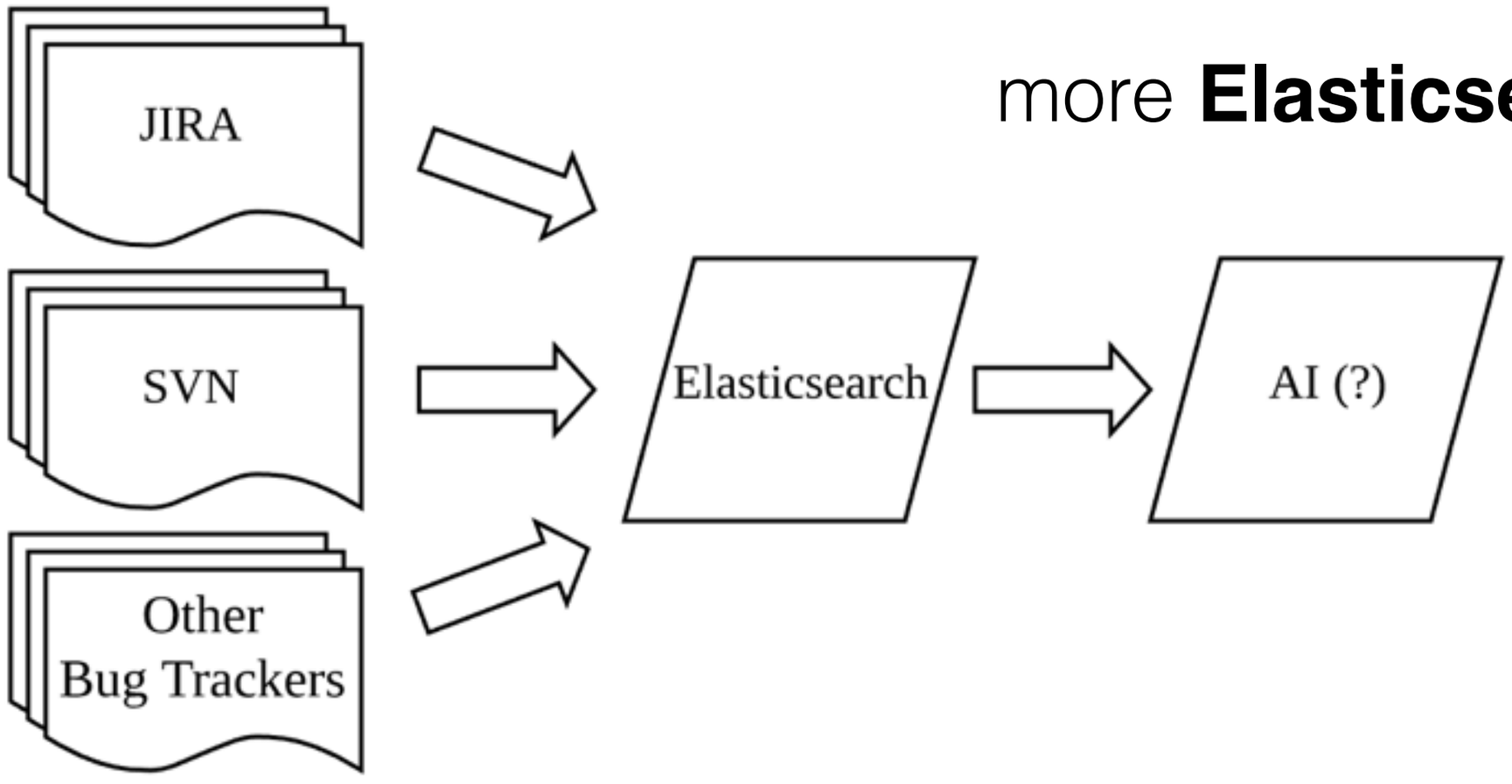
Pipeline

Recap: **textract**



Pipeline

Recap: **textract**



more **Elasticsearch** !!

elasticsearch

- NoSQL, schema-free or schema-***flexible***
- JSON everywhere
- Query DSL (Domain-specific Language)
- Search engine in nature

https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html

elasticsearch

```
GET dota2/_search
{
  "query": {

    "filtered": {
      "filter": {
        "and": [
          {
            "term": {
              "heroname": "drow"
            }
          }
        ]
      }
    }
  },
  "aggs": {}
}
```

```
{
   "took": 2,
   "timed_out": false,
   "_shards": {
      "total": 5,
      "successful": 5,
      "failed": 0
   },
   "hits": {
      "total": 1,
      "max_score": 1,
      "hits": [
         {
            "_index": "dota2",
            "_type": "agust",
            "_id": "AU-N0y9iNzuaTKNspu3X",
            "_score": 1,
            "_source": {
               "heroname": "DROW RANGER",
               "class": "Ranged - Carry",
               "skills": {
                  "main": [
                     "Frost Arrows",
                     "Gust",
                     "Precision Aura"
                  ],
                  "ulti": "Marksmanship"
               },
               "ability": "agility",
               "items": [
                  "mitem1",
                  " mitem2",
                  "mitem3",
                  "mitem4",
                  "mitem5"
               ],
               "dateofplay": "01/01/2012 00:00:00",
               "experienced": "false"
            }
         }
      ]
```

Approaches to AI 0.5

- UI side
  - Google-like searching experience
    - ✓ Space separator
    - ✓ Ranked search result
    - ✓ Near-real-time indexing

Approaches to AI 0.5

- Backend side:
  - Modularizing query strategy

```
query_all = {
    "query": {
        "bool": {
            "should":[
                {
                    "bool":{
                        "must_not":{
                            "exists":{
                                "field": "created"
                            }
                        }
                    }
                },
                {
                    "bool":{
                        "must": [
                            {
                                "range": {
                                    "created": {
                                        "gte": date_from,
                                        "lte": date_to
                                    }
                                }
                            },
                            {
                                "terms": {
                                    "status.keyword": status
                                }
                            }
                        ]
                    }
                }
            ]
        }
    },
    "_source": {
            "includes": search_field,
            "excludes": []
        },
    "sort": sort_list
}
```

Approaches to AI 0.5

- Backend side:
  - Modularizing query strategy

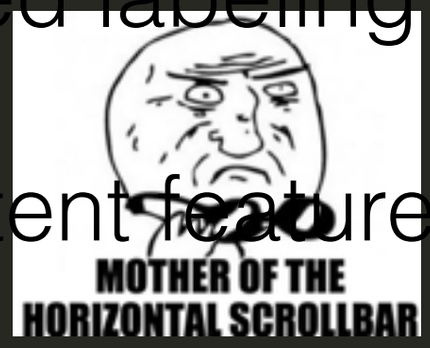
```
project_boost = {
    "function_score": {
        "query": {
            "match": {
                "project": term
            }
        },
        "boost": "200",
        "random_score": {},
        "boost_mode": "multiply"
    }
}
```

```
match_phrase_prefix = {
    "multi_match": {
        "type": "phrase_prefix",
        "query": term,
        "fields": [
            "_id",
            "summary",
            "description",
            "project"
        ]
    }
}
```

```
query_string_wildcard = {
    "query_string": {
        "query": "*{}*".format(re.sub(r'[^\w]', ' ', term)),
        "fields": [
            "issue_key",
            "summary",
            "description",
            "project"
        ]
    }
}
```

Approaches to AI 0.5

- Backend side:
  - Modularizing query strategy
    - Domain-based labeling
    - Utilizing existent features
      ex: JQL

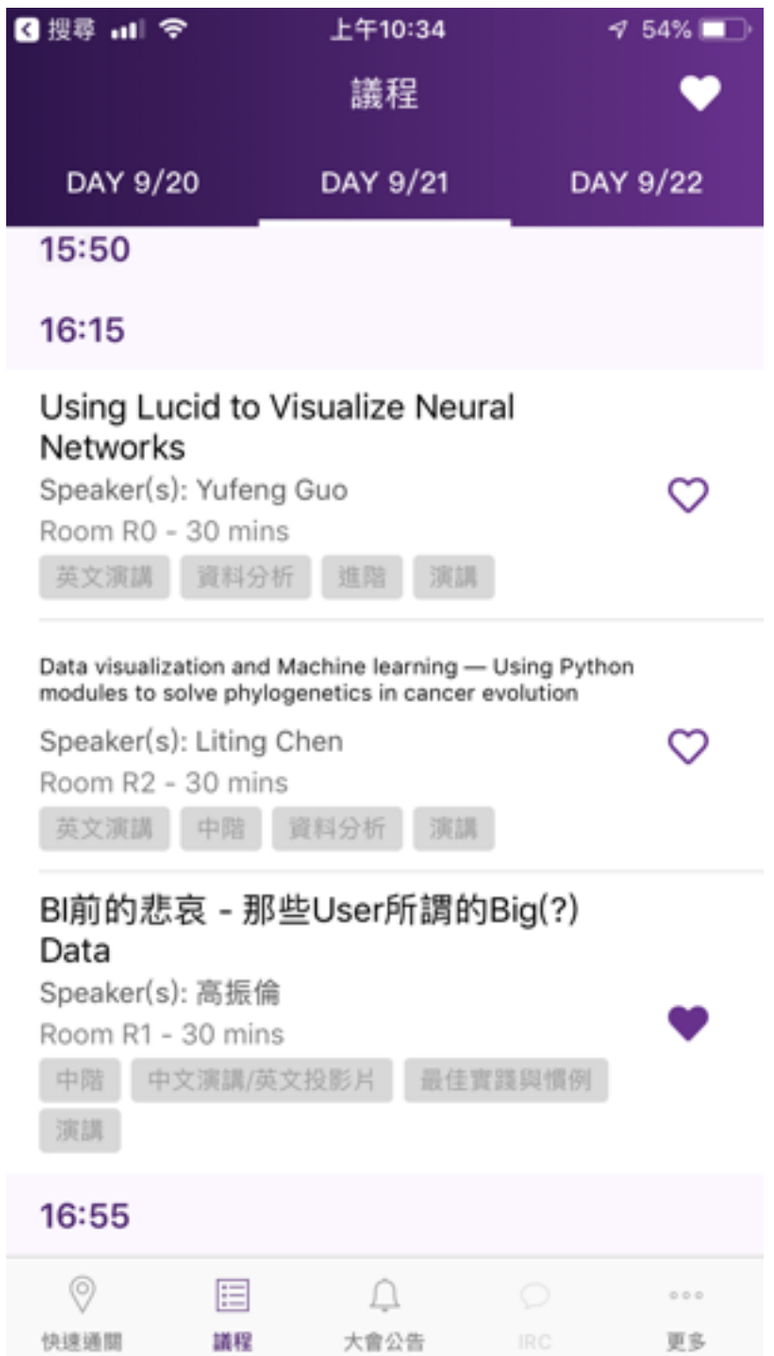



```
":{
must":[
    {
        "range":{
            "created":{
                "gte":date_from,
                "lte":date_to
            }
        }
    },
    {
        "terms":{
            "status.keyword":status
        }
    },
    {
        "bool":{
            "should": [match_issue, match_phrase, match_word, match_phrase_prefix, query_string_wildcard, match_phrase_prefix_post, query_string_wildcard_post, match_phrase_pref
        }
    }
```

Line 247, Column 3249 Spaces: 4 Python

Approaches to AI 0.5

- Backend side:
  - Modularizing query strategy
  - Domain-based labeling



Approaches to AI 0.5

- Backend side:
  - Modularizing query strategy
  - Domain-based labeling
  - Utilizing existent features
    ex: JQL


```
jql = 'summary ~ test'
issue_list = jira.search_issues(jql)
issue_list
```

```
[<JIRA Issue: key='TEST-2', id='10001'>,
 <JIRA Issue: key='TEST-1', id='10000'>]
```

https://www.atlassian.com/blog/jira-software/jql-the-most-flexible-way-to-search-jira-14

Friends: How did you write this code so beautifully ?
Me(Proudly):

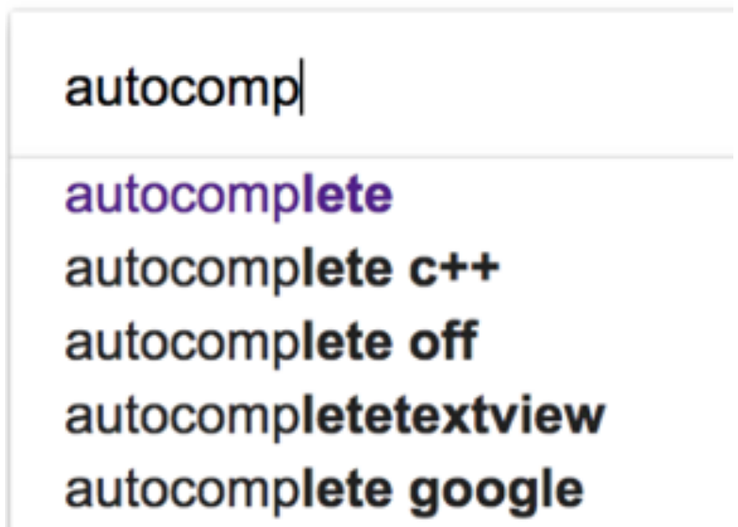
Friends: How did you write this code so beautifully ?
Me(Proudly):

AI 0.5: Must have

- Meet user expectations
- UAT (User Acceptance Testing)
- Hourly build

AI 0.5: Nice to have

- User rating 
- Auto-complete (Completion suggester)
- ~~Semantic analysis~~



Grab-and-Go

- Not Best practice but Test practice
- Not local / global optimum but User optimum
- Show your respect for frontend colleagues