

```
8
9
0     default=1.0,
1
2         )
3
4
5     global_scale_setting = FloatProperty(
6         name="Scale",
7         min=0.01, max=1000.0,
8         default=1.0,
9     )
10
11
12     def execute(self, context):
13
14         # get the folder
15         folder_path = (os.path.dirname(self.filepath))
16
17         # get objects selected in the viewport
18         viewport_selection = bpy.context.selected_objects
19
20         # get export objects
21         obj_export_list = viewport_selection
22         if self.use_selection_setting == False:
23             obj_export_list = [i for i in bpy.context.scene.objects]
24
25         # deselect all objects
26         bpy.ops.object.select_all(action='DESELECT')
27
28         for item in obj_export_list:
29             item.select = True
30             if item.type == 'MESH':
31                 file_path = os.path.join(folder_path, "{}.obj".format(item.name))
32                 bpy.ops.export_scene.obj(filepath=file_path, use_selection=True,
33                                         axis_forward=self.axis_forward_setting,
34                                         axis_up=self.axis_up_setting,
35                                         use_animation=self.use_animation_setting,
36                                         use_mesh_modifiers=self.use_mesh_modifiers_setting,
37                                         use_edges=self.use_edges_setting,
38                                         use_smooth_groups=self.use_smooth_groups_setting,
39                                         use_smooth_groups_bitflags=self.use_smooth_groups_bitflags_setting,
40                                         use_normals=self.use_normals_setting,
41                                         use_uvs=self.use_uvs_setting,
42                                         use_materials=self.use_materials_setting,
```

P Y T H O N

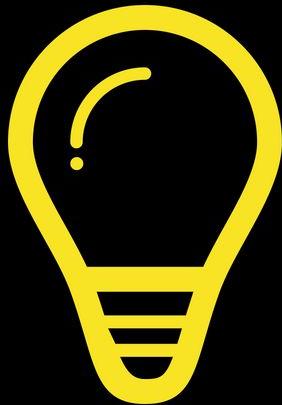
para iniciantes

GIANA DE ALMEIDA



GINA DE ALMEIDA

TI DA UAB NO IFSC - FPÓLIS
ENGAJADORA DA COMUNIDADE PYTHON
GRADUADA EM MECATRÔNICA INDUSTRIAL



NECESSIDADE

**PESSOAS QUE QUEREM APRENDER A PROGRAMAR, E AS QUE
POSSUEM DIFICULDADES DE SAIR DO NÍVEL BÁSICO.**

CARACTERÍSTICAS



OPEN SOURCE

Contribuição constante da
comunidade para vários
tipos de aplicações

CARACTERÍSTICAS

PRODUTIVIDADE + LEGIBILIDADE

Criada para produzir código bom, fácil de manter e de forma rápida



PRODUTIVIDADE

Criada para produzir
bom, fácil de manusear,
forma rápida

C A P

Python Developer's Guide

Python's Developer Guide can be found at <https://devguide.python.org/>.

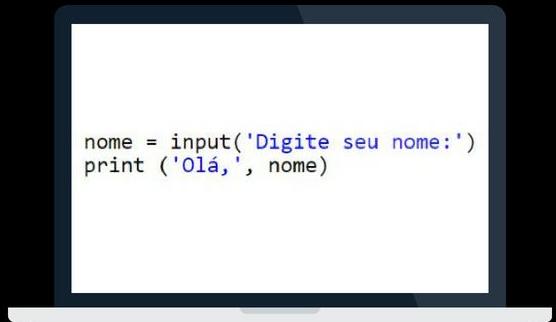
The pieces of documentation hosted here are:

- [PEPs](#) (Python Enhancement Proposals)
- [Buildbots](#)
- [Python.org Maintenance and Administration](#)

CARACTERÍSTICAS

SINTAXE SIMPLES

Faz com que a pessoa que está programando se preocupe mais com a lógica do que com a sintaxe.



CARACTERÍSTICAS

TIPAGEM DINÂMICA

Reconhece qual o tipo de variável logo que se atribui um valor a ela.

```
Python 3.6.2 (v3.6.2:5tcAsSob, Jul 8 2017, 04:57:36)
Type "help", "copyright", "credits" or "license" for more information
>>> numero = 1
>>> decimal = 0.1
>>> texto = 'um'
>>> vetor = [numero, decimal, texto]
>>> print(vetor)
[1, 0.1, 'um']
>>> type(numero)
<class 'int'>
>>> type(decimal)
<class 'float'>
>>> type(texto)
<class 'str'>
>>> type(vetor)
<class 'list'>
```

CARACTERÍSTICAS



MODULARIZAÇÃO

Estruturas completas podem ser desmontadas e divididas em componentes para reutilização em outros programas.

CARACTERÍSTICAS



FRAMEWORKS

Ferramentas pré prontas
para auxiliar em projetos
complexos

CARACTERÍSTICAS

DOCUMENTAÇÃO

Bem organizado, com opção
de diversos idiomas

CARACTERÍSTICAS

MULTIPARADIGMA

Pode ser utilizada como estruturada ou orientada a objetos por exemplo.

CARACTERÍSTICAS

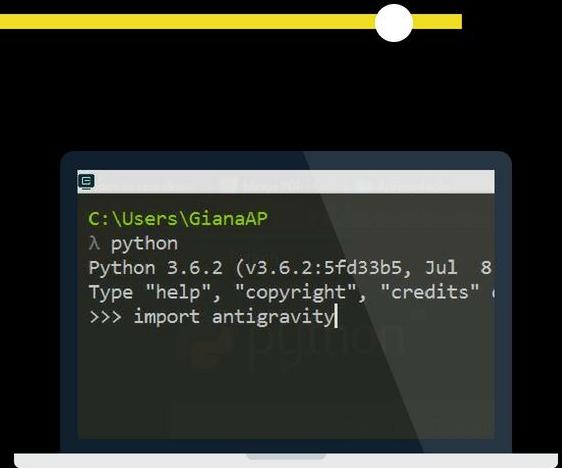
MULTIPLATAFORMA

Por ser interpretada, precisa apenas que o SO possua seu interpretador.

CARACTERÍSTICAS

DIVERTIDO

Easter eggs; Eventos locais, regionais, nacionais e internacionais;
Comunidade receptiva e maravilhosa!





MUITO MATERIAL

CUIDADO COM O EXCESSO, PODE
TE ATRAPALHAR MAIS DO QUE
AJUDAR



GAMES COM PYTHON

MÓDULO QUE FORNECE A API DA SDL E
FUNCIONALIDADES ADICIONAIS PARA
PROGRAMAÇÃO GRÁFICA



A large green Pygame logo is positioned at the top left. A cartoon dog icon is located to the right of the logo.

```
C:\Users\GianaAP\Desktop\Works\prog\Exemplos palestra\bola.py • (python) - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
bola.py
1 import sys, pygame
2 pygame.init()
3
4 tamanho      = largura, altura = 400, 640
5 velocidade   = [1, 1]
6 preto        = 0, 0, 0
7
8 tela         = pygame.display.set_mode(tamanho)
9 bola         = pygame.image.load("bola.bmp")
10 posicao_bola = bola.get_rect()
11
12 while True:
13     for event in pygame.event.get():
14         if event.type == pygame.QUIT:
15             sys.exit()
16
17     posicao_bola = posicao_bola.move(velocidade)
18
19     if (posicao_bola.left < 0) or (posicao_bola.right > largura):
20         velocidade[0] = -velocidade[0]
21
22     if (posicao_bola.top < 0) or (posicao_bola.bottom > altura):
23         velocidade[1] = -velocidade[1]
24
25     tela.fill(preto)
26     tela.blit(bola, posicao_bola)
27     pygame.display.flip()

Line 13, Column 41
```

EXEMPLO DE CÓDIGO - MOVIMENTAÇÃO DE OBJETO

pygame



EXEMPLO DE JOGO - SEVERANCE: BLADE OF DARKNESS

MATERIAL DE REFERÊNCIA:

INTRODUÇÃO:

<https://wiki.python.org.br/PyGameIntro>

TUTORIAL:

<https://bit.ly/2L3gHKJ>

PÁGINA OFICIAL:

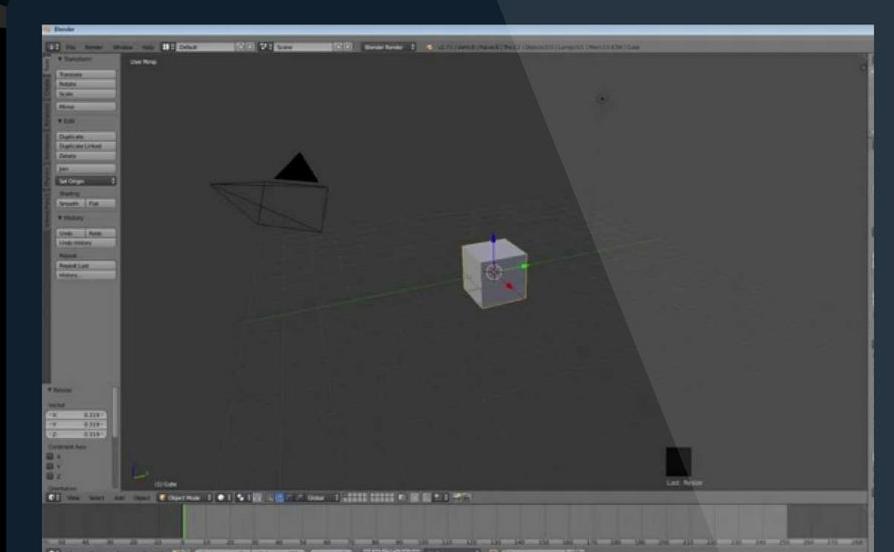
<https://www.pygame.org/docs/>



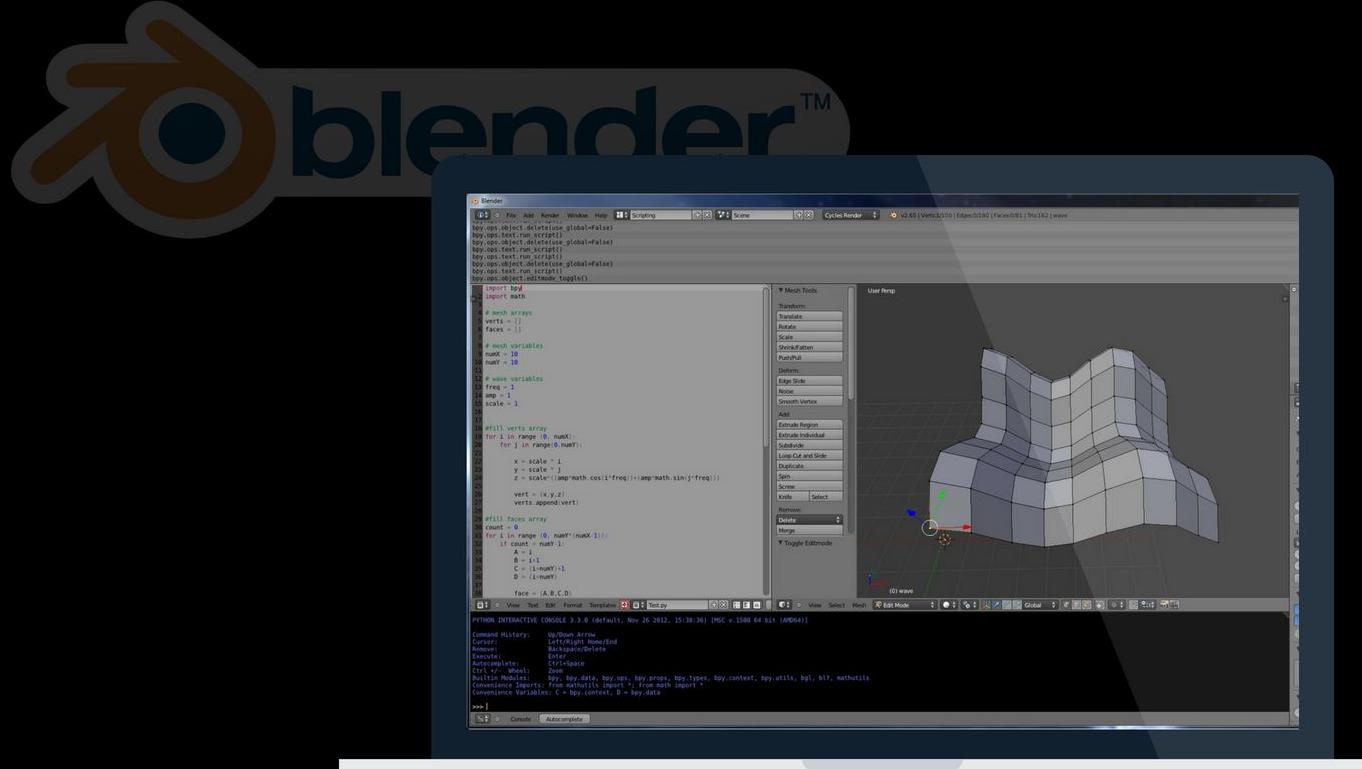


MODELAGEM 3D COM PYTHON

SOFTWARE OPEN SOURCE PARA
CRIAÇÃO DE MODELOS 3D



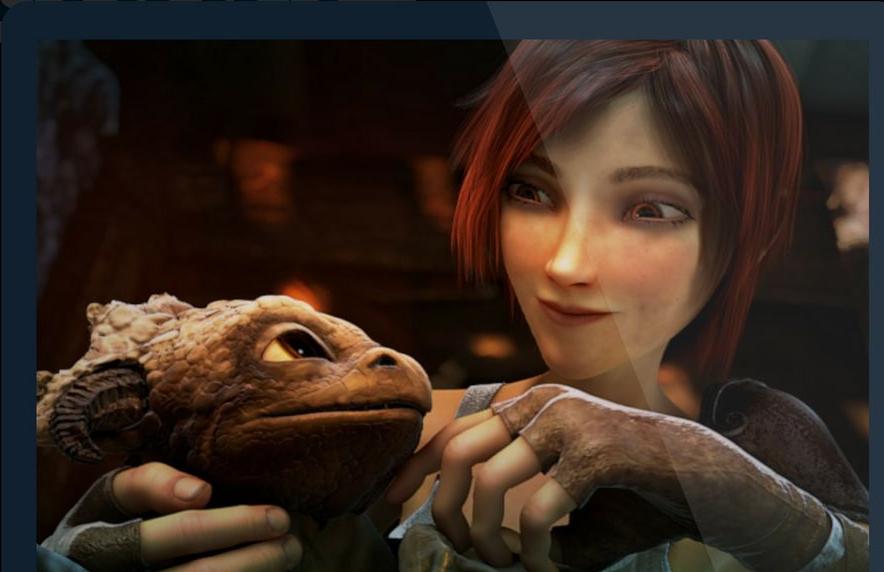
MODELAGEM PADRÃO



MODELAGEM POR SCRIPT



EXEMPLO DE ANIMAÇÃO - SINTEL



EXEMPLO DE ANIMAÇÃO - SINTEL

MATERIAL DE REFERÊNCIA:

INTRODUÇÃO - BLENDER:

<https://bit.ly/2KUv5EL>

TUTORIAL - BLENDER SCRIPTING:

<https://bit.ly/2NWD1Uz>

PÁGINA OFICIAL:

<https://www.blender.org/support/tutorials/>

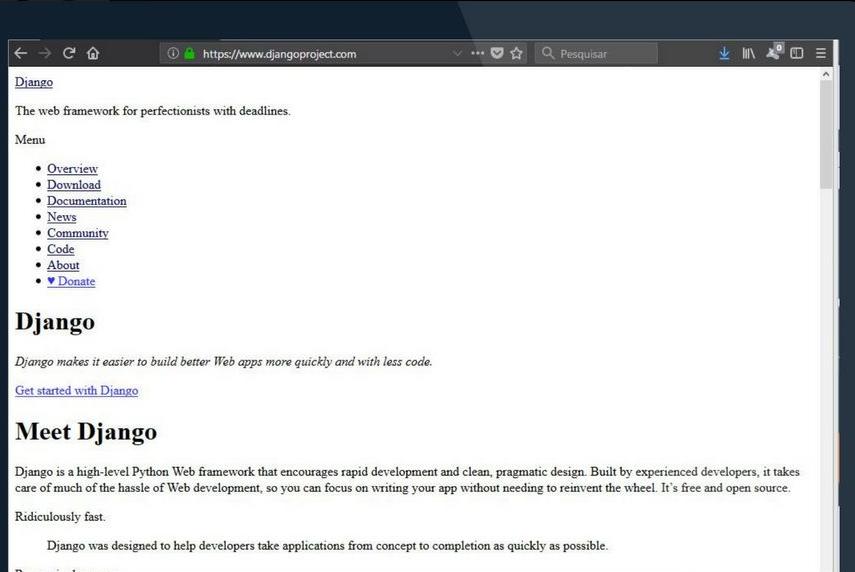




WEBSITES COM PYTHON

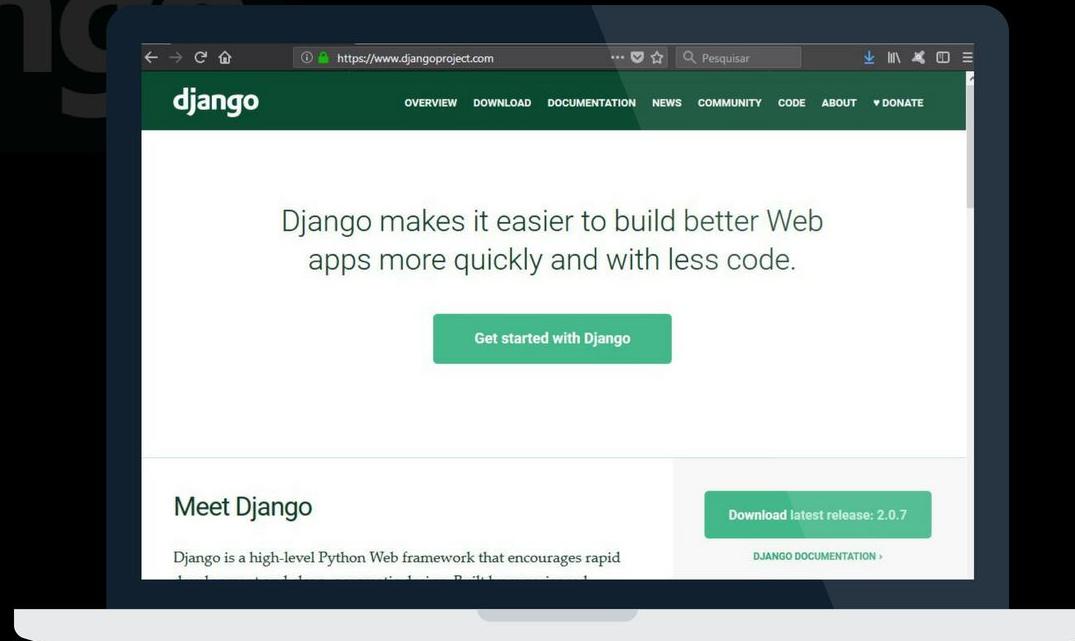
FRAMEWORK PARA A CRIAÇÃO DE
WEBSITES

django



H T M L

django



CSS

django

Django administration

Welcome, olasitarska

Home > Blog > Posts > Add post

Add post

Author: olasitarska

Title: Cras justo odio, dapibus ac facilisis in, egestas.

Text:

Cras mattis consectetur purus sit amet fermentum. Donec id elit non mi porta gravida at eget metus. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nullam quis risus eget urna mollis ornare vel eu leo. Curabitur blandit tempus porttitor. Cras mattis consectetur purus sit amet fermentum.

Fusce dapibus, tellus ac cursus commodo, tortor mauris condimentum nibh, ut fermentum massa justo sit amet risus. Curabitur blandit tempus porttitor. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec sed odio dui. Aenean eu leo quam. Pellentesque ornare sem lacinia quam venenatis vestibulum.

Created date: Date: 2014-07-07 Today |

Time: 23:07:34

Published date: Date: 2014-07-07

Time: 23:07:34

DJANGO

MATERIAL DE REFERÊNCIA:

INTRODUÇÃO - DJANGO
<https://bit.ly/2ufMPQG>

TUTORIAL DJANGO
<https://bit.ly/2hUG2p0>

PÁGINA OFICIAL
<https://docs.djangoproject.com/en/2.0/>

HTML CSS



HTML5 - TUTORIAL
<https://bit.ly/2hILOSq>

CSS3 - TUTORIAL
<https://bit.ly/2i1Y3Bg>

PÁGINAS OFICIAIS
<https://www.w3.org/TR/html52/>
<https://www.w3.org/Style/CSS/>

django



APLICATIVOS COM PYTHON

FRAMEWORK KIVY VOLTADO PARA
APLICAÇÕES.



The screenshot shows a dark-themed web page with a light blue header bar. The header bar contains the title "Widgets" in white font, followed by "Module: kivy.uix" and "Added in 1.0.0". Below the header, there is a brief description of what widgets are and their purpose. The main content area is titled "Kivy widgets can be categorized as follows:" and lists three categories with their respective sub-components:

- **UX widgets:** Classical user interface widgets, ready to be assembled to create more complex widgets.
Label, Button, CheckBox, Image, Slider, Progress Bar, TextInput, Toggle button, Switch, Video
- **Layouts:** A layout widget does no rendering but just acts as a trigger that arranges its children in a specific way. Read more on [Layouts here](#).
Anchor Layout, Box Layout, Float Layout, Grid Layout, PageLayout, RelativeLayout, Scatter Layout, Stack Layout
- **Complex UX widgets:** Non-atomic widgets that are the result of combining multiple classic widgets. We call them complex because their assembly and usage are not as generic as the classical widgets.
Bubble, Drop-Down List, FileChooser, Popup, Spinner, ListView, TabbedPanel, Video player, VKeyboard,

FUNCIONAMENTO BASEADO EM WIDGETS



```
class PongBall(Widget):
    velocity_x = NumericProperty(0)
    velocity_y = NumericProperty(0)
    velocity = ReferenceListProperty(velocity_x, velocity_y)

    def move(self):
        self.pos = Vector(*self.velocity) + self.pos

class PongApp(App):
    def build(self):
        game = PongGame()
        game.serve_ball()
        Clock.schedule_interval(game.update, 1.0 / 60.0)
        return game

if __name__ == '__main__':
    PongApp().run()
```



CÓDIGO SIMPLES, CROSS PLATAFORMA

MATERIAL DE REFERÊNCIA:

INTRODUÇÃO:

<https://bit.ly/2L1TC7L>

DOCUMENTAÇÃO EM PORTUGUÊS:

<https://bit.ly/2m4TrxD>

TUTORIAL:

<https://bit.ly/2N4ULMn>

PÁGINA OFICIAL

<https://kivy.org/#support>





PYTHON PARA MICROCONTROLADORES

HARDWARES, MICROPYTHON E
PROTÓCOLOS



ESP + MicroPython
monitorando sensor



XMPP

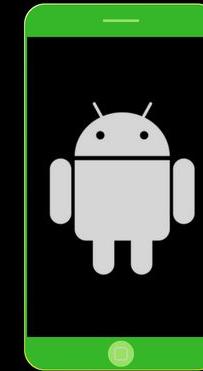


CoAP



PROTOCOLOS

INTERNET DAS COISAS



Celular recebendo informações para
determinada aplicação

MATERIAL DE REFERÊNCIA:

INTRODUÇÃO - MICROPYTHON:

<https://bit.ly/2zz4Ejo>

TUTORIAL

<https://bit.ly/2uqCeTg>

PÁGINA OFICIAL

<https://micropython.org/>

Obs: eu citei o ESP8266 na palestra,
mas roda em outros controladores
como por exemplo no Arduino ou ESP32

<https://bit.ly/2Jtfuau>





CIÊNCIA DE DADOS

MACHINE LEARNING, ESTATISTICA,
BIBLIOTECAS



- É uma linguagem de programação muito fácil de aprender.
- Linguagem de programação bastante popular nos meios acadêmicos, para o ensino de programação e para áreas de pesquisa.
- Possui IDEs bons, com muitos recursos e produtivos: Spyder, IPython Notebook, PyCharm, Eclipse, entre outros.
- Possui uma sintaxe moderna e enxuta. Possui paradigmas orientada a objetos e funcional. É uma linguagem de programação extremamente versátil.
- Possui framework de testes.
- Existe uma comunidade de desenvolvedores grande e bastante ativa. Para ter um parâmetro do tamanho desta comunidade, existem 91000 projetos hospedados no [GitHub](#).
- Grande quantidade de bibliotecas de aprendizado computacional, big data, análise estatística, computação numérica e visualização disponíveis para Python. Várias bibliotecas científicas desenvolvidas em outras linguagens de programação possuem um invólucro em Python, facilitando sua utilização.
- As principais bibliotecas para Ciência de Dados e Aprendizado Computacional são: NumPy, pandas, matplotlib e a scikit-learn.
- Possui vários pacotes para Aprendizado Profundo/Deep Learning: Lasagne, caffe, keras e tensorflow.
- Por ser uma linguagem de programação de propósito geral, o Python possui mais recursos para se criar uma aplicação final que possua partes de Aprendizado Computacional ou Ciência de Dados.

VANTAGENS DE USAR PYHTON PARA DATA SCIENCE [1]

MATERIAL DE REFERÊNCIA:

introdução:

<https://bit.ly/1KwK2pe>

CONCEITOS INICIAIS

<https://bit.ly/2ztnJU9>

ANTES DE USAR AS BIBLIOTECAS

<https://bit.ly/2L6Paoi>

BIBLIOTECAS:

<http://scikit-learn.org/>

<http://www.numpy.org/>

<https://www.scipy.org/>

<https://pandas.pydata.org/>

<https://www.statsmodels.org/stable/index.html>

<https://anaconda.org/anaconda/python>



```
8
9
0     default=1.0,
1
2         )
3
4
5     global_scale_setting = FloatProperty(
6         name="Scale",
7         min=0.01, max=1000.0,
8         default=1.0,
9     )
10
11
12     def execute(self, context):
13
14         # get the folder
15         folder_path = (os.path.dirname(self.filepath))
16
17         # get objects selected in the viewport
18         viewport_selection = bpy.context.selected_objects
19
20         # get export objects
21         obj_export_list = viewport_selection
22         if self.use_selection_setting == False:
23             obj_export_list = [i for i in bpy.context.scene.objects]
24
25         # deselect all objects
26         bpy.ops.object.select_all(action='DESELECT')
27
28         for item in obj_export_list:
29             item.select = True
30             if item.type == 'MESH':
31                 file_path = os.path.join(folder_path, "{}.obj".format(item.name))
32                 bpy.ops.export_scene.obj(filepath=file_path, use_selection=True,
33                                         axis_forward=self.axis_forward_setting,
34                                         axis_up=self.axis_up_setting,
35                                         use_animation=self.use_animation_setting,
36                                         use_mesh_modifiers=self.use_mesh_modifiers_setting,
37                                         use_edges=self.use_edges_setting,
38                                         use_smooth_groups=self.use_smooth_groups_setting,
39                                         use_smooth_groups_bitflags=self.use_smooth_groups_bitflags_setting,
40                                         use_normals=self.use_normals_setting,
41                                         use_uvs=self.use_uvs_setting,
42                                         use_materials=self.use_materials_setting,
```

P Y T H O N

para iniciantes

GIANA DE ALMEIDA

REFERÊNCIA DE LAYOUT

BANCO DE IMAGENS:
<https://www.pexels.com/>

LAYOUT:
<https://www.canva.com/>

