Week 2 Quiz

10/10 questions correct

Quiz passed!

Continue Course (/learn/r-programming/supplement/amLgW/programming-assignment-1-instructions-air-pollution)

Back to Week 2 (/learn/r-programming/home/week/2)

1.

Suppose I define the following function in R

```
cube <- function(x, n) {
        x^3
}
```

What is the result of running

```
cube(3)
```

in R after defining this function?

- The number 27 is returned

Well done!
Because 'n' is not evaluated, it is not needed even though it is a formal argument.

- A warning is given with no value returned.
- The users is prompted to specify the value of 'n'.
- An error is returned because 'n' is not specified in the call to 'cube'

2.

The following code will produce a warning in R.

```
x <- 1:10
if(x > 5) {
        x <- 0
}
```

Why?

- The expression uses curly braces.
- The syntax of this R expression is incorrect.
- You cannot set 'x' to be 0 because 'x' is a vector and 0 is a scalar.
- There are no elements in 'x' that are greater than 5

'x' is a vector of length 10 and 'if' can only test a single logical statement.

Well done!

3.

Consider the following function

```
f <- function(x) {
        g <- function(y) {
                y + z
        }
        z <- 4
        x + g(x)
}
```

If I then run in R

```
z <- 10
f(3)
```

What value is returned?

10

Well done!

7

16

4

4.

Consider the following expression:

```
x <- 5
y <- if(x < 3) {
        NA
} else {
        10
}
```

What is the value of 'y' after evaluating this expression?

NA

5

10

Well done!

3

5.

Consider the following R function

```
h <- function(x, y = NULL, d = 3L) {
        z <- cbind(x, d)
        if(!is.null(y))
                z <- z + y
        else
                z <- z + f
        g <- x + y / z
        if(d == 3L)
                return(g)
        g <- g + 10
        g
}
```

Which symbol in the above function is a free variable?

- f

Well done!

- z
- d
- L
- g

✔ 6.

What is an environment in R?

- an R package that only contains data
- a list whose elements are all functions
- a collection of symbol/value pairs

Well done!

- a special type of function

✔ 7.

The R language uses what type of scoping rule for resolving free variables?

- dynamic scoping
- lexical scoping

Well done!

- global scoping
- compilation scoping

✔ 8.

How are free variables in R functions resolved?

- The values of free variables are searched for in the environment in which the function was called

- The values of free variables are searched for in the global environment
- The values of free variables are searched for in the working directory
- The values of free variables are searched for in the environment in which the function was defined

Well done!

9.

What is one of the consequences of the scoping rules used in R?

- R objects cannot be larger than 100 MB
- All objects can be stored on the disk
- Functions cannot be nested
- All objects must be stored in memory

Well done!

10.

In R, what is the parent frame?

- It is always the global environment
- It is the environment in which a function was defined
- It is the package search list
- It is the environment in which a function was called

Well done!
