





Clase 3: Conceptos Básicos de Programación

1. Semántica vs. Sintaxis

- **Semántica:**


- Se refiere al sentido o significado que le damos al código. Por ejemplo, si estamos trabajando con operaciones matemáticas y recibimos números como entrada, también debemos retornar números para mantener la coherencia. 

- **Sintaxis:**

- Se refiere a la forma en que escribimos el código, siguiendo las reglas del lenguaje de programación. Por ejemplo, cada paréntesis abierto debe ser cerrado. 

2. Ejemplos Prácticos de Sintaxis


- **Errores de Sintaxis:**

- Si borramos un paréntesis de apertura sin cerrar, el lenguaje mostrará un error de sintaxis .

- Ejemplo:


```
print("Hola"  # Falta cerrar el paréntesis
```

- **Indentación:**

- Python es muy estricto con la indentación (espacios al inicio de una línea)
.
- Ejemplo:

```
print("Hola")  # Correcto
print("Mundo") # Incorrecto si se espera que esté ali-
neado
```



- **Errores de Indentación:**

- La indentación incorrecta puede generar errores .
- Ejemplo:

```
if True:
print("Hola")  # Error de indentación
```

3. Variables

- **Definición:**

- Las variables son contenedores para almacenar datos. Se definen con un nombre, el signo de igual  y el valor a almacenar .
- Ejemplo:

```
saludo = "Hola"
nombre = "Charlie"
```

- **Uso de Variables:**

- Podemos usar variables para almacenar y reutilizar datos 📦.
- Ejemplo:

```
saludo = "Hola"
nombre = "Charlie"
print(saludo) # Imprime: Hola
print(nombre) # Imprime: Charlie
```

- **Errores Comunes:**

- **Variable No Definida:**

```
print(nombre) # Error si 'nombre' no ha sido definido
```

- **Nombres de Variables Confusos:**

```
edad = "Charlie" # No tiene sentido usar 'edad' para un nombre
```

- **Reglas para Nombres de Variables:**


- Deben comenzar con una letra o un guion bajo `_`.
- Pueden contener letras, números y guiones bajos.
- No deben comenzar con un número ni usar palabras reservadas.
- Ejemplo válido: `saludo_2`, `nombre123`
- Ejemplo inválido: `2saludo`, `saludo-uno`

- **Palabras Reservadas:**

- No se pueden usar palabras reservadas del lenguaje para nombrar variables 🚫.
- Aquí tienes una lista de todas las palabras reservadas en Python, que no deben usarse como nombres de variables:

False	await	else	import	pass
None	break	except	in	raise
True	class	finally	is	return
and	continue	for	lambda	try
as	def	from	nonlocal	while
assert	del	global	not	with
async	elif	if	or	yield

- **Reasignación de Variables:**

- Los valores de las variables pueden ser cambiados .
- Ejemplo:

```
saludo = "Hola"
saludo = "Adiós"
print(saludo) # Imprime: Adiós
```

- **Uso de Caracteres Especiales y Números:**

- Se pueden usar guiones bajos para separar palabras.
- Ejemplo válido: `saludo_1`, `saludo_2`
- Ejemplo inválido: `1saludo`, `saludo-uno`

- **Ejemplo Adicional:**

```
saludo1 = "Hola"
saludo2 = "Mundo"
saludo1 = "Adiós"
print(saludo1) # Imprime: Adiós
print(saludo2) # Imprime: Mundo
```