



PROCESO DE GESTIÓN DE FORMACIÓN PROFESIONAL INTEGRAL FORMATO GUÍA DE APRENDIZAJE

1. IDENTIFICACIÓN DE LA GUÍA DE APRENDIZAJE

- **Denominación del programa de formación:** Variables y estructuras de control en Python.
- **Código del programa de formación:** 22810005
- **Competencia:** 220501096. Desarrollar la solución de software de acuerdo con el diseño y metodologías de desarrollo.
- **Resultados de aprendizaje a alcanzar:**
 - 220501096-1. Definir las variables a utilizar de acuerdo con el tipo de dato que deben contener.
 - 220501096-2. Formular instrucciones de entrada y salida de datos e instrucciones de proceso secuencial aplicando estándares de codificación en Python.
 - 220501096-3. Utilizar estructuras de control condicionales en Python de acuerdo con los requerimientos técnicos.
 - 220501096-4. Usar estructuras de control iterativas en Python teniendo en cuenta los ciclos requeridos por el usuario.
- **Duración de la guía:** 48 horas

2. PRESENTACIÓN

Estimados aprendices, en nombre del Servicio Nacional de Aprendizaje SENA, les damos la bienvenida al programa de formación complementaria virtual **Variables y estructuras de control en Python**.

Un lenguaje de programación es un sistema estructurado que permiten la comunicación de manera directa con una computadora. Existen muchos lenguajes de programación: Python, Java, Ruby, PHP y C++ son algunos de los lenguajes más usados en la actualidad, permitiendo incrementar puestos de trabajo para los programadores.

En este curso complementario, se ha seleccionado *Python*, un software libre, con un lenguaje sencillo, fácil de aprender y poderoso en su codificación para desarrollar aplicaciones web o móviles, juegos, gráficos y algoritmos científicos de cálculo avanzado, además, funciona en cualquier sistema operativo que integre su intérprete, siendo una herramienta útil para el área de *Machine Learning*.

Para el desarrollo de las evidencias de aprendizaje propuestas en esta guía es necesaria una lectura detallada y crítica de los componentes formativos del programa y a partir de la realización de los talleres formulados, usted se apropiará de los conocimientos presentados.

Las actividades de aprendizaje propuestas están diseñadas para realizar un trabajo autónomo sistemático, y colaborativo, con la interacción de aprendices e instructor a través de encuentros sincrónicos y asincrónicos.



3. FORMULACIÓN DE LAS ACTIVIDADES DE APRENDIZAJE

Mediante la estrategia didáctica activa de aprendizaje individual podrá aprender los fundamentos del lenguaje de programación *Python*, en lo relacionado con el uso de constantes y variables, entrada y salida de datos, instrucciones de asignación, operadores y funciones predefinidas, uso de contadores y acumuladores, sentencias condicionales y ciclos repetitivos.

3.1 Actividad de Aprendizaje 1. Definir variables en *Python* según el tipo de datos.

Duración: 6 horas

Antes de iniciar con el desarrollo de las actividades de aprendizaje y la lectura crítica de los componentes formativos, lo invitamos a participar del foro social, este foro, es un espacio de encuentro entre aprendices e instructor, donde se pueden compartir expectativas, proyectos de vida y gustos personales. Con la participación en él, se busca crear una red social de los compañeros virtuales.

Por esta razón, realice una presentación personal incluyendo: experiencia laboral, conocimientos previos y aspectos en los que considere, le ayudarán en este programa de formación en algún contexto de su vida.

Una vez revisado el componente formativo uno: **Términos asociados al tipo de datos**, desarrolle las siguientes evidencias de aprendizaje, durante la primera semana:

➤ Foro temático: **usos de Python.**

Como primera evidencia a presentar en esta actividad de aprendizaje, participe en el foro temático denominado **usos de Python**, dando respuesta al interrogante ¿Indique por lo menos uno de los usos que se le da al lenguaje de programación Python en la actualidad?

Es importante que, para participar en este foro, lea atentamente el componente formativo uno, en la sección de Introducción, analice los usos actuales del lenguaje Python y observe el video “Los usos de Python más importantes” en el siguiente enlace:
<https://www.youtube.com/watch?v=ZNxbq3DNE0>

Lineamientos para la entrega de la evidencia:

- **Producto a entregar:** participación en el foro “Usos de Python”.
- Para participar en el foro remítase al área de la actividad correspondiente e ingrese al espacio para acceder a **AA1-EV01. Foro: usos de Python.**



➤ **Evaluación. Tipos de datos en Python.**

Como segunda evidencia a desarrollar, realice la evaluación en línea compuesta por preguntas de selección múltiple con única respuesta y preguntas de Falso/Verdadero sobre los **términos asociados al tipo de datos**, de acuerdo con lo presentado en el componente formativo uno.

Lineamientos para la entrega de la evidencia:

- Para responder el cuestionario (evaluación en línea), remítase al área de la actividad correspondiente y acceda al espacio para el envío de la evidencia: **AA1-EV02. Evaluación. Tipos de datos en Python.**

3.2 Actividad de Aprendizaje 2. Codificar instrucciones de entrada y salida de datos siguiendo estándares en Python.

Duración: 18 horas

Una vez revisado el componente formativo dos: **Entrada y Salida de Datos con Python**, desarrolle las siguientes evidencias de aprendizaje:

➤ **Foro temático: Ventajas de Python**

Como primera evidencia a presentar en esta actividad participe en el foro temático denominado **“Ventajas de Python**, dando respuesta al interrogante ¿Indique por lo menos una de las ventajas de usar Python?

Es importante que, para participar en este foro, leer atentamente el componente formativo 2, en el **numeral 3.5 Funciones predefinidas de Python**, analice los usos de esas funciones y visite la página <https://entrenamiento-python-basico.readthedocs.io/es/latest/> en el enlace 5.6. Funciones integradas.

Lineamientos para la entrega de la evidencia:

- **Producto a entregar:** participación en el foro “Ventajas de Python”.
- Para participar en el foro remítase al área de la actividad correspondiente e ingrese al espacio para acceder **AA2-EV01. Foro: Ventajas de Python.**

➤ **Evaluación. Tipos de operadores en Python**

Como última evidencia a desarrollar en la actividad de aprendizaje 2, realice la evaluación en línea compuesta por relación entre columnas y por preguntas de selección múltiple con única respuesta acerca de operadores y funciones en *Python*, de acuerdo con lo presentado en el componente formativo dos.

Lineamientos para la entrega de la evidencia:

- Para responder el cuestionario (evaluación en línea), remítase al área de la actividad correspondiente y acceda al espacio para el envío de la evidencia: **AA2-EV02. Evaluación. Tipos de operadores en Python.**



3.3 Actividad de Aprendizaje 3. Utilizar estructuras de control condicionales en Python cumpliendo requerimientos técnicos.

Duración: 12 horas

Una vez revisado el componente formativo tres: **Estructuras de control condicionales**, desarrolle la siguiente evidencia de aprendizaje:

➤ **Taller 1: condicionales simples.**

Como evidencia a presentar en esta actividad de aprendizaje 3, elabore el **Taller 1: condicionales simples**. Realice lo siguiente:

- Descargue el instalador del intérprete de *Python* de la página oficial www.python.org
- Ejecute el instalador, dando clic en siguiente... siguiente...
- Digite: "IDLE" en la barra de tareas de su computador, para activar el editor de *Python*.
- Abra un nuevo archivo usando las opciones File - New File o las teclas: CTRL+N
- Digite el siguiente código del lenguaje *Python*:

```
File Edit Format Run Options Window Help
# TALLER 3 PYTHON
# AUTOR(A): XXXXXX
# FECHA:

from datetime import date
hoy = date.today()          #fecha actual
print("Hoy es el día: ", hoy)

a=int(input("digite el valor de A: "))
b=int(input("digite el valor de B: "))
if a>=b:
    print("A es mayor o igual a B")
else:
    print("A es menor que B")
print()
curso1="Requerimientos"
curso2="Algoritmos"
print("El curso1 es: ", curso1)
print("El curso2 es: ", curso2)
if curso1 == "Requerimientos" and curso2 == "Algoritmos":
    print ("Usted estudia Programación de Software")
else:
    print ("Usted estudia otro programa diferente a Programación de Software")
print()
print ("""    Final del Análisis del Programa de Formación SENA    """)
print()
frase=input("digite una oracion: ")
print("La frase en mayuscula es: ", frase.upper())
longitud=len(frase)
print("La longitud de la frase es: ", len(frase), "caracteres")
if longitud>10:
    print ("La frase contiene mas de 10 caracteres")
else:
    print("La frase contiene menos de 11 caracteres")
print()
print("FIN")
```



- f. Guarde el programa digitado usando la opción File – Save o las teclas CTRL+S.
- g. Ejecute el programa con la opción: Run – Run Module o la tecla F5
- h. Verifique los resultados.
- i. Abra un nuevo documento Word con sus datos personales y el título: AA3-EV1 TALLER 1
- j. Pegue el código final elaborado con el lenguaje Python.
- k. Capture un pantallazo de los resultados del programa y péguelo en su documento Word.
- l. Almacene el documento de Word con el nombre: AA3-EV1 Taller 1

Lineamientos para la entrega de la evidencia:

- **Producto a entregar:** Taller 1: Condicionales simples.
- **Formato:** documento PDF
- Para enviar la evidencia desarrollada: ubique el enlace para el envío de la evidencia: **AA3-EV01 Taller 1: condicionales simples.**



3.4 Actividad de aprendizaje 4. Usar estructuras de control iterativas en Python considerando los requerimientos del cliente

Duración: 12 horas

Una vez revisado el componente formativo 4: **Ciclos Iterativos con Python.**, desarrolle la siguiente evidencia de aprendizaje:

➤ **Taller 2: ciclo FOR**

Como evidencia a presentar en esta actividad de aprendizaje 4, elabore el **Taller 2: ciclo FOR**. Realice lo siguiente:

- Digite: "IDLE" en la barra de tareas de su computador, para activar el editor de Python.
- Abra un nuevo archivo usando las opciones File - New File o las teclas: CTRL+N
- Digite el siguiente código del lenguaje Python:

```
File Edit Format Run Options Window Help
# TALLER 2 PYTHON
# AUTOR(A): XXXXXXX
# FECHA:

from datetime import date
hoy = date.today() #fecha actual
print("Hoy es el día: ", hoy)
print()
print("TALLER 2 CICLOS ITERATIVOS CON LA SENTENCIA FOR")
print()
num1=int(input("digite el primer numero: "))
num2=int(input("digite el segundo numero (mayor): "))
print("ciclo para el primer numero")
for i in range(num1):
    print(i)
print("fin del ciclo")

print()
print("ciclo desde el primer numero hasta el segundo numero")
for i in range(num1,num2):
    print(i)
print("fin del ciclo")

print()
print("ciclo desde el primero hasta el segundo numero con incrementos de 2")
for i in range(num1,num2, 2):
    print(i)
print("fin del ciclo")

print()
empresa=input("digite nombre de una empresa: ")
empresa= empresa.lower()
for character in empresa:
    print(character)
print("fin del ciclo")

print()
print("FIN")
```

- Guarde el programa digitado usando la opción File – Save o las teclas CTRL+S.
- Ejecute el programa con la opción: Run – Run Module o la tecla F5
- Verifique los resultados.
- Abra un nuevo documento Word con sus datos personales y el título: AA4-EV1 TALLER 2.
- Pegue el código final elaborado con el lenguaje Python.
- Capture un pantallazo de los resultados del programa y péguelo en su documento Word.
- Almacene el documento de Word con el nombre: AA4-EV1 Taller 2

Lineamientos para la entrega de la evidencia:

- **Producto a entregar:** Taller 2: ciclo FOR
- **Formato:** documento PDF
- Para enviar la evidencia desarrollada: ubique el enlace para el envío de la evidencia: **AA4-EV01. Taller 2: ciclo FOR.**



4. ACTIVIDADES DE EVALUACIÓN

Evidencias de Aprendizaje	Criterios de Evaluación	Técnicas e Instrumentos de Evaluación
Evidencia de Desempeño AA1-EV01. Foro 1: usos de Python. Evidencia de Conocimiento: AA1-EV02. Evaluación. Tipos de datos en Python.	Diferencia las variables de acuerdo con el tipo de datos que almacenan.	Lista de chequeo Cuestionario
Evidencia de Desempeño: AA2-EV01. Foro 2: Ventajas de Python.	Codifica las instrucciones de entrada y salida de datos utilizando funciones y operadores de Python.	Lista de chequeo
Evidencias de Conocimiento: AA2-EV02. Evaluación. Tipos de operadores en Python.	Codifica instrucciones secuenciales aplicando operadores y funciones integradas de Python.	Cuestionario



Evidencias de Producto: AA3-EV01. Taller 1: condicionales simples.	Usa estructuras de control condicionales aplicando las sentencias y operadores condicionales de Python.	Lista de Verificación
Evidencias de Producto: AA4-EV01. Taller 2: ciclo FOR.	Utiliza estructuras de control aplicando las sentencias iterativas de Python	Lista de Verificación

5. GLOSARIO DE TÉRMINOS

Código Fuente: es un conjunto de instrucciones y órdenes lógicas, compuestos de algoritmos que se encuentran escritos en un determinado lenguaje de programación, las cuales deben ser interpretadas o compiladas, para permitir la ejecución de un programa informático.

Expresión: una construcción sintáctica que puede ser evaluada, hasta dar un valor. En otras palabras, una expresión es una acumulación de elementos de expresión tales como literales, nombres, accesos a atributos, operadores o llamadas a funciones, todos ellos retornando valor. A diferencia de otros lenguajes, no toda la sintaxis del lenguaje son expresiones.

Función: una serie de sentencias que retornan un valor al que las llama. También se le puede pasar cero o más argumentos los cuales pueden ser usados en la ejecución de la misma.

IDLE: es el entorno integrado de desarrollo de Python, o Integrated Development Environment for Python. IDLE es un editor básico y un entorno de intérprete que se incluye con la distribución estándar de Python.

Indentación: este término significa mover un bloque de texto hacia la derecha insertando espacios o tabuladores, para así separarlo del margen izquierdo y distinguirlo del texto adyacente; en el ámbito de la imprenta, este concepto se denomina sangrado o sangría.

Inmutable: un objeto con un valor fijo. Los objetos inmutables son números, cadenas y tuplas. Estos objetos no pueden ser alterados. Un nuevo objeto debe ser creado si un valor diferente ha de ser guardado.

Intérprete: lee un programa escrito en un lenguaje de alto nivel, instrucción a instrucción y, para cada una de ellas, efectúa una traducción a las instrucciones de código de máquina equivalentes y las ejecuta inmediatamente. Cada vez que ejecutamos el programa con un intérprete, se repite el proceso de traducción y ejecución, ya que ambos son simultáneos.



Multiparadigma: acepta diferentes paradigmas (técnicas) de programación, tales como la orientación a objetos, la programación imperativa y funcional.

Multiplataforma: significa que puede ser interpretado en diversos Sistemas Operativos como GNU/Linux, OpenBSD, sistemas privativos, entre otros.

Software libre: es el software que respeta la libertad de los usuarios y la comunidad, Significa que los usuarios tienen la libertad de ejecutar, copiar, distribuir, estudiar, modificar y mejorar el software.

Tipado dinámico: un lenguaje de tipado dinámico es aquel cuyas variables, no requieren ser definidas asignando su tipo de datos, sino que éste, se autoasigna en tiempo de ejecución, según el valor declarado.

Zen de Python: es un listado de los principios básicos del lenguaje Python, son útiles para entender y usar el lenguaje. El listado se puede encontrar digitando «import this» en la consola interactiva.

6. REFERENTES BIBLIOGRÁFICOS

Arias, A. (2019). Aprende a programar con Python. Columbia.

Buttu, M. (2016). El gran libro de Python. España: Marcombo.

Caballero, R. (2019). Big data con Python: recolección, almacenamiento y proceso. Bogotá: Madrid: Alfaomega Colombiana.

Cervantes, O. (2017). Python con aplicaciones a las matemáticas, ingeniería y finanzas. México, Alfaomega.

Cuevas, A. (2017). Python 3: Curso práctico. Bogotá: Ediciones de la U.

Guzdial, B. y Vidal, A. (2013). Introducción a la Computación y programación con Python. México: Pearson educación.

Hinojosa, A. (2016). Python paso a paso. Bogotá: Ediciones de la U.

Marzal, A. y Gracia, I. (2009). Introducción a la programación con Python. Universitat Jaume I.

Ortega, J. (2018). Hacking ético con herramientas Python. Madrid: Ra-Ma.

Pérez, A. (2016). Python fácil. México: Alfaomega Grupo Editor.

Salazar, P. (2019). Empezando a programar en Python. Bogotá: Editorial Escuela Colombiana de Ingeniería.



7. CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor (es)	Luz Amparo Blanco Medina	Experta Técnica	Regional Santander, Centro de Servicios Empresariales y Turísticos.	Noviembre de 2020
	Bertha Paola Bejarano Avila	Diseñadora Instruccional	Regional Distrito Capital, Centro para la Industria de la comunicación Gráfica.	Noviembre de 2020
	Rafael Neftali Lizcano Reyes	Asesor Pedagógico	Regional Santander, Centro Industrial del Diseño y la Manufactura	Diciembre de 2020

8. CONTROL DE CAMBIOS

	Nombre	Cargo	Dependencia	Fecha	Razón del Cambio
Autor (es)					