

Encoding Reference

CSCI 40: Strings and Files

Use this sheet for practice and open-note quizzes. Look up the values; you do not need to memorize them or convert number bases by hand.

Reading a lookup. For `H`, the decimal code is 72 and the hexadecimal code is 48. So `ord('H')` returns 72, `chr(72)` returns 'H', and `'\x48'` is 'H'. An escape beginning with `\x` always uses exactly two hexadecimal digits. Uppercase and lowercase hex digits have the same value: `'\x4C'` and `'\x4c'` both mean 'L'.

ASCII lookup. ASCII (American Standard Code for Information Interchange) assigns codes 0–127. The table shows every printable ASCII character, plus space and DEL (delete). Letters have different codes in uppercase and lowercase.

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

Common escapes. Each of these represents one character in a normal Python string.

Escape	Meaning	Decimal	Hex	Escape	Meaning	Decimal	Hex
<code>\n</code>	newline	10	0A	<code>\\</code>	backslash	92	5C
<code>\t</code>	tab	9	09	<code>\'</code>	single quote	39	27
<code>\r</code>	carriage return	13	0D	<code>\"</code>	double quote	34	22

Characters and bytes. In `'\x48'` the escape names a character; in `b'\x48'` it names a byte. ASCII characters use one byte each in UTF-8. Other characters may use more: `'café'` has four code points but uses five UTF-8 bytes. The character `é` has code point U+00E9 (decimal 233), written `'\u00e9'`; its UTF-8 bytes are `b'\xc3\xa9'`. A `\u` escape uses four hexadecimal digits; `\U` uses eight for larger code points.

`'cat'.encode('utf-8')` gives `b'cat'`. `b'cat'.decode('utf-8')` gives `'cat'`.

Reference: Python's [escape sequences](#) and [Unicode HOWTO](#).