

```

> # Examples of different data types in R
> numeric_var <- 3.14          # Numeric
> integer_var <- 10L           # Integer
> logical_var <- TRUE          # Logical
> character_var <- "Success"   # Character
> complex_var <- 2 + 3i        # Complex
>
> # Checking the type and mode
> typeof(numeric_var)         # Output: "double"
[1] "double"
> mode(numeric_var)           # Output: "numeric"
[1] "numeric"
> typeof(integer_var)         # Output: "integer"
[1] "integer"
> mode(integer_var)           # Output: "numeric"
[1] "numeric"
> source("D:/RProgramming/Class2/Self/Self_study.R")
> source("D:/RProgramming/Class2/Self/Self_study.R")
> # Using c() function to combine elements
> numeric_vector <- c(1, 2, 3)
> character_vector <- c("a", "b", "c")
> logical_vector <- c(TRUE, FALSE, TRUE)
>
> print(numeric_vector)       # Output: 1 2 3
[1] 1 2 3
> print(character_vector)
[1] "a" "b" "c"
> print(logical_vector)
[1] TRUE FALSE TRUE
> # Assigning values using different operators
> x <- 10
> 10 -> y
> z = 20
> # Examples of arithmetic operations
> sum_result <- 10 + 3        # Addition
> sub_result <- 10 - 3        # Subtraction
> div_result <- 10 / 3         # Division
> mul_result <- 10 * 3         # Multiplication
> exp_result <- 10^3           # Exponent
> mod_result <- 10 %% 3        # Modulus
> int_div_result <- 10 %/% 3   # Integer Division
>
> # Print results
> print(sum_result)           # Output: 13
[1] 13
> # Examples of relational operations
> print(4 < 2)                 # FALSE

```

```
[1] FALSE
> print(4 <= 10)      # TRUE
[1] TRUE
> print(4 >= 4)       # TRUE
[1] TRUE
> print(3 == 5)       # FALSE
[1] FALSE
> print(3 != 5)       # TRUE
[1] TRUE
> print('a' %in% c('b', 'a', 'c')) # TRUE
[1] TRUE
> # Examples of logical operations
> print(TRUE & FALSE) # FALSE
[1] FALSE
> print(TRUE | FALSE) # TRUE
[1] TRUE
> print(!TRUE)        # FALSE
[1] FALSE
> # Statistical functions
> nums <- c(10, 20, 30, 40, 50)
> print(sum(nums))     # Sum
[1] 150
> print(mean(nums))    # Mean
[1] 30
> print(min(nums))     # Minimum
[1] 10
> print(max(nums))     # Maximum
[1] 50
>
> # Logical functions
> print(any(is.na(nums))) # Check for NA values
[1] FALSE
> print(all(nums > 5))  # Check if all elements > 5
[1] TRUE
>
> # Rounding functions
> print(round(3.14159, 2)) # 3.14
[1] 3.14
> print(floor(3.7))     # 3
[1] 3
> print(ceiling(3.3))   # 4
[1] 4
```