R Basics

2023-03-01

Different ways to Print

These are different ways to print

print function

used to print single input

print automatically adds new line ‘’ at end of the input

```
print("Hello")
```

```
## [1] "Hello"
```

```
print("World!")
```

```
## [1] "World!"
```

cat function

cat concatenate multiple objects (representation) , strings by using space as a separator

created full string is given to console

cat doesn’t add newline at the end

cat doesn’t return anything (returns NULL)

cat uses much less conversions than print

```
cat("He","llo") # 'he' and 'llo' will be concatenated with space as separator
```

```
## He llo
```

```
cat("World!") # next cat command will directly send 'World!' to console without any separator
```

```
## World!
```

paste function

Concatenate vectors after converting to character

paste returns the concatenated character vector

return value can be assigned to a variable else it is printed using print function

if return value is printed then print adds new line at the end

_

```
paste("He","llo") # 'he' and 'llo' will be concatenated with space as separator and character vector is returned. This returned vector object is printed to console using print. This adds '\n' at the end of the line.
```

```
## [1] "He llo"
```

```
paste("World!") # next paste command will create another character vector 'World!' and return it. This returned character vector is printed on console using print
```

```
## [1] "World!"
```

Use variable name directly

Only writing variable name on line in script will print current value of that variable on console

```
var1 = "Hello World!"
var1
```

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

Accept user input

readline function

Works in interactive mode ONLY

Accepts input from user and returns a character vector

If user wants to read integer/numeric then returned value needs to be converted to integer / numeric

```
yourname <- readline(prompt="What is your name? ") # Assume no input is given by the user then empty character vector is returned
```

```
## What is your name?
```

```
print(yourname) # empty character vector is printed
```

```
## [1] ""
```

```
print(class(yourname)) # class of 'yourname' is 'character'
```

```
## [1] "character"
```

Working Directory of R

Directory which currently R studio is working on

R studio take this as default location to access files or store files

How to check current working directory ?

```
getwd()
```

```
## [1] "I:/R_practice/R_programming_notes"
```

How to change working directory ?

setwd function is used to set new directory

setwd is given path

path can use double back-slash “\ \” or single forward-slash “/”

preferably directory should not have name with spaces

NOTE : setwd doesn’t work well inside a R markdown or notebook

```
# path can use \\ or /
setwd("./inside_folder")
```

Install R packages

You can use commands or use GUI options to install any package

Install packages using commands

install.packages command helps install given package

Package name should be given in double quotes or single quotes

```
# install.packages('tidyverse')
```

Install packages using R Studio GUI

In R studio, main menu displayed on the top

Select Tools & click it

In Tools -> click “Install Packages” option

There will be a pop-up window

Here type the name of the package to be installed

Then package will be installed

Load packages in R

Here we load the packages in memory

Without loading packages, they can’t be used

While starting R, few base packages are pre loaded

```
#library(tidyverse)
```