

---

```
1 # Demonstrates initializing a package
2
3 # Load devtools, tidyverse for package creation
4 library(devtools)
5 library(tidyverse)
6
7 # Create a new folder for package
8 dir.create("ducksay")
9
10 # Set working directory to package directory
11 setwd("ducksay")
12
13 # Create a blank DESCRIPTION file
14 file.create("DESCRIPTION")
15
16 # Create a LICENSE file to fill in license details
17 file.create("LICENSE")
```

---

```
1 # Demonstrates required components of a DESCRIPTION file
2
3 Package: ducksay
4 Title: Duck Say
5 Description: Say hello with a duck.
6 Version: 1.0
7 Authors@R: person("Carter", "Zenke", email = "carter@cs50.harvard.edu", role = c("aut", "cre", "cph"))
8 License: MIT + file LICENSE
```

---

```
1 # Demonstrates adding on to a license template
2
3 YEAR: ...
4 COPYRIGHT HOLDER: ducksay authors
```

---

```
1 # Demonstrates creating tests
2
3 # Initialize test folder structure
4 use_testthat()
5
6 # Create testing file for `ducksay`
7 use_test("ducksay")
```

---

```
1 # Demonstrates suggesting a dependency, for testing's sake
2
3 Package: ducksay
4 Title: Duck Say
5 Description: Say hello with a duck.
6 Version: 1.0
7 Authors@R: person("Carter", "Zenke", email = "carter@cs50.harvard.edu", role = c("aut", "cre", "cph"))
8 License: MIT + file LICENSE
9 Suggests:
10     testthat (>= 3.0.0)
11 Config/testthat/edition: 3
```

```
1 # Demonstrates describing behavior of `ducksay`
2
3 describe("ducksay()", {
4   it("can print to the console with `cat`", {
5     expect_output(cat(ducksay()))
6   })
7   it("can say hello to the world", {
8     expect_match(ducksay(), "hello, world")
9   })
10 })
```

---

```
1 # Demonstrates creating a function
2
3 # Create `ducksay` file, in which to write function
4 use_r("ducksay")
```

```
1 # Demonstrates defining a function for a package
2
3 ducksay <- function() {
4   paste(
5     "hello, world",
6     ">(.)_",
7     "(____/",
8     sep = "\n"
9   )
10 }
```



---

```
1 # Demonstrates exporting and loading a package function
2
3 # Create a NAMESPACE file, in which to export ducksay function
4 file.create("NAMESPACE")
5
6 # Load function definitions for testing
7 load_all()
```

```
1 # Demonstrates declaring `ducksay` accessible to package end users
2
3 export(ducksay)
```

---

```
1 # Demonstrates running tests
2
3 test()
```

```
1 # Demonstrates checking for duck in output
2
3 describe("ducksay()", {
4   it("can print to the console with `cat`", {
5     expect_output(cat(ducksay()))
6   })
7   it("can say hello to the world", {
8     expect_match(ducksay(), "hello, world")
9   })
10  it("can say hello with a duck", {
11    duck <- paste(
12      ">(.)_",
13      "(___)/",
14      sep = "\n"
15    )
16    expect_match(ducksay(), duck, fixed = TRUE)
17  })
18 })
```

---

```
1 # Demonstrates running tests
2
3 test()
```

---

```
1 # Demonstrates documenting a function
2
3 # Create a directory for documentation
4 dir.create("man")
5
6 # Create a file in which to write documentation for ducksay
7 file.create("man/ducksay.Rd")
```

---

```
1 # Demonstrates required markup for R documentation files
2
3 \name{ducksay}
4 \alias{ducksay}
5 \title{Duck Say}
6 \description{A duck that says hello.}
7 \usage{
8 ducksay()
9 }
10 \value{
11 A string representation of a duck saying hello to the world.
12 }
13 \examples{
14 cat(ducksay())
15 }
```

---

```
1 # Demonstrates rendering documentation
2
3 ?ducksay
```



---

```
1 # Demonstrates "building" a package—converting it from source code to a bundled, shareable file
2
3 build()
```

```
1 # Demonstrates ensuring duck repeats given phrase
2
3 describe("ducksay()", {
4   it("can print to the console with `cat`", {
5     expect_output(cat(ducksay()))
6   })
7   it("can say hello to the world", {
8     expect_match(ducksay(), "hello, world")
9   })
10  it("can say hello with a duck", {
11    duck <- paste(
12      ">(.)_",
13      "(____)/",
14      sep = "\n"
15    )
16    expect_match(ducksay(), duck, fixed = TRUE)
17  })
18  it("can say any given phrase", {
19    expect_match(ducksay("quack!"), "quack!")
20  })
21 })
```

```
1 # Demonstrates taking an argument to print
2
3 ducksay <- function(phrase = "hello, world") {
4   paste(
5     phrase,
6     ">(.)_",
7     "(____)",
8     sep = "\n"
9   )
10 }
```

```
1 # Demonstrates updated markup, including specifying arguments
2
3 \name{ducksay}
4 \alias{ducksay}
5 \title{Duck Say}
6 \description{A duck that says hello.}
7 \usage{
8 ducksay(phrase = "hello, world")
9 }
10 \arguments{
11 \item{phrase}{The phrase for the duck to say.}
12 }
13 \value{
14 A string representation of a duck saying the given phrase.
15 }
16 \examples{
17 cat(ducksay())
18 cat(ducksay("quack!"))
19 }
```

---

```
1 # Demonstrates "building" a package—converting it from source code to a bundled, shareable file
2
3 build()
```

---

```
1 # Demonstrates installing a custom package
2
3 # Installs package from source, if available
4 install("ducksay")
5
6 # Installs package from bundled file, if available
7 install.packages("ducksay_1.0.tar.gz")
```

---

```
1 # Demonstrates using custom package
2
3 library(ducksay)
4
5 name <- readline("What's your name? ")
6 greeting <- ducksay(paste("hello,", name))
7 cat(greeting)
```