Parables and Pythons

Dr. Charles "Chuck" Bell

Lesson 3: 26 September 2018

CLASS AGENDA

- Bible Study
  - The Speck and the Plank
- The Golden Age of British Comedy
  - He says, “He’s already got one.”
- Computer Programming with Python
  - Hands-On Practice
    - Assignment
    - Arithmetic
    - Variables
    - Input and Output
  - New Concepts:
    - Conditional Statements

BIBLE STUDY: THE PARABLES OF JESUS

THE SPECK AND THE PLANK

- Some bible translations name this parable, "The mote and the beam."
- Mathew 7:1-5 *"Do not judge, or you too will be judged. For in the same way you judge others, you will be judged, and with the measure you use, it will be measured to you. "Why do you look at the speck of sawdust in your brother's eye and pay no attention to the plank in your own eye? How can you say to your brother, 'Let me take the speck out of your eye,' when all the time there is a plank in your own eye? You hypocrite, first take the plank out of your own eye, and then you will see clearly to remove the speck from your brother's eye.*

EXAMPLE (TRUE LIFE)

- I recently talked to someone who was very concerned about another family member. They were critical of their relative for not being able to kick the addiction of alcohol.
- It was quite a dissertation on the problems and potential solutions for the subject. This person spoke from experience because they too had overcome their addiction to alcohol.
- The problem is, this person is addicted to tobacco. So much so, they smoke constantly ceasing only to eat and sleep.
- Is this a problem?

WHAT IS THE STORY ABOUT?

- A recurring topic that, by allegory, must have been a real problem at the time – hypocrisy.
- The Bible strongly opposes the judging of one another. Judging is a dangerous practice - it defiles and destroys.
- Only God is qualified to judge. He wants us to love . . . leave the judging to him!
- Let’s take the parable apart verse-by-verse…

CRITICISM AND JUDGING (VERSES 1–2)

- *“Don’t criticize, and then you won’t be criticized! For others will treat you as you treat them.”*
- Criticism shows a lack of Christianity (James 4:12).
- Criticism shows a lack of control (I Cor. 9:27).
  - God’s people must learn to be led by the Spirit.
  - Criticism shows a lack of consecration.
- Christians are to be kind and forgiving (Eph. 4:32).

CONCERN AND JUDGING (VERSE 3)

- *Seeing the mote (small wood splinter) in a brother's eye, but not seeing the beam (log) in one's own eye.*
- Seeing another's sin: Paul told Timothy to "keep thyself pure" (I Tim. 5:22).
- See your own sins first; then another's shortcomings will fade (Rom. 14:4).
  - No one is qualified to judge others!
- Seeing another's slackness. The Bible says that God is the judge (I Cor. 4:5).
  - He will judge perfectly.

CONDEMNATION AND JUDGING (VERSE 4)

- *Should you say, ‘Friend, let me help you get that speck out of your eye,’ when you can’t even see because of the board in your own”.*
- Cleanse yourself first (II Cor. 7:1). This is a daily cleansing.
- Control yourself (Ps. 34:13).
  - Keep your lips from evil.
  - Control of the tongue often prevents trouble.
- Consecrate yourself. David’s prayer here should be ours, also.
  - (Ps. 139:23–24) *Search me, God, and know my heart; test me and know my anxious thoughts. See if there is any offensive way in me, and lead me in the way everlasting.*

CLEANSING FROM JUDGING (VERSE 5)

- *You hypocrite, first take the plank out of your own eye, and then you will see clearly to remove the speck from your brother's eye.*
- Jesus made it personal ("You hypocrite.")
  - You must cleanse your own life before you can help others.
- You must purge yourself "First cast out the beam in thine own eye."
  - See also I Peter 4:18.
- You must plan what you say and how you say it "And then shalt thou see clearly to cast out the mote out of thy brother's eye."
- Before we can help others, we must be right with God. One cannot live hypocritically and be able to help others spiritually.
  - See also James 4:8.

CONCLUSIONS

- Often those who judge others' misconduct are guilty of the same sin, and criticizing another is an attempt to cover their own wrongdoing.
- Those who judge others are not living close to God, nor are they living successful Christian lives.
- Remember: we are to love people with all their faults (because we have our own set); let God judge the fault!
- This parable, while seemingly short and clear, has a profound affect on all Christians. Why? Because people like to criticize! It's in our nature and it's wrong.
- Let's not forget Jesus' words about the adulteress brought to him by the Pharisees and lawyers (teachers of the law): *"Let any one of you who is without sin be the first to throw a stone at her."*

THE GOLDEN AGE OF BRITISH COMEDY

And now for something completely different…

- https://www.youtube.com/watch?v=GYcopzJ-T9w

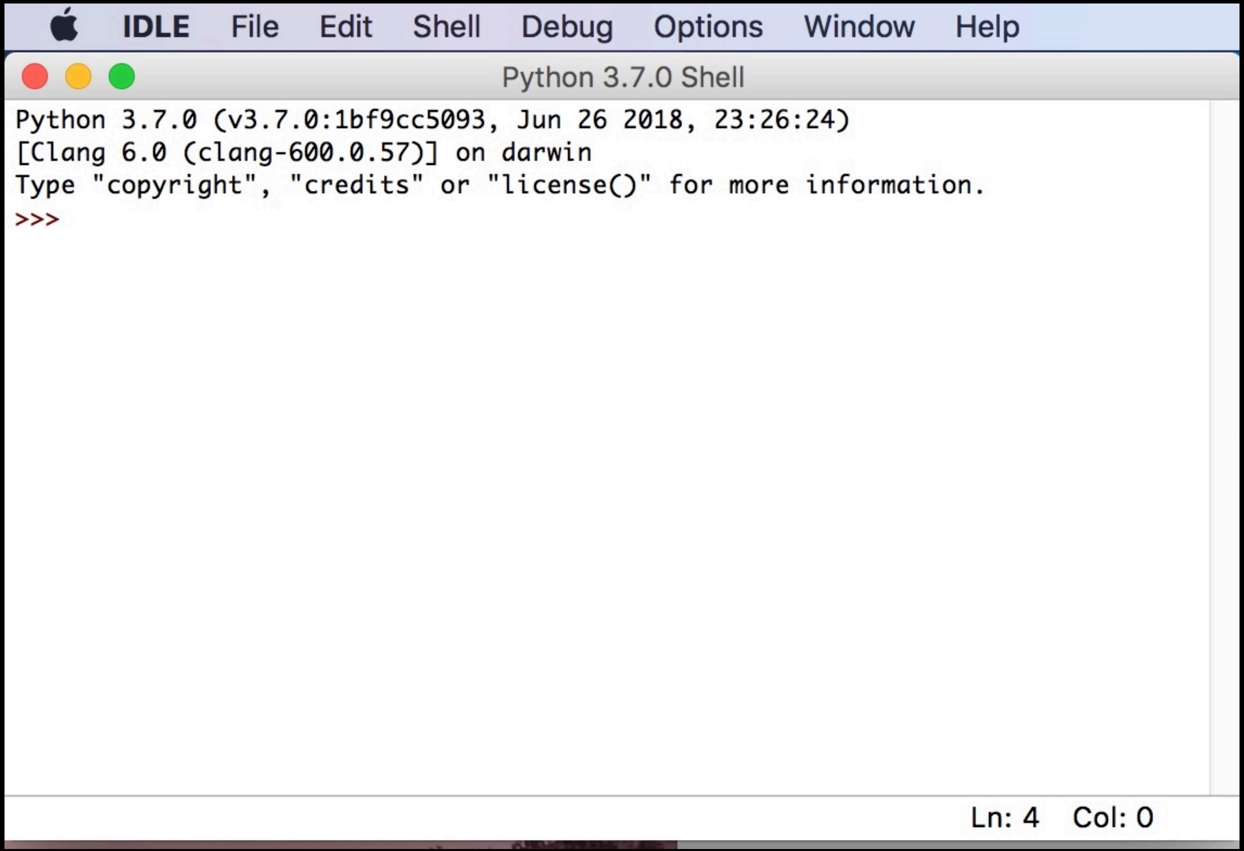
COMPUTER PROGRAMMING

Hands On Learning

THE PYTHON IDLE APPLICATION

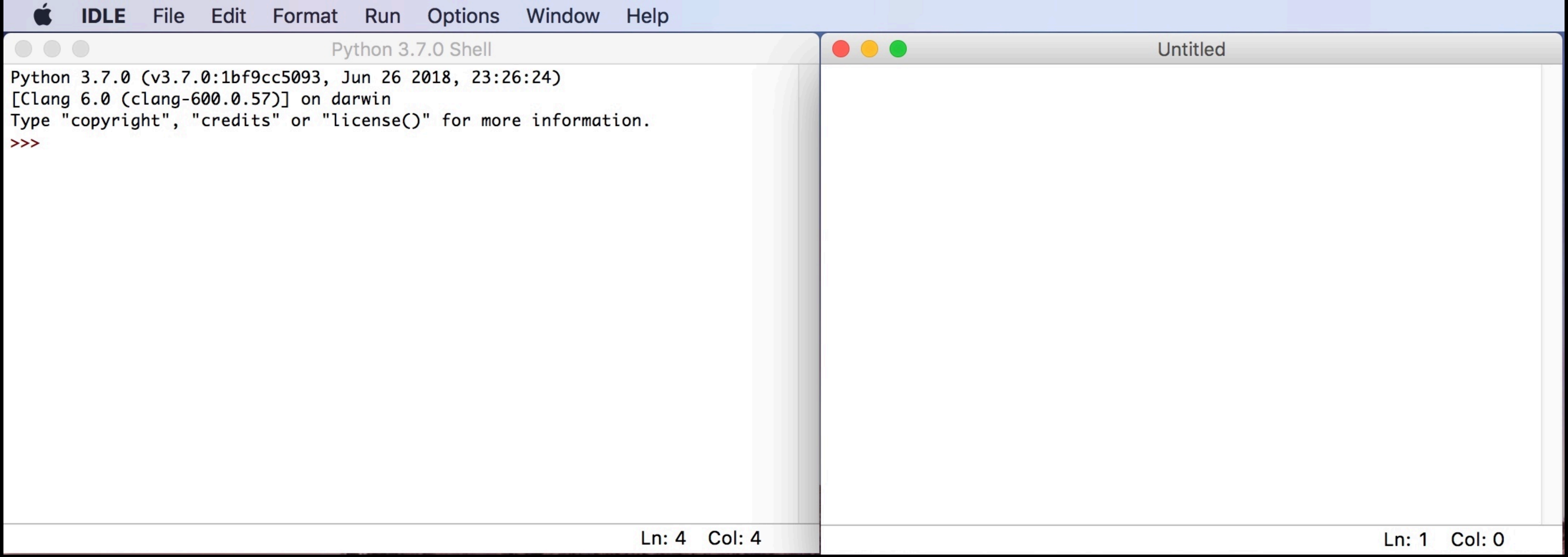
- Integrated Development Learning Environment (IDLE)
  - Available as an application on your computer
  - Has an interactive shell (the Python interpreter)
    - Same as running python from a command prompt
  - Has a file editor to allow you to create Python scripts
  - Allows you to run your scripts from the editor (nice!)
- Start the IDLE application from your start menu (or equivalent).
  - IDLE starts with the Python shell

PYTHON SHELL (IDLE)



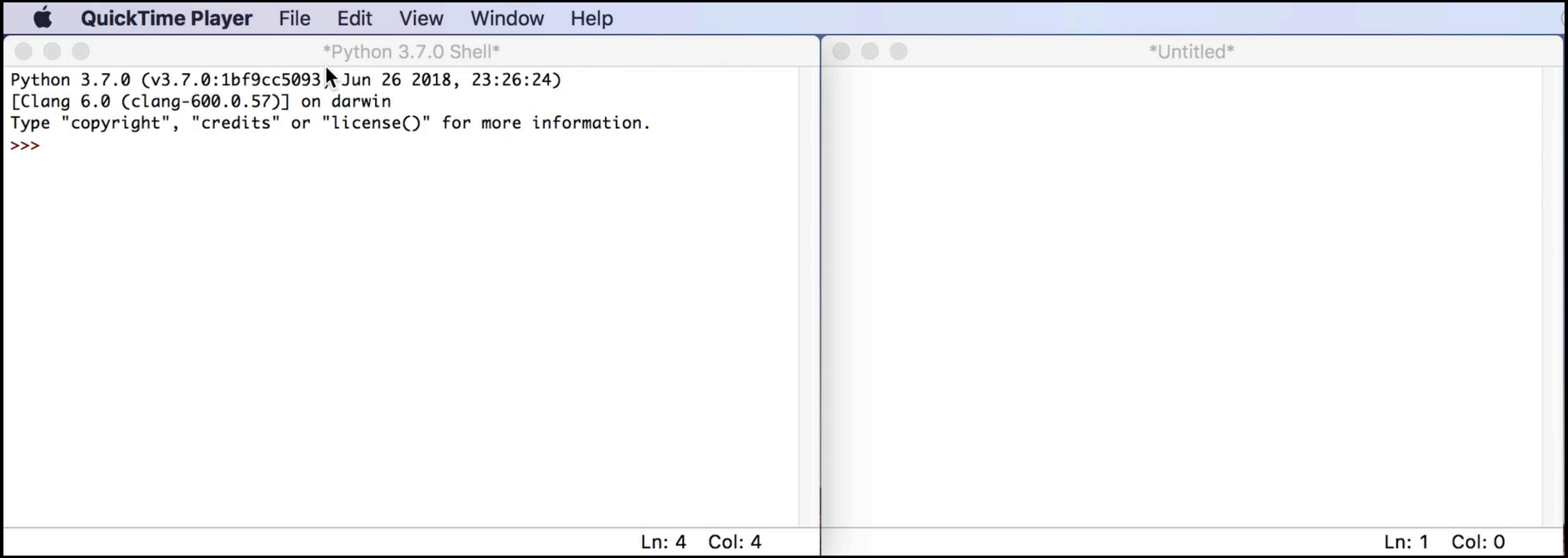
```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 26 2018, 23:26:24)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
```

PYTHON EDITOR (IDLE)



```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 26 2018, 23:26:24)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
```

DEMONSTRATION



```
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 26 2018, 23:26:24)
[Clang 6.0 (clang-600.0.57)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>>
```

HANDS ON EXPERIMENT #1
PRACTICING THE ASSIGNMENT STATEMENT

- “Numbers are fundamental to programming, and arithmetic is one of the things computers are very good at.
- We will begin by experimenting with numbers, and the best place to experiment is the Python Shell.
- Open the IDLE application on your computer.
- Type the following into the Python Shell:

```
20 * 9 / 5 + 32
```

- What number did you get?
- This example does tell us a few things: * means multiply. / means divide.
- Python does multiplication before division, and it does division before addition.

HANDS ON EXPERIMENT #2
PRACTICING THE ASSIGNMENT STATEMENT

- Next, try these statements in the shell (one at a time):

```
8 + 9 / 2 * 10
(8 + 9) / 2 * 10
(8 + 9) / (2 * 10)
```

- What numbers did you get?
- Why weren't the numbers all the same?

HANDS ON EXPERIMENT #3
PRACTICING USING VARIABLES

- Now, let’s try a variable:

```
calculatedValue = (8 + 9) / 2 * 10
```

- What happened?
- The value is ’saved’ in a variable named “`calculatedValue`”. Notice how it is spelled.
  - The variable must be on the left side and must be a single word (no spaces).
  - The variable can be as long as you like and can contain numbers and the underscore character (_).
  - Also, characters can be upper- and lowercase.
- To see the value, use the output statement – the `print()` and `format()` functions as follows.
- Remember, the format function provides the ability to substitute the value stored in the variable into the string being “printed” to the screen.

```
print("The value is {0}.".format(calculatedValue))
```

- What value did you see?

HANDS ON EXPERIMENT #4
PRACTICING GETTING INPUT FROM THE USER

- Recall, getting input from the user requires using the input() function, which provides a prompt to the user and returns the value entered.
- Try this statement in the shell:

```
chosenValue = input("Please enter a number between 1 and 6: ")
```

- What happened?
- The value the user typed in is 'saved' in a variable named "**chosenValue**".
- Once again, to see the value, use the output statement – the **print()** and **format()** functions as follows.

```
print("You entered: {0}.".format(chosenValue))
```

- What value did you see?

CONDITIONAL STATEMENTS

- Sometimes in a program, we may need to make decisions that change how the program reacts, produces output, identifies errors, warnings, etc.
- Recall from our last experiment we asked the user to enter a value between 1 and 6. How do we know whether the user did what was asked?
- We can use a conditional like this:

```
if <something is true> then
    <do operation A>
else
    <do operation B>
```

- Notice we do “A” when the condition (an expression is evaluated) is true.
- Otherwise, we do “B”.
- This is how programmers ’control’ the program flow (execution).
- Let’s try this in a more advanced example using the IDLE editor.

HANDS ON EXPERIMENT #5
USING CONDITIONALS

- Open the IDLE editor on your computer.
- Type these lines of code into the editor. (File | New File)

```
chosenValue = input("Please enter a number between 1 and 6: ")
if (chosenValue < 1) or (chosenValue > 6):
    print("ERROR: You did not enter a value between 1 and 6.")
else:
    print("You entered: {0}.".format(chosenValue))
```

- This is the same example we've just worked with only now, we're saving in a file (script).
- Go ahead and save the file. Name it "**`experiment5.py`**".
- Next, run the script using the Run | Run Module menu.
- What do you see?

HANDS ON EXPERIMENT #5
USING CONDITIONALS

- You should have gotten an error.
- Sorry, this was a dirty trick.
- Can anyone figure out why we got this error? Hint: data type.

```
RESTART: /Users/cbell/Documents/RCofC/Teaching/Parables and
Pythons/github/Parables-and-Pythons/examples/experiment5.py
Please enter a number between 1 and 6: 3
Traceback (most recent call last):
  File "/Users/cbell/Documents/RCofC/Teaching/Parables and
Pythons/github/Parables-and-Pythons/examples/experiment5.py", line 2, in
<module>
    if (chosenValue < 1) or (chosenValue > 6):
TypeError: '<' not supported between instances of 'str' and 'int'
>>>
```

- Let's try it again…

HANDS ON EXPERIMENT #5
USING CONDITIONALS

- Let’s modify the code a bit.

```
chosenValue = input("Please enter a number between 1 and 6: ")
intValue = int(chosenValue)           # add this line
if (intValue < 1) or (intValue > 6):  # change this line
    print("ERROR: You did not enter a value between 1 and 6.")
else:
    print("You entered: {0}.".format(chosenValue))
```

- Save the file.
- Next, run the script again using the Run | Run Module menu.
- What do you see?
- Try it again with an invalid number. Did you see the error?

HOMEWORK

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

No homework for week #3.

QUESTIONS OR COMMENTS?