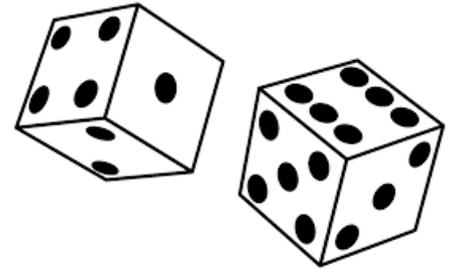
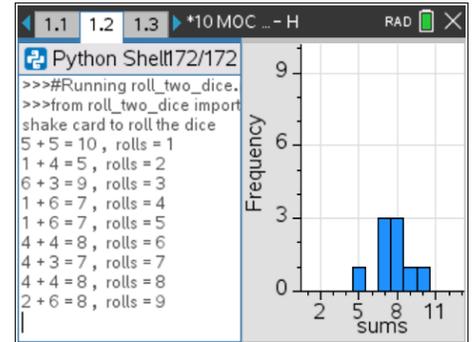


In this activity you will write a python program to collect data using the micro:bit and run the program while observing a dot plot grow on a split page of the TI-Nspire CX II.

1. This activity is a compilation of all the micro:bit skills you learned in the past three Skill Builders: write a program that uses a gesture, like 'shake' (or a button press) to collect some data, store the list as a TI-Nspire variable and...



2. ... then set up a TI-Nspire page that
 - runs the python program on one side of the screen (the python Shell) and
 - displays a dot plot (or histogram) of the collected data *as you are running the program* (using a Data & Statistics app).



3. Begin your micro:bit program with the usual imports including the **random** module and start with an empty list called **sums**:

sums = []

Immediately store this list to a TI-Nspire variable (using the same name).

store_list("sums", sums)

so that the TI-Nspire list is cleared as well.

print() some instructions to the user before the loop begins. We are going to use the 'shake' gesture to roll the dice.

```

1.1 1.2 1.3 ▶ *10 MOC ... - H RAD [ ] X
Python Shell 72/172
>>>#Running roll_two_dice.
>>>from roll_two_dice import
shake card to roll the dice
5 + 5 = 10 , rolls = 1
1 + 4 = 5 , rolls = 2
6 + 3 = 9 , rolls = 3
1 + 6 = 7 , rolls = 4
1 + 6 = 7 , rolls = 5
4 + 4 = 8 , rolls = 6
4 + 3 = 7 , rolls = 7
4 + 4 = 8 , rolls = 8
2 + 6 = 8 , rolls = 9
    
```

4. In the **while** loop body, use the gesture to
 - **toss** two dice (generate two random integers)
 - **add** them together
 - **append** the sum to the **sums** list
 - **print** the two dice values, their sum and the roll number on the TI-Nspire screen. Hint: **len(sums)** is the roll number.
 - **display** both die values on the micro:bit
 - **store** the **list** to a TI-Nspire variable

Try it now.

```

1.1 1.2 1.3 ▶ *10 MOC ... - H RAD [ ] X
*roll_two_dice.py 7/30
from random import *
from microbit import *

sums=[]
store_list("sums",sums)
print("shake card to roll the dice")

while get_key() != "esc":
    
```

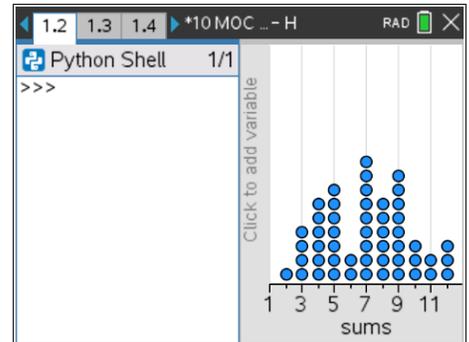



10 Minutes of Code: Python Modules

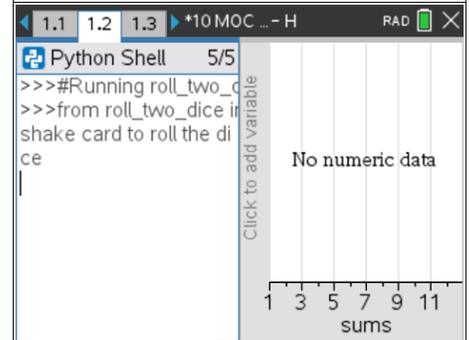
TI-NSPIRE CX II

MICRO:BIT: PAIR-O-DICE

12. Go back one page to the python Shell app (**[ctrl] [leftarrow]**) and press **[ctrl] [4]** to 'group' this app with the Data and Statistics app, creating a split-screen page with your python Shell on the left and your Data & Statistics app on the right as shown here.



13. The Shell has been 're-initialized' so pressing **[ctrl] [R]** will not re-run the program. Go back to the python Editor and press **[ctrl] [R]** to run the program. It runs in the half-screen Shell as shown here. You see 'No numeric data' on the right because the program stores an empty list right away.



As you collect the data (shake the micro:bit to roll the dice) your **sums** values appear as dots in the Data & Statistics app on the right.

Pressing **[esc]** will end the program and you can do a lot of other 1-variable data analysis in the TI-Nspire environment.

Pressing **[ctrl] [R]** again now (in the python Shell) does re-run the program.

Tip: to clear the Shell at the start of each run add the statement:

clear_history()

found on [menu] > More Modules > BBC micro:bit > Commands at the beginning of your program.

Enjoy, and remember to save your document!