Basic of R Programming Language

R is a popular programming language used for statistical computing and graphical presentation. Its most common use is to analyze and visualize data.

Why Use R?

- It is a great resource for data analysis, data visualization, data science and machine learning
- It provides many statistical techniques (such as statistical tests, classification, clustering and data reduction)
- It is easy to draw graphs in R, like pie charts, histograms, box plot, scatter plot, etc...
- It works on different platforms (Windows, Mac, Linux)
- It is open-source and free
- It has a large community support
- It has many packages (libraries of functions) that can be used to solve different problems

R & RStudio Setup on Windows

Watch this Video By Click on it 👇

Get Started

```
# Start With "Hello World"
print("Hello World!")
```

```
[1] "Hello World!"
```

```
name <- "Janak Singh"
print(name)
```

```
[1] "Janak Singh"
```

```
add = "KTM"
```

```
cat("Hello Universe!😂")
```

```
Hello Universe!😂
```

```
cat(name)
```

```
Janak Singh
```

```
paste("Hello, Elon Musk!😊")
```

```
'Hello, Elon Musk!😊'
```

```
paste("Hello", "World", sep=",")
```

```
'Hello,World'
```

User Input

```
my_name <- readline(prompt = "What is Your Name?")
class(my_name)

age <- readline(prompt="What is Your age?")
my_age <- as.integer(age)
class(my_age)
```

```
What is Your Name?Janak SIngh DHami
'character'
What is Your age?20
'integer'
```

Comments in R

Comments can be used to explain R code, and to make it more readable.

Comments starts with a #. When executing code, R will ignore anything that starts with #.

```
# This is Comment
print("How are you?") #This is also comment👌
```

```
[1] "How are you?"
```

```
# this is comment
print("Hello Guys")

#print(name) # My name
```

```
[1] "Hello Guys"
```

Variables in R

Variables in *R* are used to store data values. You can assign values to variables using the **<-** operator or the **=** operator.

Variable names must start with a letter and can include letters, numbers, and underscores.

Rules for Variable Naming:

- Must start with a letter.
- Cannot contain spaces or special characters (except _ and .).
- Case-sensitive (e.g., var and Var are different).

```
# name
# Name
# my_name
# my-name - X
# my addr = "KTM"


# case-sensitive
addr = "Nepal"
Addr = "India"

print(addr)
print(Addr)
```

```
[1] "Nepal"
[1] "India"
```

```
# Assigning values to variables

x <- 10         # Numeric value
y <- 20.5      # Decimal/float value
name <- "Alice" # Character value
isOk <- TRUE    # Logical value
com <- 3i+6   #Complex number

# Using variables
sum <- x + y

message <- paste("Name:", name, "| Sum:", sum, "| isOk:", isOk)

# Printing the result
print(message)
```

```
[1] "Name: Alice | Sum: 30.5 | isOk: TRUE"
```

- The ***paste()*** function concatenates strings.
- The ***print()*** function displays the output.

```
a <- 55L
b <- 55.5
hlo <- "Hii"
is <- TRUE
com <- 3i +7

class(a)
class(b)
class(hlo)
class(is)
class(com)
```

```
'integer'
'numeric'
'character'
'logical'
'complex'
```

```
# check variable type
class(x)
class(y)
class(com)
```

```
'numeric'
'numeric'
'complex'
```

```
x <- 1L # integer
y <- 2 # numeric

# convert from integer to numeric:
```

```
a <- as.numeric(x)

# convert from numeric to integer:
b <- as.integer(y)
```

```
p <- 5
class(p)
```

```
'numeric'
```

```
q = as.integer(p)
class(q)
```

```
'integer'
```

Multiple Variables

R allows you to assign the same value to multiple variables in one line:

```
addr1 <- aad2 <- "KTM"
```

```
print(addr1)
print(aad2)
```

```
[1] "KTM"
[1] "KTM"
```

```
# Assign the same value to multiple variables in one line
var1 <- var2 <- var3 <- "Orange"

# Print variable values
var1
var2
var3
```

```
'Orange'
'Orange'
'Orange'
```

Basic Data Types

Basic data types in R can be divided into the following types:

- **numeric** - (10.5, 55, 787)

> A numeric data type is the most common type in R, and contains any number with or without a decimal, like: 10.5, 55, 787

- **integer** - (1L, 55L, 100L, where the letter "L" declares this as an integer)

  > Integers are numeric data without decimals. This is used when you are certain that you will never create a variable that should contain decimals. To create an integer variable, you must use the letter L after the integer value

- **complex** - (9 + 3i, where "i" is the imaginary part)

  > A complex number is written with an "i" as the imaginary part

- **character** (a.k.a. string) - ("k", "R is exciting", "FALSE", "11.5")
- **logical** (a.k.a. boolean) - (TRUE or FALSE)

> We can use the class() function to check the data type of a variable.

```
# numeric
x <- 10.5
class(x)
```