

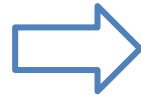


Programación  
Licenciatura en Agroinformática  
Licenciatura en Bioinformática

# Tema 02: Tipos de Datos

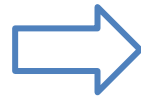
# Resolución de problemas (de nuevo)

Análisis



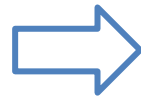
Entradas y salidas

Diseño



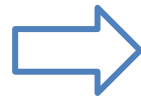
Pasos y Pseudocódigo

Implementación



Código

Depuración



Corrección de errores

# Python

- Un lenguaje de programación permite dar instrucciones a un sistema informático.
- Python es un lenguaje
  - de uso general
  - multiplataforma
  - de (muy) alto nivel.
  - interpretado
  - orientado a objetos
  - de tipado dinámico.
  - con una extensa librería de funciones.
  - con una comunidad activa.

# Ejemplo de Programa en Python

```
#Codigo para calcular la hipotenusa
cat_menor = 3
cat_mayor = 4
# "*" significa "elevado a"
hip = (cat_mayor**2 + cat_menor**2)**0.5
print (hip)
```

*Variables*

*Comentarios*

*Funciones*

*Operadores*

# Variables

- Permiten almacenar datos en memoria.
- Los nombres de las variables (y de las funciones) deben cumplir ciertas reglas:
  - Empezar con una letra.
  - El resto del nombre puede contener solo letras, números o guiones bajos ( \_ )
  - Mayúsculas y minúsculas son caracteres distintos (“Variable” y “variable” son dos nombres distintos!)
- Para asignar un valor a una variable se usa el signo igual ( = )

# Palabras reservadas

and

continue

except

global

lambda

raise

yield

as

def

exec

if

not

return

assert

del

finally

import

or

try

break

elif

for

in

pass

while

class

else

from

is

with

False

None

True

nonlocal

# Tipos de Datos

- Primitivos:
    - Enteros (int)
    - Reales (float)
    - Cadenas de caracteres (strings)
    - Verdadero / falso (boolean)
  - Abstractos:
    - Tuplas y listas
    - Diccionarios
    - Grafos
    - Arboles
    - ...
- Algoritmos y Estructuras de Datos!!*

# Tipos de Datos Numéricos

- Existen tres tipos de datos numéricos nativos en Python: **enteros**, **flotantes** y **complejos**.
- Los **enteros** pueden ser tanto positivos como negativos:
  - 5
  - -5
- Los **flotantes** son números que admiten decimales:
  - 5.0
  - -1.9
  - 6.022E23
  - 0.0000038
- No existe un límite para el largo de los números (pero para números muy grandes conviene usar notación científica).



# Cadenas de caracteres

- Permiten almacenar información en formato de texto.
- Un **string** puede almacenar cualquier tipo de carácter del código ASCII, como letras, números, signos de puntuación y espacios .
- Para definir un string se pueden utilizar tanto comillas simples ( ' ' ) como dobles ( " " ).
  - "Este es un string"
  - 'Este es otro string'
  - "223344"
  - "...!!"

# Codigo ASCII

## Caracteres ASCII de control

00	NULL	(carácter nulo)
01	SOH	(inicio encabezado)
02	STX	(inicio texto)
03	ETX	(fin de texto)
04	EOT	(fin transmisión)
05	ENQ	(consulta)
06	ACK	(reconocimiento)
07	BEL	(timbre)
08	BS	(retroceso)
09	HT	(tab horizontal)
10	LF	(nueva línea)
11	VT	(tab vertical)
12	FF	(nueva página)
13	CR	(retorno de carro)
14	SO	(desplaza afuera)
15	SI	(desplaza adentro)
16	DLE	(esc.vínculo datos)
17	DC1	(control disp. 1)
18	DC2	(control disp. 2)
19	DC3	(control disp. 3)
20	DC4	(control disp. 4)
21	NAK	(conf. negativa)
22	SYN	(inactividad sinc)
23	ETB	(fin bloque trans)
24	CAN	(cancelar)
25	EM	(fin del medio)
26	SUB	(sustitución)
27	ESC	(escape)
28	FS	(sep. archivos)
29	GS	(sep. grupos)
30	RS	(sep. registros)
31	US	(sep. unidades)
127	DEL	(suprimir)

## Caracteres ASCII imprimibles

32	espacio	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	\$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(	72	H	104	h
41	)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[	123	{
60	<	92	\	124	
61	=	93	]	125	}
62	>	94	^	126	~
63	?	95	_		

## ASCII extendido (Página de código 437)

128	Ç	160	á	192	Ł	224	Ó
129	ü	161	í	193	ł	225	ô
130	é	162	ó	194	Ł	226	Ô
131	â	163	ú	195	ł	227	ò
132	ä	164	ñ	196	—	228	ō
133	à	165	Ñ	197	+	229	Õ
134	ã	166	ª	198	ā	230	μ
135	ç	167	º	199	Ä	231	þ
136	ê	168	¿	200	Ł	232	þ
137	ë	169	®	201	Ł	233	ú
138	è	170	¬	202	Ł	234	û
139	ï	171	½	203	Ł	235	ù
140	î	172	¼	204	Ł	236	ý
141	ì	173	í	205	=	237	ÿ
142	Ä	174	«	206	Ł	238	—
143	Å	175	»	207	▣	239	˘
144	É	176		208	ð	240	≡
145	æ	177		209	Ð	241	±
146	Æ	178		210	Ê	242	≡
147	ø	179		211	È	243	¼
148	ö	180		212	È	244	¶
149	ò	181	À	213	ı	245	§
150	û	182	Ā	214	ı̇	246	÷
151	ü	183	Ă	215	ı̇ı̇	247	˚
152	ÿ	184	Ą	216	ı̇ı̇ı̇	248	˚
153	Ō	185	Ć	217	ı̇ı̇ı̇ı̇	249	˚
154	Ū	186	Ĉ	218	ı̇ı̇ı̇ı̇ı̇	250	˚
155	ø	187	Ç	219	ı̇ı̇ı̇ı̇ı̇ı̇	251	˚
156	£	188	Ɔ	220	ı̇ı̇ı̇ı̇ı̇ı̇ı̇	252	˚
157	Ø	189	€	221	ı̇ı̇ı̇ı̇ı̇ı̇ı̇ı̇	253	˚
158	x	190	¥	222	ı̇ı̇ı̇ı̇ı̇ı̇ı̇ı̇ı̇	254	■
159	f	191	ſ	223	ı̇ı̇ı̇ı̇ı̇ı̇ı̇ı̇ı̇ı̇	255	nbsp

# Caracteres especiales comunes

- Para ingresar ciertos elementos en un string es necesario utilizar “caracteres de escape”.
- Todos estos caracteres comienzan con una barra invertida (“\”)

Caracter	Funcion
\'	Comilla simple dentro del string
\"	Comilla doble dentro del string
\n	Fin de linea
\t	tabulacion

- El string *I'm learning so much today* se podría almacenar como:
  - `string1 = 'I\'m learning so much today'`
  - `string2 = "I'm learning so much today"`

# Tipos de Datos Lógicos

- Los booleanos admiten dos valores: verdadero (True) y falso (False)
- Son muy utilizados para evaluar alternativas ("si, sino").
  - `if (32 > 12)`  $\rightarrow$  True
  - `if ("arbol" > "bici")`  $\rightarrow$  False
  - `if ("vaca" > "ñandu")`  $\rightarrow$  False
- Si una variable no está vacía, al evaluarla dará True
  - `var1 = "Algo"`
  - `var2 = ""`
  - `if (var1)`  $\rightarrow$  True
  - `if (var2)`  $\rightarrow$  False

# Constantes

- Es un dato que no se modifica en toda la ejecución del programa.
- En Python no existe el tipo de dato “constante”, sino que se definen variables que luego no se modifican.
- Una convención para diferenciar constantes de variables en un código es utilizar nombres en mayúsculas
- Algunos ejemplos de constantes muy utilizadas:
  - $\pi = 3.1415$
  - $G = 9.81$
  - $E = 2.7182$

# Comentarios

- Son líneas en el código que no son leídas por el intérprete.
- Comienzan con un numeral ( # ).
- Si se desean comentar más de una línea, se pueden utilizar 3 comillas dobles ( "" )

```
# Esto es una linea de comentarios
var1 = "Esto es una variable"
# Esto es otro comentario
var2 = 7
var3 = True # Tambien puedo comentar aca
"""
Aca van
varias
lineas de
comentarios
"""
```

# Operaciones entre Variables

Operaciones aritméticas entre números (int y float):

Operación	Significado
$a + b$	Suma
$a - b$	Resta
$a * b$	Multiplicación
$a / b$	División
$a ** b$	Exponencial ("a" elevado a la "b")
$a \% b$	Módulo (resto de la división)
$a // b$	División entera

Ejemplos:

$$7 / 3 = 2.33 \quad 7 // 3 = 2 \quad 7 \% 3 = 1$$

# Operaciones entre Variables

## Asignaciones entre números (int y float)

Operación	Equivalente a	Significado
<code>a += b</code>	<code>a = a + b</code>	Asigna a "a" la suma entre "a" y "b"
<code>a -= b</code>	<code>a = a - b</code>	Asigna a "a" la resta entre "a" y "b"

Ejemplo:


```
a = 7
b = 5
a += b # ahora "a" vale 12
a -= 2 # ahora "a" vale 10
```

Este tipo de asignaciones funcionan para todas las operaciones entre números



# Reglas de Prioridad

- Se respetan las reglas de prioridad de una ecuación matemática tradicional:
  - Primero se evalúa lo que está entre paréntesis.
  - Segundo multiplicaciones, divisiones y módulos.
  - Por último, sumas y restas.
- Ejemplo:

$$3 * (5 - 2) / 7 + 12 - 7 * (4 - 1)$$


# Operaciones entre Variables

## Operaciones entre strings:

Operación	Significado
<code>a + b</code>	Concatenación
<code>a[i]</code>	Indexación
<code>a[i:j]</code>	Cortar
<code>a * b</code>	Repetición

## Ejemplos:

```
string1 = "programacion"
string2 = "2022" # Como tiene comillas, es un string
string3 = string1 + string2
inicial = string3[0] # Devuelve "p"
tresletras = string3[2:5] # Devuelve entre las 3er letra y la 6ta
```

# Comparaciones lógicas

- Dan como resultado un booleano

Operación	Resultado
5 > 9	False
5 < 9	True
5 >= 9	False
5 <= 9	True
5 == 9	False
5 != 9	True

# Tipado Dinámico

- En Python, al definir una variable no se debe indicar de qué tipo es.
- Es posible asignarle a una variable un dato de otro tipo.

```
var1 = 7
```

```
var1 = False
```

```
var1 = "tipado dinamico"
```

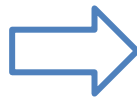
# Corriendo un código

- Existen 2 formas básicas de correr un programa:
  - Modo interactivo
  - *Scripting*
- El modo interactivo consiste en una consola en la que uno va ingresando sentencias.

```
>>> var1 = 123.45
>>> var2 = 345.67
>>> var1 / var2
0.3571325252408366
>>> var1 * var2
42672.961500000005
>>> 
```

- Un *script* es un archivo de texto con sentencias que son ejecutadas en orden por un intérprete

```
script.py
~/Documents
1 var1 = 123.45
2 var2 = 345.67
3 print (var1 / var2)
4 print (var1 * var2)
```



```
(base) matias@rfgenom003:~/Documents$ python script.py
0.3571325252408366
42672.961500000005
```