Appendix D Unicode Character Set

Fundamentally, computers simply store and manipulate numbers. Text can be processed because each character is assigned a number. The computer manipulates numbers but prints them as characters.

This appendix is about the assignment of characters to their numerical equivalents.

Understanding Encoding

Many children have played some sort of spy game that involved encoding secret messages. The encoding is usually something like this:

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
J	H	Q	S	I	A	R	Z	X	B	D	N	C	O	W	M	T	P	F	U	Y	G	K	L	V	E

The message "GO TO THE HIDEOUT" is encoded by looking up "G" in the top row and writing down "R", the letter beneath it; then looking up "O" and writing down "W"; and so on. The entire encoded message would be "RW UW UZI ZXSIWYU". Someone receiving the coded message could perform the reverse operation to recover the original message.

The computer uses a similar encoding, except it matches letters with numbers:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	...
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	...

When we type "GO TO THE HIDEOUT" into a program, the computer encodes it as 71 79 32 84 79 32 84 72 69 32 72 73 68 69 79 85 84. The spaces in the original message are encoded as 32. When it is time to print a message using, for example, `System.out.println`, the computer looks up the number 71 to discover it should display dots in the shape of "G".

Encoding Characters

A simple program that reads a line of text and displays the corresponding numeric encoding is shown in Listing D-1.

FIND THE CODE

appendices/charCodes/

Listing D-1: *A program to translate a line of text into the equivalent numeric codes*

```
1  import java.util.Scanner;
2
3  /** Translate characters into their integer equivalents.
4  *
5  * @author Byron Weber Becker */
6  public class CharacterCodes extends Object
7  {
8    public static void main(String[] args)
9    {
10     System.out.println("Type a line of text to show the Unicode encoding.");
11     System.out.println("Type \"quit\" to end.");
12
13     Scanner in = new Scanner(System.in);
14     while (true)
15     { System.out.print("> ");
16       String input = in.nextLine();
17
18       if (input.equals("quit"))
19       { break;
20       }
21
22       for (int i = 0; i < input.length(); i++)
23       { char asChar = input.charAt(i);
24         int asInt = input.charAt(i);
25         System.out.println("" + asChar + " (" + asInt + ")");
26       }
27     }
28
29   }
30 }
```

The most common encodings correspond to the ASCII character set, one of the earliest standards. They represent the character encodings from the number 0 up to 127 and are shown in Table D-1.

(table D-1)

ASCII character set

decimal	char	decimal	char	decimal	char	decimal	char
0	NUL	32	Space	64	@	96	`
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	'	71	G	103	g
8	BS	40	(	72	H	104	h
9	TAB	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	ETB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	\	124	|
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	~
31	US	63	?	95	_	127	DEL

The first column of the table contains **control characters**. One of the main uses for these characters is to control some types of printers. If the character CR (carriage return) was sent to the printer, the print head would return to the beginning of the line. The LF character (line feed) moves the paper up one line.

Some of the control characters are still used and have escape sequences so they can be easily inserted into a string. These are shown in Table D-2.

Escape Sequence	Description	Escape Sequence	Description
\n	newline (LF)	\r	return (CR)
\b	backspace (BS)	\\	backslash
\f	form feed (FF)	\'	single quote
\t	tab (TAB)	\"	double quote

(table D-2)

Escape sequences for selected control characters

The last three exist so that we can insert the backslash, single quote, and double quote into strings. For example, if you really did want to print a backslash followed by the character n, you couldn't simply write:

```
System.out.println("\n");
```

because that would print a newline character. Instead, you would need to write

```
System.out.println("\\n");
```

The \\ is interpreted as a single backslash character. The n is considered as just the letter n.