

"Mas dá pra abrir no Excel?" Exportando para txt, csv e JSON com Python


Renata D'Avila

python
BRASIL
[14] 

17 a 22
DE OUTUBRO **2018**
NATAL | RIO GRANDE DO NORTE



rsip22 = 'Renata D'Avila'

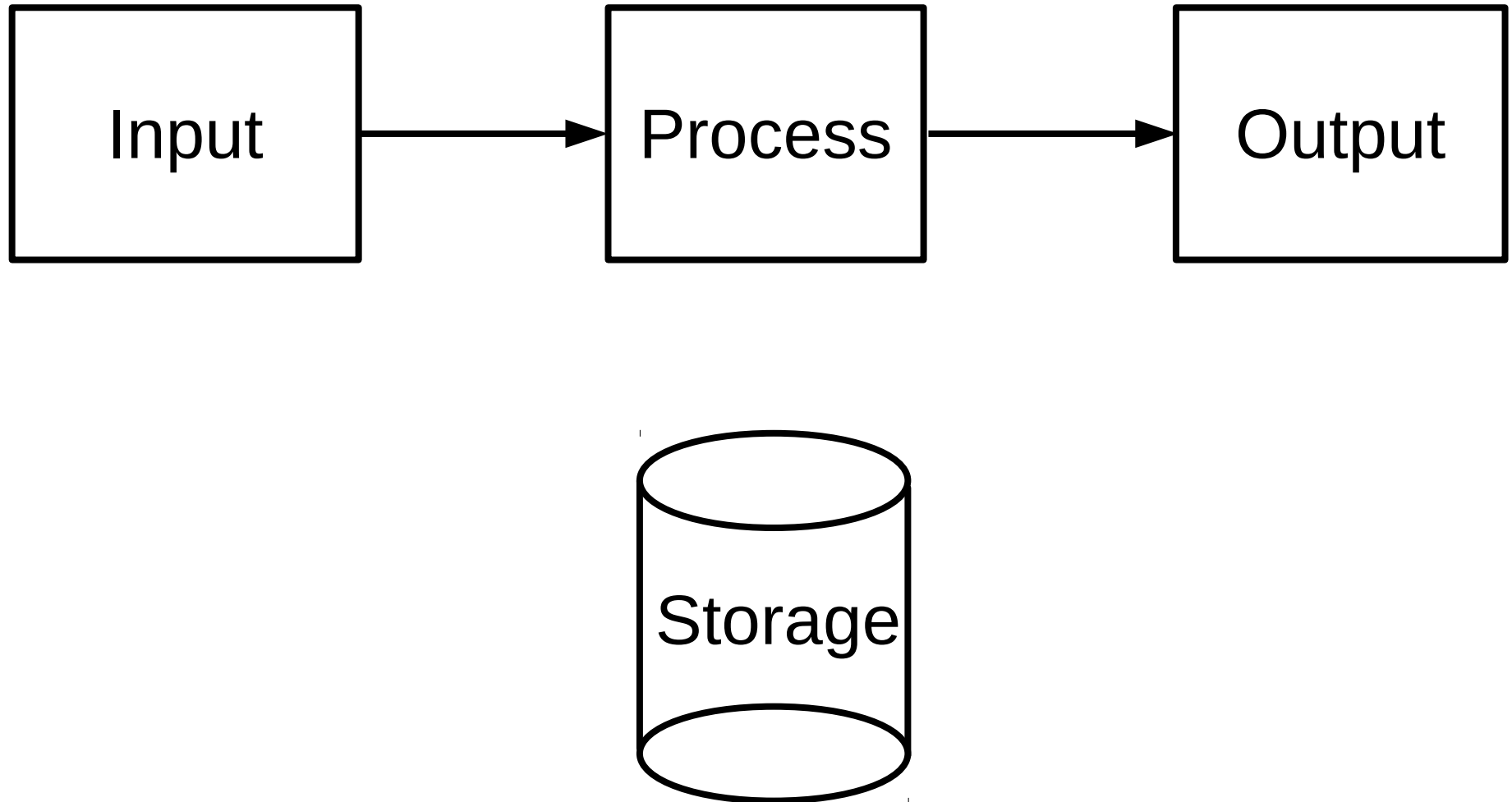
- Técnica em Informática para Internet
- Estagiária Outreachy com projeto Debian
(Dez 2017 – Mar 2018)
- PyLadies Porto Alegre
- Organizadora/Treinadora Django Girls
- Gosta de resolver problemas usando  python™



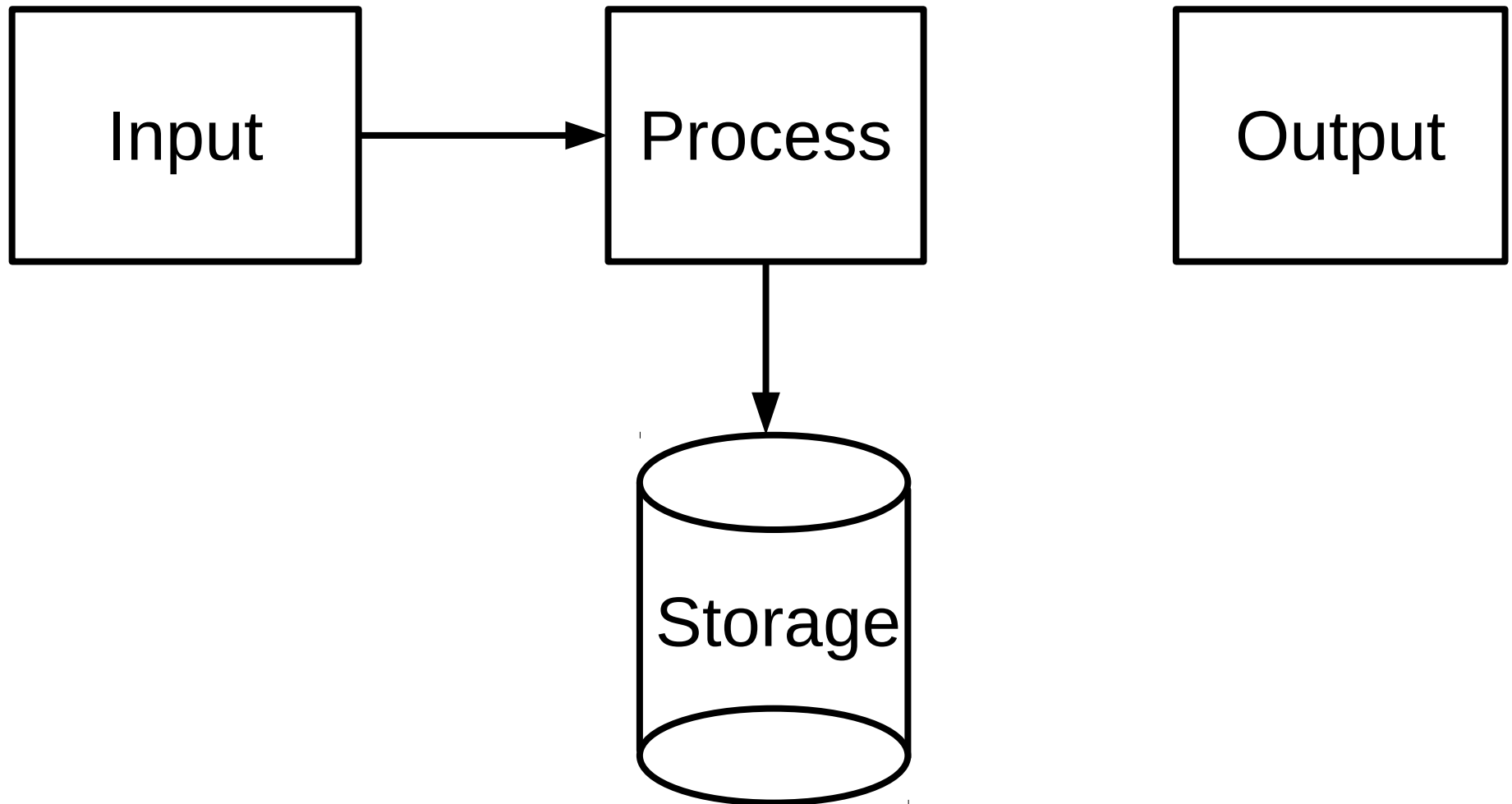
Motivações

- “Mas vai dar para abrir no Excel?”
- Mostrar como exportar dados para diversos formatos de arquivos (TXT, CSV, JSON) usando módulos próprios do Python (Standard Library)
- *Aviso de conteúdo: os códigos e soluções exibidos foram simplificados para entendimento mais fácil (e para caber nos slides).*

Computador



Opção 1: Redirecionar toda saída para o armazenamento



Exportando para arquivo, formato texto (.txt)

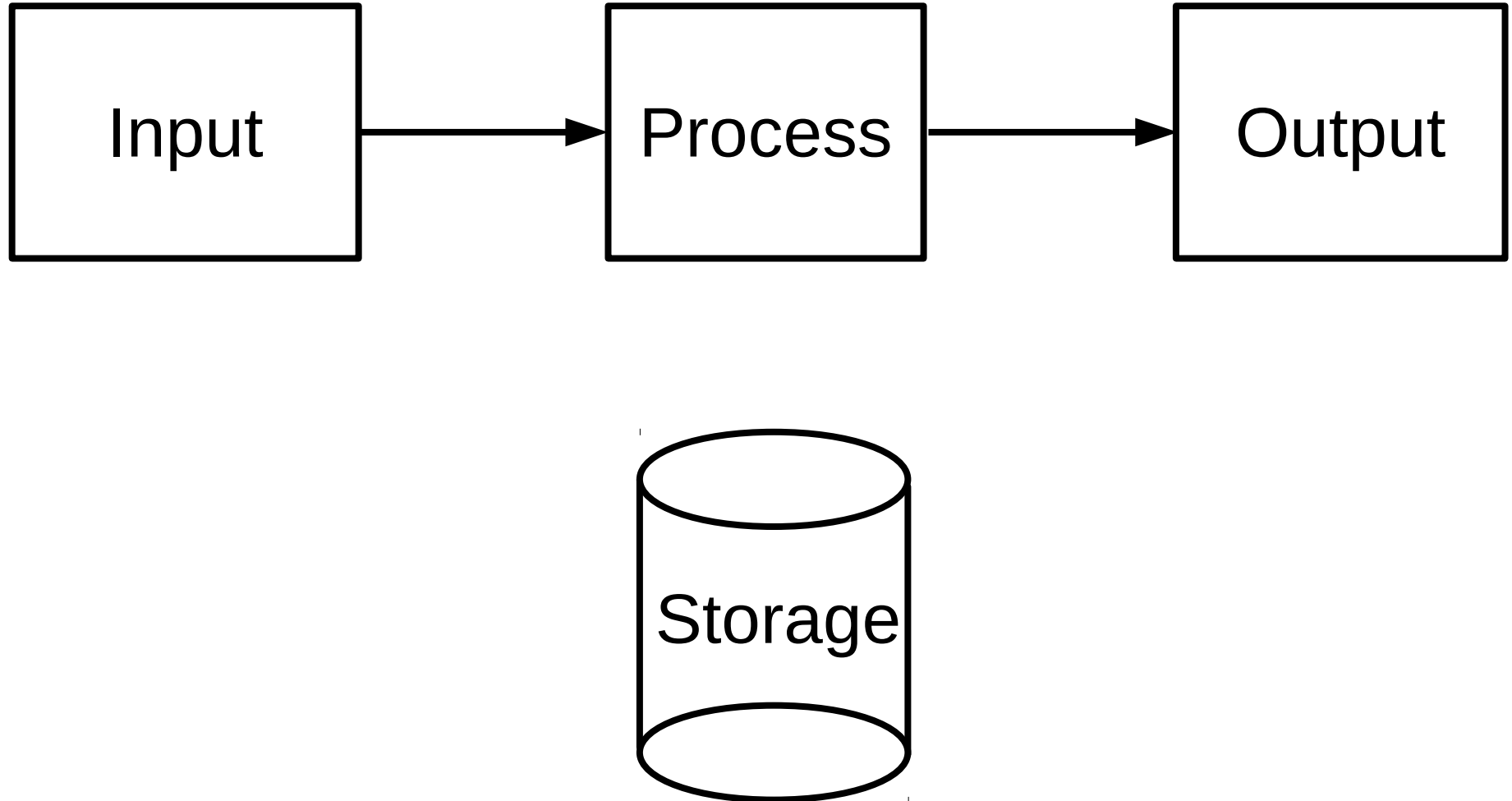
- Exportar para armazenamento (arquivo texto) todo resultado do processamento:

```
new_list = ['pybr', 'natal', '2018', 'palestras']
```

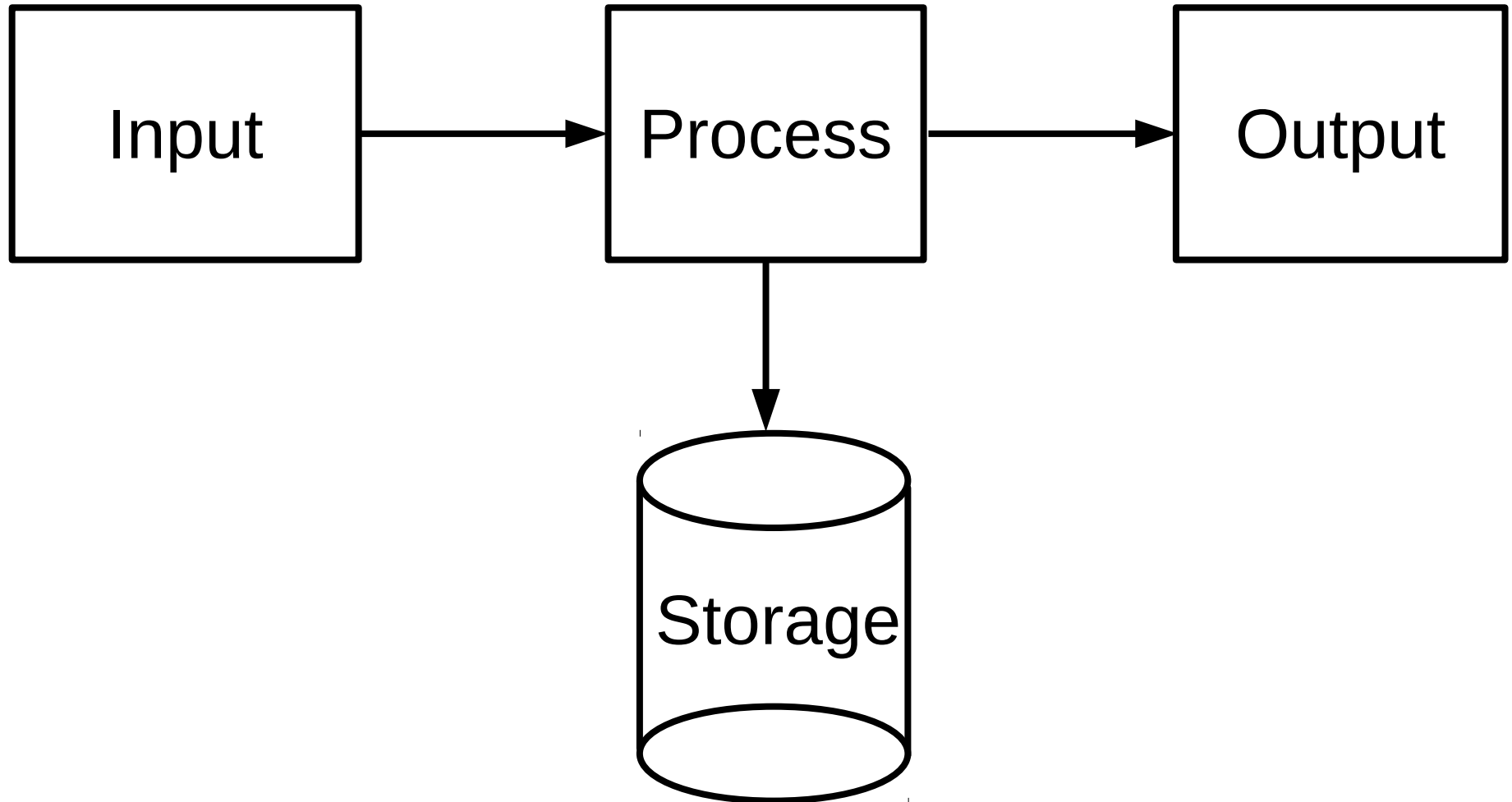
```
with open('all_output.txt', 'w') as fhand:
```

```
    fhand.write('\n'.join(str(x) for x in new_list))
```

Computador



Opção 2: Envia parte da saída para o armazenamento



Exportando para arquivo, formato texto (.txt)

- Exportar para armazenamento (arquivo) parte da saída do programa, manter output para status do processamento.
- `contextlib.redirect_stdout()`
 - Gerenciador de contexto para temporariamente redirecionar a saída do sistema (`sys.stdout`) para outro arquivo ou objeto-como-arquivo.

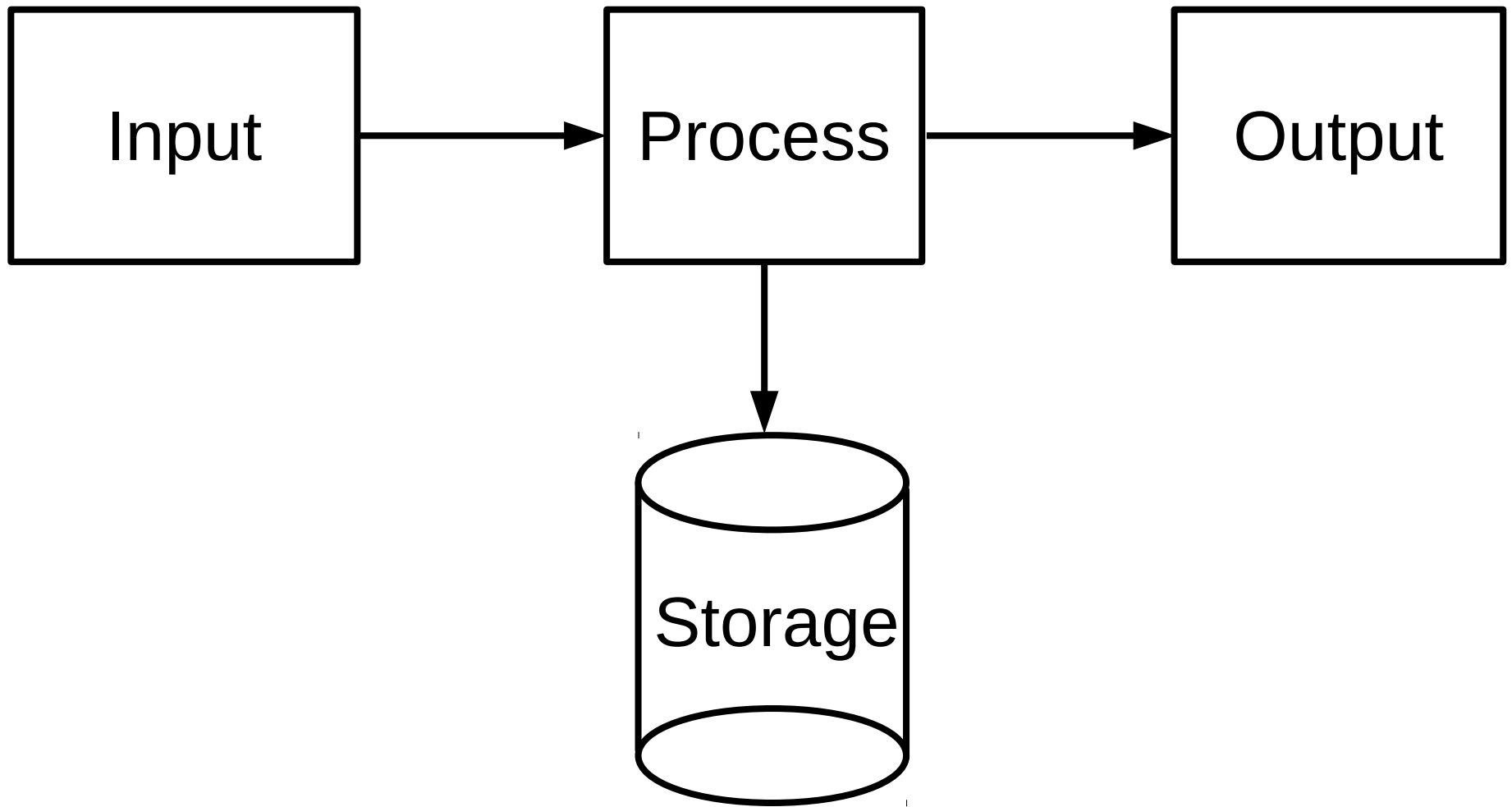
```
import os
import sys
from contextlib import redirect_stdout
```

```
STDOUT_FD = sys.stdout.fileno()
```

```
new_list = ['pybr', 'natal', '2018', 'palestras']
```

```
count = 0
```

```
with open('python_brasil.txt', 'w') as fhand:
    with redirect_stdout(fhand):
        for item in new_list:
            count = count + 1
            os.write(STDOUT_FD,
                      bytes('Line written: {}'.format(count) + '\n', 'utf8'))
            fhand.write(str(item)+'\n')
```



Exportando para arquivo, formato CSV

- O que é um arquivo CSV?

É um arquivo que contém conjuntos de dados de texto simples separados por vírgulas (**comma separated values**). **Arquivos CSV** muitas vezes são abertos por programas de planilhas como Microsoft Excel ou **OpenOffice Calc** para serem organizados em células, ou são utilizados para transferir dados entre banco de dados.

Traduzido de: <https://fileinfo.com/extension/csv>

Exportando para arquivo, formato CSV

- Exemplos CSV:

Item	Cost	Sold	Profit
Keyboard	\$10.00	\$16.00	\$6.00
Monitor	\$80.00	\$120.00	\$40.00
Mouse	\$5.00	\$7.00	\$2.00
,,Total	\$48.00		

PUNE	20th October 2018
Zaria, Kaduna	17th March 2018
Marseille	7th October 2017
Sao Leopoldo	20th October 2018
Malate, Ilorin	26th October 2018
Douala	26th October 2018
Kuwait City	26th October 2018

Exportando para arquivo, formato CSV

- Ferramenta: módulo CSV da biblioteca padrão do Python (Standard Library)
 - Implementa classes para ler e escrever dados tabulares no formato CSV
- Usando `csv.writer` (escrever listas em arquivo CSV):

Exportando para arquivo, formato CSV

```
import csv
```

```
locations = [('PUNE', '20th October 2018'),  
             ('Sao Leopoldo', '20th October 2018'),  
             ('Oko , Anambra', '3rd November 2018'),  
             ('Abeokuta, Ogun', '3rd November 2018')]
```

```
filename = 'django_girls.csv'
```

```
with open(filename, 'w', newline='') as csvfile:  
    write_location = csv.writer(csvfile, delimiter=';', dialect='excel')  
  
    for city in locations:  
        write_location.writerow(city)
```

Exportando para arquivo, formato CSV

- Usando `csv.DictWriter`:
 - Mapeia **listas de dicionários** para linhas de saída a serem escritas no arquivo


```
import csv
```

```
locations = {'PUNE': '20th October 2018',  
            'Sao Leopoldo': '20th October 2018',  
            'Malate, Ilorin': '26th October 2018',  
            'Douala': '26th October 2018',  
            'Kuwait City': '26th October 2018',  
            'Lafia, Nasarawa': '27th October 2018',  
            'Delta': '28th October 2017'}
```

```
with open('dg_2018.csv', 'w', newline='') as csvfile:  
    fieldnames = ['City', 'Date']  
    writer = csv.DictWriter(csvfile, fieldnames=fieldnames)  
  
    writer.writeheader()
```

```
for name in locations.items():  
    writer.writerow({'City': name[0], 'Date': name[1]})
```

Exportando para arquivo, formato JSON

- O que é um arquivo JSON?

É um arquivo que **armazena estruturas de dados** simples e objetos em um formato de Notação de Objeto JavaScript (ou JavaScript Object Notation, **JSON**) que é **um formato padronizado de intercâmbio de dados**. É usado primariamente para **transmitir dados entre uma aplicação web e um servidor**. Arquivos JSON são leves, baseados em texto, legíveis para humanos, e podem ser editados em um editor de texto.

Traduzido de: <https://fileinfo.com/extension/json>

Exportando para arquivo, formato JSON

- JSON é construído em duas¹ estruturas:
 - Lista ordenada de valores (array ou vetor)
 - Coleção de pares nome/valor (objetos)

Array	Objetos
<pre>["Bayelsa", "Mbale", "Salerno", "Sokoto", "Bauchi", "Mendoza"]</pre>	<pre>{ "current_user_url": "https://api.github.com/user", "emojis_url": "https://api.github.com/emojis", "events_url": "https://api.github.com/events", "feeds_url": "https://api.github.com/feeds", "team_url": "https://api.github.com/teams", "user_url": "https://api.github.com/users/{user}" }</pre>

¹ Fonte: <https://json.org>

Exportando para arquivo, formato JSON

- Fácil de traduzir objetos Python de e para strings de dados JSON em arquivos.
- Python possui a biblioteca *json* para parsear o formato e ler os dados dos objetos.

`json.load()`



Deserializa um arquivo **texto** ou **binário** contendo JSON para um objeto Python.

`json.loads()`



Deserializa *s* (str, bytes ou instância bytearray) contendo JSON para um objeto Python.

`json.dump()`



Serializa *obj* como JSON stream para um objeto Python parecido-com-arquivo (file-like).

```
import json
```

```
import os
```

```
locations = [('PUNE', '20th October 2018'),  
             ('Sao Leopoldo', '20th October 2018'),  
             ('Oko , Anambra', '3rd November 2018'),  
             ('Abeokuta, Ogun', '3rd November 2018')]
```

```
file_path = os.getcwd() + '/' + 'test_django_girls' + '.json'
```

```
with open(file_path, 'w', encoding='utf-8') as json_file:  
    json.dump(locations, json_file, indent=1)
```

Exportando para arquivo, formato JSON

- Validando as estruturas JSON criadas:

<https://jsonvalidatortool.com>

Alguns problemas que você pode encontrar

- Caracteres estranhos:

[

[

"S\u00e3o Leopoldo",

"20th October 2018"],

[

"Niter\u00f3i",

"19th May 2018"],

[

"JO\u00c3O PESSOA",

"May 2018"]

]



```
import json
```

```
import sys
```

```
locations = [('PUNE', '20th October 2018'),  
             ('São Leopoldo', '20th October 2018'),  
             ('Niterói', '19th May 2018'),  
             ('JOÃO PESSOA', 'May 2018'),  
             ('Oko , Anambra', '3rd November 2018'),  
             ('Abeokuta, Ogun', '3rd November 2018')]
```

```
file_path = sys.path[0] + '/' + 'test_django_girls' + '.json'
```

```
with open(file_path, 'w', encoding='utf-8') as json_file:  
    json.dump(locations, json_file, ensure_ascii=False, indent=1)
```


Alguns problemas que você pode encontrar

- “Células” que possuem o mesmo conteúdo do delimitador (por ex. vírgula) e quebram seu código ou dificultam leitura no Excel

```
import csv
```

```
locations = [('Oko , Anambra', '3rd November 2018'),  
             ('Abeokuta, Ogun', '3rd November 2018')]
```

```
with open('django_girls.csv', 'w', newline='') as csvfile:
```

```
    write_location = csv.writer(csvfile, delimiter=';', dialect='excel')
```

```
    for city in locations:
```

```
        write_location.writerow(city)
```

Alguns problemas que você pode encontrar

- Erros na quantidade de nomes dos campos (fieldnames) ao utilizar csv.DictWriter

```
writer = csv.DictWriter(csvfile,  
                        fieldnames=fieldnames,  
                        extrasaction='ignore')
```

E, afinal, dá pra abrir no Excel?

Bônus: Pandas

- Receitas e ferramentas para ler CSV:

<https://pandas.pydata.org/pandas-docs/stable/cookbook.html#csv>

- Exportar Series e DataFrames para CSV com o método `to_csv()`:

<https://pandas.pydata.org/pandas-docs/stable/io.html#io-store-in-csv>

- Exportar DataFrame para arquivo Excel:

<https://pandas.pydata.org/pandas-docs/stable/io.html#writing-excel-files-to-disk>

```
import csv
import pandas as pd
import selenium as sl
from selenium.webdriver.support.wait import WebDriverWait
```

```
driver = sl.webdriver.Firefox()
driver.get('https://djangogirls.org/events/')
delay = 60
locations_and_dates = dict()
```

```
get_city = WebDriverWait(driver, delay).until(
    lambda x: x.find_elements_by_class_name("overlay"))
```

```
for item in get_city:
    item_text = item.text.splitlines()
    try:
        city, date = item_text[0], item_text[1]
        locations_and_dates[city] = date
    except IndexError:
        continue
```

```
new_dic_dg = list()
```

```
for name, value in locations_and_dates.items():
    new_dic_dg.append({"city": name, "date": value})
```

```
new_df = pd.DataFrame(new_dic_dg).to_csv('dg_with_pandas.csv', index=False, sep=';')
```

```
driver.close()
```

O que mais?

Aprenda mais:

<https://eli.thegreenplace.net/2015/redirecting-all-kinds-of-stdout-in-python/>

https://docs.python.org/3/library/contextlib.html?highlight=redirect_stdout#contextlib.redirect_stdout

Obrigada!

 Renata D'Avila

 rsip22

Aberta a oportunidades!

renatadavila@pm.me