

# List exercises

In the exercises below we cover the basics of lists. Before proceeding, first read section 6.1-6.2 of [An Introduction to R](#), and the help pages for the `sum`, `length`, `strsplit`, and `setdiff` functions.

Answers to the exercises are available [here](#).



**Learn more** about lists in the online courses [Learn By Example: Statistics and Data Science in R](#) , [The Comprehensive Statistics and Data Science with R Course](#) and [R Programming: Advanced Analytics In R For Data Science](#)

## Exercise 1

If:

```
p <- c(2,7,8), q <- c("A", "B", "C") and
```

```
x <- list(p, q),
```

then what is the value of `x[2]`?

- a. NULL
- b. "A" "B" "C"
- c. "7"

## Exercise 2

If:

```
w <- c(2, 7, 8)
```

```
v <- c("A", "B", "C")
```

```
x <- list(w, v),
```

then which R statement will replace "A" in `x` with "K".

- a. `x[[2]] <- "K"`
- b. `x[[2]][1] <- "K"`
- c. `x[[1]][2] <- "K"`

## Exercise 3

If `a <- list ("x"=5, "y"=10, "z"=15)`, which R statement will give the sum of all elements in `a`?

- a. `sum(a)`

b. `sum(list(a))`  
c. `sum(unlist(a))`

#### **Exercise 4**

If `Newlist <- list(a=1:10, b="Good morning", c="Hi")`, write an R statement that will add 1 to each element of the first vector in `Newlist`.

#### **Exercise 5**

If `b <- list(a=1:10, c="Hello", d="AA")`, write an R expression that will give all elements, except the second, of the first vector of `b`.

#### **Exercise 6**

Let `x <- list(a=5:10, c="Hello", d="AA")`, write an R statement to add a new item `z = "NewItem"` to the list `x`.

#### **Exercise 7**

Consider `y <- list("a", "b", "c")`, write an R statement that will assign new names "one", "two" and "three" to the elements of `y`.

#### **Exercise 8**

If `x <- list(y=1:10, t="Hello", f="TT", r=5:20)`, write an R statement that will give the length of vector `r` of `x`.

#### **Exercise 9**

Let `string <- "Grand Opening"`, write an R statement to split this string into two and return the following output:

```
[[1]]  
[1] "Grand"
```

```
[[2]]  
[1] "Opening"
```

#### **Exercise 10**

Let:

```
y <- list("a", "b", "c") and
```

```
q <- list("A", "B", "C", "a", "b", "c").
```

Write an R statement that will return all elements of q that are not in y, with the following result:

```
[[1]]
```

```
[1] "A"
```

```
[[2]]
```

```
[1] "B"
```

```
[[3]]
```

```
[1] "C"
```

Want some extra practice with lists? Please take a look [here](#)