R Documentation:

Basic Syntax:

```
> myString <- "Hello, World!"
> print ( myString)
[1] "Hello, World!"
```

Here first statement defines a string variable myString, where we assign a string "Hello, World!" and then next statement print() is being used to print the value stored in variable myString.

Data Types:

- Vectors
- Lists
- Matrices
- Arrays
- Factors
- Data Frames

Data Type	Example	Verify

Logical	TRUE, FALSE	`v <- TRUE`
		`print(class(v))`
		it produces the following result −
		`[1] "logical"`

Numeric	12.3, 5, 999	`v <- 23.5`
		`print(class(v))`
		it produces the following result −
		`[1] "numeric"`

Integer	2L, 34L, 0L	`v <- 2L`
		`print(class(v))`
		it produces the following result −
		`[1] "integer"`

Complex	3 + 2i	`v <- 2+5i`
		`print(class(v))`
		it produces the following result −
		`[1] "complex"`

Character	'a' , '"good", "TRUE", '23.4'	`v <- "TRUE"`
		`print(class(v))`
		it produces the following result −
		`[1] "character"`

Raw	"Hello" is stored as 48 65 6c 6c 6f	`v <- charToRaw("Hello")`
		`print(class(v))`
		it produces the following result −
		`[1] "raw"`

Variables

The variables can be assigned values using leftward, rightward and equal to operator. The values of the variables can be printed
using print() or cat() function. The cat() function combines multiple items into a continuous print output.

```
# Assignment using equal operator.
var.1 = c(0,1,2,3)

# Assignment using leftward operator.
var.2 <- c("learn","R")

# Assignment using rightward operator.
c(TRUE,1) -> var.3

print(var.1)
cat ("var.1 is ", var.1 ,"\n")
cat ("var.2 is ", var.2 ,"\n")
cat ("var.3 is ", var.3 ,"\n")
```

Loops

For:

```
for (value in vector) {
   statements
}
```

Example:

```
v <- LETTERS[1:4]
for ( i in v) {
   print(i)
}
```

While:

```
while (test_expression) {
   statement
}
```

Example:

```
v <- c("Hello","while loop")
cnt <- 2

while (cnt < 7) {
   print(v)
   cnt = cnt + 1
}
```

Repeat:

```
repeat {
   commands
   if(condition) {
      break
   }
}
```

Example:

```
v <- c("Hello","loop")
cnt <- 2

repeat {
   print(v)
   cnt <- cnt+1
```

```
   if(cnt > 5) {
      break
   }
}
```

Data Structures:

Vector:

```
# Create a vector.
apple <- c('red','green',"yellow")
print(apple)

# Get the class of the vector.
print(class(apple))
```

Lists:

```
# Create a list.
list1 <- list(c(2,5,3),21.3,sin)

# Print the list.
print(list1)
```

matrix:

```
# Create a matrix.
M = matrix( c('a','a','b','c','b','a'), nrow = 2, ncol = 3, byrow
= TRUE)
print(M)
```

array:

```
a <- array(c('green','yellow'),dim = c(3,3,2))
print(a)
```

Functions:

```
function_name <- function(arg_1, arg_2, ...) {
   Function body
}
```

Example:

```
# Create a function to print squares of numbers in sequence.
new.function <- function(a) {
   for(i in 1:a) {
      b <- i^2
      print(b)
   }
}

# Call the function new.function supplying 6 as an argument.
new.function(6)
```