

CLASE 1: SINTAXIS, VARIABLES Y TIPOS DE DATOS

CURSO DE PYTHON

Fundamentos esenciales para comenzar a programar en Python desde cero

PROF: KEVIN T.A. | ING. SOFTWARE



INTRODUCCION

EN ESTA PRIMERA CLASE, APRENDEREMOS LOS PILARES DE CUALQUIER LENGUAJE DE PROGRAMACIÓN: LA SINTAXIS, LAS VARIABLES Y LOS TIPOS DE DATOS. ESTOS SON LOS BLOQUES BÁSICOS QUE NOS PERMITIRÁN COMUNICAR NUESTRAS IDEAS A LA COMPUTADORA DE MANERA EFECTIVA, ¡Y CON PYTHON COMO NUESTRO LENGUAJE, TODO SE VUELVE MÁS SENCILLO Y DIVERTIDO! ”

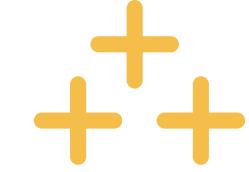


OBJETIVO

¿Qué valores? 

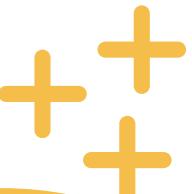
OBJETIVO GENERAL:

- Introducir a los estudiantes en los fundamentos de la programación utilizando Python, enfocados en entender la sintaxis básica, el uso de variables y los tipos de datos para resolver problemas simples.



OBJETIVOS ESPECÍFICOS:

1. Familiarizar a los estudiantes con la sintaxis básica de Python, incluyendo cómo escribir, ejecutar y entender un programa simple.
2. Introducir el concepto de variables y tipos de datos, y cómo se utilizan para almacenar y manipular información en un programa.
3. Desarrollar habilidades iniciales de resolución de problemas mediante el uso de estructuras sencillas en Python.
4. Establecer una base sólida para continuar aprendiendo programación, asegurando que los estudiantes comprendan los elementos esenciales para escribir código funcional.



SINTAXIS

¿Qué es la sintaxis de un lenguaje de programación?

Como en el español donde las palabras tienen un orden o como en el inglés las palabras tienen otro orden, de forma similar, en Python (y en cualquier otro lenguaje de programación), debemos seguir un orden y una estructura específicos para que el código funcione. Si no respetamos esa sintaxis, el programa mostrará errores y no se ejecutará correctamente.

```
# Python                                // C++  
print("Hola en Python")                 cout<<"Hola en C++";
```

```
// Java                                 // JavaScript  
System.out.println("Hola en Java");    console.log("Hola en JS");
```

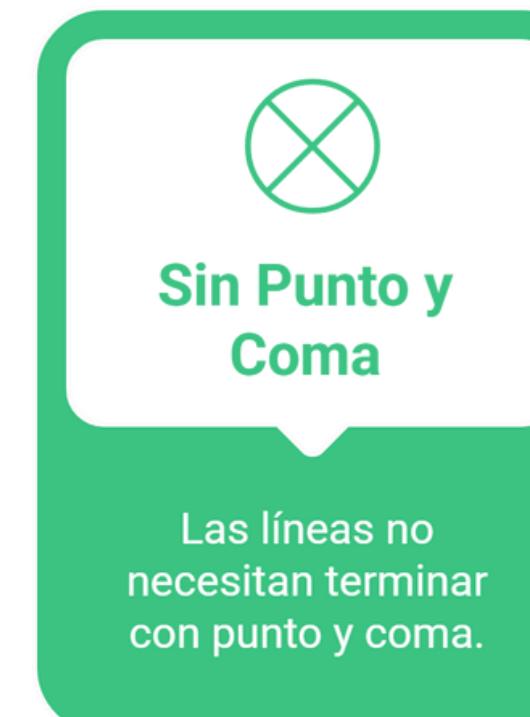


SINTAXIS EN PYTHON

¿Qué es la sintaxis en Python?

La sintaxis en Python es el conjunto de reglas que debemos seguir al escribir código para que el programa funcione correctamente.

Python se caracteriza por tener una sintaxis **simple, clara y muy parecida al lenguaje humano**, lo que lo hace ideal para principiantes.



VARIABLES

¿Qué es una variable?

Una variable es un espacio de memoria donde se guardara un valor, el cual será utilizado luego.

En Python no es necesario declarar el tipo de la variable.

Sintaxis para declarar variables:

```
nombre = "Ana"          # variable tipo cadena (string)
edad = 20                # variable tipo entero (int)
altura = 1.65            # variable tipo decimal (float)
es_estudiante = True    # variable tipo booleano (bool)
```

Reglas para nombrar variables en Python:

1. Debe comenzar con una letra o un guion bajo `_`, nunca con un número.
2. No puede contener espacios, se recomienda usar guiones bajos: `nombre_completo` (snake case)
3. Distingue entre mayúsculas y minúsculas: `Nombre` y `nombre` son variables diferentes.
4. No puede usar palabras reservadas de Python como `if`, `for`, `class`, etc.



REPRESENTACIÓN DE ESPACIO DE VARIABLE

¿Dónde está la variable o donde la ubico?

La variable se guarda en un espacio de memoria, pero no es como que se guarde ordenadamente, sino que tiene un lugar asignado y este puede ser ubicado. ¡Imaginemos casas como lugares para guardar variables y el valor de la variable como una familia que vive dentro de esa casa!

Representación visual



Av. Universitaria
Mz C1 1893
Pueblo Libre
Lima
Perú



Av. Riva Agüero
Mz B3 125
El Agustino
Lima
Perú



Av. Mariscal Cáceres
Mz A5 496
Iquitos
Loreto
Perú

Representación en código: `personas = {"Jose", "Pedro", "María"}`



LA NATURALEZA CONTEXTUAL DE LAS VARIABLES

Las variables no solo almacenan datos, sino que su tipo depende del contexto y propósito con el que se utiliza esa información. Esto significa que no basta con identificar el valor de una variable; también debemos considerar cómo será interpretado o usado en un programa.

Ejemplo

Número: **943617892**

1. Si se usa como un contador o para realizar operaciones matemáticas:

- **Tipo:** Entero (int).
- **Uso:** Sumar, restar o realizar cálculos.
- **Ejemplo:** Incrementar un contador en 1.

2. Si representa un número de teléfono:

- **Tipo:** Cadena (string).
- **Uso:** Almacenar como texto, ya que no se realizarán operaciones matemáticas con él.
- **Ejemplo:** Formatear como "+51 943-617-892".

3. Si se trata de una clave única en una base de datos:

- **Tipo:** Cadena o, en algunos casos, identificador (UUID).
- **Uso:** Referenciar algo específico, no operar matemáticamente.

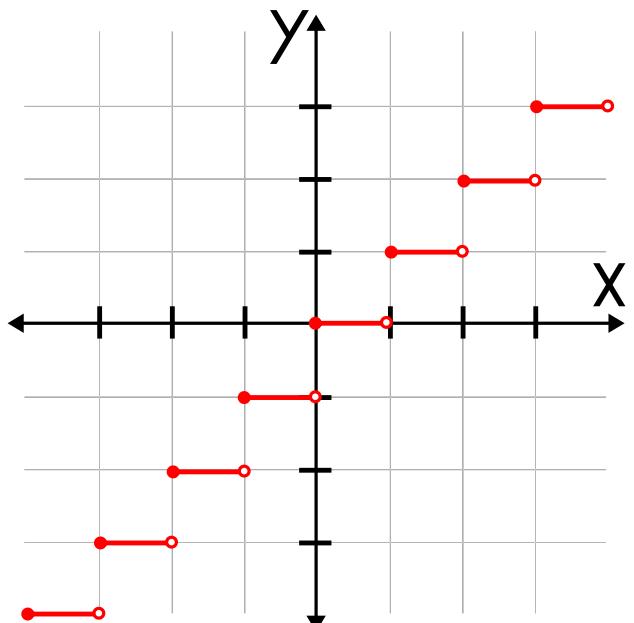


TIPOS DE VARIABLES.

Enteros (int)

Se utilizan para almacenar números enteros, sin decimales.

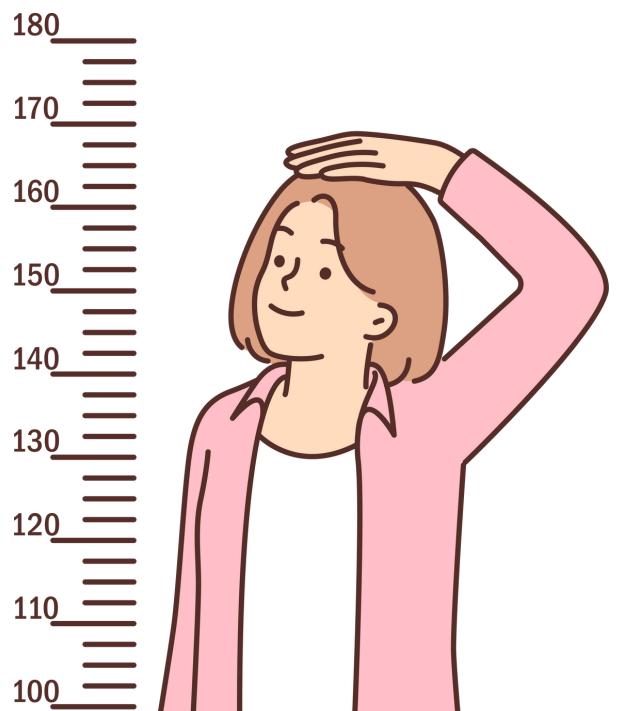
- Ejemplos:
 - Cantidad: 10 (número de manzanas).
 - Edad: 25 (años de una persona).
 - Años: 2024 (año actual).
 - Días: 365 (días en un año).



Reales (float o double)

Se usan para valores con decimales, como mediciones o cálculos precisos.

- Ejemplos:
 - Altura: 1.75 metros.
 - Peso: 68.5 kilogramos.
 - Promedios: 89.3 (promedio de calificaciones).
 - Medidas: 3.1416 (valor de π).



Caracter (char)

Almacena un solo símbolo, letra o número en formato ASCII.

- Ejemplos:
 - Símbolos: 'A', '5', '#'.

Caracteres ASCII imprimibles				ASCII extendido (Página de código 437)			
32	espacio	64	@	128	ç	160	á
33	!	65	A	129	ú	161	í
34	"	66	B	130	é	162	ó
35	#	67	C	131	â	163	û
36	\$	68	D	132	ã	164	ñ
37	%	69	E	133	à	165	N
38	&	70	F	134	à	166	ª
39	'	71	G	135	ç	167	º
40	(72	H	136	è	168	ξ
41)	73	I	137	ë	169	®
42	*	74	J	138	è	170	™
43	+	75	K	139	í	171	½
44	,	76	L	140	í	172	¼
45	-	77	M	141	í	173	i
46	.	78	N	142	À	174	«
47	/	79	O	143	ò	175	»
48	0	80	P	144	É	176	ò
49	1	81	Q	145	æ	177	ò
50	2	82	R	146	Æ	178	ò
51	3	83	S	147	ò	179	—
52	4	84	T	148	ö	180	—
53	5	85	U	149	ò	181	Á
54	6	86	V	150	ú	182	À
55	7	87	W	151	ù	183	À
56	8	88	X	152	ÿ	184	©
57	9	89	Y	153	ö	185	í
58	:	90	Z	154	Ü	186	—
59	:	91	[155	ø	187	—
60	<	92	\	124]	156	£
61	=	93	I	125	}	157	Ø
62	>	94	^	126	~	158	×
63	?	95	_	190	¥	222	—
				159	f	191	nnbsp



TIPOS DE VARIABLES.

Cadenas (string)

Se usan para almacenar texto, como palabras o frases.

- Ejemplos:
 - Palabras: "Hola".
 - Oraciones: "El cielo es azul".
 - Texto sin sentido: "xkjsdf98wq".



Booleano (bool)

Solo tiene dos valores posibles, usado para tomar decisiones lógicas.

- Ejemplos:
 - Verdadero: true (es mayor de edad).
 - Falso: false (el usuario no ingresó la contraseña correcta).

TRUE

FALSE



OPERADORES ARITMÉTICOS

Los operadores son símbolos especiales que se utilizan para realizar operaciones sobre variables y valores. Python incluye varios tipos de operadores que nos permiten manipular datos, comparar valores y tomar decisiones.

Operador	Descripcion	Ejemplo	Ejemplo real
+	Suma	$a+b$	$15+7 = 22$
-	Resta	$a-b$	$15-7 = 8$
*	Multiplicación	$a*b$	$15*7 = 105$
/	Division	a/b	$15/7 = 2.1428$
//	División Exacta	$a//b$	$15//7 = 2$
%	Modulo	$a \% b$	$15 \% 7 = 1$
**	Potencia	$a ** b$	$15 ** 7 = 170859375$



OPERADORES DE COMPARACIÓN

Permiten comparar dos valores y devuelven un resultado booleano (True o False).

Operador	Descripción	Ejemplo	Ejemplo real
<code>==</code>	Igual a	<code>a == b</code>	<code>8 == 6</code> (False)
<code>!=</code>	Distinto de	<code>a != b</code>	<code>8 != 6</code> (True)
<code>></code>	Mayor que	<code>a > b</code>	<code>8 > 6</code> (True)
<code><</code>	Menor que	<code>a < b</code>	<code>8 < 6</code> (False)
<code>>=</code>	Mayor o igual que	<code>a >= b</code>	<code>8 >= 6</code> (True)
<code><=</code>	Menor o igual que	<code>a <= b</code>	<code>8 <= 6</code> (False)



OPERADORES DE ASIGNACIÓN

Sirven para asignar valores a una variable.

Operador	Descripción	Ejemplo	Ejemplo real
=	Asignación simple	x = 5	Asigna el valor 5 a x
+=	Suma y asignación	x += 3	x += 3 →
-=	Resta y asignación	x -= 2	x -= 2 → 3
*=	Multiplicación y asignación	x *= 4	x *= 4 → 20
/=	División y asignación	x /= 2	x /= 2 → 2.5
//=	División entera y asignación	x //= 3	x //= 3 → 1
%=	Módulo y asignación	x %= 2	x %= 2 → 1
**=	Potencia y asignación	x **= 3	x **= 3 → 125



EJERCICIO PARA PENSAR

1. Imagina un lenguaje de programación sin tipos de datos.

- ¿Qué ventajas podría tener?
- ¿Qué problemas causarían si todo fuera lo mismo?
- ¿Cómo te afectaría como programador?



2. Una variable se llama así porque puede variar, pero ¿qué pasaría si solo declararas variables que nunca cambian?

- ¿Serían realmente “variables”?
- ¿Para qué sirven variables inmutables?
- ¿Existen cosas en la vida real que parecen variables pero son constantes?

3. Hoy tenemos enteros, cadenas, booleanos, flotantes...

- ¿Qué tipo de dato falta?
- ¿Qué datos no sabemos representar todavía en los lenguajes actuales?
- ¿Cómo se llamaría ese nuevo tipo?



NOTA: RESOLVER CON LA MENTE





UNIVERSIDAD NACIONAL MAYOR DE
SAN MARCOS
Universidad del Perú, DECANA DE AMÉRICA



"HOY EMPEZAMOS DESDE LO MÁS BÁSICO, PERO
RECUERDA: TODOS LOS GRANDES
PROGRAMADORES COMENZARON IGUAL. ESTOY
AQUÍ PARA ACOMPAÑARTE EN CADA PASO.
CONFÍA EN EL PROCESO Y EN TI. ¡LO MEJOR
ESTÁ POR VENIR!"

