003-Control-statements

March 18, 2024

1 TP 03 - R Control Statements - 3/4

- Written by Alexandre Gazagnes
- Last update: 2024-02-01
- Based on https://www.w3schools.com/r/default.asp

1.1 If Statement

[ ]:
```r
# Basic if / else
a <- 200
b <- 33

if (b > a) { # if b is greater than a
  print ("b is greater than a")
} else { # if b is not greater than a
  print("b is not greater than a")
}
```

[ ]:
```r
# Nested if else
x <- 41
if (x > 10) {
  print("Above ten")
  if (x > 20) { # and if inside the first if
    print("and also above 20!")
  } else {
    print("but not above 20.")
  }
} else {
  print("below 10.")
}
```

[ ]:
```r
# Using if + and
a <- 200
b <- 33
c <- 500

if (a > b & c > a) { # both conditions are true
  print("Both conditions are true")
```

```
}
```

[ ]:
```
# Using if + or
a <- 200
b <- 33
c <- 500

if (a > b | a > c) { # at least one condition is true
  print("At least one of the conditions is true")
}
```

[ ]:
```
# Using in operator
fruits <- c("apple", "banana", "cherry")
"banana" %in% fruits # TRUE
```

1.2 While loops

[ ]:
```
# Basic while
i <- 1
while (i < 6) {
  print(i)
  i <- i + 1 # increment
}
```

[ ]:
```
# break
i <- 1
while (i < 6) {
  print(i)
  i <- i + 1
  if (i == 4) {
    break # stop the loop
  }
}
```

[ ]:
```
# next
i <- 0
while (i < 6) {
  i <- i + 1
  if (i == 3) {
    next # skip the loop
  }
  print(i)
}
```

1.3 For loops

[ ]:
```r
# Basic for
fruits <- list("apple", "banana", "cherry")

for (x in fruits) { # for each item in the list
  print(x)
}
```

[ ]:
```r
# Using for + range
for (x in 1:10) { # using a range
  print(x)
}
```

[ ]:
```r
# Using for + beak
fruits <- list("apple", "banana", "cherry")
for (x in fruits) {
  if (x == "cherry") {
    break # stop the loop
  }
  print(x)
}
```

[ ]:
```r
# Nested for
adj <- list("red", "big", "tasty")
fruits <- list("apple", "banana", "cherry")
for (x in adj) {
  for (y in fruits) {
    print(paste(x, y))
  }
}
```

[ ]:
```r
# for next
fruits <- list("apple", "banana", "cherry")
for (x in fruits) {
  if (x == "banana") {
    next # skip the loop
  }
  print(x)
}
```

1.4 Functions

[ ]:
```r
# Basic function
my_function <- function() { # create a function with the name my_function
  print("Hello World!")
}
```

[ ]:
```r
# Using arguments
my_function <- function(fname) { #Here, fname is the argument
  paste(fname, "Griffin")
}

my_function("Peter")
```

[ ]:
```r
# will you get an error?
my_function()
```

[ ]:
```r
# Default arguments
my_function <- function(country = "Norway") {
  paste("I am from", country)
}

my_function("Sweden")
```

[ ]:
```r
# will you get an error?
my_function()
```

[ ]:
```r
# Using return statement
my_function <- function(x) {
  return (5 * x)
}

print(my_function(3))
```

[ ]:
```r
# Thanks to the return statement, the function will return the value 15.
y = 12 + my_function(3)
y
```