**Print Output**

Example 1:

print("Hello World!")

Example 2:

for (x in 1:10) {

print(x)

}

**Comments**

* Comments can be used to explain R code, and to make it more readable. It can also be used to prevent execution when testing alternative code.
* Comments starts with a #. When executing code, R will ignore anything that starts with #.

Example:

# This is a comment

print("Hello World!") # This is also a comment

**Variables**

* Variables are containers for storing data values.
* R does not have a command for declaring a variable.
* A variable is created the moment you first assign a value to it.
* To assign a value to a variable, use the <- sign. To output (or print) the variable value, just type the variable name

Example 1:

name <- "Pranjal"

age <- 28

print(name)

print(age)

To concatenate, or join, two or more elements, use the paste() function

Example 2:

text <- "awesome"

paste("R is", text)

Example 3:

text1 <- "R is"

text2 <- "awesome"

paste(text1, text2)

Example 4:

num1 <- 5

num2 <- 10

num1 + num2

Example 5:

# Assign the same value to multiple variables in one line

var1 <- var2 <- var3 <- "Orange"

**Rules for R variables are:**

* A variable name must start with a letter and can be a combination of letters, digits, period(.)

and underscore (\_). If it starts with period(.), it cannot be followed by a digit.

* A variable name cannot start with a number or underscore (\_)
* Variable names are case-sensitive (age, Age and AGE are three different variables)
* Reserved words cannot be used as variables (TRUE, FALSE, NULL, if...)

**Data Types:**

* Variables can store data of different types, and different types can do different things.
* In R, variables do not need to be declared with any particular type, and can even change type after they have been set

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

numeric - (10.5, 55, 787)

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

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

character (a.k.a. string) - ("k", "R is exciting", "FALSE", "11.5")

logical (a.k.a. boolean) - (TRUE or FALSE)

# Numeric (default type for numbers)

num\_var <- 42.5

print(paste("Numeric:", num\_var))

print(paste("Type:", class(num\_var)))

# Integer

int\_var <- 42

print(paste("Integer:", int\_var))

print(paste("Type:", class(int\_var)))

# Character (text/string)

char\_var <- "Hello, R!"

print(paste("Character:", char\_var))

print(paste("Type:", class(char\_var)))

# Logical (boolean values)

log\_var <- TRUE

print(paste("Logical:", log\_var))

print(paste("Type:", class(log\_var)))

# Factor (categorical data)

factor\_var <- factor(c("Male", "Female", "Female", "Male"))

print("Factor:")

print(factor\_var)

print(paste("Type:", class(factor\_var)))

**Type Conversion**

You can convert from one type to another with the following functions:

* as.numeric()
* as.integer()
* as.complex()

**Maths Functions**

Arithmetic Operators

* Addition: +
* Subtraction: -
* Multiplication: \*
* Division: /
* Exponentiation: ^
* Integer Division: %/%
* Modulo: %%

Trigonometric Functions

* Sine: sin()
* Cosine: cos()
* Tangent: tan()
* Arcsine: asin()
* Arccosine: acos()
* Arctangent: atan()

Logarithmic Functions

* Natural logarithm: log()
* Base 10 logarithm: log10()
* Base 2 logarithm: log2()

Exponential Functions

* Exponential: exp()
* Square Root and Absolute Value
* Square root: sqrt()
* Absolute value: abs()

Rounding Functions

* Round to nearest integer: round()
* Round down to nearest integer: floor()
* Round up to nearest integer: ceiling()

Statistical Functions

* Mean: mean()
* Median: median()
* Mode: mode()
* Variance: var()
* Standard deviation: sd()
* Correlation: cor()
* Covariance: cov()

Other Useful Functions

* Factorial: factorial()
* Combinations: choose()
* Permutations: perm()
* Random number generation: runif(), rnorm(), rbinom(), etc.

**String Functions**

# Concatenation

name <- "Alice"

greeting <- "Hello"

message <- paste(greeting, name)

cat("Concatenation:", message, "\n")

# Length

text <- "This is a sample text"

length <- nchar(text)

cat("Length:", length, "\n")

# Substring extraction

substring <- substr(text, 8, 13)

cat("Substring extraction:", substring, "\n")

# Character replacement

new\_text <- gsub("sample", "example", text)

cat("Character replacement:", new\_text, "\n")

# Case conversion

upper\_case <- toupper(text)

lower\_case <- tolower(text)

cat("Upper case:", upper\_case, "\n")

cat("Lower case:", lower\_case, "\n")

# Trimming

text\_with\_spaces <- " This text has spaces "

trimmed\_text <- trim(text\_with\_spaces)

cat("Trimming:", trimmed\_text, "\n")

# Regular expressions

pattern <- "text"

contains\_pattern <- grepl(pattern, text)

cat("Regular expressions:", contains\_pattern, "\n")

# Fixed string matching

library(stringr)

contains\_pattern <- str\_detect(text, pattern)

cat("Fixed string matching:", contains\_pattern, "\n")

# sprintf()

formatted\_string <- sprintf("The value of pi is approximately %.2f", pi)

cat("sprintf():", formatted\_string, "\n")

# str\_pad()

padded\_string <- str\_pad("Hello", 10, side = "right", pad = "-")

cat("str\_pad():", padded\_string, "\n")

# str\_split()

words <- strsplit(text, " ")[[1]]

cat("str\_split():", words, "\n")

# str\_wrap()

wrapped\_text <- str\_wrap(text, width = 15)

cat("str\_wrap():", wrapped\_text, "\n")

# str\_trim()

trimmed\_text <- str\_trim(text\_with\_spaces)

cat("str\_trim():", trimmed\_text, "\n")

**Operators**

* Arithmetic
* Relational
* Logical

**# Arithmetic operators**

x <- 10

y <- 5

# Addition

result <- x + y

cat("Addition:", result, "\n")

# Subtraction

result <- x - y

cat("Subtraction:", result, "\n")

# Multiplication

result <- x \* y

cat("Multiplication:", result, "\n")

# Division

result <- x / y

cat("Division:", result, "\n")

# Exponentiation

result <- x ^ y

cat("Exponentiation:", result, "\n")

# Integer division

result <- x %/% y

cat("Integer division:", result, "\n")

# Modulo

result <- x %% y

cat("Modulo:", result, "\n")

**# Relational operators**

x <- 10

y <- 5

# Equal to

result <- x == y

cat("Equal to:", result, "\n")

# Not equal to

result <- x != y

cat("Not equal to:", result, "\n")

# Greater than

result <- x > y

cat("Greater than:", result, "\n")

# Less than

result <- x < y

cat("Less than:", result, "\n")

# Greater than or equal to

result <- x >= y

cat("Greater than or equal to:", result, "\n")

# Less than or equal to

result <- x <= y

cat("Less than or equal to:", result, "\n")

**# Logical operators**

x <- TRUE

y <- FALSE

# AND

result <- x & y

cat("AND:", result, "\n")

# OR

result <- x | y

cat("OR:", result, "\n")

# NOT

result <- !x

cat("NOT:", result, "\n")

**Conditional Statements**

# Example of if-else statement

age <- 18

if (age >= 18) {

print("You are an adult.")

} else {

print("You are a minor.")}

# Example of if-else-if statement

grade <- 85

if (grade >= 90) {

print("You got an A.")

} else if (grade >= 80) {

print("You got a B.")

} else if (grade >= 70) {

print("You got a C.")

} else {

print("You need to study harder.")

}

# Example of switch statement

day <- "Monday"

switch(day,

"Monday" = print("It's Monday!"),

"Tuesday" = print("It's Tuesday!"),

"Wednesday" = print("It's Wednesday!"),

"Thursday" = print("It's Thursday!"),

"Friday" = print("It's Friday!"),

"Saturday" = print("It's Saturday!"),

"Sunday" = print("It's Sunday!"),

print("Invalid day"))

**While Loop**

Example 1:

count <- 0

while (count < 5) {

print(count)

count <- count + 1 }

Example 2:

principal <- 10000 # Initial deposit amount

rate <- 0.05 # Annual interest rate

time <- 5 # Number of years

# Calculate compound interest

amount <- principal

years <- 0

while (years < time) {

interest <- amount \* rate

amount <- amount + interest

years <- years + 1

}

# Print the final amount

cat("Final amount after", time, "years:", amount, "\n")

**For loop**

Example 1:

number <- 5

factorial <- 1

for (i in 1:number) {

factorial <- factorial \* i

}

cat("Factorial of", number, ":", factorial, "\n")

Example 2:

numbers <- c(10, 20, 30, 40, 50)

sum <- 0

count <- 0

for (number in numbers) {

sum <- sum + number

count <- count + 1 }

average <- sum / count

cat("Average of the numbers:", average, "\n")

**Functions**

* A function is a block of code which only runs when it is called.
* You can pass data, known as parameters, into a function.
* A function can return data as a result.

Example 1:

my\_function <- function() { # create a function with the name my\_function

print("Hello World!")

}

my\_function()

Example 2:

my\_function <- function(fname) {

paste(fname, " Pranjal ")

}

my\_function("Alex")

my\_function("Bravo")

my\_function("Kevin")

Example 3:

my\_function <- function(fname, lname) {

paste(fname, lname)

}

my\_function("Pranjal", "Srivastava")

Example 4:

my\_function <- function(country = "Norway") {

paste("I am from", country)

}

my\_function("USA")

my\_function("India")

my\_function() # will get the default value, which is Norway

my\_function("Japan")

Example 5:

my\_function <- function(x) {

return (5 \* x)

}

print(my\_function(3))

print(my\_function(5))

print(my\_function(9))