

```
Console ~/
| You are amazing!

===== | 95%

| Finally, let's pretend you'd like to view the contents of a variable that you created e
arlier, but you
| can't seem to remember if you named it my_div or myDiv. You could try both and see what
works, or...

...

===== | 97%

| You can type the first two letters of the variable name, then hit the Tab key (possibly
more than once).
| Most programming environments will provide a list of variables that you've created that
begin with 'my'.
| This is called auto-completion and can be quite handy when you have many variables in y
our workspace. Give
| it a try. (If auto-completion doesn't work for you, just type my_div and press Enter.)

> my_div
[1] 3.478505 3.181981 .146460

| Perseverance, that's the answer.

===== | 100%
```

```
Console ~/
| Use unlink("testdir", recursive = TRUE).

> unlink("testdir", recursive = TRUE)

| You got it!

===== | 95%

| Take nothing but results. Leave nothing but assumptions. That sounds like 'Take nothing but pictures. Leave
| nothing but footprints.' But it makes no sense! Surely our readers can come up with a better motto . . .

...

===== | 98%

| In this lesson, you learned how to examine your R workspace and work with the file system of your machine from
| within R. Thanks for playing!

...

===== | 100%

| Would you like to receive credit for completing this course on Coursera.org?

1: Yes
2: No

Selection: |
```

```
Console ~/ |
| If instead we want our vector to contain 10 repetitions of the vector (0, 1, 2), we can do rep(c(0, 1, 2),
| times = 10). Go ahead.
> rep(c(0,1,2), times=10)
[1] 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2 0 1 2
| You are doing so well!
|=====| 96%
| Finally, let's say that rather than repeating the vector (0, 1, 2) over and over again, we want our vector to
| contain 10 zeros, then 10 ones, then 10 twos. We can do this with the 'each' argument. Try rep(c(0, 1, 2),
| each = 10).
> rep(c(0,1,2), each=10)
[1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2
| You are amazing!
|=====| 100%
| Would you like to receive credit for completing this course on Coursera.org?
1: Yes
2: No
Selection: |
```

```
Console ~/ |
| Since the character vector LETTERS is longer than the numeric vector 1:4, R simply recycles, or repeats, 1:4
| until it matches the length of LETTERS.
...
|=====| 95%
| Also worth noting is that the numeric vector 1:4 gets 'coerced' into a character vector by the paste()
| function.
...
|=====| 97%
| We'll discuss coercion in another lesson, but all it really means is that the numbers 1, 2, 3, and 4 in the
| output above are no longer numbers to R, but rather characters "1", "2", "3", and "4".
...
|=====| 100%
| Would you like to receive credit for completing this course on Coursera.org?
1: Yes
2: No
Selection: |
```

```
Console ~/1 ↵

| You nailed it! Good job!

|=====| 90%

| Now that we've got NAs down pat, let's look at a second type of missing value -- NaN, which stands for 'not a
| number'. To generate NaN, try dividing (using a forward slash) 0 by 0 now..

> 0/0
[1] NaN

| Your dedication is inspiring!

|=====| 95%

| Let's do one more, just for fun. In R, Inf stands for infinity. What happens if you subtract Inf from Inf?

> Inf-Inf
[1] NaN

| Keep up the great work!

|=====| 100%

| Would you like to receive credit for completing this course on Coursera.org?

1: No
2: Yes

Selection:
```

```
Console ~/1 ↵

bar
2

| Great job!

|=====| 95%

| Likewise, we can specify a vector of names with vect[c("foo", "bar")]. Try it out.

> vect[c("foo", "bar")]
foo bar
11 2

| You are amazing!

|=====| 97%

| Now you know all four methods of subsetting data from vectors. Different approaches are best in different
| scenarios and when in doubt, try it out!

...

|=====| 100%

| Would you like to receive credit for completing this course on Coursera.org?

1: Yes
2: No

Selection: |
```