IV TH UNIT: VARIABLES AND DATA TYPES IN R

1. Allowed characters are alphanumeric '_','.'

2. Always start with alphabets

3. No special characters are allowed like $ & etc

predefined constants:

```
> pi
[1] 3.141593

> letters
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n"
[15] "o" "p" "q" "r" "s" "t" "u" "v" "w" "x" "y" "z"

> LETTERS
 [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N"
[15] "O" "P" "Q" "R" "S" "T" "U" "V" "W" "X" "Y" "Z"

> month.name
 [1] "January"   "February"  "March"     "April"
 [5] "May"       "June"      "July"      "August"
 [9] "September" "October"   "November"  "December"

> month.abb
 [1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep"
[10] "Oct" "Nov" "Dec"

>
```

Data types that are available in the R.

R has logical data types which take either a value of true or false, it supports integer data types which is the set of all integers and numeric which is set of all real numbers. We can also define complex variables, R supports set of all the complex numbers. Also, we can have a character data type where you have all the alphabets and special characters which are under the window of basic data types of
characters.There are several task that can be done using data types.

```
> typeof(1)
[1] "double"

> typeof(1.5)
[1] "double"

> typeof("2021-11-16")
[1] "character"

> is.character("2021-11-16")
[1] TRUE

> is.character(as.date("2021-11-16"))
Error in as.date("2021-11-16") : could not find function "as.date"

> is.character(as.Date("2021-11-16"))
[1] FALSE

> as.complex(2)
[1] 2+0i
```

```
> as.numeric("2")
[1] 2

> as.numeric("a")
[1] NA
Warning message:
NAs introduced by coercion
```

BASIC OBJECTS:

1. vector: A vector is an ordered collection of same data types.

2. list: list is ordered collection of object themselves

3. Data Frames: data frame is a generic tabular object

VECTORS:

```
> # VECTORS EXAMPLE
> X=c(2.3,4.6,1.2,7.8)
> print(X)
[1] 2.3 4.6 1.2 7.8

>
```

LIST: List is a generic object consisting of ordered collection of objects. List can be a list of vectors, list of matrices, list of characters and list of functions and so on.

```
> #List Example: Employee Details:
> ID=c(1,2,3,4)
> EMP.NAME=c("man","RAG","kat","SHA")
> NUM.EMP=4
> EMP.LIST(list(ID,EMP.NAME,NUM.EMP))
Error in EMP.LIST(list(ID, EMP.NAME, NUM.EMP)) :
  could not find function "EMP.LIST"
> EMP.LIST=(list(ID,EMP.NAME,NUM.EMP))
> EMP.LIST
[[1]]
[1] 1 2 3 4

[[2]]
[1] "man" "RAG" "kat" "SHA"

[[3]]
[1] 4
```

All the components of a list can be named and you can use that names to access the components of the list.

```
> EMP.LIST=list("id"=ID,"empname"=EMP.NAME,"empno"=NUM.EMP)
> EMP.LIST$empname
[1] "man" "RAG" "kat" "SHA"
> EMP.LIST$empno
[1] 4

>
```

You can also access the components of the list using indices. To access the top level components of a list you have to use double slicing operator which is two square brackets and if you want access the lower or inner level components of a list you have to use another square bracket along with the double slicing operator.

```
> print(EMP.LIST[1])
$id
[1] 1 2 3 4

> print(EMP.LIST[2])
$empname
[1] "man" "RAG" "kat" "SHA"

> print(EMP.LIST[3])
$empno
[1] 4

> print(EMP.LIST[4])
$<NA>
NULL


> print(EMP.LIST[[1]][1])
[1] 1
> print(EMP.LIST[[2]][1])
[1] "man"
> print(EMP.LIST[[2]][2])
[1] "RAG"
> print(EMP.LIST[[1]][2])
[1] 2
> print(EMP.LIST[[3]][1])
[1] 4
> print(EMP.LIST[[1]][3])
[1] 3
> print(EMP.LIST[[3]][3])
[1] NA

>
```

Two lists can be concatenated using the concatenation function.

```
> emp.ages=list("ages"=c(25,30,26,27))
> EMP.LIST=c(EMP.LIST,emp.ages)
> EMP.LIST
$id
[1] 1 2 3 4

$empname
[1] "man" "RAG" "kat" "SHA"

$empno
[1] 4

$ages
[1] 25 30 26 27
```