

Introducción al Lenguaje de Programación Java

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

- Tipos numéricos enteros
- Tipos numéricos de punto flotante
- Booleanos
- Caracteres

3 Lectura de datos



Agenda

1 Identificadores y variables

- Identificadores
- Variables

2 Tipos de datos primitivos

- Tipos numéricos enteros
- Tipos numéricos de punto flotante
- Booleanos
- Caracteres

3 Lectura de datos



Introducción

En el mundo de los lenguajes de programación, existen principalmente dos tipos de lenguajes:

- Lenguajes de **tipado estático**, donde el tipo de cada variable y expresión deben definirse de forma explícita, es decir, en el momento en que se declara una variable, se debe indicar a que tipo de dato pertenece. Ejemplo: C, C++, Java.
- Lenguajes de **tipado dinámico**, en estos lenguajes, las variables pueden recibir diferentes tipos de datos a lo largo del tiempo y no es necesario especificar un tipo de dato para las variables. Ejemplo: Ruby, Python.



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 Java se escriben como secuencias de caracteres alfanuméricos del alfabeto inglés o el guión bajo (*underscore*) (`_`), tales que su primer símbolo no es un dígito. Aunque en Java no hay límite en la longitud del identificador, es aconsejable que tenga de 4 a 15 caracteres.



Identificadores II

Un identificador válido 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 Java:

```
abstract  assert  boolean  break  byte  case
catch  char  class  const  continue  default
do  double  else  enum  extends  false  final
finally  float  for  goto  if  implements
import  instanceof  int  interface  long  native
new  null  package  private  protected  public
return  short  static  strictfp  String  super
switch  synchronized  this  throw  throws
transient  true  try  void  volatile  while
```



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 Java 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

- Tipos numéricos enteros
- Tipos numéricos de punto flotante
- Booleanos
- Caracteres

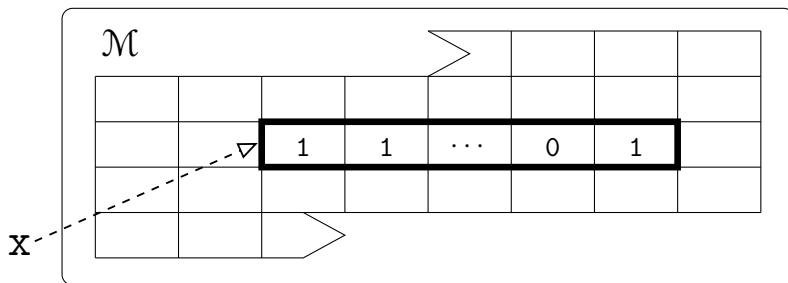
3 Lectura de datos



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

```
T x;
```

Donde \mathbb{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

¹Por 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

- Tipos numéricos enteros
- Tipos numéricos de punto flotante
- Booleanos
- Caracteres

3 Lectura de datos



En programación existe un cierto tipo de datos destinados a gestionar información básica, como números, caracteres y datos de tipo verdadero/falso. Estos tipos de datos son conocidos como primitivos o escalares, pues están definidos en el lenguaje de programación Java y porque de ellos se pueden derivar otros tipos de datos definidos por el programador.

Java soporta 8 tipos de datos primitivos, de estos tipos primitivos, seis de ellos están destinados a facilitar el trabajo con números. Podemos agrupar los datos de tipo numérico en dos categorías:

- **Tipos numéricos enteros:** Este tipo de datos permite operar exclusivamente con números enteros, sin parte decimal.
- **Tipos numéricos de punto flotante:** Además de números enteros, este tipo de datos permite operar también con números racionales.



Agenda

1 Identificadores y variables

- Identificadores
- Variables

2 Tipos de datos primitivos

- Tipos numéricos enteros
- Tipos numéricos de punto flotante
- Booleanos
- Caracteres

3 Lectura de datos



Tipos numéricos enteros I

En Java existen cuatro tipos destinados a almacenar números enteros. La única diferencia entre ellos es el número de bytes usados para su almacenamiento y en consecuencia, el rango de valores que es posible representar con ellos. Todos ellos emplean una representación que permite el almacenamiento de números negativos y positivos.

- **byte:** como lo indica su nombre, emplea un solo byte (8 bits) de almacenamiento. Ésto le permite almacenar valores en el rango $[-128, 127]$.
- **short:** usa el doble de almacenamiento que el tipo byte, lo cual hace posible representar cualquier valor en el rango $[-32.768, 32.767]$.



Tipos numéricos enteros II

- **int:** emplea 4 bytes de almacenamiento y es el tipo de dato entero más usado. El rango de valores que puede representar va de -2^{31} a $2^{31}-1$. O lo que es lo mismo, valores en el rango $[-2.147.483.648, 2.147.483.647]$.
- **long:** es el tipo entero de mayor tamaño ya que usa 8 bytes (64 bits), cuenta con un rango de valores desde -2^{63} a $2^{63}-1$. Es decir, valores en el rango $[-9.223.372.036.854.775.808, 9.223.372.036.854.775.807]$.



Tipos numéricos enteros III

Literales enteros

Ejemplo

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

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



Tipos numéricos enteros IV

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 entero en Java.

```
int i = 0;  
byte j = 1;  
long n = 587987987;  
short t = -10;  
int p = -1;
```



Agenda

1 Identificadores y variables

- Identificadores
- Variables

2 Tipos de datos primitivos

- Tipos numéricos enteros
- Tipos numéricos de punto flotante
- Booleanos
- Caracteres

3 Lectura de datos



Tipos numéricos de punto flotante I

Los tipos numéricos de punto flotante permiten representar números tanto muy grandes como muy pequeños además de números decimales (reales). Java dispone de 2 tipos concretos en esta categoría:

- **float:** conocido como tipo de precisión simple, emplea un total de 32 bits. Con este tipo de datos es posible representar números en el rango de 1.4×10^{-45} a 3.4028235×10^{38} .
- **double:** sigue un esquema de almacenamiento similar al anterior, pero usando 64 bits en lugar de 32. Esto le permite representar valores en el rango de 4.9×10^{-324} a $1.7976931348623157 \times 10^{308}$.



Tipos numéricos de punto flotante II

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:

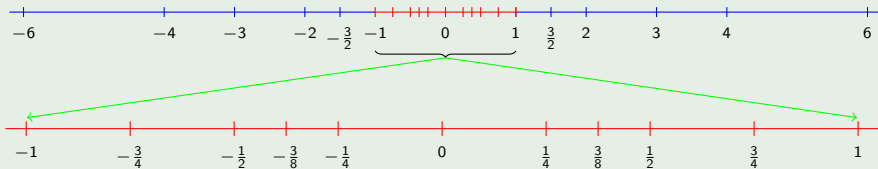


Tipos numéricos de punto flotante III

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.

Tipos numéricos de punto flotante IV

Literales reales

Ejemplo

Los literales reales, es decir, la sintaxis de los valores que pueden ser asignados a las variables de tipo real que soporta Java 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



Tipos numéricos de punto flotante V

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 real

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

Agenda

1 Identificadores y variables

- Identificadores
- Variables

2 Tipos de datos primitivos

- Tipos numéricos enteros
- Tipos numéricos de punto flotante
- Booleanos
- Caracteres

3 Lectura de datos



Booleanos I

Los booleanos en Java se codifican con la palabra `boolean` 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 Java se utiliza la expresión

```
boolean x;
```

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 Java sólo hay dos literales para representar los valores lógicos, estos son:

<code>true</code>	<code>false</code>
-------------------	--------------------

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 boolean

```
boolean b = true;  
boolean flag = true;  
boolean exp = false;  
boolean isPrime = false;
```



Agenda

1 Identificadores y variables

- Identificadores
- Variables

2 Tipos de datos primitivos

- Tipos numéricos enteros
- Tipos numéricos de punto flotante
- Booleanos
- Caracteres

3 Lectura de datos



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

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



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	Hex	Unicode 	Char 	Description
128	80	U+00C7	Ç	latin capital letter c with cedilla
129	81	U+00FC	ü	latin small letter u with diaeresis
130	82	U+00E9	é	latin small letter e with acute
131	83	U+00E2	â	latin small letter a with circumflex
132	84	U+00E4	ä	latin small letter a with diaeresis
133	85	U+00E0	à	latin small letter a with grave
134	86	U+00E5	å	latin small letter a with ring above
135	87	U+00E7	ç	latin small letter c with cedilla
136	88	U+00EA	ê	latin small letter e with circumflex
137	89	U+00EB	ë	latin small letter e with diaeresis
138	8A	U+00E8	è	latin small letter e with grave
139	8B	U+00EF	ï	latin small letter i with diaeresis
140	8C	U+00EE	î	latin small letter i with circumflex
141	8D	U+00EC	ì	latin small letter i with grave
142	8E	U+00C4	Ä	latin capital letter a with diaeresis
143	8F	U+00C5	Å	latin capital letter a with ring above



Caracteres IX



Tabla de códigos ASCII 144-159

Dec	Hex	Unicode 	Char 	Description
144	90	U+00C9	É	latin capital letter e with acute
145	91	U+00E6	æ	latin small ligature ae
146	92	U+00C6	Æ	latin capital ligature ae
147	93	U+00F4	ô	latin small letter o with circumflex
148	94	U+00F6	ö	latin small letter o with diaeresis
149	95	U+00F2	ò	latin small letter o with grave
150	96	U+00FB	û	latin small letter u with circumflex
151	97	U+00F9	ù	latin small letter u with grave
152	98	U+00FF	ÿ	latin small letter y with diaeresis
153	99	U+00D6	Ö	latin capital letter o with diaeresis
154	9A	U+00DC	Ü	latin capital letter u with diaeresis
155	9B	U+00A2	¢	cent sign
156	9C	U+00A3	£	pound sign
157	9D	U+00A5	¥	yen sign
158	9E	U+20A7	₧	peseta sign
159	9F	U+0192	ƒ	latin small letter f with hook



Caracteres X

Tabla de códigos ASCII 160-175

Dec	Hex	Unicode 	Char 	Description
160	A0	U+00E1	á	latin small letter a with acute
161	A1	U+00ED	í	latin small letter i with acute
162	A2	U+00F3	ó	latin small letter o with acute
163	A3	U+00FA	ú	latin small letter u with acute
164	A4	U+00F1	ñ	latin small letter n with tilde
165	A5	U+00D1	Ñ	latin capital letter n with tilde
166	A6	U+00AA	a	feminine ordinal indicator
167	A7	U+00BA	o	masculine ordinal indicator
168	A8	U+00BF	¿	inverted question mark
169	A9	U+2310	⌞	reversed not sign
170	AA	U+00AC	¬	not sign
171	AB	U+00BD	½	vulgar fraction one half
172	AC	U+00BC	¼	vulgar fraction one quarter
173	AD	U+00A1	¡	inverted exclamation mark
174	AE	U+00AB	«	left-pointing double angle quotation mark
175	AF	U+00BB	»	right-pointing double angle quotation mark



Caracteres XI



Tabla de códigos ASCII 176-191

Dec	Hex	Unicode	Char	Description
176	B0	U+2591	░	light shade
177	B1	U+2592	▒	medium shade
178	B2	U+2593	▓	dark shade
179	B3	U+2502		box drawings light vertical
180	B4	U+2524	├	box drawings light vertical and left
181	B5	U+2561	┤	box drawings vertical single and left double
182	B6	U+2562	├┤	box drawings vertical double and left single
183	B7	U+2556	└	box drawings down double and left single
184	B8	U+2555	┘	box drawings down single and left double
185	B9	U+2563	├┘	box drawings double vertical and left
186	BA	U+2551	└└	box drawings double vertical
187	BB	U+2557	┘└	box drawings double down and left
188	BC	U+255D	└┘	box drawings double up and left
189	BD	U+255C	┘└	box drawings up double and left single
190	BE	U+255B	└┘	box drawings up single and left double
191	BF	U+2510	┐	box drawings light down and left



Caracteres XII



Tabla de códigos ASCII 192-207

Dec	Hex	Unicode 	Char 	Description
192	C0	U+2514	┐	box drawings light up and right
193	C1	U+2534	└	box drawings light up and horizontal
194	C2	U+252C	┘	box drawings light down and horizontal
195	C3	U+251C	┌	box drawings light vertical and right
196	C4	U+2500	—	box drawings light horizontal
197	C5	U+253C	┐	box drawings light vertical and horizontal
198	C6	U+255E	└	box drawings vertical single and right double
199	C7	U+255F	┘	box drawings vertical double and right single
200	C8	U+255A	┐	box drawings double up and right
201	C9	U+2554	└	box drawings double down and right
202	CA	U+2569	┘	box drawings double up and horizontal
203	CB	U+2566	┐	box drawings double down and horizontal
204	CC	U+2560	┌	box drawings double vertical and right
205	CD	U+2550	=	box drawings double horizontal
206	CE	U+256C	┐	box drawings double vertical and horizontal
207	CF	U+2567	└	box drawings up single and horizontal double



Caracteres XIII



Tabla de códigos ASCII 208-223

Dec	Hex	Unicode 	Char 	Description
208	D0	U+2568	⌏	box drawings up double and horizontal single
209	D1	U+2564	⌏	box drawings down single and horizontal double
210	D2	U+2565	⌏	box drawings down double and horizontal single
211	D3	U+2559	⌏	box drawings up double and right single
212	D4	U+2558	⌏	box drawings up single and right double
213	D5	U+2552	⌏	box drawings down single and right double
214	D6	U+2553	⌏	box drawings down double and right single
215	D7	U+256B	⌏	box drawings vertical double and horizontal single
216	D8	U+256A	⌏	box drawings vertical single and horizontal double
217	D9	U+2518	└	box drawings light up and left
218	DA	U+250C	┐	box drawings light down and right
219	DB	U+2588	■	full block
220	DC	U+2584	▀	lower half block
221	DD	U+258C	▄	left half block
222	DE	U+2590	▬	right half block
223	DF	U+2580	▀	upper half block



Caracteres XIV



Tabla de códigos ASCII 224-239

Dec	Hex	Unicode 	Char 	Description
224	E0	U+03B1	α	greek small letter alpha
225	E1	U+00DF	ß	latin small letter sharp s
226	E2	U+0393	Γ	greek capital letter gamma
227	E3	U+03C0	π	greek small letter pi
228	E4	U+03A3	Σ	greek capital letter sigma
229	E5	U+03C3	σ	greek small letter sigma
230	E6	U+00B5	μ	micro sign
231	E7	U+03C4	τ	greek small letter tau
232	E8	U+03A6	Φ	greek capital letter phi
233	E9	U+0398	Θ	greek capital letter theta
234	EA	U+03A9	Ω	greek capital letter omega
235	EB	U+03B4	δ	greek small letter delta
236	EC	U+221E	∞	infinity
237	ED	U+03C6	φ	greek small letter phi
238	EE	U+03B5	ε	greek small letter epsilon
239	EF	U+2229	∩	intersection



Caracteres XV

Tabla de códigos ASCII 240-255

Dec	Hex	Unicode 	Char 	Description
240	F0	U+2261	≡	identical to
241	F1	U+00B1	±	plus-minus sign
242	F2	U+2265	≥	greater-than or equal to
243	F3	U+2264	≤	less-than or equal to
244	F4	U+2320	∫	top half integral
245	F5	U+2321	∫	bottom half integral
246	F6	U+00F7	÷	division sign
247	F7	U+2248	≈	almost equal to
248	F8	U+00B0	°	degree sign
249	F9	U+2219	•	bullet operator
250	FA	U+00B7	•	middle dot
251	FB	U+221A	√	square root
252	FC	U+207F	ⁿ	superscript latin small letter n
253	FD	U+00B2	²	superscript two
254	FE	U+25A0	■	black square
255	FF	U+00A0		no-break space



Caracteres XVI

Los `ASCII` en Java se codifican con la palabra `char` y su declaración es la siguiente

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

```
char x;
```

lo que sirve para declarar que la variable x pertenece al conjunto de los `ASCII`. Una variable `char` solo puede almacenar un caracter a la vez.



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 Java (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 carácter, no se sabe que valor tiene, por eso es necesario inicializar la variable. Los siguientes son ejemplos de inicializaciones de variables de tipo char:

```
char c = ' ';  
char new_line = '\n';  
char tab = '\t';  
char letra = 'a';  
char character = 'A';
```

```
char value = 70;  
char nine = '9';  
char htab = 9;  
char cero = '0';  
char at = '@';
```



Agenda

1 Identificadores y variables

- Identificadores
- Variables

2 Tipos de datos primitivos

- Tipos numéricos enteros
- Tipos numéricos de punto flotante
- Booleanos
- Caracteres

3 Lectura de datos



Lectura de datos I

La mayoría de veces los programas deben interactuar con el usuario, es decir, el usuario debe proporcionar información (nombres, teléfonos, direcciones...etc), la cual debe almacenarse en variables con el fin de poder usar éstos datos en el programa. Para realizar estas operaciones de lectura de variables Java cuenta con la clase Scanner, la cual permite leer todos los tipos de datos primitivos. Para utilizar Scanner, en la parte superior del código se debe agregar lo siguiente:

```
import java.util.Scanner;
```



Lectura de datos II

leer datos con Scanner I

Para poder leer datos es necesario crear una instancia del objeto Scanner, tal y como se muestra a continuacion:

```
Scanner sc = new Scanner(System.in);
```

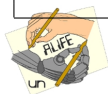
Dependiendo del tipo de dato que se vaya a leer se usan diferentes formas de lectura. Para variables numéricas:

Tipo de Dato

byte
short
int
long
float
double

Forma de lectura

```
byte b = Byte.parseByte(sc.nextLine());  
short s = Short.parseShort(sc.nextLine());  
int i = Integer.parseInt(sc.nextLine())  
long l = Long.parseLong(sc.nextLong());  
float f = Float.parseFloat(sc.nextLine());  
double d = Double.parseDouble(sc.nextLine());
```



Lectura de datos III

leer datos con Scanner II

Para variables booleanas y de tipo char:

Tipo de Dato

boolean
char

Forma de lectura

```
boolean b = Boolean.parseBoolean(sc.nextLine());  
char c = sc.nextLine().charAt(0);
```

Nota

Es necesario definir con precision el tipo de dato que se leerá, ya que si la forma de lectura usada no coincide con el tipo de dato ingresado, se originaran errores o comportamientos no deseados en el programa.



Lectura de datos IV

Ejemplo

```
// Se importa el paquete donde se encuentra Scanner
import java.util.Scanner;

// instancia del objeto Scanner
Scanner sc = new Scanner (System.in);

// leer datos de tipo byte
byte b = Byte.parseByte(sc.nextLine());

// leer datos de tipo short
short s = Short.parseShort(sc.nextLine());

// leer datos de tipo int
int i = Integer.parseInt(sc.nextLine());
```

Lectura de datos IV

Ejemplo

```
// leer datos de tipo long
long l = Long.parseLong(sc.nextLine());

// leer datos de tipo float
float f = Float.parseFloat(sc.nextLine());

// leer datos de tipo double
double d = Double.parseDouble(sc.nextLine());

// leer datos de tipo boolean
boolean b = Boolean.parseBoolean(sc.nextLine());

// leer datos de tipo char
char c = sc.nextLine().charAt(0);
```