Last Lecture

✓ Introduced **Natural language processing** and **Chatbots**

✓ We built **chatbots** using
  • `print(...)` and `input(prompt)`
  • `data type` `str`
  • `variables`
  • `string formatting`
  • `list`
  • `random module`
From last lecture: Review Questions

1. How do you print in Python?
2. What is the command or function to get input from the user?
3. How do you concatenate two words in Python?
4. How do you store an input from the user?
5. What symbol do we use to assign a value to a variable?
6. What is an example of a chatbot?
From last lecture: Review Questions

7. How do we make a list in Python?
8. What module do we need to import to randomly choose something from a list?
9. How can we test smaller pieces of our Python code?
10. What does a dot after a module name do?
11. What is the only kind of symbol we can have in a variable name?
From last lecture: Your turn!

- **Step 1 - Problem Statement**
  - Write a chatbot that prints random fortune cookie quotes.
Possible Solution

```python
# randomFortuneCookieQuote.py
#
# Description: Chatbot that prints random fortune cookie quotes
#
# Author: Anne Lavergne
# Date: W Sept. 13 2023

import random

# Create the fortune cookie quotes
fortuneQuotes = [
    "You will have a long and happy life.",
    "You will have many children.",
    "You will be successful in business.",
    "You will find true love."
]

# Select one fortune cookie quote randomly
randomFortune = random.choice(fortuneQuotes)

# and print it
print(randomFortune)
```
Notice …

• Location of `import` statement

• Left-aligned code

• `input(prompt)` returns a string
  • A string has the data type `str`

• Good Programming Style - GPS
  • Give your variables descriptive names
  • Length of our lines of code: 80 characters
GPS and user interaction

• As a user, which one would you prefer?

User interaction # 1

```python
>>> name = input()
Anne
>>> print(name)
Anne
```  

User interaction # 2

```python
>>> name = input("Please, enter your first name: ")
Please, enter your first name: Anne
>>> print("You have entered " + "'" + name + "'" + " as your first name.")
You have entered 'Anne' as your first name.
```  

• Why?
Today’s Menu

- We shall introduce *branching*
  - A way of controlling the *execution flow* when the program is executed
Let’s introduce Branching (Python conditional statements)

- **Step 1 - Problem Statement**
  - Create a chatbot that asks a user how their day is going, and make a comment that changes depending on how the user answered.
How can we improve our program?

• Introducing the Boolean operator or
How can we improve our program?

• How to make our program more robust?
  • User can answer with a word which can be written with upper and/or lower case letters
    • Example: Yes, yes, YES
    • Introducing string functions `upper()` and `lower()`
  • User can include blank spaces in their answers
    • Example: "Yes" or "yes" and these answers do not match "YES"
    • Introducing string function `strip()`
How can we improve our program?

- Introducing the Boolean operators `in` and `not in`
Notice …

• Conditional statements like
  
  ```python
  if response == "Great!"
  ```

  contain a Boolean expression:

  ```python
  response == "Great!"
  ```

  and Boolean expressions can be **True** or **False**.

• We could write:

  ```python
  result = response == "Great!"
  ```

  in our Python program

• Variable `result` would have the data type: `bool`

• `bool` is short for **Boolean**
Review: Conditional statements

**if statement**

Syntax #1:

```
if condition:
    some statement(s)
```

This small empty space made of a few white space (or blank) characters is called an **indentionation** and its presence is **very important** in a conditional statement in Python.
Review: Conditional statements

if else statement

Syntax #2:

```
if condition:
    some statement(s)
else:
    some other statement(s)
```

Indentation

Boolean expression
Your turn!

- Step 1 - Problem Statement
  - Write a login program, which allows a user to login in with a password
Review Questions

1. What is wrong with this code?
   
   ```python
   3_colours = ["blue", "green", "white", "pink"]
   ```

2. Why do we want to put an `import` statement at the beginning (top) of a program?

3. When a program executes, what is first executed?

4. Is the `if` part of a conditional statement mandatory?
Review Questions

5. Is the `else` part of a conditional statement mandatory?

6. What is wrong with this code fragment?
   ```python
   if color = "purple":
       print("Cool!")
   ```

7. How could this code be improved?
   ```python
   favBand = input()
   print("Oh, I like this band!")
   ```
Next Lecture

• More on conditional statements and Boolean expressions (conditions)
  • What if there are many conditions (many branches)?
  • What if we are dealing with integers?
  • Can these conditional statements be nested?