# Character Strings - Converting Strings

- Its is very common to convert character case
  - ➔ To all upper case or all lower case
- The ToUpper () function converts from lowercase to upper case
- The ToLower () function converts from uppercase to lower Case

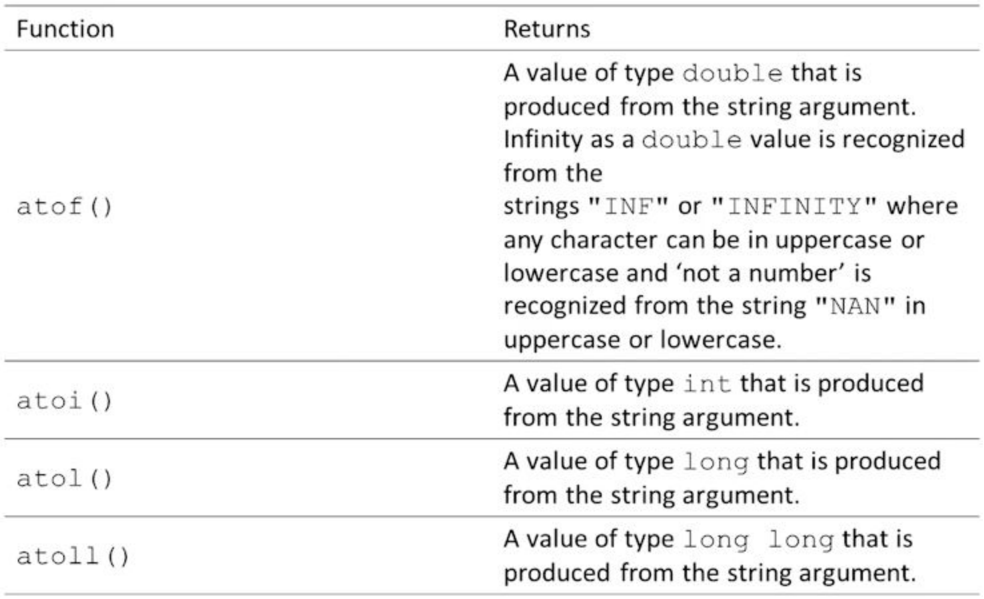
- Both functions return either the converted  character or the character for characters that are already in the correct case or are not convertible such as punctuation characters
- This is how you convert a string to uppercase

```
for (int i = 0 ; (buf[i] = (char) toupper(buf [i] != ‘\0’; ++i);
```

- This loop will convert the entire string in the buf array to uppercase by stepping through the string one character at a time
  - ➔ Loop stops when it reaches the string termination character ‘\0’
  - ➔ The cast to type char is there because toupper() returns type int

- You can use the function toupper() in combination with the strstr() function to find out whether one string occurs in another, ignoring case.

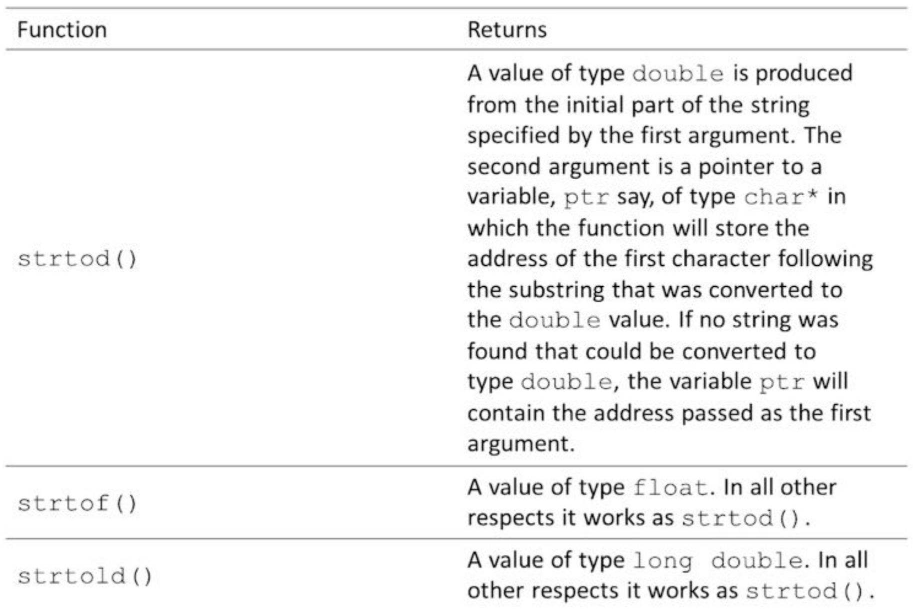
## Converting  Strings To numbers



| Function | Returns |
|---|---|
| `atof()` | A value of type `double` that is produced from the string argument. Infinity as a `double` value is recognized from the strings "`INF`" or "`INFINITY`" where any character can be in uppercase or lowercase and ‘not a number’ is recognized from the string "`NAN`" in uppercase or lowercase. |
| `atoi()` | A value of type `int` that is produced from the string argument. |
| `atol()` | A value of type `long` that is produced from the string argument. |
| `atoll()` | A value of type `long long` that is produced from the string argument. |

The stdlib.h header file declares functions that you can use to convert a string to a numeric value
For all four functions, leading whitespace is ignored.
Char value_str[] = “98.4”;

Double value = atof(value_str);



| Function | Returns |
|---|---|
| `strtod()` | A value of type `double` is produced from the initial part of the string specified by the first argument. The second argument is a pointer to a variable, `ptr` say, of type `char*` in which the function will store the address of the first character following the substring that was converted to the `double` value. If no string was found that could be converted to type `double`, the variable `ptr` will contain the address passed as the first argument. |
| `strtof()` | A value of type `float`. In all other respects it works as `strtod()`. |
| `strtold()` | A value of type `long double`. In all other respects it works as `strtod()`. |

```
#include <stdlib.h>
#include <stdio.h>

int main ()
{
    double value = 0 ;
    char str[] = "3.5 2.5 1.26";          // The string to be
converted
    char *pstr = str;                     // Pointer to the
string to be converted
    char *ptr = NULL;                     // Pointer to character
position after conversion

    while(1)
    {
        value = strtod(pstr, &ptr);        //Convert starting at
pstr
        if(pstr == ptr)                   //pstr stored if no
conversion..
            break;                        //...so we are done
            else
            {
                printf(" %f", value);     //Output the resultant
value
```

```
                    pstr = ptr;                // Store Start for next
conversation
               }
     }
}
```