

Introduccion a Python

Guia completa para estudiantes y principiantes

Aprende los conceptos base

variables, condiciones, bucles, funciones y buenas practicas



Python como herramienta para crear soluciones digitales

1. Objetivos de esta guia

Esta guia explica Python desde cero con lenguaje claro, ejemplos y diagramas. La idea es que puedas leerla como material de estudio, repasar conceptos y usarla como referencia rapida.

Ruta de aprendizaje



Que aprenderas

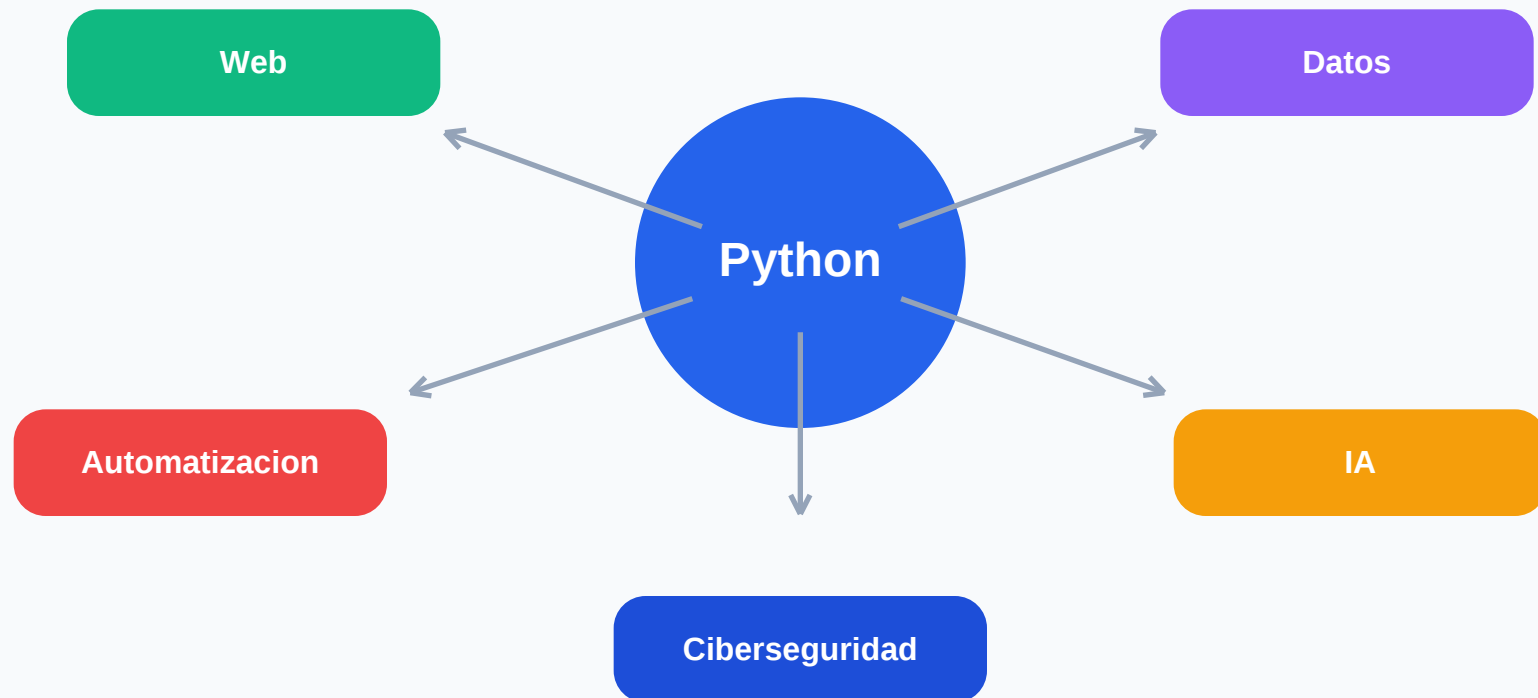
Al terminar, sabras reconocer la estructura de un programa simple, leer ejemplos basicos y entender por que Python se usa tanto en tecnologia.

Consejo de estudio

No intentes memorizar todo. Lee, prueba ejemplos y vuelve a esta guia cuando necesites recordar una idea.

2. Que es Python

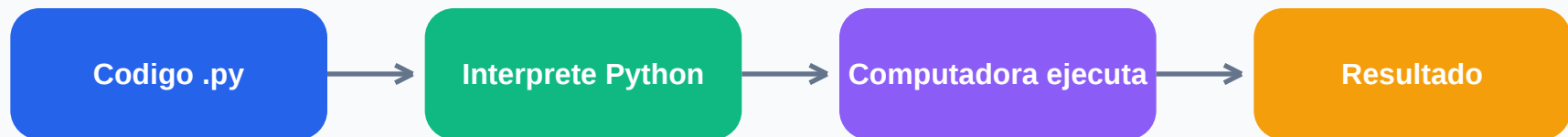
Python es un lenguaje de programacion usado para crear instrucciones que una computadora puede ejecutar. Es popular porque su sintaxis es clara, se lee facilmente y permite construir desde scripts pequenos hasta sistemas mas avanzados.



Idea clave: Python no es solo para programadores expertos. Tambien es una excelente puerta de entrada para aprender logica, resolver problemas y comprender mejor herramientas digitales.

3. Como se ejecuta un programa

Cuando escribes código Python, el programa pasa por una serie de pasos antes de mostrar un resultado. Esta idea ayuda a entender por qué un error en el código puede detener la ejecución.



Archivo ejemplo.py

```
nombre = "Ana"  
print("Hola", nombre)
```

Salida esperada

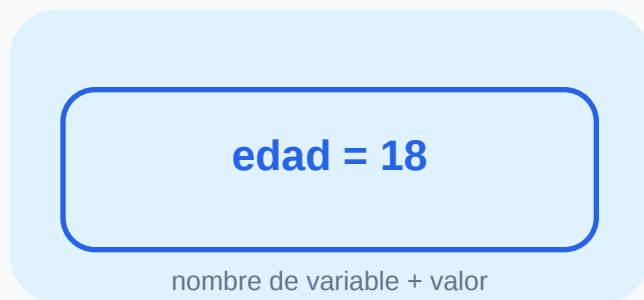
El programa muestra un mensaje en pantalla. Si escribiste todo correctamente, verás algo parecido a: Hola Ana.

Importante

El interprete lee instrucciones. Si encuentra una palabra mal escrita o una estructura incorrecta, puede mostrar un mensaje de error.

4. Variables y tipos de datos

Una variable es un nombre que guarda un valor. Puedes imaginarla como una caja con etiqueta: la etiqueta es el nombre de la variable y adentro esta el dato.



Tipo	Ejemplo	Uso
str	"ByteNova"	texto
int	25	numero entero
float	3.14	numero decimal
bool	True	verdadero/falso

Variables

```
nombre = "Dylan"  
edad = 20  
activo = True  
print(nombre, edad)
```

Las variables permiten reutilizar informacion. Si cambias el valor de una variable, todo el codigo que la usa puede trabajar con el nuevo dato.

5. Operadores y expresiones

Los operadores permiten hacer operaciones con datos. En Python puedes sumar, comparar, unir textos y crear expresiones logicas.

Aritmeticos

+ - * /

calculos

Comparacion

== != > <

preguntas

Logicos

and or not

condiciones

Asignacion

= += -=

guardar cambios

Operacion

```
precio = 100
descuento = 20
total = precio - descuento
print(total) # 80
```

Comparacion

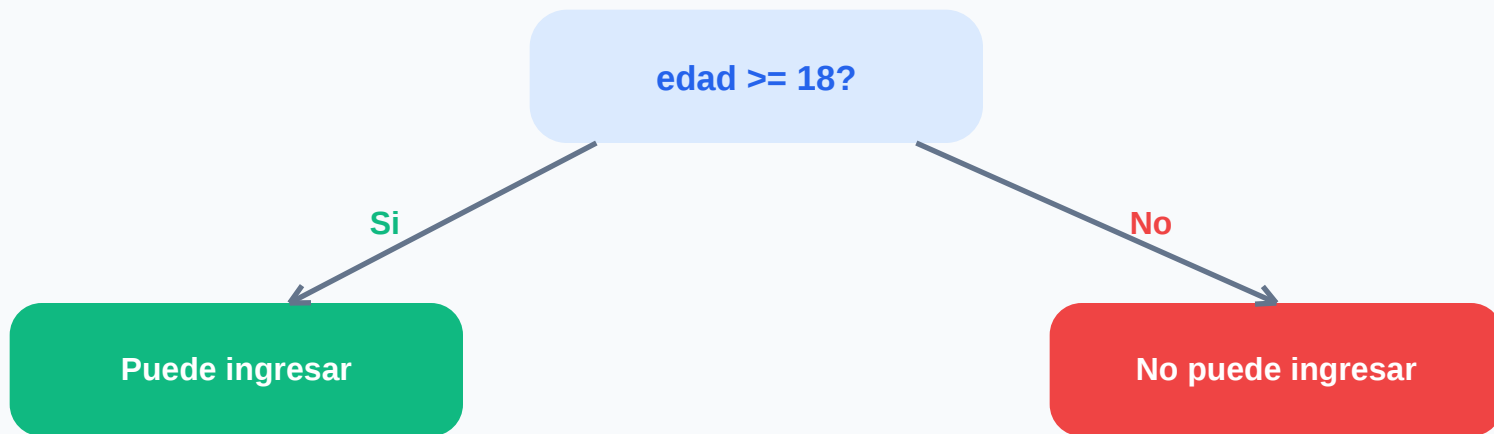
```
edad = 18
puede_entrar = edad >= 18
print(puede_entrar)
```

Tip

Una expresion es una combinacion de valores, variables y operadores que produce un resultado.

6. Condiciones: if, elif y else

Las condiciones permiten que un programa tome decisiones. Si una condicion se cumple, se ejecuta una parte del codigo; si no, se puede ejecutar otra.



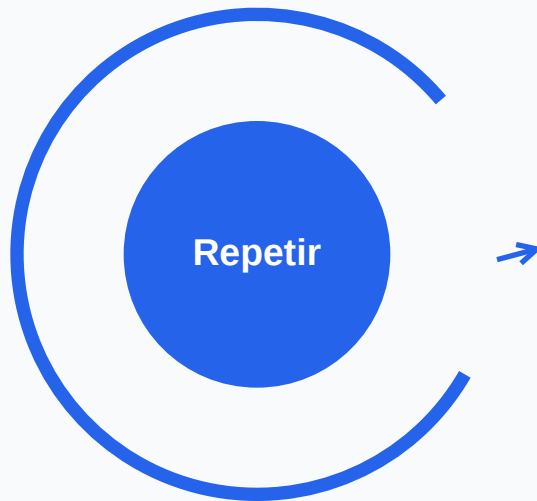
Decision con if

```
edad = 17
if edad >= 18:
    print("Puede ingresar")
else:
    print("No puede ingresar")
```

La indentacion es el espacio al inicio de una linea. En Python es muy importante porque indica que instrucciones pertenecen al if o al else.

7. Bucles: repetir instrucciones

Un bucle permite repetir instrucciones. Python tiene bucles for y while. Un for suele usarse cuando sabes que vas a recorrer una coleccion; un while se usa mientras una condicion siga siendo verdadera.



hasta terminar la tarea

Bucle for

```
for numero in range(1, 4):  
    print(numero)
```

Bucle while

```
contador = 1  
while contador <= 3:  
    print(contador)  
    contador += 1
```

Cuidado

Un while mal diseñado puede convertirse en un bucle infinito si la condicion nunca cambia.

8. Listas y diccionarios

Las listas guardan varios valores en orden. Los diccionarios guardan informacion usando pares de clave y valor. Ambas estructuras ayudan a organizar datos.

Lista con posiciones



Lista

```
temas = ["HTML", "CSS", "Python"]  
print(temas[0]) # HTML  
temas.append("IA")
```

Diccionario

```
usuario = {  
    "nombre": "Ana",  
    "edad": 18  
}  
print(usuario["nombre"])
```

Recuerda: en una lista, el primer elemento tiene índice 0. Este detalle es muy comun en programacion.

9. Funciones: organizar el código

Una función es un bloque de código que realiza una tarea específica. Sirve para no repetir instrucciones y para ordenar mejor un programa.



Funcion con return

```
def saludar(nombre):  
    mensaje = "Hola " + nombre  
    return mensaje  
  
resultado = saludar("ByteNova")  
print(resultado)
```

Ventaja

Las funciones hacen que el código sea más limpio, reutilizable y fácil de corregir.

10. Errores y buenas practicas

Los errores son parte normal del aprendizaje. Un mensaje de error no significa fracaso; significa que Python encontro algo que necesita corregirse.

Tipo de error	Ejemplo	Que revisar
Sintaxis	falta un parentesis	simbolos, comillas, dos puntos
Indentacion	espacios mal puestos	bloques if, for, funciones
Nombre	variable no definida	ortografia del nombre
Logica	resultado inesperado	orden de pasos y condiciones

Buenas practicas

Usa nombres claros, escribe poco codigo por vez, prueba con frecuencia y comenta partes importantes.

Como depurar

Lee el mensaje de error, revisa la linea indicada y compara con un ejemplo correcto.

11. Mini proyecto y checklist

Para practicar, puedes crear un pequeno programa que pida datos, haga una decision y muestre un resultado. Este tipo de ejercicio une varios conceptos en una sola tarea.

Mini proyecto: registro simple

```
nombre = input("Tu nombre: ")
edad = int(input("Tu edad: "))

if edad >= 18:
    print(nombre, "puede registrarse")
else:
    print(nombre, "debe esperar")
```

Checklist de estudio

- ☐ Comprendo que es una variable
- ☐ Puedo usar print e input
- ☐ Entiendo if y else
- ☐ Se que es un bucle
- ☐ Puedo crear una funcion
- ☐ Se leer errores basicos

Siguiente paso en ByteNova

Practica modificando el mini proyecto: agrega una variable nueva, cambia la condicion o muestra un mensaje diferente. Puedes cambiar textos, condiciones o agregar nuevas respuestas.