SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

Universidad JOSÉ VASCONCELOS
CHIAPAS
Educar para Trascender

Capacitación Digital.mx

Guía de Estudio

Fundamentos de Python para Oficinistas

Introducción

Python es un lenguaje de programación poderoso, fácil de aprender y ampliamente utilizado en tareas cotidianas, como la automatización de procesos y el análisis de datos. Esta guía de estudio está diseñada para ayudarte a preparar el cuestionario "Fundamentos de Python para Oficinistas". Cada sección cubre los conceptos clave y ofrece explicaciones claras y ejemplos prácticos para que puedas comprender mejor cómo funciona Python.

1. ¿Qué es Python?

Python es un lenguaje de programación de alto nivel, lo que significa que es más fácil de leer y escribir en comparación con otros lenguajes. Es conocido por su sintaxis sencilla y su enfoque en la legibilidad del código. Python se utiliza en diversas áreas como desarrollo web, análisis de datos, automatización de tareas y más.

---

2. Mostrar mensajes en pantalla

La función `print()` es la más básica y común en Python. Permite mostrar mensajes en pantalla. Por ejemplo:

```
print("Hola, Mundo!")
```

Este código mostrará: **Hola, Mundo!**

---

3. Variables y asignación de valores

Las variables son espacios en memoria donde puedes almacenar datos. Se crean utilizando el signo igual (=). Por ejemplo:

```
x = 5
nombre = "Ana"
```

En este caso, `x` almacena el valor 5 y `nombre` almacena el texto "Ana".

---

4. Tipos de datos

En Python, los datos pueden tener diferentes tipos:

- **Enteros (int):** Números sin decimales, como `42`.
- **Texto (str):** Secuencias de caracteres, como "Hola".
- **Decimales (float):** Números con punto decimal, como `3.14`. Puedes verificar el tipo de dato usando la función `type()`:

```
type(42)  # Devuelve: <class 'int'>
```

---

5. Interacción con el usuario

La función `input()` se utiliza para pedir información al usuario. Devuelve siempre un texto (string). Por ejemplo:

```
nombre = input("¿Cómo te llamas? ")
print("Hola,", nombre)
```

Este código solicitará el nombre del usuario y luego lo saludará.

---

6. Operadores matemáticos

Python permite realizar operaciones matemáticas básicas como suma (+), resta (−), multiplicación (*) y división (/). Por ejemplo:

```
print(3 + 2)  # Resultado: 5
print(10 / 2) # Resultado: 5.0
```

Ten en cuenta que la división siempre devuelve un valor decimal.

---

7. Condicionales (if)

Los condicionales permiten ejecutar código sólo si se cumple una condición. Por ejemplo:

```
if 10 > 5:
    print("Diez es mayor que cinco")
```

En este caso, se mostrará el mensaje porque la condición `10 > 5` es verdadera.

---

8. Listas

Una lista es una colección de elementos que se almacena entre corchetes (`[]`). Por ejemplo:

```
numeros = [1, 2, 3]
print(numeros[0])  # Resultado: 1
```

Los elementos de una lista comienzan a contarse desde el índice `0`.

---

9. Ciclos (for)

Los ciclos `for` se utilizan para repetir acciones. Por ejemplo:

```
for i in range(3):
    print(i)
```

Esto mostrará:

```
0
1
2
```

---

10. Funciones

Las funciones son bloques de código reutilizables que se definen con `def`. Por ejemplo:

```
def saludar():
    print("Hola")
```

```
saludar()
```

El resultado será: **Hola**.

---

11. Manipulación de cadenas de texto

Python ofrece muchas herramientas para trabajar con texto. Por ejemplo, `upper()` convierte el texto a mayúsculas:

```
texto = "hola"
print(texto.upper())  # Resultado: HOLA
```

---

12. Longitud de cadenas y listas

La función `len()` te dice cuántos elementos tiene una lista o cuántos caracteres tiene un texto:

```
len("Hola")  # Resultado: 4
len([1, 2, 3])  # Resultado: 3
```

---

13. Concatenación de texto

Puedes unir textos con el operador +:

```
print("Python es " + "genial")  # Resultado: Python es genial
```

Conclusión

Con estos conceptos, estás listo para enfrentar el cuestionario y aplicar Python en tus tareas diarias. Practicar con ejemplos es la mejor manera de consolidar tu aprendizaje. ¡Explora y diviértete con Python!

Referencias

1. Van Rossum, G., & Drake, F. L. (2009). *Python Tutorial*. Python Software Foundation. Recuperado de https://docs.python.org/es/3/tutorial/
2. Sweigart, A. (2019). *Automate the Boring Stuff with Python*. No Starch Press.
3. Real Python. (n.d.). *Python Basics*. Recuperado de https://realpython.com/python-basics/