

```

177         ),
178         default='Y',
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.
190
191         # get objects selected in the viewport
192         viewport_selection = bpy.context.sel
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
198
199         # deselect all objects
200         bpy.ops.object.select_all(action='DE

```

**Python para**  
**Novatos: Comienza tu**  
**Viaje en el Mundo del**  
**Código**

# Python para Novatos: Comienza tu Viaje en el Mundo del Código

Python es un lenguaje de programación, interpretado y de propósito general, diseñado para ser fácil de leer y escribir. Su sintaxis clara y sencilla permite a los desarrolladores expresar mejor sus conceptos.

Este lenguaje admite múltiples paradigmas de programación, incluyendo la programación orientada a objetos. Además, cuenta con una amplia biblioteca estándar que facilita tareas como el desarrollo web, la automatización de procesos y el análisis de datos.

Para comenzar tu viaje en el mundo de la programación con Python, es esencial familiarizarte con los conceptos básicos que forman este lenguaje.

# 1. Tipos de Datos y Variables

En Python, los tipos de datos definen la naturaleza de los valores que puedes manipular. Los principales son:

- **Enteros (int):** Números sin decimales.
  - Ejemplo: edad = 30
- **Flotantes (float):** Números con decimales.
  - Ejemplo: precio = 19.99
- **Cadenas de texto (str):** Secuencias de caracteres
  - Ejemplo: nombre = "Ana"
- **Booleanos (bool):** Valores de verdad, True o False.
  - Ejemplo: es\_estudiante = True

Las variables son contenedores que almacenan datos. En Python, no es necesario declarar el tipo de una variable; el intérprete lo infiere automáticamente

```
nombre = "Carlos" # cadenas de texto
edad = 25          # Entero
altura = 1.75      # Flotante
es_estudiante = False # Booleano
```

## 2. Condicionales

Las estructuras condicionales permiten ejecutar bloques de código basados en condiciones específicas. En Python, utilizamos if, elif y else para este propósito

```
edad = 18
```

```
if edad >= 18:
    print("Eres mayor de edad")
elif edad >= 13:
    print("Eres un adolescente")
else:
    print("Eres un niño")
```

# 3. Bucles

Los bucles son estructuras que repiten un bloque de código mientras se cumpla una condición. Python ofrece dos tipos principales:

- Bucle for: Itera sobre una secuencia (como una lista o un rango de números).

```
for i in range(1, 6):           # Itera del 1 al 5
    print(i)
```

- Bucle while: Ejecuta un bloque de código mientras una condición sea verdadera.

```
contador = 1
```

```
while contador <= 5:
    print(contador)
```