Thank you Ella!

Why was the computer cold?

Because it left its Windows open!

Source: https://www.ducksters.com/jokes/computer.php

CMPT 120

Lecture 8 – Chatbots

Iteration statement, `for` loop, `in` and `range( )`

Feedback from In-Class Activity 1

- Thank you!
- GPS
  - Comments
  - User interaction

Last Lecture

- Feedback from Assignment 0
- Improving grade-to-letter grade converter
  - Robustness -> User input Validation
  - Efficiency
- Step 4 Testing and Errors
- Our Guessing Game:

Your turn!

- **Step 1 - Problem Statement**
  - Write a guessing game, which allows a user to guess a number between 1 and 10.

Lecture 6 …
Robustness - User Validation

- What if the user enters "banana"?
  - Misbehaving user versus well-behaved user
    - Testing our new version of our guessing game:
      1. Test case 1 : input != number to guess
         - How to know the number to guess?
           - The trick is to …
      2. Test case 2 : input == number to guess
      3. Test case 3 : invalid input: 53 (> 10)
      4. Test case 4 : invalid input: -21 (< 1)
      5. Test case 5 : invalid input: "banana"

Robustness - User Validation

- Test cases:
  - Test data
  - Expected result
  - Actual result

- Testing our new version of our guessing game:
  1. Test case 1 : input != number to guess
  2. Test case 2 : input == number to guess
  3. Test case 3 : invalid input: 53 (> 10)
  4. Test case 4 : invalid input: -21 (< 1)
  5. Test case 5 : invalid input: "banana"

Readings Review

1. What does this code output:

A list is a **sequence**

```
movies = ["Superman", "Frozen", "X-Men"]
for movie in movies:
  print(movie,"!")
```

Readings Review

2. What does this code output:

```
for i in range(3):
    print(i)
```

The range function creates a **sequence**

Readings Review

3. What does this code output:

```
name = "Anne"
for char in name:
    print(char)
```

A string is a **sequence**

From a previous lecture: The `in` keyword (and `not in`)

```
# howIsItGoing-v2.py
#
# Description: Chatbot that asks a user how their day is going, and make
#              a comment that changes depending on how the user answered.
#
# Author: Anne Lavergne
# Date: M Jan. 15 2024


# Ask user how their day is going
# Read user's reply
userReply = input("Hey! How's your day going? ").strip().lower()

# Some possible replies when all is well!
wellResponses = ["great!", "fine!", "good!", "ok!"]

# Some possible replies when all is not well!
notWellResponses = ['not so good!', 'not so well!', 'terrible!', 'bad!']

# Make comment if user's day is going well
if userReply in wellResponses:
    print("Glad to hear!")
    print("Mine too!")
# Make comment if user's day is not going well
elif userReply in notWellResponses:
    print("Oh! Sorry to hear!")
else:
    # Make another type of comment otherwise
    print("Oh! I see ... !")
```

Careful: This **`in`** behaves differently than the **`in`** used in the **`for`** loop!

Repeated code -> Bad idea!

- What do you mean by ***repeated code***?
- If the **problem statement** is: List some movies, then Solution 1 would solve the problem using repeated code -> bad idea!
  - Solution 1:

```
print("Superman")
print("Frozen")
print("X-Men")
```

This semester, let's not do this! ☹

- Solution 2 would not -> good idea!
  - Solution 2:

```
movies = ["Superman", "Frozen", "X-Men"]
for movie in movies:
    print(movie,"!")
```

This semester, let's do this! ☺

Today's Menu
Improving our guessing game

- Wouldn't it be nice to play our guessing game many times without having to press **Run** over and over again?

- **New Problem Statement**
  - Write a guessing game, which allows a player to guess a number between 1 and 10 **in 3 guesses!**

- **Let's get coding!**

Review - Syntax of a `for` loop

Can be a string

Can be a list

```
<statement outside (before) the loop>
for <iterating variable> in <sequence> :
    <first statement to be repeated>
    <second statement to be repeated>
    ...
    <last statement to be repeated>
<statement outside (after) the loop>
```

Can be produced using range(...)

Review - Syntax of a `for` loop

```
<statement outside (before) the loop>
for <iterating variable> in <sequence> :
        <first statement to be repeated>
        <second statement to be repeated>
        ...
        <last statement to be repeated>
<statement outside (after) the loop>
```

- **Important** – About Indentation
  - ***Statements inside the loop*** (i.e., statements executed at each iteration of the loop) are the statements indented to the right with respect to the **`for`** keyword
  - ***Statements outside the loop*** (before and after the loop) are the statements that are **not** indented to the right with respect to the **`for`** keyword – these statements are considered to be at the same level of indentation as the **`for`** loop (as left-aligned as the **`for`** loop))

Review Questions

1. What are the **keywords** needed to make a **loop**?
2. In a **loop**, what do you need to do to the code that you want to repeat?
3. True or False? Functions (methods) can be **chained** from **left** to **right**?
4. Can you create a **List** containing all **variables**? Some **variables**?

Reflection: How would you go about answering this questions?

Next Lecture

- Introducing another field of study in Computing Science: **Cryptography and Encryption**

- Can we build programs that create secret(encrypted) messages using
  - Arithmetic operators
  - String indexing and slicing mechanism
  - etc…

- Let's see ☺