Strings = arrays of characters
- lowercase → 'a' to 'z'
- uppercase → 'A' to 'B'
- digits → '0' to '9'
- Null → '\0'
- Space, Escape

ASCII : American Standard Code for Information Interchange

→ Every character has a decimal code.

'A' → 65 , 'Z' → 90
'a' → 97 , 'z' → 122
'0' → 48 , '9' → 57

→ Remember

NULL → 0

☆ Unicode is superset of ASCII, Java uses Unicode

Java char → 2 Byte ∴ Unicode 2^{16}
C++ char → 1 Byte ∴ ASCII $2^8 = 128$ (0-127)

String s = "hello";

In Java
Array of characters does not need NULL termination,
In C++ NULL ('\0') termination is needed

CODE to make String in Java

Output

```
// print str:
System.out.println(str);                    hello

// print length
System.out.println(str.length());           5

// can also print using loop, but not using str[i].
for(int i=0; i<str.length(); i++)           h
    System.out.println(str.charAt(i));      e
                                            l
                                            l
                                            o
```

Advantage of strings over character arrays
→ Strings have some special functions

① Toggle Case:

Approach: if S.charAt(i) >= 'A' && S.charAt(i) <= 'Z'
add('a' - 'A');
else
subtract('a' - 'A');

CODE

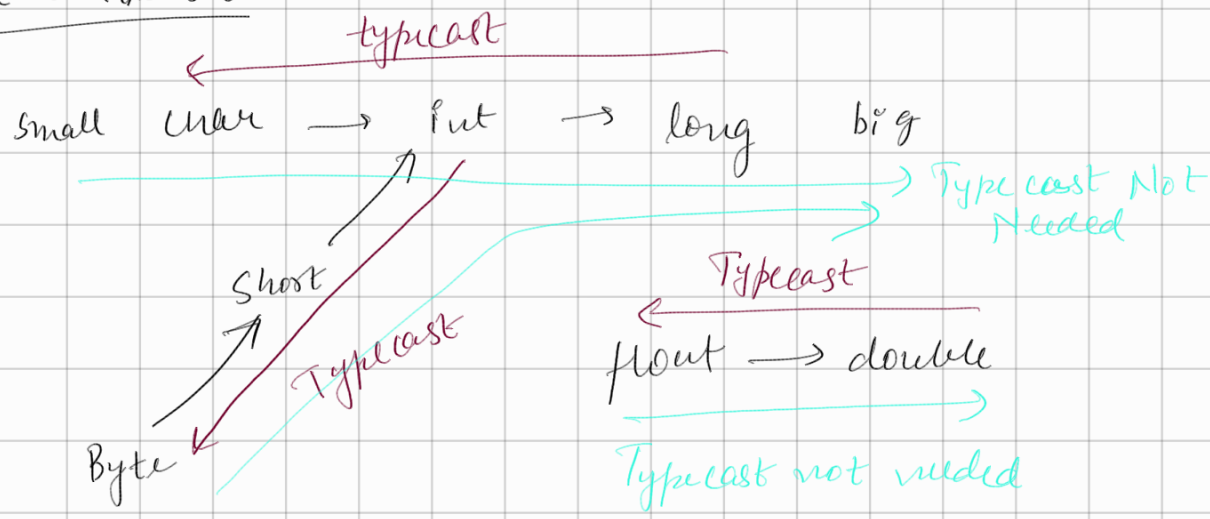
```
String res="";

for(int i=0; i<str.length(); i++){
    char ch = str.charAt(i);
    if(ch>='A' && ch<='Z')
        res += (char)(ch+32);
    else
        res += (char)(ch-32);
```

}

return res;

Type casting / conversion

Type Conversion



long to int ⇒ big to small value

Java → compile time error

Soln ⇒ type cast int x = (int) y;

perform only when sure that no loss occurs.

(2) Print all Palindromic Substrings.

Eg:

"a b c c"

Approach

① Traverse for all substrings

② check if Palindrome

true ↙ ↘ false

Print move on

CODE

```
for (int i=0; i< s.length(); i++) {
    for (int j=i ; j< s.length(); j++) {
        if ( isPalin (s.substring (i, j+1)) {
            System.out.println (s.substring (i, j+1));
        }
    }
}.
```

isPalin function:

```
int left = 0, right = s.length() - 1;
while ( left < right){
    if ( s.charAt[left] != s.charAt [right])
        return false;
    left ++;
    right --;
}
return true;
```