# Evidence 1

To represent the grid of 8 rows and 15 columns, a **two-dimensional array** can be used, where each cell of the grid represents a square metre of the grid.

The contents of each square metre of the pond can be represented using a **character**:

* . represents one square metre of water
* S represents the stone impact position
* etc.

**Procedures and Functions:**

* CreatePond() creates a grid of 8 rows and 15 columns, representing the pond, then initialises the pond by setting every square metre of the pond to be water, represented by .. It returns the grid created, representing the pond. **[function]**
* DisplayPond(pond) displays the current contents of the pond, given an input of pond. **[procedure]**
* ThrowStone(pond) throws the stone at a certain position input by the user, given an input of pond. It then displays the pond via the DisplayPond procedure. It returns the grid representing the pond after the stone has been thrown. **[function]**

# Evidence 2

def CreatePond():

pond = [['.' for i in range(15)] for j in range(8)]

return pond

def DisplayPond(pond):

display = ""

for row in pond:

for cell in row:

display += cell

display += "\n"

print(display)

def ThrowStone(pond):

x = input("Enter X coordinate <1 to 15> : ")

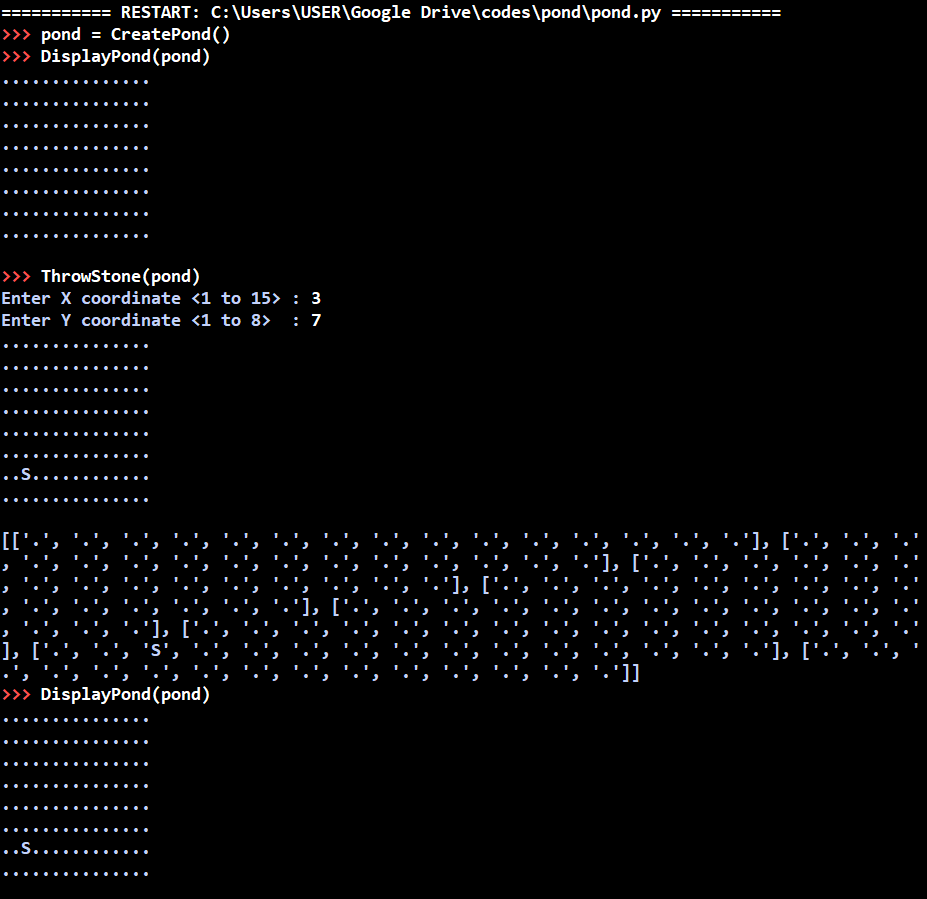
y = input("Enter Y coordinate <1 to 8> : ")

pond[int(y) - 1][int(x) - 1] = 'S' # pond array is 0-based, coordinates are 1-based

DisplayGrid(pond) # display the current state of the pond

return pond

# Evidence 3



# Evidence 4

import random

import math

def CreatePond():

pond = [['.' for i in range(15)] for j in range(8)]

inserted = 0

while inserted < 3:

x = math.floor(random.random() \* 15) + 1 # generates random x-coordinate from 1 to 15

y = math.floor(random.random() \* 8) + 1 # generates random y-coordinate from 1 to 8

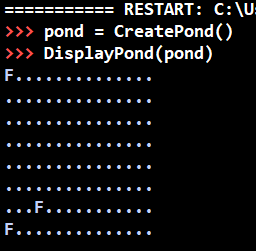
if (pond[y - 1][x - 1] != 'F'): # prevents fish from spawning at same spot

pond[y - 1][x - 1] = 'F'

inserted += 1

return pond

# Evidence 5



# Evidence 6

def FeedFish(pond):

x = input("Enter X coordinate <1 to 15> : ")

y = input("Enter Y coordinate <1 to 8> : ")

if pond[int(y) - 1][int(x) - 1] == 'F': # there is a fish present

pond[int(y) - 1][int(x) - 1] = 'H' # happy fish

else:

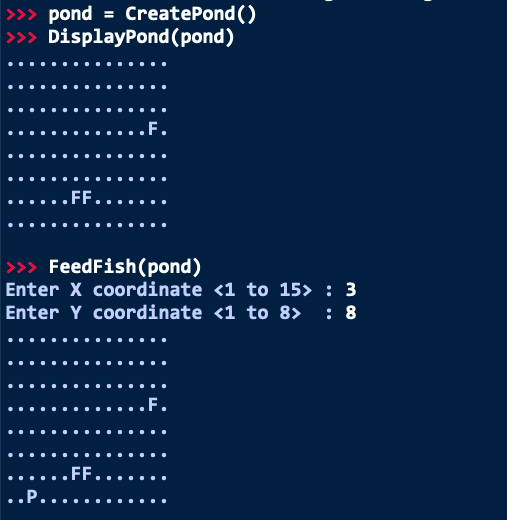
pond[int(y) - 1][int(x) - 1] = 'P' # uneaten pellet

DisplayPond(pond) # display the current state of the pond

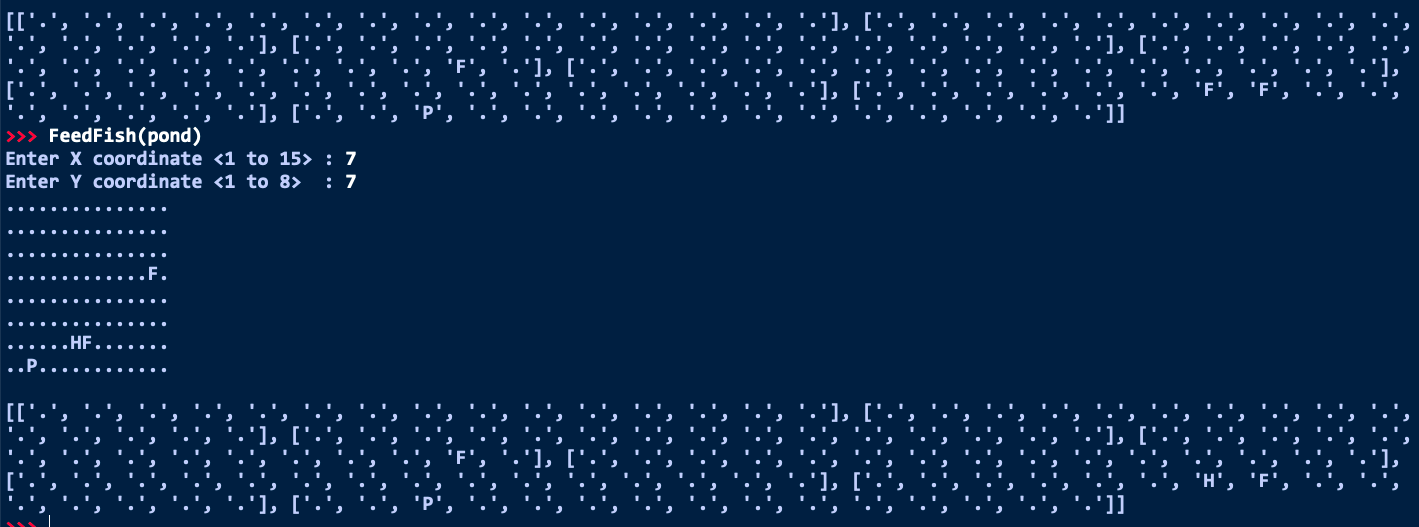
return pond

# Evidence 7

**Case which did not feed a fish:**

****

**Case where a fish was fed:**

****