



INTRODUCCIÓN A PYTHON: TIPOS DE DATOS, VARIABLES, OPERADORES Y EXPRESIONES

10110 – FUNDAMENTOS DE COMPUTACIÓN Y PROGRAMACIÓN

10145 - FUNDAMENTOS DE PROGRAMACIÓN PARA INGENIERÍA



RESUMEN DE CONTENIDOS

OPERADORES ARITMÉTICOS Y PRECEDENCIA



OPERACIÓN	OPERADOR	ARIDAD	ASOCIATIVIDAD	PRECEDENCIA
EXPONENCIACIÓN	**	BINARIA	DERECHA	1
IDENTIDAD	+	UNARIA	-	2
NEGACIÓN	-	UNARIA	-	2
MULTIPLICACIÓN	*	BINARIA	IZQUIERDA	3
DIVISIÓN	/	BINARIA	IZQUIERDA	3
DIVISIÓN ENTERA	//	BINARIA	IZQUIERDA	3
MÓDULO	%	BINARIA	IZQUIERDA	3
SUMA	+	BINARIA	IZQUIERDA	4
RESTA	-	BINARIA	IZQUIERDA	4



TIPOS NUMÉRICOS

- En Python existen tres tipos de datos para representar números: `int`, `float` y `complex`
 - **Enteros** (`int`): Representan números enteros, positivos y negativos
 - **Números de punto flotante** (`float`): Representan números no enteros, no son una representación perfecta del conjunto matemático \mathbb{R} , también se les llama flotantes
 - **Números complejos** (`complex`): Representan números que tienen parte real e imaginaria, sin embargo, tampoco son una representación totalmente fiel del conjunto matemático \mathbb{C}



OTROS TIPOS DE DATO

- **Booleano** (**bool**): Se utiliza para representar valores lógicos
 - Un Booleano sólo puede ser **True** o **False** dependiendo del caso
 - Internamente se representan como números, 0 para **False** y 1 para **True**
- **String** (**str**): Se utiliza para almacenar cadenas de texto
 - Se identifican fácilmente porque inician y cierran con comillas. Por ejemplo: **'HOLA'**, **"2.7"**, **'EL5398'**
 - Se utilizan para capturar entradas para el programa o para informar de resultados a un usuario



FUNCIONES DE *TYPECASTING*

FUNCIÓN	EJEMPLO	RESULTADO
<code>int()</code>	<code>int('450')</code>	450
<code>float()</code>	<code>float(" -3.5 ")</code>	-3.5
<code>complex()</code>	<code>complex(2,3)</code>	(2+3j)
<code>bool()</code>	<code>bool(3.8)</code> <code>bool('False')</code>	True True
<code>str()</code>	<code>str(4.3440000)</code> <code>str(False)</code>	'4.344' 'False'



ASIGNACIÓN

- Una asignación es una sentencia con la siguiente estructura:

`<identificador> = <expresión>`

- Reglas de un identificador:

- El primer caracter no puede ser un dígito
- Puede llevar letras, dígitos y el caracter subrayado (`_`)
- No puede coincidir con las palabras reservadas de Python:

`and, assert, break, class,
continue, def, del, elif, else,
except, exec, finally, for,
from, global, if, import, in,
is, lambda, not, or, pass,
raise, return, try, while,
yield`

- Puede ser:

- Un valor constante (un número o texto)
- Una operación entre números
- Una operación entre variables y constantes previamente declaradas
- Mezcla entre operaciones, variables y números



EJERCICIOS



EJERCICIO PROPUESTO 1

- ¿Con qué valor termina la variable resultado en este caso?

```
valor = 12
```

```
resultado = valor + resultado
```

```
resultado = float(resultado ** 2)
```

```
resultado = resultado - valor % 3
```



EJERCICIO PROPUESTO 2

- Asumiendo que tengo el siguiente código y que no puedo definir nuevas variables, ni nuevos valores
 - $a = 1$
 - $b = 2$
- ¿Qué instrucciones debo escribir para que se cumpla que:
 - $a = 2$ y
 - $b = 1$?



TAREAS PARA TRABAJO AUTÓNOMO

1. Revisar el apunte:

- Tipos de datos, operadores y expresiones en Google Colab (Disponible en: https://github.com/PROGRA-FING-USACH/2023-1/blob/main/Lecturas/02_Input_output.ipynb)



¿CONSULTAS?