

```
if operation == "MIRROR_Z" :
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end -add back the deselected
mirror_ob.select= 1
modifier_ob.select=1
bpy.context.scene.objects.active = modifier_ob
print("Selected" + str(modifier_ob)) # modifier ob
    mirror_ob.select = 0
done = bpy.context.selected_objects[0]
done.select = True
```

Back end nivel intermedio

Programación con Python



A lo largo de los años los lenguajes de programación han ido evolucionando para que sean más sencillos de entender y de implementar. Dentro de estos lenguajes se encuentra Python, el cual es utilizado entre muchas otras cosas para realizar el back end de un desarrollo web siendo este un lenguaje con gran capacidad y con gran facilidad de escritura, que proporcionará ahorro en líneas de código y en utilización de recursos.

Instalación y configuración

Python es un lenguaje de programación que se ha vuelto muy popular con el paso de los años y esto es gracias a su simplicidad para la programación, observa el siguiente ejemplo:

Para crear un programa que nos diga “Hola mundo” en C, se utiliza el siguiente código:

```
#include <stdio.h>

int main(void)
{
    printf("Hello Word");
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Para Python lo único que vas a colocar es lo siguiente:

```
print("Hello Word");
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Como puedes observar, solo necesitas una línea de código para hacer lo mismo que en 5 líneas de código. Lo anterior ayudará en tus desarrollos back end, ya que podrás ahorrar muchas líneas de código, lo cual se traduce a menor tiempo en cuanto a programación, por lo tanto, una menor inversión en los desarrollos que realices.

Muchas empresas utilizan este lenguaje, ya que es código abierto y su implementación es muy sencilla. Algunas de estas empresas son:

- YouTube.
- Reddit.
- Spotify.

De acuerdo con Amos, Bader, Jablonski y Heisler (2020), Python puede ser instalado tanto en Windows, como en Linux y MacOS, a continuación, revisarás como realizar la instalación en cada uno de ellos.

Para Windows debes seguir los siguientes pasos:

1. Ingresá a la siguiente liga:

Python. (2022). *Download the latest versión of Windows.* Recuperado de <https://www.postman.com/>

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

2. Haz clic en Windows



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

3. Haz clic en “Latest Python 3 Release- Python 3.10.7”

Python »» Downloads »» Windows

Python Releases for Windows

- [Latest Python 3 Release - Python 3.10.7](#)
- [Latest Python 2 Release - Python 2.7.18](#)

Stable Releases <ul style="list-style-type: none">▪ Python 3.7.14 - Sept. 6, 2022 Note that Python 3.7.14 cannot be used on Windows XP or earlier.<ul style="list-style-type: none">▪ No files for this release.▪ Python 3.8.14 - Sept. 6, 2022 Note that Python 3.8.14 cannot be used on Windows XP or earlier.<ul style="list-style-type: none">▪ No files for this release.▪ Python 3.9.14 - Sept. 6, 2022	Pre-releases <ul style="list-style-type: none">▪ Python 3.11.0rc2 - Sept. 12, 2022<ul style="list-style-type: none">▪ Download Windows embeddable package (32-bit)▪ Download Windows embeddable package (64-bit)▪ Download Windows embeddable package (ARM64)▪ Download Windows installer (32-bit)▪ Download Windows installer (64-bit)▪ Download Windows installer (ARM64)▪ Python 3.11.0rc1 - Aug. 8, 2022
--	---

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

4. Desplaza la página hasta debajo de todo y haz clic en “Windows Installer 64 bits”

tions are specified.

[Full Changelog](#)

Files

Version	Operating System	Description	MD5 Sum	File Size	GPG	Sigstore	
Gzipped source tarball	Source release		1aea68575c0e97bc83ff8225977b0d46	26006589	SIG	CRT	SIG
XZ compressed source tarball	Source release		b8094f007b3a835ca3be6bdf8116cccc	19618696	SIG	CRT	SIG
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	4c89649f6ca799ff29f1d1dffccb9393	40865361	SIG	CRT	SIG
Windows embeddable package (32-bit)	Windows		7e4de22bfe1ebd333b2c691ec2c1fcee	7615330	SIG	CRT	SIG
Windows embeddable package (64-bit)	Windows		7f90f8642c1b19cf02bce91a5f4f9263	8591256	SIG	CRT	SIG
Windows help file	Windows		643179390f5f5d9d6b1ad66355c795bb	9355326	SIG	CRT	SIG
Windows installer (32-bit)	Windows		58755d6906f825168999c83ce82315d7	27779240	SIG	CRT	SIG
Windows installer (64-bit)	Windows	Recommended	bfbef8467c7e3504f3800b0fe94d9a3e6	28953568	SIG	CRT	SIG

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

5. Abre el instalador y haz clic en Install Now.



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

6. Sigue los pasos del instalador y una vez terminado busca la carpeta donde se descargó Python y ábrelo.

7. Busca el archivo llamado IDLE. dicho archivo es un ambiente interactivo que te permitirá escribir el código directamente en Python y probarlo en el momento.

Para la instalación en Linux deberás seguir los siguientes pasos:

1. Ingresa a la línea de comandos y escribe el siguiente código:

```
$Python3 --version
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

2. Te aparecerá la versión que tienes instalada de Python, por lo tanto, no necesitarás realizar ninguna otra acción.

3. Para abrir el IDLE debes de colocar el siguiente comando:

```
$idle-python3.10
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

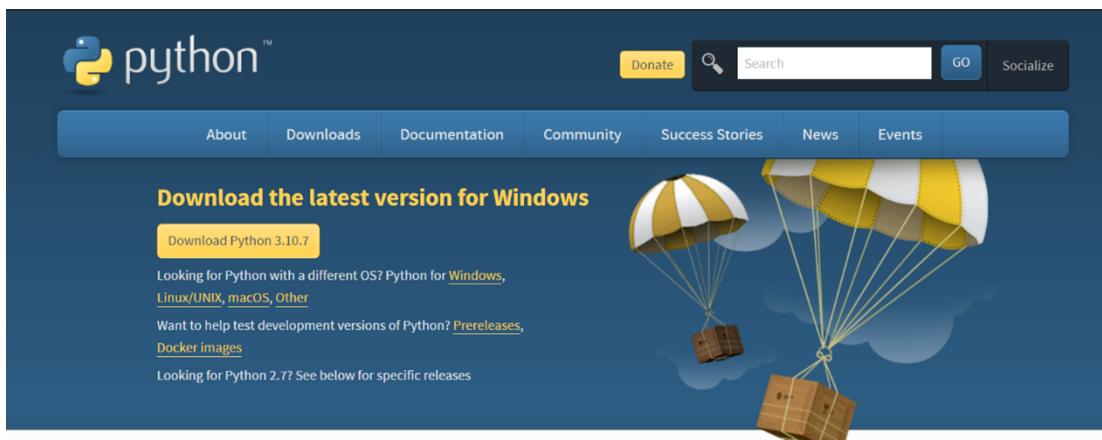
Para instalar en MacOS deberás realizar lo siguiente:

1. Ingresa a la siguiente liga:

Python. (2022). Download the latest versión of Windows. Recuperado de <https://www.postman.com/>

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

2. Haz clic en MacOS.



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Explicación

3. Haz clic en “Latest Python 3 Release- Python 3.10.7”.

Python Releases for macOS

- [Latest Python 3 Release - Python 3.10.7](#)
- [Latest Python 2 Release - Python 2.7.18](#)

Stable Releases

- [Python 3.7.14 - Sept. 6, 2022](#)
 - No files for this release.
- [Python 3.8.14 - Sept. 6, 2022](#)
 - No files for this release.
- [Python 3.9.14 - Sept. 6, 2022](#)
 - No files for this release.
- [Python 3.10.7 - Sept. 6, 2022](#)
 - Download [macOS 64-bit universal2 installer](#)
- [Python 3.10.6 - Aug. 2, 2022](#)
 - Download [macOS 64-bit universal2 installer](#)

Pre-releases

- [Python 3.11.0rc2 - Sept. 12, 2022](#)
 - Download [macOS 64-bit universal2 installer](#)
- [Python 3.11.0rc1 - Aug. 8, 2022](#)
 - Download [macOS 64-bit universal2 installer](#)
- [Python 3.11.0b5 - July 26, 2022](#)
 - Download [macOS 64-bit universal2 installer](#)
- [Python 3.11.0b4 - July 11, 2022](#)
 - Download [macOS 64-bit universal2 installer](#)
- [Python 3.11.0b3 - June 1, 2022](#)
 - Download [macOS 64-bit universal2 installer](#)

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

4. Baja toda la página y haz clic en donde dice “macOS 64-bit universal2 installer”

Files

Version	Operating System	Description	MD5 Sum	File Size	PGP	Sigstore
Gzipped source tarball	Source release		1aea68575c0e97bc83ff8225977b0d46	26006589	SIG	CRT SIG
XZ compressed source tarball	Source release		b8094f007b3a835ca3be6bdf8116cccc	19618696	SIG	CRT SIG
macOS 64-bit universal2 installer	macOS	for macOS 10.9 and later	4c89649f6ca799ff29f1d1dffccb9393	40865361	SIG	CRT SIG
Windows embeddable package (32-bit)	Windows		7e4de22bfe1e6d333b2c691ec2c1fce	7615330	SIG	CRT SIG
Windows embeddable package (64-bit)	Windows		7f90f8642c1b19cf02bce91a5f4f9263	8591256	SIG	CRT SIG
Windows help file	Windows		643179390f5fd9d6b1ad66355c795bb	9355326	SIG	CRT SIG
Windows installer (32-bit)	Windows		58755d6906fb25168999c83ce82315d7	27779240	SIG	CRT SIG
Windows installer (64-bit)	Windows	Recommended	bfbef467c7e3504f3800b0fe94d9a3e6	28953568	SIG	CRT SIG

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Explicación

5. Abre el instalador y sigue los pasos del instalador.
6. Una vez instalado abre el buscador y haz clic en aplicaciones.
7. Haz clic en el folder de Python 3.10.7 y después haz doble clic en el ícono de IDLE.

Variables, tipos de datos y condicionales

Como en todo lenguaje de programación es importante conocer que es lo que podemos hacer con dicho lenguaje. A continuación, verás como declarar variables, acceder a su información, los tipos de datos que existen y como se usan las condicionales.

Comenzarás con las variables. De acuerdo con Amos, Bader, Jablonski y Heisler (2020), una variable es algo que puedes asignar un valor y después te puedes referir a ese valor durante la programación.

En el caso de un back end, una variable te podría ayudar para mostrar lo solicitado por un usuario, por ejemplo: imagina que un usuario ingresa a una página web donde puede realizar una búsqueda acerca de lo que quiere comprar. Podrías usar una variable para almacenar lo que el usuario escribió y buscar en las bases de datos lo que el usuario necesita para mostrárselo.

Para crear una variable utilizarás el siguiente código:

```
>>>nombredelvariable = valordelvariable
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

```
>>>message = 'Hello, World!'
print(message)
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Explicación

El código anterior mostrará el siguiente resultado:

Hello, World!

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Una de las ventajas que tiene Python es la gran variedad de tipos de datos que posee. A continuación, verás dichos tipos y su utilidad en back end. De acuerdo con Python (s.f.), los tipos de datos son:

1. Datetime: es un módulo en el cual podrás manipular fechas y horas.

Dentro de este módulo encontrarás las siguientes opciones:

- a. **MINYEAR:** el año más pequeño que se puede insertar es 1.
- b. **MAXYEAR:** El año más grande que se puede insertar es 9999.
- c. **Date:** fecha formada por año, mes y día.
- d. **Time:** tiempo conformado por horas, minutos, segundos, microsegundos.
- e. **Datetime:** la combinación de fecha y hora.
- f. **Tzinfo:** información de zona horaria.

2. ZoneInfo: te permitirá obtener la zona horaria donde se encuentre el usuario.

3. Calendar: en este módulo podrás generar calendarios y obtener información general acerca del mismo.

4. Collections: son contenedores especializados que te permiten almacenar diferentes tipos de datos.

5. Heappq: es un algoritmo que te permitirá crear colas con prioridad, es decir, podrás crear una cola con datos que tendrán una prioridad frente al desarrollo, por ejemplo: en un desarrollo web hay tareas que son más prioritarias que otras.

6. Bisect: te permite mantener una lista ordenada, aunque se agregue un nuevo elemento.

7. Array: objeto que te permite guardar muchos datos del mismo tipo.

Explicación

Una vez que ya conoces todos los tipos de datos y las variables en Python procederás al tema de las condicionales. Como todo lenguaje de programación una condicional está basada en si cierta cosa se cumple o no. Las condicionales te servirán para realizar validaciones dentro de tu back end, por ejemplo: si la conexión a la base de datos fue exitosa, si el valor introducido por el usuario existe dentro de la base de datos, etc.

La sintaxis de una condicional en Python es la siguiente:

```
if condicion:  
    # ejecutar un código
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

La sintaxis de un `else` en Python es la siguiente:

```
if condicion:  
    # ejecutar un código  
else:  
    # ejecutar un código distinto
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

La sintaxis de un `elif` en Python es la siguiente:

```
if condicion:  
    # ejecutar un código  
elif otra_condicion:  
    # ejecutar otro código  
else:  
    # ejecutar un código distinto
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Explicación

A continuación, se muestra un ejemplo del uso de las condicionales al momento de querer obtener los datos en una base de datos:

```
import MySQLdb

DB_HOST = 'localhost'
DB_USER = 'root'
DB_PASS = 'mysqlroot'
DB_NAME = 'a'

def run_query(query=''):
    datos = [DB_HOST, DB_USER, DB_PASS, DB_NAME]

    conn = MySQLdb.connect(*datos) # Conectar a la base de datos
    cursor = conn.cursor()        # Crear un cursor
    cursor.execute(query)         # Ejecutar una consulta

    if query.upper().startswith('SELECT'):
        data = cursor.fetchall()  # Traer los resultados de un select
    else:
        conn.commit()            # Hacer efectiva la escritura de datos
        data = None

    cursor.close()               # Cerrar el cursor
    conn.close()                 # Cerrar la conexión

    return data
```

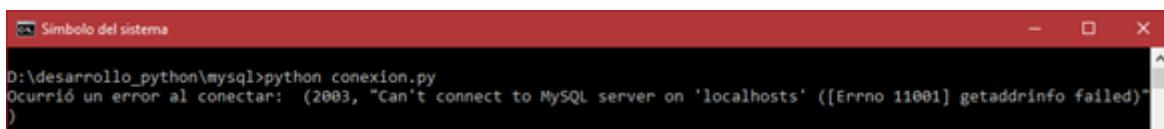
Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Si la conexión es exitosa arrojará el siguiente mensaje:

Conexión correcta

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

En caso de que la conexión no se haya realizado, se obtendrá el siguiente mensaje:



Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Explicación

Bucles, funciones y listas

De acuerdo con Prieto (s.f.), los bucles permitirán realizar ciertas acciones que requieras que se repitan hasta que una condición se cumpla o deje de cumplirse. Por ejemplo: en una página web cuando el usuario busca algún producto podrías crear un bucle que tenga la condición de buscar en la base de datos hasta que encuentre el producto que el usuario buscó.

Dentro de Python existen 2 tipos de bucles:

- For
- While

El bucle `while` funciona de la siguiente manera: mientras una expresión sea verdadera se ejecutará el código que se encuentra dentro del bucle.

Su sintaxis es la siguiente:

```
while <expr>:  
    <statement(s)>
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

La sintaxis de un `else` en el bucle `while` es la siguiente:

```
while <expr>:  
    <statement(s)>  
else:  
    <additional_statement(s)>
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

El bucle `for` cambia un poco su forma de operar; ya que este bucle hace que se ejecuten ciertas acciones para cada elemento que se le pase al bucle.

Explicación

Su sintaxis es la siguiente:

```
for i in <collection>
    <loop body>
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

A continuación, se muestra un ejemplo utilizando un bucle para consultar un dato en una base de datos de películas:

```
import pymysql
try:
    conexion = pymysql.connect(host='localhost',
                                user='root',
                                password='',
                                db='peliculas')
    try:
        with conexion.cursor() as cursor:
            consulta = "SELECT id, titulo, anio FROM peliculas WHERE anio > %s;"
            cursor.execute(consulta, (2000))

            # Con fetchall traemos todas las filas
            peliculas = cursor.fetchall()

            # Recorrer e imprimir
            for pelicula in peliculas:
                print(pelicula)
    finally:
        conexion.close()
except (pymysql.err.OperationalError, pymysql.err.InternalError) as e:
    print("Ocurrió un error al conectar: ", e)
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

El resultado será el siguiente:

```
D:\desarrollo_python\mysql>python consultar_where.py
(2, 'Ready Player One', 2018)
(3, 'It', 2017)
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Una vez que ya conoces los bucles existentes para Python, procederás a revisar las funciones y las listas.

De acuerdo con Bustamante (2021), las funciones como has visto anteriormente te permiten crear una solución más modular, esto es gracias a que el código dentro de la función lo puedes reutilizar y ejecutar el número de veces que tu deseas.

Explicación

En Python, la sintaxis es la siguiente:

```
def nombrefunción():
    bloque de código
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

En cuanto a las listas, se definen como un contenedor donde puedes almacenar diferentes elementos que estén relacionados.

Para crear una lista debes usar la siguiente sintaxis:

```
>>>nombradelista[valor1,valor2...valorn]
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

A continuación, se muestra el ejemplo de una función creada para insertar un dato en la base de datos de películas:

```
def insert_pelicula():
    sqls = []
        nombre = 'La vida es bella'
        an='1997'
    for i in range(10):
        titulo = nombre
        anio= an
        sql = "insert into peliculas(titulo,anio) value ('%s','%s')%(nombre,an)"
        sqls.append(sql)

    return sql
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Explicación

El resultado en la base de datos será el siguiente:

Id	Titulo	Anio
1	Titanic	1997
2	The Dark Night	2008
3	El rey león	1994
4	Avatar	2009
5	SkyFall	2012
6	Toy Story 4	2019
7	La vida es bella	1997

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

API-REST con Python

Ya que tienes bien establecidas las bases de Python podrás continuar con la creación de un API-REST en este lenguaje. De acuerdo con Van Schooneveld (s.f.), un API REST es una arquitectura en la cual se define la comunicación entre el cliente y el servidor a través de la red, en otras palabras, como se comunica el front end y el back end.

Para que Python interactúe con API'S REST necesitas primero instalar los requerimientos de HTTP, es decir el GET, POST, PUT, etc.

Para realizar dicha instalación deberás insertar el siguiente código en el IDLE de Python:

```
$ python -m pip install requests
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Para utilizar el método GET deberás utilizar el siguiente código:

```
>>> import requests
>>> api_url = "URLAPIREST"
>>> response = requests.get(api_url)
>>> response.json()
{valores a enviar a la API}
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Explicación

A continuación, se muestra el método GET para obtener la longitud y latitud de una ubicación a través de la API de Google Maps:

```
import requests

URL = "http://maps.googleapis.com/maps/api/geocode/json"

location = "Galerías metepec"

PARAMS = {'address':location}

r = requests.get(url = URL, params = PARAMS)

data = r.json()

latitude = data['results'][0]['geometry']['location']['lat']
longitude = data['results'][0]['geometry']['location']['lng']
formatted_address = data['results'][0]['formatted_address']

print("Latitude:%s\nLongitude:%s\nFormatted Address:%s"
      %(latitude, longitude,formatted_address))
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

El resultado sería el siguiente:

```
Latitude:19.25899318301106
Longitude: -99.6204757027363
Formated address:Galerias Metepec, Metepec, Toluca, 52141, Mexico
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Para el método POST (utilizado para agregar nueva información en la API) deberás utilizar el siguiente código:

```
>>> import requests
>>> api_url = "URLAPI"
>>> todo = {"Valores a agregar"}
>>> response = requests.post(api_url, json=todo)
>>> response.json()
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Explicación

A continuación, se muestra un ejemplo del método POST para agregar un nuevo usuario y una contraseña en una API:

```
import requests
auth_data = {'email': 'usuarioprueba@gmail.com', 'pass': '1234'}
resp = requests.post('Nombre de la API', data=auth_data)
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Para el método PUT (utilizado para actualizar un valor existente) deberás usar el siguiente código:

```
>>> import requests
>>> api_url = "URLAPI"
>>> response = requests.get(api_url)
>>> response.json()
{Valor a cambiar}

>>> todo = {Valor actualizado}
>>> response = requests.put(api_url, json=todo)
>>> response.json()
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

A continuación, se muestra un ejemplo para implementar el método PUT tomando el ejemplo del usuario y contraseña que se acaban de agregar con el método POST:

```
import requests
dato_actualizar={'email': 'usuarioprueba@gmail.com', 'pass': '123456'}
resp = requests.put('Nombre de la API', data=dato_actualizar)
```

Esta pantalla se obtuvo directamente del software que se está explicando en la computadora, para fines educativos.

Lo anterior lo podrás utilizar siempre y cuando requieras consumir los resultados de una API que ya existe, pero si la API de tu desarrollo web no existe; deberás primero crearla para después poder utilizar los métodos en Python y explotar al máximo los datos que ahí se encuentren.

Tecmilenio no guarda relación alguna con las marcas mencionadas como ejemplo. Las marcas son propiedad de sus titulares conforme a la legislación aplicable, se utilizan con fines académicos y didácticos, por lo que no existen fines de lucro, relación publicitaria o de patrocinio.

Python es un lenguaje que ha agarrado mucha fuerza en los últimos años gracias a su facilidad para programar y a su facilidad de entendimiento. Dicho lenguaje permitirá establecer conexiones directas con APIs REST existentes, de las cuales podrás obtener la información que el front end esté solicitando.

¿Consideras que Python es un lenguaje útil?

¿Consideras que Python puede soportar toda la infraestructura de un back end?

Referencias bibliográficas

- Amos, D., Bader, D., Jablonski, J., y Heisler, F. (2020). *Python Basics: A Practical Introduction to Python 3.* (4^a ed.). Estados Unidos: Real Python.
- Python. (s.f.). *Python 3.10.7 documentation.* Recuperado de <https://docs.python.org/3.10/>
- Python. (2022). *Download the latest versión of Windows.* Recuperado de <https://www.postman.com/>
- Prieto, R. (s.f.). *Cómo hacer bucles en Python.* Recuperado de <https://www.raulprietofernandez.net/blog/programacion/como-hacer-bucles-en-python>
- Bustamante, S. (2021). *Guía de funciones de Python con ejemplos.* Recuperado de <https://www.freecodecamp.org/espanol/news/guia-de-funciones-de-python-con-ejemplos/>
- Van Schooneveld, J. (s.f.). *Python and REST APIs: Interacting With Web Services.* Recuperado de <https://realpython.com/api-integration-in-python/>

Para saber más

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

Lecturas

Para conocer más acerca de **Python**, te sugerimos leer lo siguiente:

- Lozano, J. (s.f.). *list python – Listas en Python: El tipo list y operaciones más comunes.* Recuperado de <https://j2logo.com/python/tutorial/tipo-list>
- Lozano, J. (s.f.). *Tutorial de cómo crear un API REST en Python con Flask.* Recuperado de <https://j2logo.com/flask/tutorial-como-crear-api-rest-python-con-flask/>

Videos

Para conocer más acerca de **Python**, te sugerimos revisar lo siguiente:

- AnderCode. (2021, 01 de junio). *Creación de API REST Web Service con Python y MySQL - 1* [Archivo de video] Recuperado de <https://www.youtube.com/watch?v=wvojFzvmmrY>
- IBM Technology. (2020, 23 de octubre). *¿Qué es una API REST?* [Archivo de video]. Recuperado de <https://www.youtube.com/watch?v=lsMQRaeKNDk>

Checkpoints

Asegúrate de:

- Comprender la utilidad de Python en el back end para utilizar menos recursos al momento de ejecutar, así como una forma más fácil de implementación.
- Comprender como utilizar Python para conectarse con APIs REST.
- Comprender la importancia de un lenguaje de programación para programar el back end.

Requerimientos Técnicos

- Programación en servidores.
- Bases de back end.
- Bases de APIs.
- Python.

Prework

- Descarga Python programa:

Python. (2022). *Download the latest versión of Windows.* Recuperado de <https://www.python.org/downloads/>

Los siguientes enlaces son externos a la Universidad Tecmilenio, al acceder a ellos considera que debes apegarte a sus términos y condiciones.

La obra presentada es propiedad de ENSEÑANZA E INVESTIGACIÓN SUPERIOR A.C. (UNIVERSIDAD TECMILENIO), protegida por la Ley Federal de Derecho de Autor; la alteración o deformación de una obra, así como su reproducción, exhibición o ejecución pública sin el consentimiento de su autor y titular de los derechos correspondientes es constitutivo de un delito tipificado en la Ley Federal de Derechos de Autor, así como en las Leyes Internacionales de Derecho de Autor.

El uso de imágenes, fragmentos de videos, fragmentos de eventos culturales, programas y demás material que sea objeto de protección de los derechos de autor, es exclusivamente para fines educativos e informativos, y cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por UNIVERSIDAD TECMILENIO.

Queda prohibido copiar, reproducir, distribuir, publicar, transmitir, difundir, o en cualquier modo explotar cualquier parte de esta obra sin la autorización previa por escrito de UNIVERSIDAD TECMILENIO. Sin embargo, usted podrá bajar material a su computadora personal para uso exclusivamente personal o educacional y no comercial limitado a una copia por página. No se podrá remover o alterar de la copia ninguna leyenda de Derechos de Autor o la que manifieste la autoría del material.