Parables and Pythons

Dr. Charles "Chuck" Bell

Lesson 7: 24 October 2018

CLASS AGENDA

- **NOTICE: No class next week, but services will be held in the dock!**
- Bible Study
  - The wise and foolish virgins and the closed door
- The Golden Age of British Comedy
  - ‘tis but a scratch
- Computer Programming with Python
  - Review:
    - List Methods
  - Hands-On Practice
    - Dictionaries
- Homework

BIBLE STUDY: THE PARABLES OF JESUS

THE WISE AND FOOLISH VIRGINS MATTHEW 25:1–13

“At that time the kingdom of heaven will be like ten virgins who took their lamps and went out to meet the bridegroom. Five of them were foolish and five were wise. The foolish ones took their lamps but did not take any oil with them. The wise ones, however, took oil in jars along with their lamps. The bridegroom was a long time in coming, and they all became drowsy and fell asleep. “At midnight the cry rang out: ‘Here’s the bridegroom! Come out to meet him!’ “Then all the virgins woke up and trimmed their lamps. The foolish ones said to the wise, ‘Give us some of your oil; our lamps are going out.’

THE WISE AND FOOLISH VIRGINS MATTHEW 25:1–13

"'No,' they replied, 'there may not be enough for both us and you. Instead, go to those who sell oil and buy some for yourselves.' "But while they were on their way to buy the oil, the bridegroom arrived. The virgins who were ready went in with him to the wedding banquet. And the door was shut. "Later the others also came. 'Lord, Lord,' they said, 'open the door for us!' "But he replied, 'Truly I tell you, I don't know you.' "Therefore keep watch, because you do not know the day or the hour.

THE CLOSED DOOR LUKE 13:25

Once the owner of the house gets up and closes the door, you will stand outside knocking and pleading, ‘Sir, open the door for us.' "But he will answer, ‘I don’t know you or where you come from.’

WHAT IS THE STORY ABOUT?

- The parable in Mathew 25:1-13 and Luke 13:25 are the same moral teaching.
  - One has more story and relates to societal situations of the times (and present day).
  - The other has a more pointed story (and much shorter) but teaches the same.
- This parable stresses the importance of being ready for the coming of Christ. He will come again—John 14:1–6.
- All the prophecies in Matt. 24 concerning His coming are in the process of being fulfilled.
- He wants His people to be awake and alert. Satan seeks to put God’s people to sleep spiritually, so they will not be ready.
- Let’s take the parable apart verse-by-verse…plus a little more to the story.

PEOPLE (VERSES 1–5)

- **Story** (verse 1): A parable to teach us readiness.
- **Symbols** (verses 2–4)
  - **The foolish.** They took no extra oil, thinking that they had plenty (cf. Rev. 3:17).
  - **The wise.** They were prepared with an extra supply of oil. They did not depend upon past resources.
- **Sleeping** (verse 5): While waiting for the wedding, they all slept. The wise went to sleep prepared, the foolish, unprepared (cf. Prov. 27:1).

PLAN (VERSES 6–9)

- **Preparation** (verses 6–7): “The bridegroom cometh; go ye out to meet him.” They trimmed their lamps. We, as Christians, must trim our spiritual lives in readiness for His coming—Matt. 24:44.
- **Pleading** (verse 8): The foolish asked the wise to give them some of their oil. They were dependent on others for entrance to the wedding feast. Each person must have a personal relationship with Jesus Christ.
- **Personal** (verse 9): The wise told the foolish to go and buy oil for themselves. No one can accept Christ for you. Each person must be born again—John 3:1–8.

PROBLEM (VERSES 10–12)

- **Problem** (verse 10): Note: “The door was shut.” God gave the people in Noah’s day time to prepare—Gen. 6:3. Some day the door will be closed (see Prov. 29:1; Isa. 55:6).
- **Plea** (verse 11): “Open to us.” The wise could not open the door. Only Christ can open the door—John 14:6.
- **Punishment** (verse 12): Note: “I know you not.” Only those who are born again will gain entrance—John 3:1–8; Mark 16:16.

PREPARATION (VERSE 13)

- Prepare by watching!
- **Watch your character** (I Cor. 7:1): You are developing good or bad habits. It takes a lifetime to build good Christian character.
- **Watch your consecration** (Rom. 12:1-2): Caleb, though 85 years old, could say that he had fully followed the Lord—Deut. 1:36. God wants full surrender, full dedication.
- **Watch your companions** (II Cor. 6:17): You are known by your associations. Choose Christian friends—Amos 3:3.

CONCLUSIONS

- Are you awake?
  - Are you actively watching?
- Do you have enough oil?
  - The oil is a symbol of the Holy Spirit. Paul tell us to be filled with the Spirit—Eph. 5:18.
- Don't depend on past experiences or blessings.
  - Each day, renew the touch of God upon your life.
- Learn to walk in the Spirit . . . pray in the Spirit . . . live in the Spirit.
  - As the Holy Spirit controls your life, you will be ready when the bridegroom comes.
- **More importantly: be prepared for Christ's return.**
  - Do put off until tomorrow what you need to do today because tomorrow may be too late.

THE GOLDEN AGE OF BRITISH COMEDY

TV
14
LV

L LANGUAGE

V VIOLENCE

IT CONTAINS STRONG LANGUAGE & VIOLENCE
VIEWER DISCRETION ADVISED

COMPUTER PROGRAMMING

Hands On Learning

REVIEW: METHODS

```
my_list = ['a',1,True]
my_list.append('z')
```

the object that
we are calling the
method with

the name of
the method

arguments to
the method

REVIEW: LIST METHODS

- A list is mutable and can change:

```
my_list[0]='a'  #index assignment
my_list.append(), my_list.extend()
my_list.pop()
my_list.insert(), my_list.remove()
my_list.sort()
my_list.reverse()
```

- Most of these methods ***do not return a value***
- This is because lists are mutable, so the methods modify the list directly. No need to return anything.

DICTIONARIES

- Store ***pairs*** of entries called ***items***
  { 'CS' : '743-713-3350', 'UHPD' : '713-743-3333'}
- Each pair of entries contains
  - A ***key***
  - ***A value***
- Key and values are separated by a colon
- Paris of entries are separated by commas
- Dictionary is enclosed within curly braces

USE

- Keys must be ***unique*** within a dictionary
  - No ***duplicates***
- If we have

        age = {'Alice' : 25, 'Bob' :28}

  then

        age['Alice'] is **25**

  and

        age[Bob'] is **28**

DICTIONARIES ARE MUTABLE

- `>>> age = {'Alice' : 25, 'Bob' : 28}`
- `>>> saved = age`
- `>>> age['Bob'] = 29`

-
  ```
  >>> age
  {'Bob': 29, 'Alice': 25}
  ```

-
  ```
  >>> saved
  {'Bob': 29, 'Alice': 25}
  ```

KEYS MUST BE UNIQUE

- >>> age = {'Alice' : 25, 'Bob' : 28, 'Alice' : 26}
- >>> age
  {'Bob': 28, 'Alice': 26}

DISPLAYING CONTENTS

- >>> age = {'Alice' : 25, 'Carol': 'twenty-two'}
- >>> age.items()
  dict_items([('Alice', 25), ('Carol', 'twenty-two')])
- >>> age.keys()
  dict_keys(['Alice', 'Carol'])
- age.values()
  dict_values([28, 25, 'twenty-two'])

UPDATING DIRECTORIES

- >>> age = {'Alice': 26 , 'Carol' : 22}
- >>> age.update({'Bob' : 29})
- >>> age
  {'Bob': 29, 'Carol': 22, 'Alice': 26}
- >>> age.update({'Carol' : 23})
- >>> age
  {'Bob': 29, 'Carol': 23, 'Alice': 26}

RETURNING A VALUE

-
  ```
  >>> age = {'Bob': 29, 'Carol': 23, 'Alice': 26}
  ```

-
  ```
  >>> age.get('Bob')
  29
  ```

-
  ```
  >>> age['Bob']
  29
  ```

REMOVING A SPECIFIC ITEM

-
  ```
  >>> a = {'Alice' : 26, 'Carol' : 'twenty-two'}
  ```

-
  ```
  >>> a
  {'Carol': 'twenty-two', 'Alice': 26}
  ```

-
  ```
  >>> a.pop('Carol’)
  'twenty-two'
  ```

-
  ```
  >>> a
  {'Alice': 26}
  ```

REMOVING A SPECIFIC ITEM

-
  ```
  >>> a.pop('Alice')
  26
  ```

-
  ```
  >>> a
  {}
  ```

-
  ```
  >>>
  ```

REMOVE A RANDOM ITEM

- >>> age = {'Bob': 29, 'Carol': 23, 'Alice': 26}
- >>> age.popitem()
  ('Bob', 29)
- >>> age
- {'Carol': 23, 'Alice': 26}
- >>> age.popitem()
  ('Carol', 23)
- >>> age
  {'Alice': 26}

SUMMARY

- Strings, lists, tuples, sets and dictionaries all deal with aggregates
- Two big differences
  - ***Lists*** and ***dictionaries*** are ***mutable***
    - Unlike strings, tuples and sets
  - ***Strings***, ***lists*** and ***tuples*** are ***ordered***
    - Unlike sets and dictionaries

HOMEWORK

All homework assignments can be handed in on hardcopy (with your name at the top) or emailed to me at drcharlesbell@gmail.com.

HOMEWORK #5 - REVIEW

- This is the correct solution to the modified dice simulator assignment:

```
import random
print("Welcome to the Yahtzee dice roller!")
dice_stack = []
dice_queue = []
roll_number = 1
repeat = 'y'
while (repeat == 'y') or (repeat == 'Y'):
    dice = [0,0,0,0,0]
    print("Rolling the dice...")
    for i in range(0, 5):
        dice[i] = random.randint(1, 6)
    print("You rolled the following dice:")
    dice.sort()
    print(dice)
    print()
    dice_stack.insert(0, (roll_number, dice))
    dice_queue.append((roll_number, dice))
    repeat = input("Would you like to roll again? [Y/n]: ")
    print()
    roll_number = roll_number + 1
print("Stack:\n{0}".format(dice_stack))
print("Queue:\n{0}".format(dice_queue))
print("bye!")
```

HOMEWORK #5 - REVIEW

```
$ python3 ./yahtzee2.py
Welcome to the Yahtzee dice roller!
Rolling the dice...
You rolled the following dice:
[1, 2, 3, 4, 6]
Would you like to roll again? [Y/n]: y
Rolling the dice...
You rolled the following dice:
[1, 2, 4, 4, 5]
Would you like to roll again? [Y/n]: y
Rolling the dice...
You rolled the following dice:
[1, 4, 4, 4, 5]
Would you like to roll again? [Y/n]: n
Stack:
[(3, [1, 4, 4, 4, 5]), (2, [1, 2, 4, 4, 5]), (1, [1, 2, 3, 4, 6])]
Queue:
[(1, [1, 2, 3, 4, 6]), (2, [1, 2, 4, 4, 5]), (3, [1, 4, 4, 4, 5])]
bye!
```

HOMEWORK ASSIGNMENT #6

- (2 points) Create a script named ‘rolodex1.py’ that prompts the user for the name, address, and phone number for at least 5 entries. Print out the dictionary.
  - **`Hint: you will need a dictionary with at least 7 fields (first_name, last_name, street, city, state, zip, phone)`**
  - **`Hint: each entry in the rolodex is a dictionary, but the rolodex can be a list.`**
- (2 bonus points) Print the names in the dictionary in last name, first name order. Note: entering the data in order does not count towards these bonus points.
- (1 bonus point) Make the address a dictionary.
- (4 bonus points) Permit the user to enter more than one phone number and store them in the dictionary. Show a printout of the dictionary.
  - **`print(rolodex)`**

QUESTIONS OR COMMENTS?