Project 1: Mastermind

Elliott C

Barcelona, October 2019

What’s it all about and those all important rules

- One player (computer) selects a secret combination of coloured pegs. They can use the same colour more than once
- The second player (user) needs to guess that combination
- Feedback is provided after each guess
  - For each **correct** colour that appears in the code but **incorrect** position **1 WHITE** peg awarded
  - For each **correct** colour and **position** combination 1 **BLACK** peg awarded
- The feedback from the prior guess should help inform the next guess
- Second player only has a fixed number of guesses

Workflow

- Welcome the player to the game and explain how it works
- Computer to generate the ‘code’ of 3 numbers from a choice of 5.
  - Represents 3 colours from choice of 5, repeats are allowed
- Ask the user to input the colour guess of their first position
  - Repeat for 2nd and 3rd positions
  - Colours translated into numbers
- After all guesses inputted, the player’s guess will be compared to the CPU’s code
- Assess current guess and provide feedback
  - If the user guess matches computer generated code then we have a winner (and stop game)
- If exact match not found then user asked to enter their next guess
  - Repeats for 5 rounds

Challenges

- Getting to know the game!
- Branched logic
    - Winner, white peg(s), black peg(s)
- Resetting WHITE/ BLACK peg feedback at the beginning of next guess
- Too many functions!
- Terminating colours

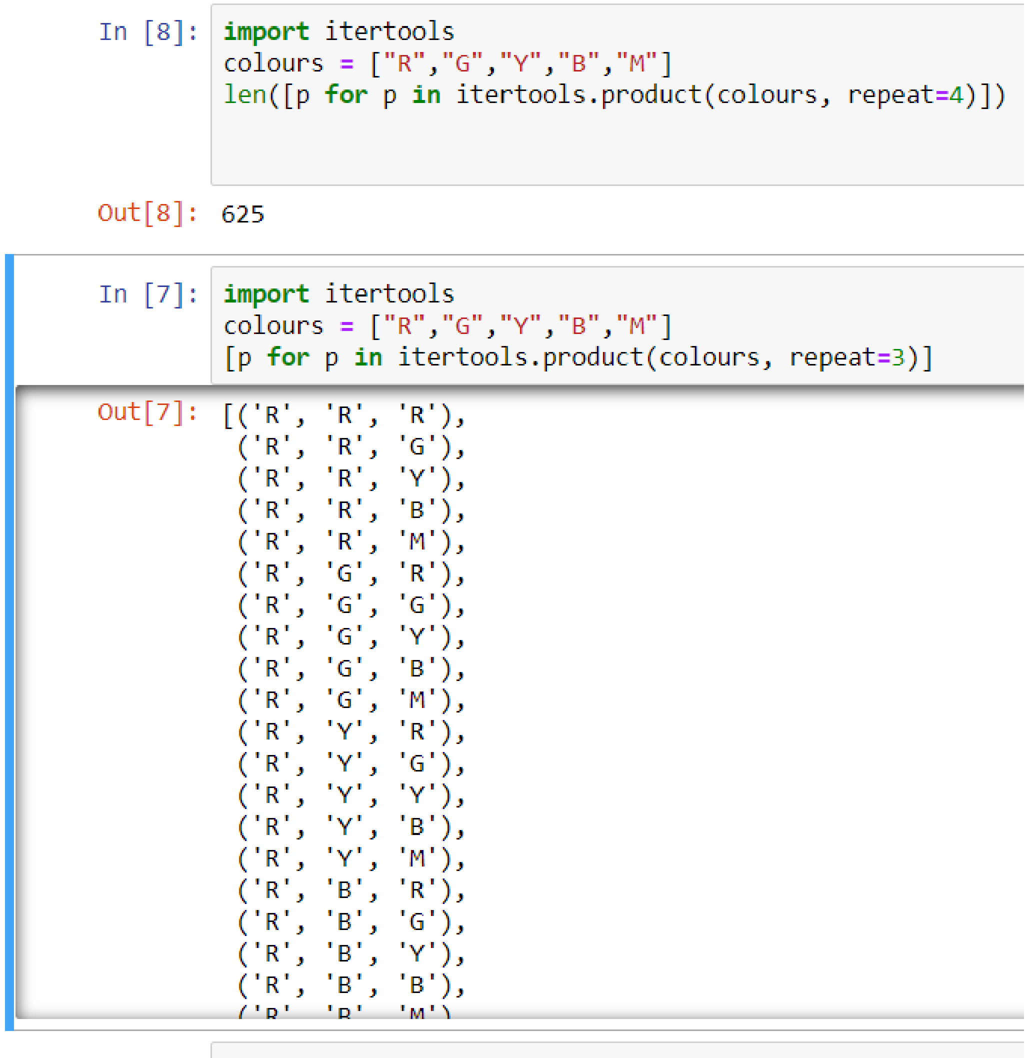
Learnings

- “Measure twice, cut once”
- Spend more time planning
  - Think about the specific functions before coding
- Use the new skills learned during the classes

Future Improvements

- Increase the length of the code that needs to be cracked
- Allow user to define the length of code
- Improve visuals
  - i.e. for WHITE/ BLACK peg feedback
- Use more efficient code
- Display probability of success before each guess

itertools



```
In [8]:  import itertools
         colours = ["R","G","Y","B","M"]
         len([p for p in itertools.product(colours, repeat=4)])

Out[8]:  625
```

```
In [7]:  import itertools
         colours = ["R","G","Y","B","M"]
         [p for p in itertools.product(colours, repeat=3)]

Out[7]:  [('R', 'R', 'R'),
          ('R', 'R', 'G'),
          ('R', 'R', 'Y'),
          ('R', 'R', 'B'),
          ('R', 'R', 'M'),
          ('R', 'G', 'R'),
          ('R', 'G', 'G'),
          ('R', 'G', 'Y'),
          ('R', 'G', 'B'),
          ('R', 'G', 'M'),
          ('R', 'Y', 'R'),
          ('R', 'Y', 'G'),
          ('R', 'Y', 'Y'),
          ('R', 'Y', 'B'),
          ('R', 'Y', 'M'),
          ('R', 'B', 'R'),
          ('R', 'B', 'G'),
          ('R', 'B', 'Y'),
          ('R', 'B', 'B'),
          ('R', 'B', 'M')
```

Demo