

Universidade Federal de Uberlândia
Engenharia de Controle e Automação / Engenharia Mecatrônica
Sistemas Embarcados II / Sistemas Digitais para Mecatrônica
Prof. Éder Alves de Moura
Semana 04 – Programação com Python



1. Essa atividade consiste em assistir os vídeos do curso introdutório de Python, disponível em:

<https://www.youtube.com/watch?v=k9TUPpGqYTo&list=PL-osiE8oTeTskrapNbzxhwoFUiLCjGgY7&index=2>

e reproduzir os códigos de cada aula. Os vídeos da lista estão numerados de 1 até 26. Você deve reproduzir **APENAS** os códigos disponíveis nos vídeos de 2 até 20 (exceto os dos vídeos 12 e 13). Você deverá criar um arquivo 'pyXX.py', onde o XX corresponde ao número do vídeo na lista, contendo todo o desenvolvimento proposto.

02)

```
1 message = 'Hello World'
2
3 for i in range(len(message)):
4     print('message(i) =', message[i])
5     print('i= ', i)
6     print("");
7
8
9 message = message.replace('World', "Universe")
10 print(message)
11 print("")
12
13 greeting = 'Hello'
14 name = 'Michael'
15 message = '{}, {}. Welcome!'.format(greeting, name)
16 print(message)
17 print("")
18
19
20 message = f'{greeting}, {name.upper()}. Welcome!'
21 print(message)
22 print("")
23
24 print(dir(name))
```

```

message(i) = H
i= 0

message(i) = e
i= 1

message(i) = l
i= 2

message(i) = l
i= 3

message(i) = o
i= 4

message(i) = 
i= 5

message(i) = W
i= 6

message(i) = o
i= 7

message(i) = r
i= 8

message(i) = l
i= 9

message(i) = d
i= 10

Hello Universe

Hello, Michael. Welcome!

Hello, MICHAEL. Welcome!

['__add__', '__class__', '__contains__', '__delattr__', '__dir__', '__doc__', '__eq__',
'__format__', '__ge__', '__getattribute__', '__getitem__', '__getnewargs__', '__gt__',
'__hash__', '__init__', '__init_subclass__', '__iter__', '__le__', '__len__', '__lt__',
'__mod__', '__mul__', '__ne__', '__new__', '__reduce__', '__reduce_ex__', '__repr__',
'__rmod__', '__rmul__', '__setattr__', '__sizeof__', '__str__', '__subclasshook__',
'capitalize', 'casefold', 'center', 'count', 'encode', 'endswith', 'expandtabs',
'find', 'format', 'format_map', 'index', 'isalnum', 'isalpha', 'isascii', 'isdecimal',
'isdigit', 'isidentifier', 'islower', 'isnumeric', 'isprintable', 'isspace', 'istitle',
'isupper', 'join', 'ljust', 'lower', 'lstrip', 'maketrans', 'partition', 'replace',
'rfind', 'rindex', 'rjust', 'rpartition', 'rsplit', 'rstrip', 'split', 'splitlines',
'startswith', 'strip', 'swapcase', 'title', 'translate', 'upper', 'zfill']
PS C:\Users\Nicol\OneDrive\Bureau\travail\sistemas digitais>

```

03)

```

1  print('+', 3+2, ' ; -', 3-2, ' ; *', 3*2, ' ; /', 3/2, ' ; //', 3//2, ' ; **', 3**2, ' ; %', 3%2)
2  message = ""
3  for i in range(10):
4      message += "i = "
5      message += str(i)
6      message += ': '
7      if(i%2==0):
8          message += "par ; "
9      else:
10         message += "impar ; "
11 print("")
12 print(message)
13
14 num = 3.14159265359
15 message = ""
16 for i in range(5):
17     message += "i = "
18     message += str(i)
19     message += ': '
20     message += str(round(num, i))
21     message += '; '
22 print("")
23 print(message)
24
25 print('')
26 print(2>1)

```

```

+ 5 ; - 1 ; * 6 ; / 1.5 ; // 1 ; ** 9 ; % 1

i = 0: par ; i = 1: impar ; i = 2: par ; i = 3: impar ; i = 4: par ; i = 5: impar ; i = 6: par ; i = 7: impar ; i = 8: par ; i = 9: impar ;

i = 0: 3.0; i = 1: 3.1; i = 2: 3.14; i = 3: 3.142; i = 4: 3.1416;

True

```

04)

```
1  lista = [1, 2, 5, 4]
2
3  for i in lista:
4      print(i)
5  print('')
6
7  for i in range(len(lista)):
8      print(i, lista[i])
9      print('')
10
11 print('-1', lista[-1])
12 print(":2", lista[:2])
13 print(["2:", lista[2:]])
14
15 print('Append, insert and extend')
16 lista.append(6)
17 print(lista)
18 lista.insert(2,7)
19 print(lista)
20
21 lista2 = [8,9]
22 lista.append(lista2)
23 print(lista)
24 lista = [1, 2, 5, 4]
25 lista.extend(lista2)
26 print(lista)
27
28 print("")
29 print('pop')
30 lista.pop()
31 print(lista)
32 pop = lista.pop()
33 print(lista, pop)
34
35 print("")
36 print('sort, inversa and index')
37 lista.sort()
38 print(lista)
39 lista.reverse()
40 print(lista, lista.index(2))
41
42 print("")
43 print('str operations')
44 lista = ['a', 'b', 'c', 'd']
45 print(' ', '.join(lista))
46 print(' ', '.join(lista).split(' ', ))
47
```

```
42 print("")
43 print('str operations')
44 lista = ['a', 'b', 'c', 'd']
45 print(' '.join(lista))
46 print(' '.join(lista).split(' '))
47
48 print("")
49 print("list1 = list2 ou list1 = list2.copy()")
50 list1 = [1, 2, 3, 4]
51 list2 = list1
52 print(list1, list2)
53 list1[0] = 0
54 list1[3] = 5
55 print(list1, list2)
56
57
58 print("")
59 list1 = [1, 2, 3, 4]
60 list2 = list1.copy()
61 print(list1, list2)
62 list1[0] = 0
63 list1[3] = 5
64 print(list1, list2)
65
66 #tuple nao pode ser modificado, so consultado e os elementos de set não são organizados
67 emptyList = []
68 emptyList = list()
69 emptyTuple = ()
70 emptyTuple = tuple()
71 emptySet = {}
72 emptySet = set()
```

```

1
2
5
4

0 1

1 2

2 5

3 4

-1 4
:2 [1, 2]
2: [5, 4]
Append, insert and extend
[1, 2, 5, 4, 6]
[1, 2, 7, 5, 4, 6]
[1, 2, 7, 5, 4, 6, [8, 9]]
[1, 2, 5, 4, 8, 9]

pop
[1, 2, 5, 4, 8]
[1, 2, 5, 4] 8

sort, inversa and index
[1, 2, 4, 5]
[5, 4, 2, 1] 2

str operations
a, b, c, d
['a', 'b', 'c', 'd']

list1 = list2 ou list1 = list2.copy()
[1, 2, 3, 4] [1, 2, 3, 4]
[0, 2, 3, 5] [0, 2, 3, 5]

[1, 2, 3, 4] [1, 2, 3, 4]
[0, 2, 3, 5] [1, 2, 3, 4]

```

05)

```

1  student = {'name': 'Jhon', 'age': 25, 'courses': ['Math', 'CompSci']}
2  print(student['name'])
3  print(student['courses'])
4  print(student)
5  print(student.get('phone'))
6  print(student.get('phone', 'not found'))
7
8  student['phone'] = '555-5555'
9  student['name'] = 'Jane'
10
11 print(student)
12
13 print("")
14
15 student.update({'name': "aaa", 'age': 26, 'phone': '556-6689'})
16 print(student)
17
18 age = student.pop('age')
19
20 print(student, age)
21
22 print("")
23 print(len(student))
24 print('keys:', student.keys())
25 print('values:', student.values())
26 print('items:', student.items())
27
28 print("")
29 ✓ for key, value in student.items():
30     print(key, value)

```

```

Jhon
['Math', 'CompSci']
{'name': 'Jhon', 'age': 25, 'courses': ['Math', 'CompSci']}
None
not found
{'name': 'Jane', 'age': 25, 'courses': ['Math', 'CompSci'], 'phone': '555-5555'}

{'name': 'aaa', 'age': 26, 'courses': ['Math', 'CompSci'], 'phone': '556-6689'}
{'name': 'aaa', 'courses': ['Math', 'CompSci'], 'phone': '556-6689'} 26

3
keys: dict_keys(['name', 'courses', 'phone'])
values : dict_values(['aaa', ['Math', 'CompSci'], '556-6689'])
items : dict_items([('name', 'aaa'), ('courses', ['Math', 'CompSci']), ('phone', '556-6689')])

name aaa
courses ['Math', 'CompSci']
phone 556-6689
PS C:\Users\Wicol\OneDrive\Bureau\travail\sistemas digitais>

```

06)

```
1 conditions = [False, None, 0, 1, 10, True]
2
3 for condition in conditions:
4     print('condition =', str(condition))
5     if condition:
6         print('evalutaion : True')
7     else:
8         print('evaluation : False')
9     print("")
```

```
condition = False
evaluation : False

condition = None
evaluation : False

condition = 0
evaluation : False

condition = 1
evalutaion : True

condition = 10
evalutaion : True

condition = True
evalutaion : True
```

07)


```
1  nums = [1, 2, 3, 4, 5, 6]
2
3
4  for num in nums:
5      print(num)
6      if num==3:
7          break
8      elif num == 2:
9          continue
10     print('reloop')
11
12  x = 0
13  while x < 10: # ou True
14      if x==5:
15          break
16      print(x)
17      x +=1
18
```

```
1
reloop
2
3
0
1
2
3
4
```

08)

```

1  def hello_func():
2      """ function that returns hello world"""
3      print('Hello World !')
4      print("")
5
6  hello_func()
7  print(hello_func)
8
9
10 def student_info(*args, **kwargs):
11     print(args)
12     print(kwargs)
13
14
15 courses = ["Math", "Art"]
16 info = {'name': "Jhon", 'age': 22}
17 print("")
18 student_info(courses, info)
19 print("")
20 student_info(*courses, **info)
21
22

```

```

Hello World !

<function hello_func at 0x000001A648FF6D30>

(['Math', 'Art'], {'name': 'Jhon', 'age': 22})
{}

('Math', 'Art')
{'name': 'Jhon', 'age': 22}

```

09)

```

import my_module

courses = ['History', 'Math', 'Physics', 'CompSci']

index = my_module.find_index(courses, 'Math')
print(index)

print("")
from my_module import find_index, test
import sys

index = find_index(courses, 'Math')
print(index, test)

print("")
print(sys.path)

print("")
import os
print(os.getcwd())

import webbrowser
import hashlib
webbrowser.open("https://www.youtube.com/watch?v=CqvZ3vGoGs0&list=PL-osiE80TeTskrapNbzxhwoFUiLCjGgY7&index=9&ab_channel=CoreySchafer")

```

```

Imported my module
1
1 Test String

['C:\\Users\\Nicol\\OneDrive\\Bureau\\travail\\systemas digitais', 'C:\\Program Files\\WindowsApps\\PythonSoftwareFoundation.Python.3.8.3.8.2880.0_x64__qbz5n2kfra8p0\\python38.zip', 'C:\\Program Files\\WindowsApps\\PythonSoftwareFoundation.Python.3.8.3.8.2880.0_x64__qbz5n2kfra8p0\\DLLs', 'C:\\Program Files\\WindowsApps\\PythonSoftwareFoundation.Python.3.8.3.8.2880.0_x64__qbz5n2kfra8p0\\lib', 'C:\\Users\\Nicol\\AppData\\Local\\Microsoft\\WindowsApps\\PythonSoftwareFoundation.Python.3.8.3.8.2880.0_x64__qbz5n2kfra8p0', 'C:\\Users\\Nicol\\AppData\\Local\\Packages\\PythonSoftwareFoundation.Python.3.8.3.8.2880.0_x64__qbz5n2kfra8p0\\LocalCache\\Local-packages\\Python38\\site-packages', 'C:\\Program Files\\WindowsApps\\PythonSoftwareFoundation.Python.3.8.3.8.2880.0_x64__qbz5n2kfra8p0\\lib\\site-packages']

C:\\Users\\Nicol\\OneDrive\\Bureau\\travail\\systemas digitais

```

10)

```

1 import os
2
3 print(os.getcwd())
4
5 os.chdir(r'C:\Users\Nicol\OneDrive\Bureau\travail\systemas digitais')
6
7 print(os.getcwd())
8
9
10 os.mkdir("OS-demo-2")
11
12 print(os.listdir())
13 os.rename('OS-demo-2', 'OS-demo-1')
14 print(os.listdir())
15 os.rmdir('OS-demo-1')
16
17
18 from datetime import datetime
19 print("")
20 mod_time = os.stat('demo.txt').st_mtime
21 print(datetime.fromtimestamp(mod_time))
22
23
24 print("")
25 for dirpath, dirnames, filenames in os.walk(r'C:\Users\Nicol\OneDrive\Bureau\travail\systemas digitais'):
26     print('Current Path:', dirpath)
27     print('Directories:', dirnames)
28     print('Files', filenames)
29     print()
30
31
32 print(os.environ.get('HOME'))
33

```

```

C:\Users\Nicol\OneDrive\Bureau\travail\systemas digitais
['demo.txt', 'my_module.py', 'OS-demo-2', 'py02.py', 'py03.py', 'py04.py', 'py05.py', 'py06.py', 'py07.py', 'py08.py', 'py09.py', 'py10.py', 'Semana 02 - Ambiente de Programacao Linux.pdf', 'semana2.docx', 'semana4.docx', '__pycache__', '-$semana4.docx']
['demo.txt', 'my_module.py', 'OS-demo-1', 'py02.py', 'py03.py', 'py04.py', 'py05.py', 'py06.py', 'py07.py', 'py08.py', 'py09.py', 'py10.py', 'Semana 02 - Ambiente de Programacao Linux.pdf', 'semana2.docx', 'semana4.docx', '__pycache__', '-$semana4.docx']

2022-01-05 21:26:04.959069

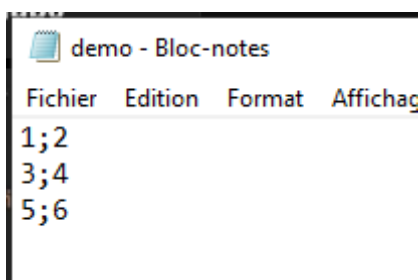
Current Path: C:\Users\Nicol\OneDrive\Bureau\travail\systemas digitais
Directories ['__pycache__']
Files ['demo.txt', 'my_module.py', 'py02.py', 'py03.py', 'py04.py', 'py05.py', 'py06.py', 'py07.py', 'py08.py', 'py09.py', 'py10.py', 'Semana 02 - Ambiente de Programacao Linux.pdf', 'semana2.docx', 'semana4.docx', '-$semana4.docx']

Current Path: C:\Users\Nicol\OneDrive\Bureau\travail\systemas digitais\_pycache__
Directories []
Files ['my_module.cpython-38.pyc']

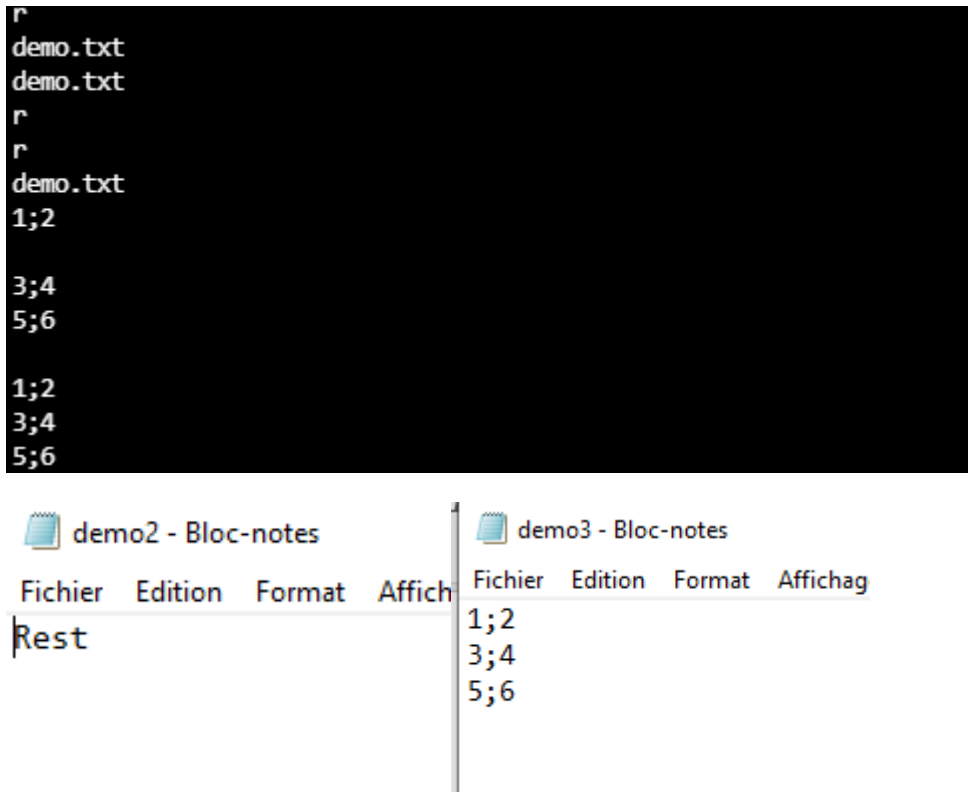
None

```

11)



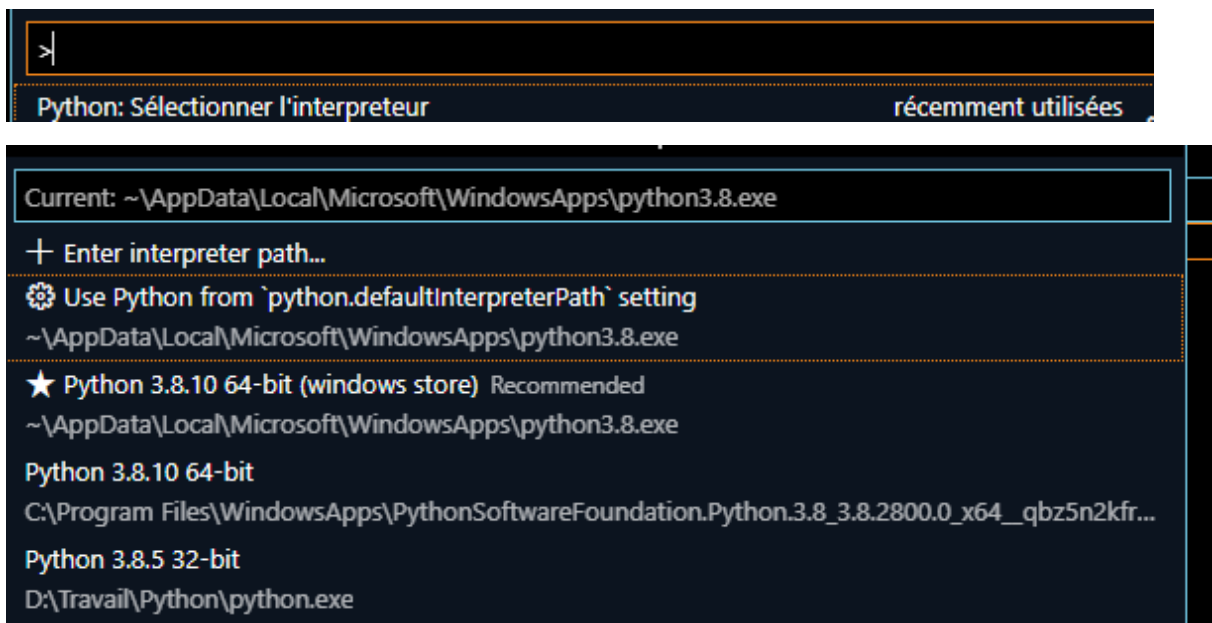
```
1  f = open('demo.txt', 'r')
2  print(f.mode)
3  print(f.name)
4  f.close()
5
6  print(f.name)
7  print(f.mode)
8
9
10 with open('demo.txt', 'r') as f:
11     print(f.mode)
12     print(f.name)
13     f_contents = f.readline()
14     print(f_contents)
15
16     f_contents = f.readline()
17     print(f_contents, end = '')
18     f_contents = f.readline()
19     print(f_contents, end = '')
20
21
22 print('')
23 print('')
24
25
26 with open('demo.txt', 'r') as f:
27     for line in f:
28         print(line, end='')
29
30 with open('demo2.txt', 'w') as f:
31     f.write('Test')
32     f.seek(0)
33     f.write('R')
34
35 with open('demo.txt', 'r') as rf:
36     with open('demo3.txt', 'w') as wf:
37         for line in rf:
38             wf.write(line)
39
```



12) Para achar o local do interpreter python com o Visual Studio Code, eu faço:

Ctrl+Maj+P >> Python: Sélectionner l'interpreteur >> Python 3.8.10 60-bit (windows store)

Tambem tenho outras versões do Python instaladas no meu computador por se for preciso



14)

```
import os

os.chdir(r'C:\Users\Nicol\OneDrive\Bureau\travail\systemas digitaïs')

for f in os.listdir():
    file_name, file_ext = os.path.splitext(f)
    print(file_name, file_ext)
    #Pluto - Our Solar System - #10.mp4
    #f_title, f_course, f_num = file_name.split('-')
    #f_title.strip()
    #f_course.strip()
    #f_num.strip()
    #print('{}-{}-{}{}'.format(f_num, f_course, f_title, f_ext))
    #10-Our Solar System-Pluto.mp4
```

```
demo .txt
['demo']
demo2 .txt
['demo2']
demo3 .txt
['demo3']
my_module .py
['my_module']
py02 .py
['py02']
py03 .py
['py03']
py04 .py
['py04']
py05 .py
['py05']
py06 .py
['py06']
py07 .py
['py07']
py08 .py
['py08']
py09 .py
['py09']
py10 .py
['py10']
py11 .py
['py11']
py12 .py
['py12']
py14 .py
['py14']
Semana 02 - Ambiente de Programacao Linux .pdf
['Semana 02 ', ' Ambiente de Programacao Linux']
semana2 .docx
['semana2']
semana4 .docx
['semana4']
__pycache__
['__pycache__']
~$semana4 .docx
['~$semana4']
~\RL0867 .tmp
['~\RL0867']
```

15)

```

1  import csv
2
3  with open('names.csv', 'r') as csv_file:
4      csv_reader = csv.reader(csv_file)
5      #next(csv_reader) #pula a linha 1
6      with open('new_names.csv', 'w') as new_file:
7          #field_names = ['first_name', 'last_name', 'email']
8
9          #csv_writer = csv.DictWriter(new_file, fieldnames=field_names, delimiter =
10         csv_writer = csv.DictWriter(new_file, delimiter = '\t')
11         #csv_writer.writeheader()
12         for line in csv_reader:
13             csv_writer.writerow(line)
14
15

```

16)

```

1  import csv
2
3  html_output = ''
4  names = []
5
6  with open('patrons.csv', 'r') as data_file:
7      csv_data= csv.reader(data_file)
8
9      print(csv_data)
10     next(csv_data)
11     next(csv_data)
12     #print(list(csv_data))
13     for line in csv_data:
14         line = line[0][1:].split(';')
15         if line[0] == 'No Reward':
16             break
17         names.append(f"{line[0]} {line[1]}")
18
19
20 html_output += f'<p>There are currently {len(names)} public contributors. Thank You !</p>'
21 html_output += '\n<ul>'
22
23 for name in names:
24     html_output += f'\n\t<li>{name}</li>'
25
26 html_output += '\n</ul>'
27
28 print(html_output)

```



```
<_csv.reader object at 0x000001FC06056280>
<p>There are currently 30 public contributors. Thank You !</p>
<ul>
  <li>John Doe</li>
  <li>Dave Smith</li>
  <li>Mary Jacobs</li>
  <li>Jane Stuart</li>
  <li>Tom Wright</li>
  <li>Steve Robinson</li>
  <li>Nicole Jacobs</li>
  <li>Jane Wright</li>
  <li>Jane Doe</li>
  <li>Kurt Wright</li>
  <li>Kurt Robinson</li>
  <li>Jane Jenkins</li>
  <li>Neil Robinson</li>
  <li>Tom Patterson</li>
  <li>Sam Jenkins</li>
  <li>Steve Stuart</li>
  <li>Maggie Patterson</li>
  <li>Maggie Stuart</li>
  <li>Jane Doe</li>
  <li>Steve Patterson</li>
  <li>Dave Smith</li>
  <li>Sam Wilks</li>
  <li>Kurt Jefferson</li>
  <li>Sam Stuart</li>
  <li>Jane Stuart</li>
  <li>Dave Davis</li>
  <li>Sam Patterson</li>
  <li>Tom Jefferson</li>
  <li>Jane Stuart</li>
  <li>Maggie Jefferson</li>
</ul>
```

17)

```

py17.py > ...
1  import datetime
2  import pytz
3  tday = datetime.date.today()
4  print(tday.weekday())
5  print(tday.isoweekday())
6
7  tdelta = datetime.timedelta(days=7)
8
9  print(tday-tdelta)
10
11 bday = datetime.date(2016,9,24)
12
13 till_bday = bday-tday
14 print(till_bday)
15
16 print("")
17 t = datetime.time(9,30,45,100000)
18 print(t.hour)
19
20 dt_today = datetime.datetime.today()
21 dt_now = datetime.datetime.now()
22 dt_utcnow = datetime.datetime.utcnow()
23
24 print(dt_today, dt_now, dt_utcnow)
25
26 print()
27 dt_utcnow = datetime.datetime(2016, 7, 27, 12, 30, 45, tzinfo=pytz.UTC)
28 print(dt_utcnow)
29 dt_mtn = dt_utcnow.astimezone(pytz.timezone('US/Mountain'))
30

```

```

2
3
2021-12-29
-1929 days, 0:00:00

9
2022-01-05 22:49:11.813988 2022-01-05 22:49:11.813988 2022-01-06 01:49:11.813988

2016-07-27 12:30:45+00:00

```

18)

```

1  import builtins
2
3  print(dir(builtins))
4
5  m = min([5,1.4,2,3])
6  print(m)
7
8  def test(z):
9      x = 'local x'
10     print(z)
11
12     test('local z')
13
14     def outer():
15         x = 'outer x'
16
17         def inner():
18             x = 'inner x'
19             print(x)
20
21         inner()
22         print(x)
23
24     outer()

```

```

['ArithmeticError', 'AssertionError', 'AttributeError', 'BaseException', 'BlockingIOError', 'BrokenPipeError', 'BufferError', 'BytesWarning', 'ChildProcessError', 'ConnectionResetError', 'DeprecationWarning', 'EOFError', 'Ellipsis', 'EnvironmentError', 'Exception', 'False', 'FileExistsError', 'FileNotFoundError', 'ImportError', 'ImportWarning', 'IndentationError', 'IndexError', 'InterruptedError', 'IsADirectoryError', 'KeyError', 'KeyboardInterrupt', 'LookupError', 'NotADirectoryError', 'NotImplemented', 'NotImplementedError', 'OSError', 'OverflowError', 'PendingDeprecationWarning', 'PermissionError', 'ProcessLookupError', 'RuntimeWarning', 'StopAsyncIteration', 'StopIteration', 'SyntaxError', 'SyntaxWarning', 'SystemError', 'SystemExit', 'TabError', 'TimeoutError', 'UnicodeDecodeError', 'UnicodeError', 'UnicodeTranslateError', 'UnicodeWarning', 'UserWarning', 'ValueError', 'Warning', 'WindowsError', 'ZeroDivisionError', '_NameError', '_NameWarning', '_PackageError', '_SpecError', '_abs', '_all', '_any', '_ascii', '_bin', '_bool', '_breakpoint', '_bytearray', '_bytes', '_callable', '_chr', '_classmethod', '_dir', '_divmod', '_enumerate', '_eval', '_exec', '_execfile', '_exit', '_filter', '_float', '_format', '_frozenset', '_getattr', '_globals', '_hasattr', '_hash', '_help', '_len', '_license', '_list', '_locals', '_map', '_max', '_memoryview', '_min', '_next', '_object', '_oct', '_open', '_ord', '_pow', '_print', '_property', '_quit', '_range', '_e', '_sorted', '_staticmethod', '_str', '_sum', '_super', '_tuple', '_type', '_vars', '_zip']
1.4
local z
inner x
outer x

```

20)

```

2
3  try:
4      f = open('testfile.txt')
5  except FileNotFoundError as e :
6      print('Sorry, This file does not exist')
7      print(e)
8  except Exception as e:
9      print(e)
10 else:
11     print(f.read())
12     f.close()
13 finally:
14     print("executing Finally...")

```

```
Sorry, This file does not exist
[Errno 2] No such file or directory: 'testfile.txt'
executing Finally...
```

22) O pip já foi configurado e instalei os módulos no meu computador, pois eu precisei dele para instalar as bibliotecas comuns.

Para instalar ele usei 'pip install all' e depois usei 'pip install {nome biblioteca}'

23)

```
PS C:\Users\Nicol\OneDrive\Bureau\travail\sistemas digitais> pip install requests
Collecting requests
  Downloading requests-2.27.1-py2.py3-none-any.whl (63 kB)
    |████████████████████| 63 kB 309 kB/s
Requirement already satisfied: certifi>=2017.4.17 in c:\users\nicol\appdata\local\pac
kages\pythonsoftwarefoundation.python.3.8_qbz5n2kfra8p0\localcache\local-packages\pyt
hon38\site-packages (from requests) (2020.6.20)
Collecting idna<4,>=2.5
  Downloading idna-3.3-py3-none-any.whl (61 kB)
    |████████████████████| 61 kB 961 kB/s
Collecting charset-normalizer~2.0.0
  Downloading charset_normalizer-2.0.10-py3-none-any.whl (39 kB)
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.7-py2.py3-none-any.whl (138 kB)
    |████████████████████| 138 kB 1.6 MB/s
Installing collected packages: urllib3, idna, charset-normalizer, requests
  WARNING: The script normalizer.exe is installed in 'C:\Users\Nicol\AppData\Local\Pa
ckages\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\LocalCache\local-packages\Py
thon38\Scripts' which is not on PATH.
  Consider adding this directory to PATH or, if you prefer to suppress this warning,
  use --no-warn-script-location.
Successfully installed charset-normalizer-2.0.10 idna-3.3 requests-2.27.1 urllib3-1.2
6.7
WARNING: You are using pip version 21.0.1; however, version 21.3.1 is available.
You should consider upgrading via the 'C:\Users\Nicol\AppData\Local\Microsoft\Windows
Apps\PythonSoftwareFoundation.Python.3.8_qbz5n2kfra8p0\python.exe -m pip install --up
grade pip' command.
```

```
1  import requests
2
3  r = requests.get('https://xkcd.com/353/')
4
5  print(r)
6  print(r.text)
7
8  r = requests.get('https://imgs.xkcd.com/comics/python.png')
9
10 with open('comic.png', 'wb') as f:
11     f.write(r.content)
12
13 print('status code',r.status_code)
14 print('headers',r.headers)
15 print('url',r.url)
16 print('text',r.text)
17
18 print()
19 print()
20
21 payload = {'username':'corey', 'password': 'testing'}
22 r = requests.post('https://httpbin.org/post', data=payload)
23 r_dict = r.json()
24
25 print(r_dict['form'])
```

```
</head>
<body>
<div id="topContainer">
<div id="topLeft">
<ul>
<li><a href="/archive">Archive</a></li>
<li><a href="https://what-if.xkcd.com">What If?</a></li>
<li><a href="https://blog.xkcd.com">Blog</a></li>
<li><a href="/how-to/">How To</a></li>
<li><a href="https://store.xkcd.com">Store</a></li>
<li><a rel="author" href="/about">About</a></li>
<li><a href="/atom.xml">Feed</a> &bull; <a href="/newsletter/">Email</a></li>
<li><a href="https://twitter.com/xkcd/">TW</a> &bull; <a href="https://www.facebook.com/TheXKCD/">FB</a> &bull; <a href="https://www.instagram.com/xkcd/">IG</a></li>
</ul>
</div>
```

```
url https://imgs.xkcd.com/comics/python.png
text PNG
```

IHDR+s0lv+gAMA0000cHRMZ000000B0:0co0pHYsd!!0!@00ebIIDATx@EQU]000\$000;
Ji00000000A00PK00QD[K(jX00h00F000000P00O00hk)E0000000E0-000B-A0000
000mmA(000000FX0nNhnj-000000DJI000P[000000v00000A0000Pi00000DK
'00vmmm00Kkhi~0Ki0000h0-0-/0000(0000BeP000Y000V000000jkk0jXB[[[K
0000hc\0)~0000NOuH00005000{000000l000-g00000{400G00N000ux000z00400
00000000{{00v000Amc00000005}00G00N/C00Z000000u0000000Y[00,/0000▼00abc
00000^L00勝0000▼000[['0]0z00000000S0>yyp000000vr000Ol000費0;0nv?0000

```

ng">https://imgs.xkcd.com/comics/python.png</a>

<div id="transcript" style="display: none">[[ Guy 1 is talking to Guy 2, who is float
ing in the sky ]]
Guy 1: You're flying! How?
Guy 2: Python!
Guy 2: I learned it last night! Everything is so simple!
Guy 2: Hello world is just <code>print "Hello, World!"</code>
Guy 1: I dunno... Dynamic typing? Whitespace?
Guy 2: Come join us! Programming is fun again! It's a whole new world up here!
Guy 1: But how are you flying?
Guy 2: I just typed <code>import antigravity</code>
Guy 1: That's it?
Guy 2: ...I also sampled everything in the medicine cabinet for comparison.
Guy 2: But i think this is the python.
{{ I wrote 20 short programs in Python yesterday. It was wonderful. Perl, I'm l
eaving you. }}</div>
</div>
<div id="bottom" class="box">

<map id="comicmap" name="comicmap">
<area shape="rect" coords="0,0,100,100" href="/150/" alt="Grownups"/>
<area shape="rect" coords="104,0,204,100" href="/730/" alt="Circuit Diagram"/>
<area shape="rect" coords="208,0,308,100" href="/162/" alt="Angular Momentum"/>
<area shape="rect" coords="312,0,412,100" href="/688/" alt="Self-Description"/>
<area shape="rect" coords="416,0,520,100" href="/556/" alt="Alternative Energy Revolu
tion"/>
</map>
<br />
<a href="//xkcd.com/1732/"></a>
<br />
</div>
<!--

```

```

{'password': 'testing', 'username': 'corey'}

```

2. A biblioteca numpy representa a base da manipulação de dados matriciais na linguagem Python e é amplamente utilizada em várias outras bibliotecas. O vídeo disponibilizado na sequência apresenta um tutorial sobre o seu uso na solução de vários problemas de engenharia.

Reproduza os códigos **APENAS** dos vídeos das seções 1, 2 e 3. Para cada vídeo, você deverá criar um arquivo 'numpy0X.py', onde o X corresponde à seção, e apresentar o desenvolvimento de todos os códigos desenvolvidos durante o tutorial.

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3  a1 = np.array([4,6,8,2])
4  a2 = np.zeros(10)
5  a3 = np.ones(4)
6  a4 = np.random.random(10)
7  a5 = np.random.randn(10)
8  a6 = np.linspace(0, 10, 100)
9  a7 = np.arange(0, 10, 0.2)
10 print("array operations")
11 print(a1*2)
12 print(2*a1>10)
13 print(1/a4 + a4)
14 plt.plot(a6, a6**2)
15 plt.show()
16 plt.hist(a4)
17 plt.show()
18 def f(x):
19     return x**2 * np.sin(x) / np.exp(-x)
20
21 plt.plot(a7, f(a7))
22
23 print()
24 print("Array indexing/slicing")
25 a1 = np.array([2,4,6,8,10])
26 print('a1[2]', a1[2])
27 print('a1[2:]', a1[2:])
28 print('a1[:-2]', a1[:-2])
29 print('a1[::2]', a1[::2])
30 print('a1>3', a1>3)
31 print('a1[a1>3]', a1[a1>3])
32 names = np.array(['Jim', 'Luke', 'Josh', 'Pete'])
33 first_letter_j = np.vectorize(lambda s: s[0])(names)=='J'
34 print(names[first_letter_j])
35 print('a1%4', a1%4)
36 print('a1%4==0', a1%4==0)
37 print('a1[a1%4==0]', a1[a1%4==0])
38

```



```

print()
print("Calculus/ statistical Functions")
a1 = 2*np.random.randn(10000) + 10
print('np.mean(a1)',np.mean(a1))
print('np.std(a1)',np.std(a1))
print('np.percentile(a1, 80)',np.percentile(a1, 80))
x = np.linspace(1, 10, 100)
y = 1/x**2 * np.sin(x)
dydx = np.gradient(y,x)
y_integral = np.cumsum(y) * (x[1]-x[0])
plt.plot(x,y)
plt.plot(x,dydx)
plt.plot(x,y_integral)

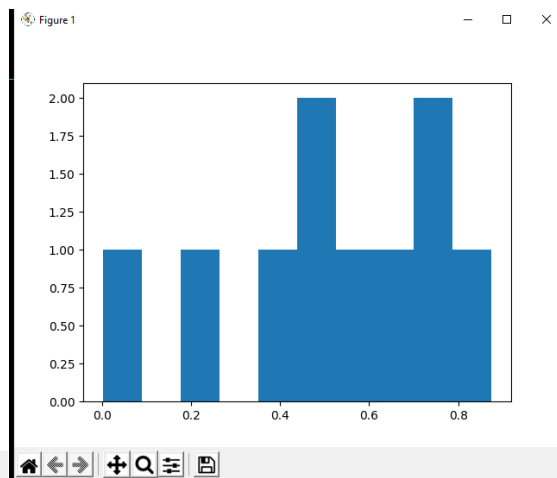
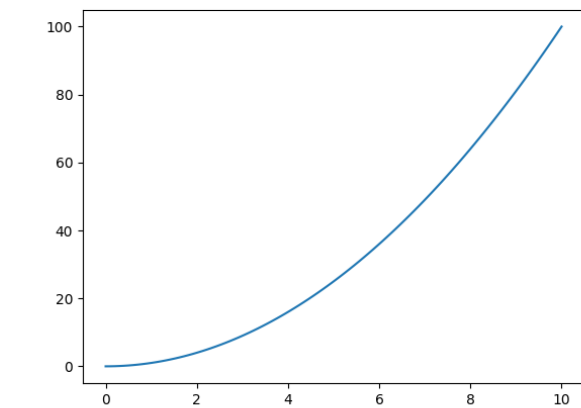
```

```

[False True True False]
[ 2.69288131 544.89303053  2.09126031  2.52596773  2.85468454
 2.34382387  2.11832054  2.01787167  4.48134455  2.21637365]

```

Figure 1



```

Array indexing/slicing
a1[2] 6
a1[2:] [ 6  8 10]
a1[:-2] [2 4 6]
a1[::2] [ 2  6 10]
a1>3 [False  True  True  True  True]
a1[a1>3] [ 4  6  8 10]
['Jim' 'Josh']
a1%4 [2 0 2 0 2]
a1%4==0 [False  True False  True False]
a1[a1%4==0] [4 8]

Calculus/ statistical Functions
np.mean(a1) 9.967125976961254
np.std(a1) 1.9931523563581341
np.percentile(a1, 80) 11.647024038799481

```

3. A biblioteca `matplotlib` é usada para a apresentação de gráficos e é semelhante ao comando `plot` do Matlab. O vídeo disponibilizado na sequência apresenta alguns exemplos básicos de uso.

Reproduza os códigos **APENAS** dos vídeos das seções de 1 até 6. Para cada vídeo, você deverá criar um arquivo '`plot0X.py`', onde o X corresponde à seção, e apresentar o desenvolvimento dos códigos desenvolvidos durante o tutorial.

```

1  import numpy as np
2  import matplotlib.pyplot as plt
3
4  x = [1,2,3,4,5,6,7,8,9,10]
5  y = [1,2,3,4,5,6,7,8,9,10]
6  plt.scatter(x,y)
7  plt.show()
8
9  x1 = np.arange(-1000,1000,1)
10
11 plt.plot(x1,x1**2)
12 plt.show()
13
14 plt.plot(x1, -x1**3+4)
15 plt.show()
16

```

