1

Guess My Number

Introduction

In this worksheet we will create a version of the *Guess my Number* game:

Guess my Number

- The computer will generate a secret random number between 1 and 20 inclusive.
- We will have 5 attempts to guess the number.
- After each guess the computer will tell us to go "lower" of "higher" with our next guess.

This game is simple but when coding this in python we will cover a lot of important concepts in programming:

① **Storing information**

We start with the storing of information (or data, as programmers like to call it). When storing data there are two questions you need to ask yourself

- What type of information do i want to store?
- What is the information that I want to store?

② **Outputting messages**

There is no point in storing information if we can't output — or print — it out at a later point. We will use the **`print`** command for this.

③ **Generating randomness**

There are many situations in programming where we want some random behaviour. The python module `random` will help us with this task.

④ **Repeating, repeating, repeating … commands**

We want the computer to repeat some commands a number of times. We will use the **`for`** statement and **`range`** command for this.

⑤ **Making decisions**

Finally, we need the computer to be able to make decisions and to execute different commands based on the outcome of that decision — here we will use the **`if`** statements for this.

With the above concepts covered we can build our game, Then you can try the extensions / variations suggested.

CoderDojo, Tramore, Waterford. (`kmurphy@wit.ie`)

1 Storing information

Create a new file with the following contents and save as `Storing_Information.py`.

Storing_Information.py

```
1 x = 5
2 print(x)
3 print(x * 4)
4
5 x = "5"
6 print(x)
7 print(x * 4)
```

So what happened when we run this and why?

- In line 1 we stored some data — the number 5 — in a variable (think box) called `x`. Now for the rest of the program whenever we what to get the contents of the box `x` we just type the name of the box, `x`.
- So in line 2, we wanted to output the information stored in `x` so we typed `print (x)`
- In line 3, we multiplied the contents of `x` by `4` and outputted the result. Since `x` contained `5` the result was $5 \times 4 = 20$.
- In line 5, we stored new data into — a string containing the character 5 — in our variable `x`. There are two important things to note
  - Storing new data into a box, replaces any existing data.
  - A string is a sequence of characters (letters, digits, symbols, etc) surrounded by quotes.
- In line 6, we print out the contents of box `x` we get `"5"`. Hopefully, no surprises there.
- In line 7, something weird happens, the multiplication by 4 did not result in 20. Instead we got `"5555"`. Why?
  - .Python does different things depending on type type of data/information it has. For example multiplying a number by 4 does the multiplication you learnt in school, but multiplying a string by by 4 repeats the string 4 times.

  ?

We save information/data using
name of box = data to be saved

- storing numbers versus storing strings

2 Basic Game

This section only contains notes on items for discussion — it need a lot of work before it can be used as standalone.

2.1 Outputting messages

- print plain message
- print message with placeholders

2.2 Generating randomness

- importing modules
- `random.randint`

2.3 Repeating, repeating, repeating ... commands

2.4 Making decisions

3 Extensions / Variations

OK, now that the basic game is finished, why not try some extensions …

3.1 Wild card guesses

In this version of the game the computer generates **two** random numbers at the start.

- the secret number, stored in `secret`
- a second integer in range 1 to 20, stored in `wildCard`

Whenever the human player inputs the wild card guess, the computer is allowed pick a new random secret number without telling the human player.

3.2 Warmer or colder

Rather then replying with “lower” of “higher” , the computer could reply with “colder” or “warmer”. The rules for this are ‘

- Computer replies with “colder” if the current guess is further away from the secret number than the previous guess.
- Computer replies with “warmer” if the current guess is closer to the secret number than the previous guess.