R Basics

R Reserved words

Reserved words in R programming are a set of words that have special meaning and cannot be used as an identifier (variable name, function name etc.).

Here is a list of reserved words in the R's parser.

the reserved words in R can be viewed by

```
> ?reserved
> help("reserved")
```

Reserved words in R

if	else	repeat	while	function	
for	in	next	break	TRUE	
FALSE		NULL	Inf	NaN	NA
NA_integer_		NA_real_		NA_complex_	NA_character_ …

Among these words, `if`, `else`, `repeat`, `while`, `function`, `for`, `in`, `next` and `break` are used for conditions, loops and user defined functions.

They form the basic building blocks of programming in R.

`TRUE` and `FALSE` are the logical constants in R.

`NULL` represents the absence of a value or an undefined value.

`Inf` is for "Infinity", for example when 1 is divided by 0 whereas `NaN` is for "Not a Number", for example when 0 is divided by 0.

`NA` stands for "Not Available" and is used to represent missing values.

R is a case sensitive language. Which mean that `TRUE` and `True` are not the same.

While the first one is a reserved word denoting a logical constant in R, the latter can be used a variable name.

```
> TRUE<-1
Error in TRUE <- 1 : invalid (do_set) left-hand side to assignment
> True<-1
> True
[1] 1
> TRUE
[1] TRUE
```

Variables in R

Variables are used to store data, whose value can be changed according to our need. Unique name given to variable (function and objects as well) is identifier.

Rules for writing Identifiers in R

1. Identifiers can be a combination of letters, digits, period (.) and underscore (_).
2. It must start with a letter or a period. If it starts with a period, it cannot be followed by a digit.
3. Reserved words in R cannot be used as identifiers.

Valid identifiers in R

`total`, `Sum`, `.fine.with.dot`, `this_is_acceptable`, `Number5`

Invalid identifiers in R

`tot@l`, `5um`, `_fine`, `TRUE`, `.0ne`

Constants in R

Constants, as the name suggests, are entities whose value cannot be altered. Basic types of constant are numeric constants and character constants.

Numeric Constants

All numbers fall under this category. They can be of type `integer`, `double` or `complex`.

It can be checked with the `typeof()` function.

Numeric constants followed by `L` are regarded as `integer` and those followed by `i` are regarded as `complex`.

```
> typeof(5)
[1] "double"
> typeof(5L)
[1] "integer"
> typeof(5i)
[1] "complex"
```

Numeric constants preceded by `0x` or `0X` are interpreted as hexadecimal numbers.

```
> 0xff
[1] 255
> 0XF+1
[1] 16
```

Character Constants

Character constants can be represented using either single quotes (') or double quotes (") as delimiters.

```
> 'example'
[1] "example"
> typeof("5")
[1] "character"
```

Built-in Constants

Some of the built-in constants defined in R along with their values is shown below.

```
> LETTERS
 [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L"
[13] "M" "N" "O" "P" "Q" "R" "S" "T" "U" "V" "W" "X"
[25] "Y" "Z"
> letters
 [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l"
[13] "m" "n" "o" "p" "q" "r" "s" "t" "u" "v" "w" "x"
[25] "y" "z"
> pi
[1] 3.141593
> month.name
 [1] "January"   "February"  "March"     "April"
 [5] "May"       "June"      "July"      "August"
 [9] "September" "October"   "November"  "December"
> month.abb
 [1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug"
 [9] "Sep" "Oct" "Nov" "Dec"
```

But it is not good to rely on these, as they are implemented as variables whose values can be changed.

```
> pi
[1] 3.141593
> pi<-56
> pi
[1] 56
```

R Operators

R has several operators to perform tasks including arithmetic, logical and bitwise operations.

R has many operators to carry out different mathematical and logical operations.

Operators in R can mainly be classified into the following categories.

Arithmetic operators
Relational operators
Logical operators
Assignment operators

R Arithmetic Operators

These operators are used to carry out mathematical operations like addition and multiplication. Here is a list of arithmetic operators available in R.

Arithmetic Operators in R

Operator	Description
+	Addition
–	Subtraction
*	Multiplication
/	Division

^	Exponent
%%	Modulus (Remainder from division)
%/%	Integer Division

```
> x<-5
> y<-16
> x+y
[1] 21
> x-y
[1] -11
> x*y
[1] 80
> y/x
[1] 3.2
> y%/%x
[1] 3
> y%%x
[1] 1
> y^x
[1] 1048576
```

R Relational Operators

Relational operators are used to compare between values. Here is a list of relational operators available in R.

Operator	Description
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to

```
> x<-5
> y<-16
> x<y
[1] TRUE
> x>y
[1] FALSE
> y>=20
[1] FALSE
> y==16
[1] TRUE
> x!=5
[1] FALSE
```

Operation on Vectors

The above mentioned operators work on vectors. The variables used above were in fact single element vectors.

We can use the function `c()` (as in concatenate) to make vectors in R.

```
> x<-c(2,8,3)
> y<-c(6,4,1)
> x+y
[1]  8 12  4
> x>y
[1] FALSE  TRUE  TRUE
```

When there is a mismatch in length (number of elements) of operand vectors, the elements in shorter one is recycled in a cyclic manner to match the length of the longer one.

R will issue a warning if the length of the longer vector is not an integral multiple of the shorter vector.

```
> x<-c(2,1,8,3)
> y <- c(9,4)
> x+y
[1] 11  5 17  7
> x+c(1,2,3)
[1]  3  3 11  4
Warning message:
In x + c(1, 2, 3) :
  longer object length is not a multiple of shorter object length
```

R Logical Operators

Logical operators are used to carry out Boolean operations like `AND`, `OR` etc.

Operator	Description
!	Logical NOT

&	Element-wise logical AND
&&	Logical AND
|	Element-wise logical OR
||	Logical OR

Operators `&` and `|` perform element-wise operation producing result having length of the longer operand.

But `&&` and `||` examines only the first element of the operands resulting into a single length logical vector.

Zero is considered `FALSE` and non-zero numbers are taken as `TRUE`

```
> x <- c(TRUE,FALSE,0,6)
> y <- c(FALSE,TRUE,FALSE,TRUE)
> !x
[1] FALSE  TRUE  TRUE FALSE
> x&y
[1] FALSE FALSE FALSE  TRUE
> x&&y
[1] FALSE
> x|y
[1]  TRUE  TRUE FALSE  TRUE
> x||y
[1] TRUE
```

R Assignment Operators

These operators are used to assign values to variables.

Assignment Operators in R

Operator	Description
<-, <<-, =	Leftwards assignment
->, ->>	Rightwards assignment

The operators `<-` and `=` can be used, almost interchangeably, to assign to vari able in the same environment.

The `<<-` operator is used for assigning to variables in the parent environments (more like global assignments). The rightward assignments, although availabl e are rarely used.

```
> x<-5
> x
[1] 5
> x=9
> x
[1] 9
> 10->x
> x
[1] 10
```

R Program to Take Input From User

When we are working with R in an interactive session, we can use `readline()` function to take input from the user (terminal).

This function will return a single element character vector.

So, if we want numbers, we need to do appropriate conversions.

For this you have to go to File >New file >R script

In R script Window

Write the following set of Script

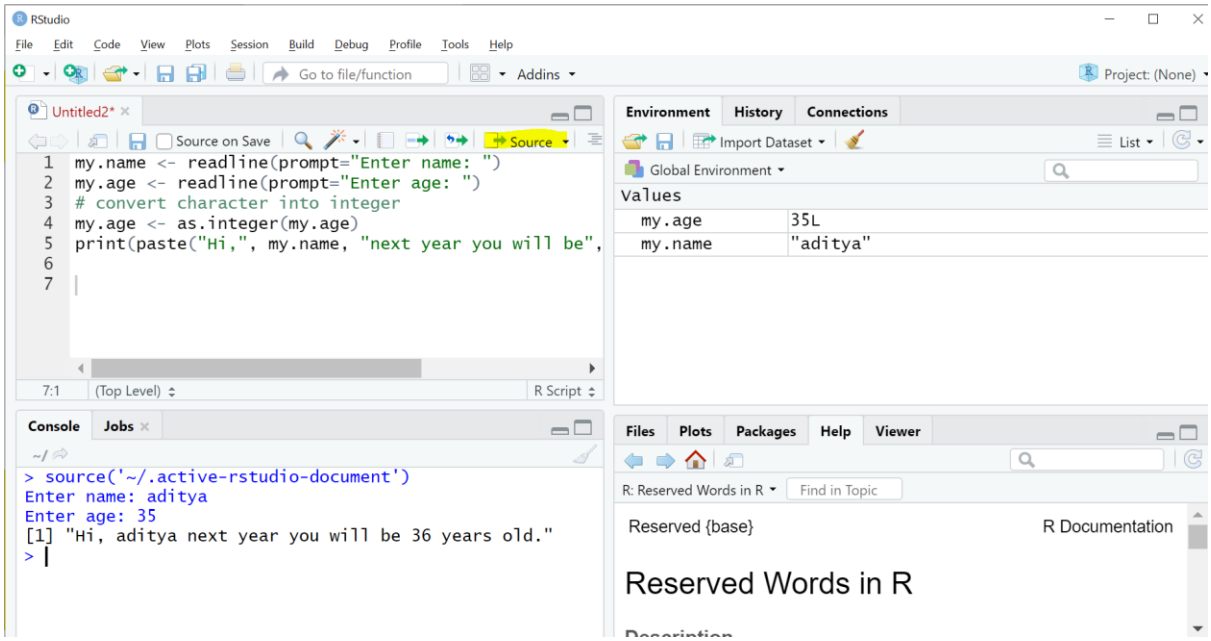
```
my.name <- readline(prompt="Enter name: ")
my.age <- readline(prompt="Enter age: ")
# convert character into integer
my.age <- as.integer(my.age)
print(paste("Hi,", my.name, "next year you will be", my.age+1, "years old."))
```

ctrl+shift+s source the contents of the active document

ctrl+shift+enter source the contents of the active document(with echo) click source button

for running specific like press ctrl+R in RGui

and ctrl+Enter in Rstudio


RStudio
File Edit Code View Plots Session Build Debug Profile Tools Help
Go to file/function
Addins
Project: (None)
Untitled2*
Source on Save
Source
1 my.name <- readline(prompt="Enter name: ")
2 my.age <- readline(prompt="Enter age: ")
3 # convert character into integer
4 my.age <- as.integer(my.age)
5 print(paste("Hi,", my.name, "next year you will be",
6
7
7:1
(Top Level)
R Script
Environment History Connections
Import Dataset
List
Global Environment
Values
my.age 35L
my.name "aditya"
Console Jobs
~/
> source('~/.active-rstudio-document')
Enter name: aditya
Enter age: 35
[1] "Hi, aditya next year you will be 36 years old."
>
Files Plots Packages Help Viewer
R: Reserved Words in R
Find in Topic
Reserved {base}
R Documentation
Reserved Words in R