class04.R

sbhwang

2022-02-04

```
# 2. Simple Calculations
5+3
```

```
## [1] 8
```

```
5-3
```

```
## [1] 2
```

```
5*3
```

```
## [1] 15
```

```
5/3
```

```
## [1] 1.666667
```

```
# 3. objectName <- value
x <- 3*4
x
```

```
## [1] 12
```

```
this_is_a_really_long_name <- 2.5
this_is_a_really_long_name
```

```
## [1] 2.5
```

```
r_rocks <- 2^3
# rrocks --> object name recognizes _
# R_rocks --> object name differentiates capital letters
r_rocks
```

```
## [1] 8
```

```
# 4. Calling Functions
seq(1, 10)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

```
seq(1, 10, by=2)
```

```
## [1] 1 3 5 7 9
```

```
# useful functions documentation: example()
example(seq)
```

```
##
## seq> seq(0, 1, length.out = 11)
##  [1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
##
## seq> seq(stats::rnorm(20)) # effectively ’along’
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20
##
## seq> seq(1, 9, by = 2)     # matches ’end’
## [1] 1 3 5 7 9
##
## seq> seq(1, 9, by = pi)    # stays below ’end’
## [1] 1.000000 4.141593 7.283185
##
## seq> seq(1, 6, by = 3)
## [1] 1 4
##
## seq> seq(1.575, 5.125, by = 0.05)
##  [1] 1.575 1.625 1.675 1.725 1.775 1.825 1.875 1.925 1.975 2.025 2.075 2.125
## [13] 2.175 2.225 2.275 2.325 2.375 2.425 2.475 2.525 2.575 2.625 2.675 2.725
## [25] 2.775 2.825 2.875 2.925 2.975 3.025 3.075 3.125 3.175 3.225 3.275 3.325
## [37] 3.375 3.425 3.475 3.525 3.575 3.625 3.675 3.725 3.775 3.825 3.875 3.925
## [49] 3.975 4.025 4.075 4.125 4.175 4.225 4.275 4.325 4.375 4.425 4.475 4.525
## [61] 4.575 4.625 4.675 4.725 4.775 4.825 4.875 4.925 4.975 5.025 5.075 5.125
##
## seq> seq(17) # same as 1:17, or even better seq_len(17)
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17
```

```
date()
```

```
## [1] "Fri Feb  4 11:08:54 2022"
```

```
# 5. Getting help in R
# log computes logarithms by default natural logarithms
help(log)
?log

# Used when you do not know the name of the function
help.search("cross tabulate")
??"cross tabulate"

example(log)
```

```
##
## log> log(exp(3))
## [1] 3
##
## log> log10(1e7) # = 7
## [1] 7
##
## log> x <- 10^-(1+2*1:9)
##
## log> cbind(x, log(1+x), log1p(x), exp(x)-1, expm1(x))
##            x
##  [1,] 1e-03 9.995003e-04 9.995003e-04 1.000500e-03 1.000500e-03
##  [2,] 1e-05 9.999950e-06 9.999950e-06 1.000005e-05 1.000005e-05
##  [3,] 1e-07 1.000000e-07 1.000000e-07 1.000000e-07 1.000000e-07
##  [4,] 1e-09 1.000000e-09 1.000000e-09 1.000000e-09 1.000000e-09
##  [5,] 1e-11 1.000000e-11 1.000000e-11 1.000000e-11 1.000000e-11
##  [6,] 1e-13 9.992007e-14 1.000000e-13 9.992007e-14 1.000000e-13
##  [7,] 1e-15 1.110223e-15 1.000000e-15 1.110223e-15 1.000000e-15
##  [8,] 1e-17 0.000000e+00 1.000000e-17 0.000000e+00 1.000000e-17
##  [9,] 1e-19 0.000000e+00 1.000000e-19 0.000000e+00 1.000000e-19
```

```r
# 6. Vectors, Vectorization, and Indexing
length(3.1)
```

```
## [1] 1
```

```r
length(1:4)
```

```
## [1] 4
```

```r
# Vectorization
x <- c(56, 95.3, 0.4)
x
```

```
## [1] 56.0 95.3  0.4
```

```r
y <- c(3.2, 1.1, 0.2)
y
```

```
## [1] 3.2 1.1 0.2
```

```r
x+y
```

```
## [1] 59.2 96.4  0.6
```

```r
x-y
```

```
## [1] 52.8 94.2  0.2
```

```r
x/y
```

```
## [1] 17.50000 86.63636  2.00000
```

```r
sqrt(x)
```

```
## [1] 7.4833148 9.7621719 0.6324555
```

```r
round(sqrt(x))
```

```
## [1]  7 10  1
```

```r
# round to 1 decimal point
round(sqrt(x), 1)
```

```
## [1] 7.5 9.8 0.6
```

```r
# round to 3 decimal point
round(sqrt(x), 3)
```

```
## [1] 7.483 9.762 0.632
```

```r
# Vector Indexing
x <- c(56, 95.3, 0.4)
x[2]
```

```
## [1] 95.3
```

```r
x[1]
```

```
## [1] 56
```

```r
x[4]
```

```
## [1] NA
```

```r
x[3] <- 0.5
x
```

```
## [1] 56.0 95.3  0.5
```