Understanding ASCII Values in Python

What is ASCII?

ASCII stands for American Standard Code for Information Interchange. It is a standardized system to represent characters (like letters, numbers, and symbols) using numeric codes. Computers internally store and process characters as numbers. ASCII values bridge the gap between human-readable text and computer-readable data.

Why Use ASCII?

- Computers only understand numbers (binary). ASCII allows them to work with text.

- Makes it easier to store and transmit text.

- Useful in sorting, encrypting, comparing, or processing text.

How Does ASCII Work?

Each character is mapped to a unique number (from 0 to 127). For example:

- 'A' = 65

- 'a' = 97

- '0' = 48

- ' ' (space) = 32

- '@' = 64

How to Use ASCII in Python:

- Use ord() to get ASCII value of a character.

  Example: ord('A') = 65

- Use chr() to get character from ASCII value.

Example: chr(65) = 'A'

- ASCII values can be used in loops, comparisons, etc.

```
Examples:
1. Get ASCII value:
   print(ord('B'))  # Output: 66

2. Get character from ASCII:
   print(chr(100))  # Output: 'd'

3. Loop through A-Z:
   for i in range(65, 91):
       print(chr(i), end=' ')  # Output: A B C ... Z
```

Understanding Special ASCII Characters (like Backspace):

ASCII 8 represents the backspace character. In Python:

```
print("hello9" + chr(8))
```

This prints "hello9", because backspace only moves the cursor back. It doesn't delete anything unless something overwrites the character.

To actually remove the '9', you can do:

```
print("hello9\b \b")
```

This moves back, prints a space to erase '9', and moves back again.

ASCII Table (0 - 127):

Decimal	Char	Description

Understanding ASCII Values in Python

0	-	Null (NUL)
1	-	Start of Heading (SOH)
2	-	Start of Text (STX)
3	-	End of Text (ETX)
4	-	End of Transmission (EOT)
5	-	Enquiry (ENQ)
6	-	Acknowledge (ACK)
7	-	Bell (BEL)
8	-	Backspace (BS)
9	-	Horizontal Tab (TAB)
10	-	Line Feed (LF)
11	-	Vertical Tab (VT)
12	-	Form Feed (FF)
13	-	Carriage Return (CR)
14	-	Shift Out (SO)
15	-	Shift In (SI)
16	-	Data Link Escape (DLE)
17	-	Device Control 1 (DC1)
18	-	Device Control 2 (DC2)
19	-	Device Control 3 (DC3)
20	-	Device Control 4 (DC4)
21	-	Negative Acknowledge (NAK)
22	-	Synchronous Idle (SYN)
23	-	End of Transmission Block (ETB)
24	-	Cancel (CAN)
25	-	End of Medium (EM)
26	-	Substitute (SUB)
27	-	Escape (ESC)
28	-	File Separator (FS)
29	-	Group Separator (GS)

Understanding ASCII Values in Python

30	-	Record Separator (RS)
31	-	Unit Separator (US)
32	' '	Printable Character
33	'!'	Printable Character
34	'"'	Printable Character
35	'#'	Printable Character
36	'$'	Printable Character
37	'%'	Printable Character
38	'&'	Printable Character
39	"'"	Printable Character
40	'('	Printable Character
41	')'	Printable Character
42	'*'	Printable Character
43	'+'	Printable Character
44	','	Printable Character
45	'-'	Printable Character
46	'.'	Printable Character
47	'/'	Printable Character
48	'0'	Printable Character
49	'1'	Printable Character
50	'2'	Printable Character
51	'3'	Printable Character
52	'4'	Printable Character
53	'5'	Printable Character
54	'6'	Printable Character
55	'7'	Printable Character
56	'8'	Printable Character
57	'9'	Printable Character
58	':'	Printable Character
59	';'	Printable Character

Understanding ASCII Values in Python

60	'<'	Printable Character
61	'='	Printable Character
62	'>'	Printable Character
63	'?'	Printable Character
64	'@'	Printable Character
65	'A'	Printable Character
66	'B'	Printable Character
67	'C'	Printable Character
68	'D'	Printable Character
69	'E'	Printable Character
70	'F'	Printable Character
71	'G'	Printable Character
72	'H'	Printable Character
73	'I'	Printable Character
74	'J'	Printable Character
75	'K'	Printable Character
76	'L'	Printable Character
77	'M'	Printable Character
78	'N'	Printable Character
79	'O'	Printable Character
80	'P'	Printable Character
81	'Q'	Printable Character
82	'R'	Printable Character
83	'S'	Printable Character
84	'T'	Printable Character
85	'U'	Printable Character
86	'V'	Printable Character
87	'W'	Printable Character
88	'X'	Printable Character
89	'Y'	Printable Character

Understanding ASCII Values in Python

90	'Z'	Printable Character
91	'['	Printable Character
92	'\\'	Printable Character
93	']'	Printable Character
94	'^'	Printable Character
95	'_'	Printable Character
96	'`'	Printable Character
97	'a'	Printable Character
98	'b'	Printable Character
99	'c'	Printable Character
100	'd'	Printable Character
101	'e'	Printable Character
102	'f'	Printable Character
103	'g'	Printable Character
104	'h'	Printable Character
105	'i'	Printable Character
106	'j'	Printable Character
107	'k'	Printable Character
108	'l'	Printable Character
109	'm'	Printable Character
110	'n'	Printable Character
111	'o'	Printable Character
112	'p'	Printable Character
113	'q'	Printable Character
114	'r'	Printable Character
115	's'	Printable Character
116	't'	Printable Character
117	'u'	Printable Character
118	'v'	Printable Character
119	'w'	Printable Character

120	'x'	Printable Character
121	'y'	Printable Character
122	'z'	Printable Character
123	'{'	Printable Character
124	'|'	Printable Character
125	'}'	Printable Character
126	'~'	Printable Character
127	-	Delete (DEL)