String

Strings are sequence of characters.

The string in rust can be classified into the following –

- String literal (&str)
- String Object (String)

String literal

String literals (&str) are used when the value of a string is known at compile time. String literals are a set of characters, which are hardcoded into a variable.

```
fn main() {
   let name:&str="OneTwoCoding";
   println!("{}",name);
}
```

String literals are static by default. This means that string literals are guaranteed to be valid for the duration of the entire program.

String object

To create a String object, we can use any of the following syntax −

- String::new()
- String::from()

This creates a string with some default value passed as parameter to the from() method.

String Slicing

String slicing is all about fetching a substring (part of a string) from a given string.

Therefore, we need to specify the starting and ending index of a String. Index starts from 0.

The minimum index value is 0 and the maximum index value is the size of the data structure. NOTE that the last index will not be included in final string.

The diagram below shows a sample string Abhishek, that has 8 characters. The index of the first character is 0 and that of the last character is 7.

A	b	h	i	s	h	e	k
0	1	2	3	4	5	6	7

```
fn main(){
    let name = "Abhishek";
    let n = &name[0..2];
    println!("{}",n);
}
```

String concatenation

1. Format!()
2. +

Format!()

We can use it for both string literals and string object

```
fn main(){
    let name = "Abhishek";
    let name_2 = "Kumar";
    let con = format!("{} {}",name,name_2);
    println!("{}",con);
}
```

+

Only used for string object

```
fn main(){
    let name = "Abhishek".to_string();
    let name_2 = " Kumar".to_string();
    let con =  name + &name_2;
    println!("{}",con);
}
```

String methods

Methods are used for checking and modifying the string.

- new()   ---- Creates a new empty String.
- from()   ---- Creates a new String having default value.
- to_string() ---- Converts string literal to string object.
- replace() ---- Replaces all matches of a pattern with another string.
- push() ---- Appends the given char to the end of this String.
- push_str() --- Appends a given string onto the end of this String.
- len() ---- Returns the length of the String.