

Char data-type ✓

- Scanner Class → supports `nextInt()`, `nextDouble()`, `nextC()`, `nextLine()`,
 - doesn't support `nextChar()`
 - `char ch = sc.nextC();` `charAt(0);`

```
Scanner scn = new Scanner(System.in);  
                                (one value)
```

```
{ char ch = scn.next().charAt(0); }
```

```
System.out.println(ch);
```

Character Methods ()

isLetter()

Syntax boolean isLetter (char ch);


```
System.out.println(Character.isLetter('c')); → true  
System.out.println(Character.isLetter('1')); → false
```

```
public static void main(String[] args) {  
    System.out.println(Character.isLetter('c')); → true  
    System.out.println(Character.isLetter('1')); → false  
    System.out.println(Character.isLetter('#')); → false  
    System.out.println(Character.isLetter('F')); → true
```

alphabetical character → a to z
A to Z

1
int

1
char

isDigit() → char type of data

Syntax → boolean isDigit(char ch)
(0 - 9)

```
System.out.println(Character.isDigit('1')); → true  
System.out.println(Character.isDigit('g')); → false  
System.out.println(Character.isDigit('9')); → true  
System.out.println(Character.isDigit('99')); → error  
(0 to 9)
```

isWhitespace()

Syntax → boolean isWhitespace(char ch);

```
System.out.println(Character.isWhitespace('e')); → false  
System.out.println(Character.isWhitespace(' ')); → true  
System.out.println(Character.isWhitespace('\n')); → true  
System.out.println(Character.isWhitespace('\t')); → true
```

Space bar → ' ' → one space

tab → '---' → four space

isUpperCase()

Syntax → boolean

isUpperCase(char ch);

true
false

```
public static void main(String[] args) {  
    System.out.println(Character.isUpperCase('c')); → false  
    System.out.println(Character.isUpperCase('G')); — true
```

isLowerCase()

Syntax → boolean

isLowerCase(char ch);

```
public static void main(String[] args) {  
    System.out.println(Character.isLowerCase('c')); — true  
    System.out.println(Character.isLowerCase('G')); — false
```

toUpperCase();

Syntax → char

toUpperCase(char ch);

↓ c to C

- any letter either small case or capital case
it will convert everything to
capital case/uppercase.

```
public static void main(String[] args) {  
    System.out.println(Character.toUpperCase('c')); → C  
    System.out.println(Character.toUpperCase('G')); → G  
    System.out.println(Character.toUpperCase('f')); → F  
    System.out.println(Character.toUpperCase('d')); → D
```

toLowerCase();

Syntax → char

toLower Case(char ch);

G → g

Syntax → char → lower case →

G → g

```
public static void main(String[] args) {  
    System.out.println(Character.toLowerCase('c'));
```

→ c

```
System.out.println(Character.toLowerCase('G'));
```

→ g

```
System.out.println(Character.toLowerCase('f'));
```

→ f

```
System.out.println(Character.toLowerCase('d'));
```

→ d

to String(); data type

Syntax → String → to string (char ch)

Geekster →

“G” “e” “e” “k” “s” “t” “e” “r”

combination

String → is a bunch of character

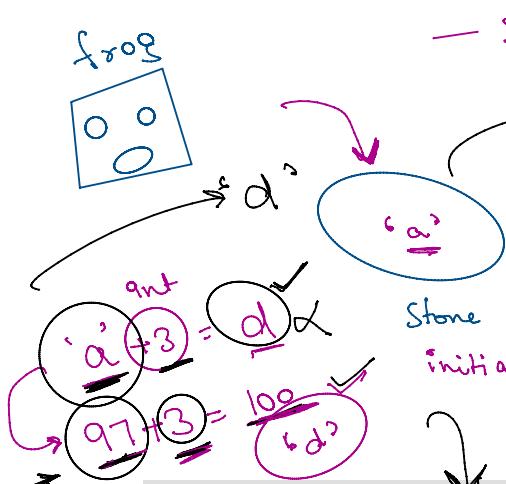
“c” → “c”
char string

```
public static void main(String[] args) {
```

```
    System.out.println(Character.toString('c'));
```

Char

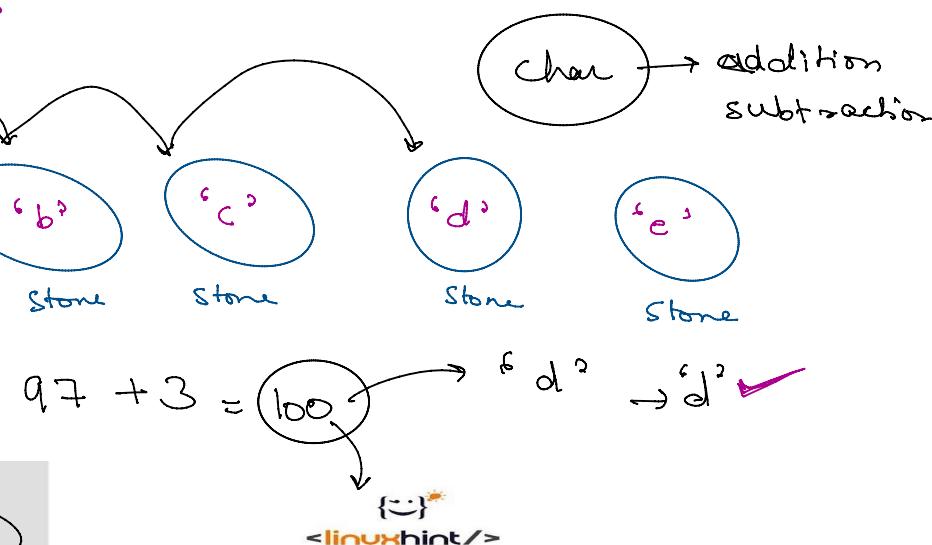
“c”
string



ASCII Table

Code Char	Code Char	Code Char	Code Char
0 NUL (null)	32 SPACE	64 @	96 `
1 SOH (start of heading)	33 !	65 A	97 a ✓
2 STX (start of text)	34 " "	66 B	98 b
3 ETX (end of text)	35 #	67 C	99 c
4 EOT (end of transmission)	36 \$	68 D	100 d
5 ENQ (enquiry)	37 %	69 E	101 e
6 ACK (acknowledge)	38 &	70 F	102 f
7 BEL (bell)	39 .	71 G	103 g
8 BS (backspace)	40 (72 H	104 h
9 TAB (horizontal tab)	41)	73 I	105 i

A → American
S → Standard
C → Code
I → Information



is Digit [0 to 9]

C — Code
I — Information
I → Interchange

encode
decode
encryption
decryption

5	ENQ (enquiry)	37 %	69 E	101 e
6	ACK (acknowledge)	38 &	70 F	102 f
7	BEL (bell)	39 '	71 G	103 g
8	BS (backspace)	40 (72 H	104 h
9	TAB (horizontal tab)	41)	73 I	105 i
10	LF (NL line feed, new line)	42 *	74 J	106 j
11	VT (vertical tab)	43 +	75 K	107 k
12	FF (NP form feed, new page)	44 .	76 L	108 l
13	CR (carriage return)	45 -	77 M	109 m
14	SO (shift out)	46 .	78 N	110 n
15	SI (shift in)	47 /	79 O	111 o
16	DLE (data link escape)	48 0	80 P	112 p
17	DC1 (device control 1)	49 1	81 Q	113 q
18	DC2 (device control 2)	50 2	82 R	114 r
19	DC3 (device control 3)	51 3	83 S	115 s
20	DC4 (device control 4)	52 4	84 T	116 t
21	NAK (negative acknowledge)	53 5	85 U	117 u
22	SYN (synchronous idle)	54 6	86 V	118 v
23	ETB (end of trans. block)	55 7	87 W	119 w
24	CAN (cancel)	56 8	88 X	120 x
25	EM (end of medium)	57 9	89 Y	121 y
26	SUB (substitute)	58 :	90 Z	122 z
27	ESC (escape)	59 :	91 [123 {
28	FS (file separator)	60 <	92 \	124
29	GS (group separator)	61 =	93]	125 }
30	RS (record separator)	62 >	94 ^	126 ~
31	US (unit separator)	63 ?	95 _	127 DEL

✓ is Digit [0 to 9]

How to convert char to int in java?

- Using ASCII value ✓
- Character.getNumericValue(); ✓
- using parseInt() method with String.valueOf(); Bridge method
- char to int by subtracting with '0' — important

Using ASCII value

```

char first ='5'; ✓ '9'
int second = first; — Ascii → '5' → 53 int → 57-48 = 9
System.out.println("char value: " + first);
System.out.println("ASCII Value" + second);
int num = second - 48; → 5
System.out.println("int value is " + num);

```

char value: 5 ✓



char value: 5
ASCII Value 53
int value is 5

47 1
48 0
49 1
50 2

```
public static void main(String[] args) {
```

```
    char first = '9'; '9'  

    int second = first; '9'  

    System.out.println("char value: " + first); 9  

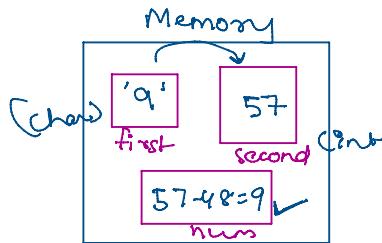
    System.out.println("ASCII Value" + second); 57  

    int num = second - 48;  

    System.out.println("int value is " + num); 9
```

char → '9' as an input
 '9' → 57 (ASCII)
 $57 - 48 \rightarrow 9$

char value: 9
ASCII Value 57
int value is 9



Character.getNumericValue()



```
char first = '5'; '5' → 5
System.out.println("char value: " + first);

int second = Character.getNumericValue(first); '5' → 5
System.out.println("int value is " + second); ✓
```

Finished in 124 ms

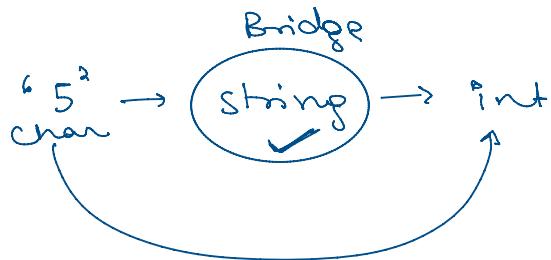
char value: 5

Finished in 124 ms

char value: 5

int value is 5

Bridge Method



- `parseInt()`
- `String.valueOf();`

```
public static void main(String[] args) {  
    char ch1 = '5';  
    char ch2 = '8';  
  
    int num1 = Integer.parseInt(String.valueOf(ch1));  
    int num2 = Integer.parseInt(String.valueOf(ch2));  
  
    System.out.println(num1); 5  
    System.out.println(num2); 8  
}
```

Annotations on the code:

- `char ch1 = '5';` and `char ch2 = '8';` are circled in green.
- `String.valueOf(ch1)` and `String.valueOf(ch2)` are circled in blue.
- `Integer.parseInt(String.valueOf(ch1))` and `Integer.parseInt(String.valueOf(ch2))` are circled in blue.
- A pink arrow labeled `ch → str` points from `ch1` to `String.valueOf(ch1)`.
- A pink arrow labeled `String → int` points from `String.valueOf(ch1)` to `num1`.
- A pink arrow labeled `5` points from `num1` to `5`.
- A pink arrow labeled `8` points from `num2` to `8`.

'5' — 5

Bridge → String

Char → String → int

Subtracting with '0';


```

char ch1 = '5';   '5'
char ch2 = '7';   '7'

int num1 = ch1 - '0'; = 53 - 48 = 5
int num2 = ch2 - '0';
              '7' - '0' = 55 - 48 = 7

```

```

System.out.println(num1);
System.out.println(num2);

```

48	0
49	1
50	2
51	3
52	4
53	5
54	6
55	7
56	8
57	9

