

Camino al Certamen 1: Resumen de Contenidos

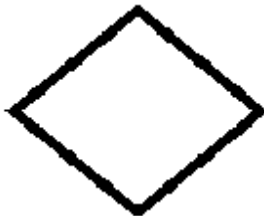
Diagramas de Flujo



Comienzo y Fin



Proceso General



Toma de Decisiones



Entrada o Salida
De Datos



Líneas de Flujo

Tipos de Datos

Números Enteros: tipo **int**

Ejemplos: [...], -2, -1, 0, 1, 2, [...]

Números Reales: tipo **float**

Ejemplos: 1/7, 3.1415, 7.35e-2

Números Complejos: tipo **complex**

Ejemplos: 3+9j, -1.4+2.7j

Valores Lógicos: tipo **bool**

Sólo 2 posibilidades: **True** o **False**

Texto (strings): tipo **str**

Ejemplo: "Este es un string"

IF - ELIF - ELSE

If (condicion1):

#Condicion1 es Verdadera

elif (condicion2):

#Condicion2 es Verdadera

elif (condicion3):

#Condicion3 es Verdadera

[...]

else:

#Ninguna condición es verdadera

Funciones

Definición def nombre(parámetros):

#código de la func.

- Las funciones **pueden o no** devolver

Algún valor (pueden o no tener un
return al final de la misma)

- En las funciones **no se usa print** para
devolver resultados, **se usa return!**

Ciclo while

while (condicion):

#Esto se ejecuta mientras la

#condición sea verdadera

Creado por Camilo Saldías

<https://github.com/csaldias/python-usm>

Precedencia de Operadores

La precedencia de los operadores determina el orden en el que son evaluados dentro de una expresión. En la tabla, los operadores están ordenados del operador con la mayor prioridad (arriba) hasta el operador con la menor prioridad (abajo.) Los operadores en un mismo cuadro son evaluados de izquierda a derecha.

Operador	Descripción
()	Paréntesis (agrupación)
f(parámetros...)	Funciones
x[indice:índice]	Operador de Rebanado
x.atributo	Atributos
**	Exponenciación
*, /, %	Multiplicación, División, Resto
+, -	Adición y Sustracción
in, not in, <, <=, >, >=, <>, !=, ==	Comparaciones, pertenencia
not x	Negación
and	Y lógico (o booleano)
or	O lógico (o booleano)