

MBA
USP
ESALQ

Desenvolvimento backend

Guilherme Lima

***A responsabilidade pela idoneidade, originalidade e licitude dos conteúdos didáticos apresentados é do professor.**

Proibida a reprodução, total ou parcial, sem autorização. Lei nº 9610/98

```

177     default="v",
178 )
179 global_scale_setting = FloatProperty(
180     name="Scale",
181     min=0.01, max=1000.0,
182     default=1.0,
183 )
184
185 def execute(self, context):
186
187     # get the folder
188     folder_path = (os.path.dirname(se
189
190     # get objects selected in the view
191     viewport_selection = bpy.context.s
192
193     # get export objects
194
195     select
196     == False.
197     i in bpy.context.scene.objects?
198
199     ion=
200
201     bin(f
202     obj(f
203     ax
204     ax
205     us
206     us
207
208     use_normals_setting,
209     use_normals_setting,
210     use_normals_setting,

```



Minhas redes

guilhermelimadev



guilherme-lima-developer





python

```
177         ),
178         default="v",
179     )
180     global_scale_setting = FloatProperty(
181         name="Scale",
182         min=0.01, max=1000.0,
183         default=1.0,
184     )
185
186     def execute(self, context):
187
188         # get the folder
189         folder_path = (os.path.dirname(self.filepath))
190
191         # get objects selected in the viewport
192         viewport_selection = bpy.context.selected_objects
193
194         # get export objects
195         obj_export_list = viewport_selection
196         if self.use_selection_setting == False:
197             obj_export_list = [i for i in bpy.context.scene.objects]
198
199         # deselect all objects
200         bpy.ops.object.select_all(action='DESELECT')
201
202         for item in obj_export_list:
203             item.select = True
204             if item.type == 'MESH':
205                 file_path = os.path.join(folder_path, "{}.obj".format(item.name))
206                 bpy.ops.export_scene.obj(filepath=file_path, use_selection=True,
207                                         axis_forward=self.axis_forward_setting,
208                                         axis_up=self.axis_up_setting,
209                                         use_animation=self.use_animation_setting,
210                                         use_mesh_modifiers=self.use_mesh_modifiers_setting,
211                                         use_edges=self.use_edges_setting,
212                                         use_smooth_groups=self.use_smooth_groups_setting,
213                                         use_smooth_groups_bitflags=self.use_smooth_groups_bitflags_setting,
214                                         use_normals=self.use_normals_setting,
215                                         use_uv=self.use_uv_setting,
216                                         use_materials=self.use_materials_setting,
```

Empresas



Empresas

Google, YouTube, Dropbox, Yahoo!, Zope Corporation, Walt Disney Feature Animation, Blender 3D, Pixar, NASA, NSA, Red Hat, Nokia, IBM, Netflix, Yelp, Intel, Spotify

Top Companies Using Python





Python é uma linguagem de programação de alto nível, interpretada de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte.





OBJETIVOS

Gustavo Henrique de Souza 373.625.128

OBJETIVOS

- Ambiente

OBJETIVOS

- Ambiente
- Criar um programa

OBJETIVOS

- Ambiente
- Criar um programa
- Tipos de dados

OBJETIVOS

- Ambiente
- Criar um programa
- Tipos de dados
- Variáveis

OBJETIVOS

- Ambiente
- Criar um programa
- Tipos de dados
- Variáveis
- Sintaxe

OBJETIVOS

- Ambiente
- Criar um programa
- Tipos de dados
- Variáveis
- Sintaxe
- Funções

OBJETIVOS

- Ambiente
- Criar um programa
- Tipos de dados
- Variáveis
- Sintaxe
- Funções
- De forma prática



python

```
177         ),
178         default="v",
179     )
180     global_scale_setting = FloatProperty(
181         name="Scale",
182         min=0.01, max=1000.0,
183         default=1.0,
184     )
185
186     def execute(self, context):
187
188         # get the folder
189         folder_path = (os.path.dirname(self.filepath))
190
191         # get objects selected in the viewport
192         viewport_selection = bpy.context.selected_objects
193
194         # get export objects
195         obj_export_list = viewport_selection
196         if self.use_selection_setting == False:
197             obj_export_list = [i for i in bpy.context.scene.objects]
198
199         # deselect all objects
200         bpy.ops.object.select_all(action='DESELECT')
201
202         for item in obj_export_list:
203             item.select = True
204             if item.type == 'MESH':
205                 file_path = os.path.join(folder_path, "{}.obj".format(item.name))
206                 bpy.ops.export_scene.obj(filepath=file_path, use_selection=True,
207                                         axis_forward=self.axis_forward_setting,
208                                         axis_up=self.axis_up_setting,
209                                         use_animation=self.use_animation_setting,
210                                         use_mesh_modifiers=self.use_mesh_modifiers_setting,
211                                         use_edges=self.use_edges_setting,
212                                         use_smooth_groups=self.use_smooth_groups_setting,
213                                         use_smooth_groups_bitflags=self.use_smooth_groups_bitflags_setting,
214                                         use_normals=self.use_normals_setting,
215                                         use_uv=self.use_uv_setting,
216                                         use_materials=self.use_materials_setting,
```


OBRIGADO!



Instagram



Linkedin