

Introducción a los Lenguajes de Programación I

Identificadores, variables, tipos

Jonatan Gómez Perdomo, Ph.D.

jgomezpe@unal.edu.co

Arles Rodríguez, Ph.D.

aerodriguezp@unal.edu.co

Camilo Cubides, Ph.D.(c)

eccubidesg@unal.edu.co

Grupo de investigación en vida artificial – Research Group on Artificial Life – (Alife)

Departamento de Ingeniería de Sistemas e Industrial

Facultad de Ingeniería

Universidad Nacional de Colombia



Agenda

1 Identificadores y variables

- Identificadores
- Variables

2 Tipos de datos primitivos

- Enteros
- Reales
- Booleanos
- Caracteres



Agenda

1 Identificadores y variables

- Identificadores
- Variables

2 Tipos de datos primitivos

- Enteros
- Reales
- Booleanos
- Caracteres



Identificadores I

Definición

Un *identificador* es una secuencia de símbolos que se utilizan como nombres de variables, funciones, arreglos, clases y otras estructuras de los lenguajes de programación.

Los identificadores en Python se escriben como secuencias de caracteres alfanuméricos del alfabeto inglés o el guión bajo (*underscore*) (`_`), tales que su primer símbolo no es un dígito. Un identificador no puede tener más de 256 símbolos.



Identificadores II

Un identificar valido debe cumplir con la condición adicional de que no pertenezca a las palabras reservadas para el lenguaje, a continuación se listan las palabras reservadas del lenguaje Python:

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



Identificadores III

Ejemplo

Las siguientes secuencias de caracteres son ejemplos de identificadores validos:

```
i  
x  
suma  
sumando1  
sumando2  
Edad  
paisDeNacimiento  
_nombre  
area_circulo
```



Identificadores IV

Ejemplo

Las siguientes secuencias de caracteres son ejemplos de secuencias que no son identificadores, ¿por qué?:

```
1er_mes  
primer nombre  
while  
p@dre  
día
```



Identificadores V

Nota

Una nota importante para el lenguaje Python es que éste es sensible a mayúsculas y minúsculas, esto quiere decir que por ejemplo los identificadores

dia	Dia	DIA
-----	-----	-----

sirven para declarar entidades (variable, funciones, etc.) que son diferentes, pues al ser la misma palabra, difiere en que algunas letras son mayúsculas en unos identificadores y en los otros no.

Nota

Cabe anotar que existen lenguajes que no son sensibles a mayúsculas y minúsculas, tales como DFD, BASIC, FORTRAN y HTML.

Agenda

1 Identificadores y variables

- Identificadores
- Variables

2 Tipos de datos primitivos

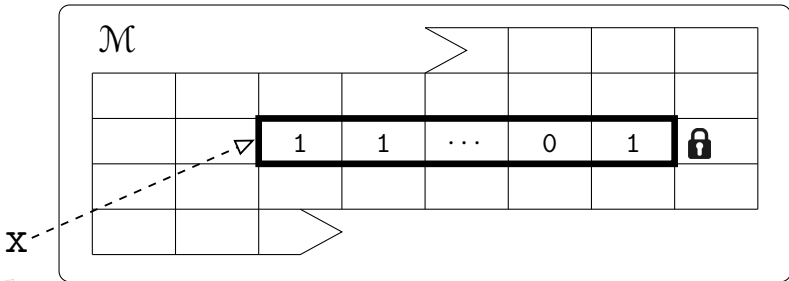
- Enteros
- Reales
- Booleanos
- Caracteres



Variables I

Definición

Una *variable* es un espacio de memoria donde se almacena un dato, un espacio donde se guarda la información necesaria para realizar las acciones que ejecutan los programas.



Variables II

Para declarar una variable se necesitan principalmente dos componentes: el nombre y el tipo de dato (opcional en algunos lenguajes). Los tipos de variables se estudian en la siguiente sección, con respecto al nombre, este simplemente debe ser un identificador valido que no sea una palabra reservada.

En general una variable se declara así

```
x: T
```

Donde T es el tipo de dato o conjunto al que pertenece la variable y x es el identificador que es el nombre de la variable.



Variables III

Una buena práctica de programación¹ es asignarle el nombre a una variable de tal manera que indique por un lado el papel que desempeña dicha variable en el algoritmo y por otro los posibles valores que almacena. Nombres de variables recomendados dependiendo del tipo de problema pueden ser:

velocidad	espacio	masa	aceleracion
exponente		termino1	valor_maximo
area_circulo		nombre_estudiante	last_name



Por buenas o mejores prácticas se entiende un conjunto coherente de acciones que han rendido bien o incluso excelente servicio en un determinado contexto.



Agenda

1 Identificadores y variables

- Identificadores
- Variables

2 Tipos de datos primitivos

- Enteros
- Reales
- Booleanos
- Caracteres



En programación existe un tipo de dato que permite aproximar el conjunto de los números enteros conocido como `int`.

Otro tipo de dato que aproxima el conjunto de los números reales se conoce como `float`.

Otro tipo de dato conocido como `str` es el que sirve para representar las diferentes letras.

Para la representación de los valores de verdad se tendrán los booleanos representados en el tipo de dato `bool`.

Estos tipos de datos son conocidos como primitivos o escalares, pues están definidos en el lenguaje de programación Python y porque de ellos se pueden derivar otros tipos de datos definidos por el programador.



Agenda

1 Identificadores y variables

- Identificadores
- Variables

2 Tipos de datos primitivos

- Enteros
- Reales
- Booleanos
- Caracteres



Enteros I

Los enteros en Python se codifican con la palabra `int` y su declaración es la siguiente

Si x es una variable algebraica que varia en el conjunto \mathbb{Z} , para definir x en el lenguaje Python se utiliza la expresión

```
x: int
```

lo que sirve para declarar que la variable x pertenece a los enteros que son representables en el lenguaje Python.



Enteros II

Literales enteros

Ejemplo

Los literales enteros, es decir, la sintaxis de los valores que pueden ser asignados a las variables de tipo `int` que soporta Python son por ejemplo:

-32768	-0	-1	-127
32768	0	1	127
+32768	+0	+1	+127



Enteros III

Ejemplo

Cuando se declara una variable de tipo entero, no se sabe que valor tiene, por eso es necesario inicializar la variable. Los siguientes son ejemplos de inicializaciones de variables de tipo `int` en Python.

```
i = 0  
j : int = 1  
n = 5  
p : int = -10  
k = -1
```



Agenda

1 Identificadores y variables

- Identificadores
- Variables

2 Tipos de datos primitivos

- Enteros
- Reales
- Booleanos
- Caracteres



Reales I

Los reales en Python se codifican con la palabra `float` y su declaración es la siguiente

Si x es una variable algebraica que varia en el conjunto \mathbb{R} , para definir x en el lenguaje Python se utiliza la expresión

```
x: float
```

lo que sirve para declarar que la variable x pertenece a los reales que son representables en el lenguaje Python.



Reales II

El subconjunto de los números reales que pueden ser representados en el lenguaje Python, es un subconjunto propio de los racionales, que se representan con 64 bits (8 bytes) y que usan un tipo de codificación definida por el *IEEE standard for Binary Floating-Point Arithmetic 754* de 1985, los valores distintos de 0 de este conjunto varían en el rango

$$-1.7976931348623157 \times 10^{+308} \leq x \leq -2.2250738585072014 \times 10^{-308}$$

y

$$2.2250738585072014 \times 10^{-308} \leq x \leq 1.7976931348623157 \times 10^{+308}$$

que dan una precisión científica de 15 dígitos.



Reales III

Densidad y distribución de los números reales de máquina I

Los números reales son densos en ellos mismos, es decir, que dados dos reales distintos siempre se puede encontrar uno distinto que se encuentre entre ellos dos.

Los números reales de máquina no son densos, esto sucede por que los números reales de máquina son finitos y por lo tanto, existen números distintos para los cuales no hay números entre ellos. Además, la mayoría de los números se acumulan alrededor del 0 y hacia los extremos superior e inferior se encuentran más dispersos.

Ejemplo

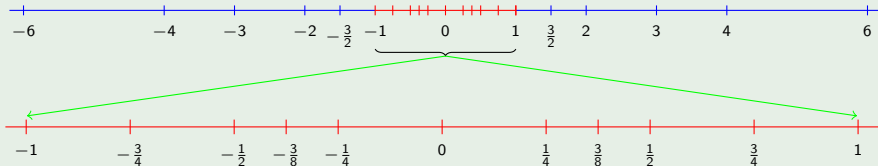
Para una máquina muy sencilla que utiliza una representación en base 2, que utiliza 5 bits; de estos 1 para el signo, 3 para el exponente y 1 para la mantisa. El conjunto de los números de esta máquina son los siguientes:

Reales IV

Densidad y distribución de los números reales de máquina II

Ejemplo (continuación)

$$\left\{ -6, -4, -3, -2, -\frac{3}{2}, -1, -\frac{3}{4}, -\frac{1}{2}, -\frac{3}{8}, -\frac{1}{4}, 0, \frac{1}{4}, \frac{3}{8}, \frac{1}{2}, \frac{3}{4}, 1, \frac{3}{2}, 2, 3, 4, 6 \right\}$$



Obsérvese como alrededor del origen se acumulan muchos números y hacia los extremos se encuentran más dispersos.

Reales V

Literales reales

Ejemplo

Los literales reales, es decir, la sintaxis de los valores que pueden ser asignados a las variables de tipo `float` que soporta Python son por ejemplo:

-3.14159265	-0.0	-6.02214129E+23	-6.674287e-11
3.14159265	0.0	6.02214129E23	6.674287E-11
+3.14159265	+0.0	+6.02214129e+23	+6.674287E-11



Reales VI

Ejemplo

Cuando se declara una variable de tipo real, no se sabe que valor tiene, por eso es necesario inicializar la variable. Los siguientes son ejemplos de inicializaciones de variables de tipo float

```
e = 2.7182818284
gamma: float = 0.577215664901
phi = 1.61803398874989
a: float = +1.0
X = -1.0
coordenada_1: float = -2.5
const_Boltzmann = 1.3806488E-23
Luz: float = 2.998e+8
Avogadro = +6.02214129e+23
G: float = 6.67384e-11
Plank = 6.62606896E-34
```

Agenda

1 Identificadores y variables

- Identificadores
- Variables

2 Tipos de datos primitivos

- Enteros
- Reales
- Booleanos
- Caracteres



Booleanos I

Los booleanos en Python se codifican con la palabra `bool` y su declaración es la siguiente

Si x es una variable algebraica que varia en el conjunto \mathbb{B} , para definir x en el lenguaje Python se utiliza la expresión

```
x: bool
```

lo que sirve para declarar que la variable x pertenece al conjunto de los booleanos ($\mathbb{B} = \{V, F\}$).



Booleanos II

Literales booleanos

Como sólo hay dos valores de verdad V y F , en Python sólo hay dos literales para representar los valores lógicos, estos son:

True	False
------	-------

donde la cadena True representa el valor de verdad V y la cadena False representa el valor de verdad F .



Booleanos III

Ejemplo

Cuando se declara una variable de tipo booleano, no se sabe que valor tiene, por eso es necesario inicializar la variable. Los siguientes son ejemplos de inicializaciones de variables de tipo bool

```
b = True  
flag: bool = True  
exp = False  
isPrime: bool = False
```



Booleanos IV

En Python la representación interna de los valores booleanos se hace mediante el uso de valores enteros `int` de 32 bits de la siguiente manera: el valor lógico F se representa con el entero 0 (cero) y el valor lógico V se representa con cualquier entero distinto de 0, por costumbre se usa el entero 1 (uno). De lo anterior se obtienen las siguientes equivalencias lógicas:

`True` \Leftrightarrow 1

`False` \Leftrightarrow 0



Agenda

1 Identificadores y variables

- Identificadores
- Variables

2 Tipos de datos primitivos

- Enteros
- Reales
- Booleanos
- Caracteres



Caracteres I

Los caracteres representan los símbolos definidos por el ASCII (*American Standard Code for Information Interchange*). Los caracteres se representan con 8 bits (1 byte), lo que ofrece 256 símbolos distintos. El conjunto ASCII cumple con la siguiente característica

$$\text{ASCII} \supseteq \left\{ \begin{array}{l} !, ", \#, \$, \%, \&, ', (,), *, +, ,, -, ., /, \\ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \\ :, ;, <, =, >, ?, @, \\ A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z, \\ [, \backslash,], \wedge, _ , ' , \\ a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z, \\ \{, |, \}, \sim \end{array} \right\}$$



Caracteres II

Tabla de códigos ASCII 0-15

A continuación se presenta una serie de tablas con todos símbolos ASCII.

DEC	Símbolo	Descripción
0	NUL	Null char
1	SOH	Start of Heading
2	STX	Start of Text
3	ETX	End of Text
4	EOT	End of Transmission
5	ENQ	Enquiry
6	ACK	Acknowledgment
7	BEL	Bell

DEC	Símbolo	Descripción
8	BS	Back Space
9	HT	Horizontal Tab
10	LF	Line Feed
11	VT	Vertical Tab
12	FF	Form Feed
13	CR	Carriage Return
14	SO	Shift Out/X-On
15	SI	Shift In/X-Off



Caracteres III

Tabla de códigos ASCII 16-31

DEC	Símbolo	Descripción
16	DLE	Data Line Escape
17	DC1	Device Control 1 (oft. XON)
18	DC2	Device Control 2
19	DC3	Device Control 3 (oft. XOFF)
20	DC4	Device Control 4
21	NAK	Negative Acknowledgement
22	SYN	Synchronous Idle
23	ETB	End of Transmit Block
24	CAN	Cancel
25	EM	End of Medium
26	SUB	Substitute
27	ESC	Escape
28	FS	File Separator
29	GS	Group Separator
30	RS	Record Separator
31	US	Unit Separator



Caracteres IV

Tabla de códigos ASCII 32-47

DEC	Símbolo	Descripción
32	[SPACE]	Space
33	!	Exclamation mark
34	"	Double quotes (or speech marks)
35	#	Number
36	\$	Dollar
37	%	Procenttecken
38	&	Ampersand
39	'	Single quote
40	(Open parenthesis (or open bracket)
41)	Close parenthesis (or close bracket)
42	*	Asterisk
43	+	Plus
44	,	Comma
45	-	Hyphen
46	.	Period, dot or full stop
47	/	Slash or divide



Caracteres V

Tabla de códigos ASCII 48-63

DEC	Símbolo	Descripción
48	0	Zero
49	1	One
50	2	Two
51	3	Three
52	4	Four
53	5	Five
54	6	Six
55	7	Seven
56	8	Eight
57	9	Nine
58	:	Colon
59	;	Semicolon
60	<	Less than (or open angled bracket)
61	=	Equals
62	>	Greater than (or close angled bracket)
63	?	Question mark



Caracteres VI

Tabla de códigos ASCII 64-95

DEC	Símb.	Descripción
64	@	At symbol
65	A	Uppercase A
66	B	Uppercase B
67	C	Uppercase C
68	D	Uppercase D
69	E	Uppercase E
70	F	Uppercase F
71	G	Uppercase G
72	H	Uppercase H
73	I	Uppercase I
74	J	Uppercase J
75	K	Uppercase K
76	L	Uppercase L
77	M	Uppercase M
78	N	Uppercase N
79	O	Uppercase O

DEC	Símb.	Descripción
80	P	Uppercase P
81	Q	Uppercase Q
82	R	Uppercase R
83	S	Uppercase S
84	T	Uppercase T
85	U	Uppercase U
86	V	Uppercase V
87	W	Uppercase W
88	X	Uppercase X
89	Y	Uppercase Y
90	Z	Uppercase Z
91	[Opening bracket
92	\	Backslash
93]	Closing bracket
94	^	Caret-circumflex
95	_	Underscore



Caracteres VII

Tabla de códigos ASCII 96-127

DEC	Símb.	Descripción
96	`	Grave accent
97	a	Lowercase a
98	b	Lowercase b
99	c	Lowercase c
100	d	Lowercase d
101	e	Lowercase e
102	f	Lowercase f
103	g	Lowercase g
104	h	Lowercase h
105	i	Lowercase i
106	j	Lowercase j
107	k	Lowercase k
108	l	Lowercase l
109	m	Lowercase m
110	n	Lowercase n
111	o	Lowercase o

DEC	Símb.	Descripción
112	p	Lowercase p
113	q	Lowercase q
114	r	Lowercase r
115	s	Lowercase s
116	t	Lowercase t
117	u	Lowercase u
118	v	Lowercase v
119	w	Lowercase w
120	x	Lowercase x
121	y	Lowercase y
122	z	Lowercase z
123	{	Opening brace
124		Vertical bar
125	}	Closing brace
126	~	Equivalency sign - tilde
127	[DEL]	Delete



Caracteres VIII

Tabla de códigos ASCII 128-143

DEC	Símbolo	Descripción
128	€	Euro sign
129		
130	,	Single low-9 quotation mark
131	f	Latin small letter f with hook
132	„	Double low-9 quotation mark
133	...	Horizontal ellipsis
134	†	Dagger
135	‡	Double dagger
136	^	Modifier letter circumflex accent
137	‰	Per mille sign
138	Š	Latin capital letter S with caron
139	<	Single left-pointing angle quotation
140	Œ	Latin capital ligature OE
141		
142	Ž	Latin captial letter Z with caron
143		



Caracteres IX

Tabla de códigos ASCII 144-159

DEC	Símbolo	Descripción
144		
145	`	Left single quotation mark
146	'	Right single quotation mark
147	``	Left double quotation mark
148	''	Right double quotation mark
149	•	Bullet
150	–	En dash
151	—	Em dash
152	~	Small tilde
153	™	Trade mark sign
154	š	Latin small letter S with caron
155	>	Single right-pointing angle quotation mark
156	œ	Latin small ligature oe
157		
158	ž	Latin small letter z with caron
159	Ÿ	Latin capital letter Y with diaeresis



Caracteres X

Tabla de códigos ASCII 160-175

DEC	Símbolo	Descripción
160		Non-breaking space
161	¡	Inverted exclamation mark
162	¢	Cent sign
163	£	Pound sign
164	¤	Currency sign
165	¥	Yen sign
166		Pipe, Broken vertical bar
167	§	Section sign
168	¨	Spacing diaeresis - umlaut
169	©	Copyright sign
170	ª	Feminine ordinal indicator
171	«	Left double angle quotes
172	¬	Not sign
173	-	Soft hyphen
174	®	Registered trade mark sign
175	—	Spacing macron - overline



Caracteres XI

Tabla de códigos ASCII 176-191

DEC	Símbolo	Descripción
176	°	Degree sign
177	±	Plus-or-minus sign
178	²	Superscript two - squared
179	³	Superscript three - cubed
180	´	Acute accent - spacing acute
181	μ	Micro sign
182	¶	Pilcrow sign - paragraph sign
183	·	Middle dot - Georgian comma
184	¸	Spacing cedilla
185	¹	Superscript one
186	º	Masculine ordinal indicator
187	»	Right double angle quotes
188	¹ / ₄	Fraction one quarter
189	¹ / ₂	Fraction one half
190	³ / ₄	Fraction three quarters
191	¿	Inverted question mark



Caracteres XII

Tabla de códigos ASCII 192-207

DEC	Símbolo	Descripción
192	À	Latin capital letter A with grave
193	Á	Latin capital letter A with acute
194	Â	Latin capital letter A with circumflex
195	Ã	Latin capital letter A with tilde
196	Ä	Latin capital letter A with diaeresis
197	Å	Latin capital letter A with ring above
198	Æ	Latin capital letter AE
199	Ç	Latin capital letter C with cedilla
200	È	Latin capital letter E with grave
201	É	Latin capital letter E with acute
202	Ê	Latin capital letter E with circumflex
203	Ë	Latin capital letter E with diaeresis
204	Ì	Latin capital letter I with grave
205	Í	Latin capital letter I with acute
206	Î	Latin capital letter I with circumflex
207	Ï	Latin capital letter I with diaeresis



Caracteres XIII

Tabla de códigos ASCII 208-223

DEC	Símbolo	Descripción
208	Ð	Latin capital letter ETH
209	Ñ	Latin capital letter N with tilde
210	Ò	Latin capital letter O with grave
211	Ó	Latin capital letter O with acute
212	Ô	Latin capital letter O with circumflex
213	Õ	Latin capital letter O with tilde
214	Ö	Latin capital letter O with diaeresis
215	×	Multiplication sign
216	Ø	Latin capital letter O with slash
217	Ù	Latin capital letter U with grave
218	Ú	Latin capital letter U with acute
219	Û	Latin capital letter U with circumflex
220	Ü	Latin capital letter U with diaeresis
221	Ý	Latin capital letter Y with acute
222	Þ	Latin capital letter THORN
223	ß	Latin small letter sharp s - ess-zed



Caracteres XIV

Tabla de códigos ASCII 224-239

DEC	Símbolo	Descripción
224	à	Latin small letter a with grave
225	á	Latin small letter a with acute
226	â	Latin small letter a with circumflex
227	ã	Latin small letter a with tilde
228	ä	Latin small letter a with diaeresis
229	å	Latin small letter a with ring above
230	æ	Latin small letter ae
231	ç	Latin small letter c with cedilla
232	è	Latin small letter e with grave
233	é	Latin small letter e with acute
234	ê	Latin small letter e with circumflex
235	ë	Latin small letter e with diaeresis
236	ì	Latin small letter i with grave
237	í	Latin small letter i with acute
238	î	Latin small letter i with circumflex
239	ï	Latin small letter i with diaeresis



Caracteres XV

Tabla de códigos ASCII 240-255

DEC	Símbolo	Descripción
240	ð	Latin small letter eth
241	ñ	Latin small letter n with tilde
242	ò	Latin small letter o with grave
243	ó	Latin small letter o with acute
244	ô	Latin small letter o with circumflex
245	õ	Latin small letter o with tilde
246	ö	Latin small letter o with diaeresis
247	÷	Division sign
248	ø	Latin small letter o with slash
249	ù	Latin small letter u with grave
250	ú	Latin small letter u with acute
251	û	Latin small letter u with circumflex
252	ü	Latin small letter u with diaeresis
253	ý	Latin small letter y with acute
254	þ	Latin small letter thorn
255	ÿ	Latin small letter y with diaeresis



Caracteres XVI

Los `ASCII` en Python se codifican con la palabra `str` y su declaración es la siguiente

Si x es una variable algebraica que varia en el conjunto `ASCII`, para definir x en el lenguaje Python se utiliza la expresión

```
x: str
```

lo que sirve para declarar que la variable x pertenece al conjunto de los `ASCII`. Los caracteres son cadenas de longitud 1.



Caracteres XVII

Existen algunos caracteres especiales que no tiene su propio símbolo en el teclado o que no se imprime el símbolo en la pantalla o que tienen un uso particular en Python (son reservados) y que son utilizados comúnmente; éstos se representan de la siguiente manera, usando el símbolo auxiliar \ (*back slash*):

`\n` : Nueva línea.

`\t` : Tabulador horizontal.

`\\` : Diagonal invertida (*back slash*).

`\'` : Imprime apóstrofo.

`\"` : Imprime Comillas.

`\b` : Retroceso (retrocede un espacio el cursor).

`\v` : Tabulador vertical (coloca el cursor justo debajo del último carácter de la línea actual).

`\r` : Retorno de carro (coloca el cursor en el primer carácter de la línea actual y sobrescribe el texto de la línea).

`\?` : Imprime el símbolo de interrogación.



Caracteres XVIII

Literales carácter

Ejemplo

Cuando se declara una variable de tipo cadena, no se sabe que valor tiene, por eso es necesario inicializar la variable. Los siguientes son ejemplos de inicializaciones de variables de tipo `str`, para indicar que se está definiendo un literal de carácter (de un sólo símbolo), se encierra el símbolo entre apóstrofes `' '` o entre comillas dobles `" "`, así como se muestra a continuación:

```
c = " "  
CH : str = "\n"  
tab = "\t"  
letra : str = "a"  
caracter = "A"
```

```
value = '\ "  
_last : str = '\ "'  
C_0 = '&'  
cero : str = '0'  
at = '@'
```