Day 5

Accepting input

You may want to be able to input text into your program.

For that, use the `input` function:

```
animal = input("What is your favorite animal? ")
print("You said: " + animal)
```

Parsing Input

To go from a string to a value like a number, you need to **parse** it.

Integer `int("123")`
Float `float("12.3")`

Guessing Game

```
from random import randint

target = randint(1, 100)
guess = 0
while ???:
  guess = int(input("Enter your guess: "))
  if ???:
    ???
  else:
    ???
```

Guessing Game Improvements

- Can you tell the user if they are too high or too low?
- Can you let the user pick the range of numbers?
- Can you have the sense HAT light up if they are wrong or right?

Recursive Functions

A function can call itself using its name.

```
def countdown(number):
  if number == 0:
    print("All done!")
  else:
    print(number)
    countdown(number - 1)

countdown(10)
```

Fibonacci

The **Fibonacci sequence** is a sequence of numbers where each number is the sum of the two proceeding ones. It starts with `0, 1`:

$$0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$$

The terms are numbered starting from zero, so $F_0 = 0$, $F_1 = 1$, $F_2 = 1$, $F_3 = 2$, and so on.

Fibonacci Function

Let's write a function that takes n and calculates F_n.

A simple algorithm for a function `fib(n)` is:

1. Check if n is zero. If so, return zero.
1. Check if n is one. If so, return one.
1. Return `fib(n - 1) + fib(n - 2)`

fib(n)

```
def fib(n):
  if n == 0:
    return 0
  if n == 1:
    return 1
  return fib(n - 1) + fib(n - 2)

for i in range(0, 10):
  print(fib(i))
```

Using the Plotter

Mu has a built-in plotter that lets you visualize data.

Here is a helper function to display a data point:

```
def plot(n):
  print((n,))
```

Plotter

Try it out:

```
for i in range(0, 100):
  plot(i * i)
```

Collatz Conjecture

For a given number n, define a sequence by repeatedly applying the following:

- If the number is 1, stop
- If the number is even, divide it by two
- If the number is odd, multiply it by three and add 1

It seems like this always ends up at 1 for any number n, but nobody has been able to prove that it always will.

```
def collatz(n):
  while n != 1:
    sleep(0.5) # To better see the plot
    plot(n)
    if n % 2 == 0:
      n = n / 2
    else:
      n = n * 3 + 1

collatz(27)
```

Lab: Compass Maze

https://tinyurl.com/wilson-pi-day-5

- For this one, you'll need to save the contents of `maze.py` in the Trinket to a file called `maze.py` next to your code.
- The easiest way to do this is to press `New`, paste the code from `maze.py` online, press save, then name the file `maze.py`.

Survey: https://tinyurl.com/wilson-pi-survey-2024