

## Models and Simulations: Problem Set 1

**Instructions:** Open Netlogo. In the interface, create a button called “setup.” Then click on the code tab and begin the following exercises. To run the program setup, click on the Interface tab, and click your setup button.

### Exercise 1: Working with strings

1. Using the WORD function, finish the following program by defining a variable called `concatenaion` that is equal to the string “Superman is awesome.” Also, print the variable `concatenaion` in the command center.

```
to setup
  let s "Superman"
  let i "is "
  let a "awesome."
  ...
end
```

**Output:**

*Superman is awesome.*

2. Using the ITEM and WORD functions, finish the following program by defining a variable called `ears` that is the string “ears.” Print `ears` to the command center.

```
to setup
  let s "superman"
  print item 3 s
  ...
end
```

**Output:**

*e*  
*ears*

3. Find three functions in the Netlogo dictionary section on strings so that the output of the program below is as the output that follows it:

```
to setup
  let mystring "Superman"
  print [1st-function-name] mystring
  print [2nd-function-name] "e" mystring
  print [2nd-function-name] "f" mystring
  print [3rd-function-name] 1 mystring "oa"
  ...
end
```

**Output:**

```
8
true
false
Soaperman
```

4. Predict what will happen when you run the following program; write your prediction down. Then execute the program in Netlogo. What happens? Explain what happens if your prediction differed from what occurred.

```
to setup
  print item 8 "Superman"
end
```

**Exercise 2: Arithmetic Operations**

In the Interface, create two different inputs. For the first, enter X as the global variable and select “Number” from the dropdown list called “Type.” For the second, enter Y as the global variable and again select “Number” as the type.

1. Using the functions +, ^, MOD, FLOOR and CEILING, write a setup program that prints  $x + y$ ,  $x^y$ ,  $x \bmod y$ ,  $\frac{y}{x}$ ,  $\frac{y}{x}$  rounded down, and  $\frac{y}{x}$  rounded up. For example, if you enter 2 for  $x$  and 7 for  $y$ , the output of your program should be

**Output:**

```
The sum of 2 and 7 is 9.
2 to the 7th power is 128.
```

*7 mod 2 is 1.*  
*2 mod 7 is 2.*  
*7/2 is 3.5.*  
*7/2 rounded down is 3.*  
*7/2 rounded up is 4.*

2. Pick some mathematical function from the “Mathematical” section of the Netlogo dictionary and output the result of that function when applied to x, y, or both x and y. For example, if you were to pick sin and apply it to x, your program should output:

**Output:**

*The sum of 2 and 7 is 9.*  
*2 to the 7th power is 128.*  
*7 mod 2 is 1.*  
*2 mod 7 is 2.*  
*7/2 is 3.5.*  
*7/2 rounded down is 3.*  
*7/2 rounded up is 4.*  
*The sin of 2 is 0.03489949670250097*

**Exercise 3: Working with Boolean Variables**

1. Predict what the output of the following program is. Write your prediction down and explain it. Then execute the program in NetLogo. What happens? If your prediction was not correct, explain what you think the program is doing.

```
to setup
  print 5 < 3
end
```

2. In the Interface, create two number inputs x and y just like you did in Exercise 2. Next, create a third input z and select “String” from the dropdown menu. Using functions from previous exercises and the functions and, or, and not, Write a program with the following output when a user enters 2 for x, 3 for y, and “cat” for z.

**Output:**

*It is false that 2 > 3.*  
*It is true that the word “cat” contains at least 3 letters.*  
*It is false that both 2 < 3 and “cat” contains the letter “b”.*

*It is true that either  $2 < 3$  or "cat" contains the letter "b".*

**Exercise 4: Working With Lists** In each of the following, you will employ some function on lists to complete the program below:

to setup

```
let mylist n-values 10 [?]
```

```
...
```

end

1. Using the `fput` and `lput` functions, add -1 to the front of `mylist` and 10 to the end of `mylist`. Print `mylist` to the command center. The output of your program ought to be:

**Output:**

```
[-1 0 1 2 3 4 5 6 7 8 9 10]
```

2. Using `n-values`, change `mylist` so that each of its elements is doubled. So `mylist` ought to be the list below. Print `mylist` to the command center.

**Output:**

```
[-1 0 1 2 3 4 5 6 7 8 9 10]
```

```
[-2 0 2 4 6 8 10 12 14 16 18 20]
```

3. Using the `filter` and `mod` functions, change `mylist` so that it contains only multiples of three. Print `mylist` to the command center. The output of your program ought to be as follows:

**Output:**

```
[-1 0 1 2 3 4 5 6 7 8 9 10]
```

```
[-2 0 2 4 6 8 10 12 14 16 18 20]
```

```
[0 6 12 18]
```

4. Above, you have been working with lists of numbers. In Netlogo, the elements of a list can be any data type. That is, you can have a list of numbers, booleans, strings ... and even lists themselves! Using the `n-values` function and only one line of code, create and print a list of length 5 such that element  $i$  of the list is the first  $i$  whole numbers. In other words, create and print the following list:

**Output:**

```
[ [] [0] [0 1] [0 1 2] [0 1 2 3] ]
```

5. Consider the list of lists you created in the previous exercise. Write a one-line program that prints the second item in the third list. That is, the third list is `[0 1 2]`, and the second item in this list is 1.