

---

```
1 # Demonstrates counting votes for 3 different candidates
2
3 mario <- as.integer(readline("Mario: "))
4 peach <- as.integer(readline("Peach: "))
5 bowser <- as.integer(readline("Bowser: "))
6
7 total <- sum(mario, peach, bowser)
8 cat("Total votes:", total)
```

---

```
1 # Demonstrates defining a function
2
3 get_votes <- function() {
4   votes <- as.integer(readline("Enter votes: "))
5   return(votes)
6 }
7
8 mario <- get_votes()
9 peach <- get_votes()
10 bowser <- get_votes()
11
12 total <- sum(mario, peach, bowser)
13 cat("Total votes:", total)
```

---

```
1 # Demonstrates R returning the last evaluated expression
2
3 get_votes <- function() {
4   votes <- as.integer(readline("Enter votes: "))
5 }
6
7 mario <- get_votes()
8 peach <- get_votes()
9 bowser <- get_votes()
10
11 total <- sum(mario, peach, bowser)
12 cat("Total votes:", total)
```

---

```
1 # Demonstrates defining a parameter
2
3 get_votes <- function(prompt) {
4   votes <- as.integer(readline(prompt))
5 }
6
7 mario <- get_votes("Mario: ")
8 peach <- get_votes("Peach: ")
9 bowser <- get_votes("Bowser: ")
10
11 total <- sum(mario, peach, bowser)
12 cat("Total votes:", total)
```

---

```
1 # Demonstrates defining a parameter with a default value
2
3 get_votes <- function(prompt = "Enter votes: ") {
4   votes <- as.integer(readline(prompt))
5 }
6
7 mario <- get_votes()
8 peach <- get_votes()
9 bowser <- get_votes()
10
11 total <- sum(mario, peach, bowser)
12 cat("Total votes:", total)
```

```
1 # Demonstrates overriding the default value of a parameter
2
3 get_votes <- function(prompt = "Enter votes: ") {
4   votes <- as.integer(readline(prompt))
5 }
6
7 mario <- get_votes("Mario: ")
8 peach <- get_votes("Peach: ")
9 bowser <- get_votes("Bowser: ")
10
11 total <- sum(mario, peach, bowser)
12 cat("Total votes:", total)
```

```
1 # Demonstrates exact argument matching
2
3 get_votes <- function(prompt = "Enter votes: ") {
4   votes <- as.integer(readline(prompt))
5 }
6
7 mario <- get_votes(prompt = "Mario: ")
8 peach <- get_votes(prompt = "Peach: ")
9 bowser <- get_votes(prompt = "Bowser: ")
10
11 total <- sum(mario, peach, bowser)
12 cat("Total votes:", total)
```

```
1 # Demonstrates anticipating invalid input
2
3 get_votes <- function(prompt = "Enter votes: ") {
4   votes <- as.integer(readline(prompt))
5   if (is.na(votes)) {
6     return(0)
7   } else {
8     return(votes)
9   }
10 }
11
12 mario <- get_votes("Mario: ")
13 peach <- get_votes("Peach: ")
14 bowser <- get_votes("Bowser: ")
15
16 total <- sum(mario, peach, bowser)
17 cat("Total votes:", total)
```



```
1 # Demonstrates ifelse as last evaluated expression
2
3 get_votes <- function(prompt = "Enter votes: ") {
4   votes <- as.integer(readline(prompt))
5   ifelse(is.na(votes), 0, votes)
6 }
7
8 mario <- get_votes("Mario: ")
9 peach <- get_votes("Peach: ")
10 bowser <- get_votes("Bowser: ")
11
12 total <- sum(mario, peach, bowser)
13 cat("Total votes:", total)
```

```
1 # Demonstrates suppressWarnings
2
3 get_votes <- function(prompt = "Enter votes: ") {
4   votes <- suppressWarnings(as.integer(readline(prompt)))
5   ifelse(is.na(votes), 0, votes)
6 }
7
8 mario <- get_votes("Mario: ")
9 peach <- get_votes("Peach: ")
10 bowser <- get_votes("Bowser: ")
11
12 total <- sum(mario, peach, bowser)
13 cat("Total votes:", total)
```

---

```
1 # Demonstrates a duck quacking 3 times
2
3 cat("quack!\n")
4 cat("quack!\n")
5 cat("quack!\n")
```

---

```
1 # Demonstrates duck quacking in an infinite loop
2
3 repeat {
4   cat("quack!\n")
5 }
```

```
1 # Demonstrates quacking 3 times with repeat
2
3 i <- 3
4 repeat {
5   cat("quack!\n")
6   i <- i - 1
7   if (i == 0) {
8     break
9   } else {
10    next
11  }
12 }
```

---

```
1 # Demonstrates removing extraneous next keyword
2
3 i <- 3
4 repeat {
5   cat("quack!\n")
6   i <- i - 1
7   if (i == 0) {
8     break
9   }
10 }
```

---

```
1 # Demonstrates a while loop, counting down
2
3 i <- 3
4 while (i != 0) {
5   cat("quack!\n")
6   i <- i - 1
7 }
```

---

```
1 # Demonstrates a while loop, counting up
2
3 i <- 1
4 while (i <= 3) {
5   cat("quack!\n")
6   i <- i + 1
7 }
```



---

```
1 # Demonstrates a for loop
2
3 for (i in c(1, 2, 3)) {
4   cat("quack!\n")
5 }
```

---

```
1 # Demonstrates a for loop with syntactic sugar
2
3 for (i in 1:3) {
4   cat("quack!\n")
5 }
```

```
1 # Demonstrates reprompting the user for valid input
2
3 get_votes <- function(prompt = "Enter votes: ") {
4   repeat {
5     votes <- suppressWarnings(as.integer(readline(prompt)))
6     if (!is.na(votes)) {
7       break
8     }
9   }
10  return(votes)
11 }
12
13 mario <- get_votes("Mario: ")
14 peach <- get_votes("Peach: ")
15 bowser <- get_votes("Bowser: ")
16
17 total <- sum(mario, peach, bowser)
18 cat("Total votes:", total)
```

```
1 # Demonstrates tightening return
2
3 get_votes <- function(prompt = "Enter votes: ") {
4   repeat {
5     votes <- suppressWarnings(as.integer(readline(prompt)))
6     if (!is.na(votes)) {
7       return(votes)
8     }
9   }
10 }
11
12 mario <- get_votes("Mario: ")
13 peach <- get_votes("Peach: ")
14 bowser <- get_votes("Bowser: ")
15
16 total <- sum(mario, peach, bowser)
17 cat("Total votes:", total)
```

```
1 # Demonstrates prompting for input in a loop
2
3 get_votes <- function(prompt = "Enter votes: ") {
4   repeat {
5     votes <- suppressWarnings(as.integer(readline(prompt)))
6     if (!is.na(votes)) {
7       return(votes)
8     }
9   }
10 }
11
12 for (name in c("Mario", "Peach", "Bowser")) {
13   votes <- get_votes(paste0(name, ": "))
14 }
```

```
1 # Demonstrates prompting for input, tallying votes in a loop
2
3 get_votes <- function(prompt = "Enter votes: ") {
4   repeat {
5     votes <- suppressWarnings(as.integer(readline(prompt)))
6     if (!is.na(votes)) {
7       return(votes)
8     }
9   }
10 }
11
12 total <- 0
13 for (name in c("Mario", "Peach", "Bowser")) {
14   votes <- get_votes(paste0(name, ": "))
15   total <- total + votes
16 }
17
18 cat("Total votes:", total)
```

---

```
1 # Demonstrates summing votes for each candidate procedurally
2
3 votes <- read.csv("votes.csv")
4
5 total_votes <- c()
6 for (candidate in rownames(votes)) {
7   total_votes[candidate] <- sum(votes[candidate, ])
8 }
9 total_votes
```

---

```
1 # Demonstrates summing votes for each voting method procedurally
2
3 votes <- read.csv("votes.csv")
4
5 total_votes <- c()
6 for (method in colnames(votes)) {
7   total_votes[method] <- sum(votes[, method])
8 }
9 total_votes
```



---

```
1 # Demonstrates summing votes for each candidate with apply
2
3 votes <- read.csv("votes.csv")
4 total_votes <- apply(votes, MARGIN = 1, FUN = sum)
5 total_votes
```

---

```
1 # Demonstrates summing votes for each voting method with apply
2
3 votes <- read.csv("votes.csv")
4 total_votes <- apply(votes, MARGIN = 2, FUN = sum)
5 total_votes
```