

به نام خدا

معرفی زبان R

استاد: فرزین یغمایی

استاد یار: امیر شکری

دانشجو: محمد مهدی اسماعیل زاده

دانشگاه سمنان

پاییز ۱۴۰۰

1. مقدمه

1.1. عملگرهای R

R عملگرهای زیادی برای انجام عملیات ریاضی و منطقی مختلف دارد. اپراتورها در R عمدتاً به دسته های زیر طبقه بندی می شوند.

• عملگرهای محاسبه ای

این عملگرها برای انجام عملیات ریاضی مانند جمع و ضرب استفاده می شوند. در زیر لیستی از عملگرهای حسابی موجود در R آمده است.

Arithmetic Operators in R

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Exponent
%%	Modulus (Remainder from division)
%/%	Integer Division

نمونه کد:

```
> 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
```

- عملگرهای رابطه ای

عملگرهای رابطه ای برای مقایسه بین مقادیر استفاده می شوند. در زیر لیستی از عملگرهای رابطه ای موجود در R آمده است.

Relational Operators in R

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

نمونه کد:

```
> x <- 5
> y <- 16

> x<y
[1] TRUE

> x>y
[1] FALSE

> x<=5
[1] TRUE

> y>=20
[1] FALSE

> y == 16
[1] TRUE

> x != 5
[1] FALSE
```

• عملگرهای منطقی

عملگرهای منطقی برای انجام عملیات بولی مانند AND، OR و غیره استفاده می شوند.

Logical Operators in R	
Operator	Description
!	Logical NOT
&	Element-wise logical AND
&&	Logical AND
	Element-wise logical OR
	Logical OR

نمونه کد:

```
> 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
```

• عملگرهای واگذاری

این عملگرها برای تخصیص مقادیر به متغیرها استفاده می شوند.

Assignment Operators in R ▪

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

نمونه کد:

```
> x <- 5

> x
[1] 5

> x = 9

> x
[1] 9

> 10 -> x

> x
[1] 10
```

1.2. متغیر و ثابت ها

از متغیرها برای ذخیره داده ها استفاده می شود که مقدار آنها را می توان با توجه به نیاز ما تغییر داد. نام منحصر به فرد داده شده به متغیر (تابع و اشیاء نیز) شناسه است.

- شناسه های معتبر در R

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

- شناسه های نامعتبر در R

`tot@1`, `5um`, `_fine`, `TRUE`, `.0ne`

- ثابت ها در R

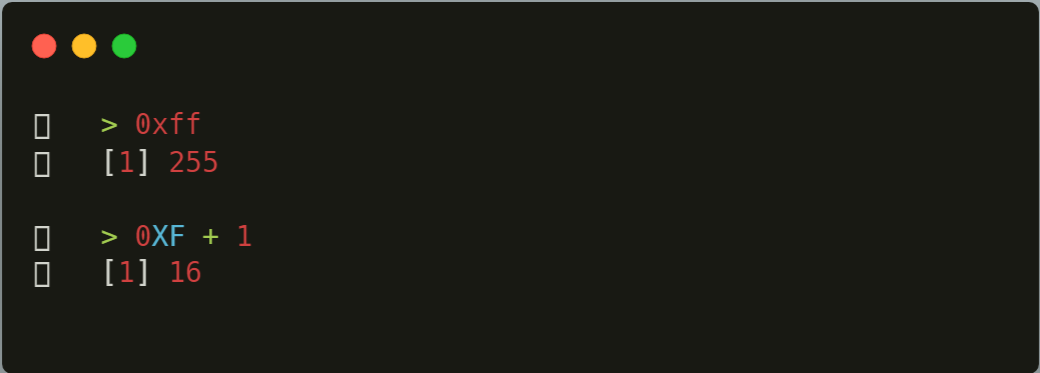
1.1.2. ثابت های عددی

```
> typeof(5)
[1] "double"

> typeof(5L)
[1] "integer"


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

ثابت های عددی که بعد از `x0` یا `X0` استفاده میشوند به عنوان اعداد هگزادسیمال تفسیر می شوند.



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

2..1.2. ثابت های کاراکتری



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


3.1.2. ثابت های داخلی



LETTERS

```
[1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S"  
[20] "T" "U" "V" "W" "X" "Y" "Z"
```

> letters

```
[1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"  
[20] "t" "u" "v" "w" "x" "y" "z"
```

> pi

```
[1] 3.141593
```

> month.name

```
[1] "January" "February" "March" "April" "May" "June"  
[7] "July" "August" "September" "October" "November" "December"
```

> month.abb

```
[1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"
```

1.3. اولویت عملگر ها

Operator Precedence in R

Operator	Description	Associativity
\wedge	Exponent	Right to Left
-x, +x	Unary minus, Unary plus	Left to Right
%%	Modulus	Left to Right

<code>*, /</code>	Multiplication, Division	Left to Right
<code>+, -</code>	Addition, Subtraction	Left to Right
<code><, >, <=, >=, ==, !=</code>	Comparisons	Left to Right
<code>!</code>	Logical NOT	Left to Right
<code>&, &&</code>	Logical AND	Left to Right
<code> , </code>	Logical OR	Left to Right
<code>->, ->></code>	Rightward assignment	Left to Right
<code><-, <<-</code>	Leftward assignment	Right to Left
<code>=</code>	Leftward assignment	Right to Left

1.4. کلمات رزرو شده در R

Reserved words in R

<code>if</code>	<code>else</code>	<code>repeat</code>	<code>while</code>	<code>function</code>
<code>for</code>	<code>in</code>	<code>next</code>	<code>break</code>	<code>TRUE</code>
<code>FALSE</code>	<code>NULL</code>	<code>Inf</code>	<code>NaN</code>	<code>NA</code>
<code>NA_integer_</code>	<code>NA_real_</code>	<code>NA_complex_</code>	<code>NA_character_</code>	<code>...</code>

2. حلقه و تصمیم

2.1. تابع های شرطی

- نحوه استفاده از if :

```
if (test_expression) {  
    statement  
}
```

- نحوه استفاده از if ...else :

```
if (test_expression) {  
    statement1  
} else {  
    statement2  
}
```

2.2. تابع شرطی ifelse()

- نحوه استفاده از ifelse():

```
ifelse(test_expression, x, y)
```

2.3. حلقه در R

- نحوه استفاده از for:

```
for (val in sequence)
{
  statement
}
```

2.4. حلقه while در R

- نحوه استفاده از while:

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

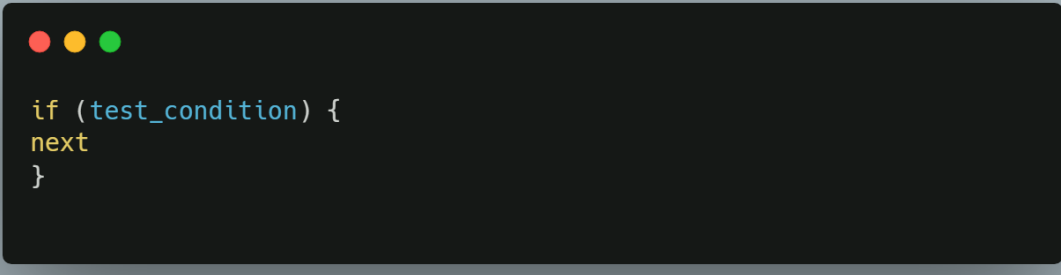
R break & next .2.5

The syntax of break statement is: •

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It displays the R break statement syntax.

```
if (test_expression) {  
  break  
}
```

The syntax of next statement is: •

A terminal window with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. It displays the R next statement syntax.

```
if (test_condition) {  
  next  
}
```

2.6. حلقه نامحدود در R

- استفاده از repeat :

```
repeat {  
  statement  
}
```


3. توابع R

3.1. توابع R

- نحوه تعریف تابع در R :

```
func_name <- function (argument) {  
  statement  
}
```

- نحوه صدا زدن تابع:



```
func_name(argument)
```

3.2. تابع return

- نحوه استفاده از تابع return :




```
return(expression)
```

3.3. تابع بازگشتی در R

3.4. تابع switch

- نحوه استفاده از تابع switch :



```
switch (expression, list)
```

4. ساختار داده ها در R

4.1. بردارها در R

بردار یک ساختار داده پایه در R است. شامل عناصری از همان نوع است. انواع داده ها می توانند منطقی، عدد صحیح، دوگانه، کاراکتر، پیچیده یا خام باشند.

نوع یک بردار را می توان با تابع `typeof()` بررسی کرد.

یکی دیگر از ویژگی های مهم یک بردار طول آن است. این تعداد عناصر در بردار با تابع `length()` قابل بررسی است.

از آنجایی که یک بردار باید عناصری از یک نوع داشته باشد، این تابع سعی می کند تا عناصر را به یک نوع وادار کند، اگر متفاوت باشند.

اجبار از انواع پایین تر به بالاتر از منطقی به عدد صحیح تا مضاعف به کاراکتر است.

نمونه کد:

```
> x <- c(1, 5, 4, 9, 0)
> typeof(x)
[1] "double"

> length(x)
[1] 5

> x <- c(1, 5.4, TRUE, "hello")

> x
[1] "1"      "5.4"    "TRUE"   "hello"

> typeof(x)
[1] "character"
```


اگر بخواهیم بردار اعداد متوالی ایجاد کنیم، عملگر **:** بسیار مفید است.

```
> x <- 1:7; x  
[1] 1 2 3 4 5 6 7  
  
> y <- 2:-2; y  
[1] 2 1 0 -1 -2
```

دنباله های پیچیده تری را می توان با استفاده از تابع **seq()** مانند تعیین تعداد نقاط در یک بازه یا اندازه گام ایجاد کرد.

```
> seq(1, 3, by=0.2)  
[1] 1.0 1.2 1.4 1.6 1.8 2.0 2.2 2.4 2.6 2.8 3.0  
  
> seq(1, 5, length.out=4)  
[1] 1.000000 2.333333 3.666667 5.000000
```

چگونه به عناصر یک بردار دسترسی پیدا کنیم؟
با استفاده از نمایه سازی برداری می توان به عناصر یک بردار دسترسی داشت. بردار مورد استفاده برای نمایه سازی می تواند منطقی، عدد صحیح یا بردار کاراکتری باشد.

4.2. لیست ها در R

لیست یک ساختار داده است که دارای اجزایی از انواع داده های مختلف است.

بردارهایی که همه عناصر از یک نوع را داشته باشند بردار اتمی و بردارهایی که عناصری از نوع متفاوت دارند فهرست نامیده می شود.

می توانیم بررسی کنیم که آیا لیستی با تابع `typeof()` است یا خیر و طول آن را با استفاده از `length()` پیدا کنیم. در اینجا نمونه ای از لیستی است که دارای سه جزء از نوع داده های مختلف است.

لیست را میتوان با تابع `list()` ساخت.

```
> x <- list("a" = 2.5, "b" = TRUE, "c" = 1:3)
```

4.3. عامل ها در R

فاکتور یک ساختار داده ای است که برای فیلدهایی استفاده می شود که فقط مقادیر از پیش تعریف شده و محدود (داده های طبقه بندی) را می گیرد. به عنوان مثال: یک فیلد داده مانند وضعیت تاهل ممکن است فقط حاوی مقادیر مجرد، متاهل، جدا شده، مطلقه یا بیوه باشد.

در چنین حالتی، ما مقادیر ممکن را از قبل می دانیم و این مقادیر از پیش تعریف شده و متمایز را سطوح می نامند. در زیر مثالی از فاکتور در R آورده شده است.

ما می توانیم با استفاده از تابع `factor()` یک فاکتور ایجاد کنیم. سطوح یک عامل در صورت عدم ارائه از داده ها استنتاج می شود.

```
> x <- factor(c("single", "married", "married", "single"));
> x

[1] single married married single
Levels: married single

> x <- factor(c("single", "married", "married", "single"), levels = c("single", "married",
"divorced"));

> x
[1] single married married single
Levels: single married divorced
```