Season 2 – Season 1 Review

# What do we remember from last year?

## My thoughts

Programming is telling a story.

Getting the computer to do what you want.

Parts to make the whole

Python is a programming language – used to tell story / get what you want

Not the only language C, Java, … many more

## Your Thoughts

## A New Energy

Homework:

### Practice may not make perfect but can often make things go smoother

### 1-2 hours per week

* **Only after all other homework**
* Standing assignment: Duplicate/extend previous class(es) activity
* Can be all-encompassing – Jack Donavin 6.251

## Last year’s project Twenty Questions – “I’m thinking of a number…”

Let’s look at YOUR example

* Let’s play it
* What are some of the program’s features?
* in detail, line by line.

# *# iteration\_7.py 12Aug2020 crs*

*Comments*

*How do they work? What do they do? Of what use are they?*

*What does this comment tell the reader?*

"""  
Show use of subroutine for game  
Support changing hi guess limit  
+ Prompt user for number  
  
+ Set up target, check for match  
Use integer values  
  
+ Check / Report too high, low,..  
  
+ Randomize the target  
  
+ Add preamble with range  
  
+ Add multiple game support  
  
+ Protect against typos - ask again  
"""

What does the “”” do? Which one?

Why do we call the group a “Doc comment”

import random

Bringing additional stuff

What does “random” bring?

target\_hi = 20 # High end of target  
target\_hi = 30 # High end of target  
###target\_hi = 5 # TFD  
target\_low = 1 # LOW END OF TARGET

variables

assignment

at the end of line comments

comments to hide code

preamble = f"""  
I'm thinking of a number between {target\_low} and {target\_hi}  
Can you guess it? Remember to press the ENTER key  
to enter your guess. Good Luck!  
"""

print(preamble)

while True:

target = random.randint(target\_low, target\_hi)

while True:

inp = input(f"Enter Guess ({target\_low} and {target\_hi}):")

print("Number:", inp)

try:  
 guess = int(inp) *# Convert to integer* except:  
 print(f"I don't recognize '{inp}' as a number"  
 " please try again.")  
 continue

if guess < target:  
 print("Sorry your input of", guess, "is too low.")  
 continue  
 if guess > target:  
 print("Sorry your input of", guess, "is too high.")  
 continue  
 if guess == target:  
 print("Congratulations", guess, "is my number!")  
 break # End this game

print("Play a new game?")  
 inp = input("Enter N to quit: ")  
 if inp == "N" or inp == "n":  
 break

print("See you next time.")

### Extensions:

1. Announce number of guesses each time, at success.
2. Announce length of time each time, at success.

## More Details

Python, like other programming languages has parts to use in telling the story

Like the parts used in tasks such as walking, reading, writing, typing, and other skills we tend to remember them and improve in their use as we use them more and more.

**Comments** – for freeform description – the back story, future ideas, concerns

Single line - # to end of line

Doc comments:

“”” … almost anything not a triple “

“””

‘’’ … usually reserved in commenting out a bunch of code

including # and “”” doc comments “””

for debugging purposes …

‘’’

**Data**

* Numbers 1 2 3 1.234 1E2
* Text strings “a string”, ‘another string’
* Groups of data [1,2,3], [“red”, ”orange”, “yellow”]

**Data Storage**

* **variables**
  + From which to get data
  + to place data
  + names to aid in organization

**Computation / manipulation – depending on data type**

+, - \*, /

**Input / Output**

* input()
* print()

**tests / comparisons**

* **==**
* **!=**
* **>, <, >=, <=**

**flow control** – what do we do next

indentation

* code at the same indentation is executed together, sequentially
* in branches, loops the conditional code is indented
* the end of a conditional code section the indentation is decreased to the precondition

**branching** – one way or another

* if
* if else
* if else elif

**looping – repeating till**

* while – repeat while condition is True
* for – repeat over a list

**change loop flow**

* break
* continue

**exceptions**

* try
* except

**functions – code in a bag**

* let someone else do some of the work
* do work someone else can use
* package code for reuse