



Parables and Pythons

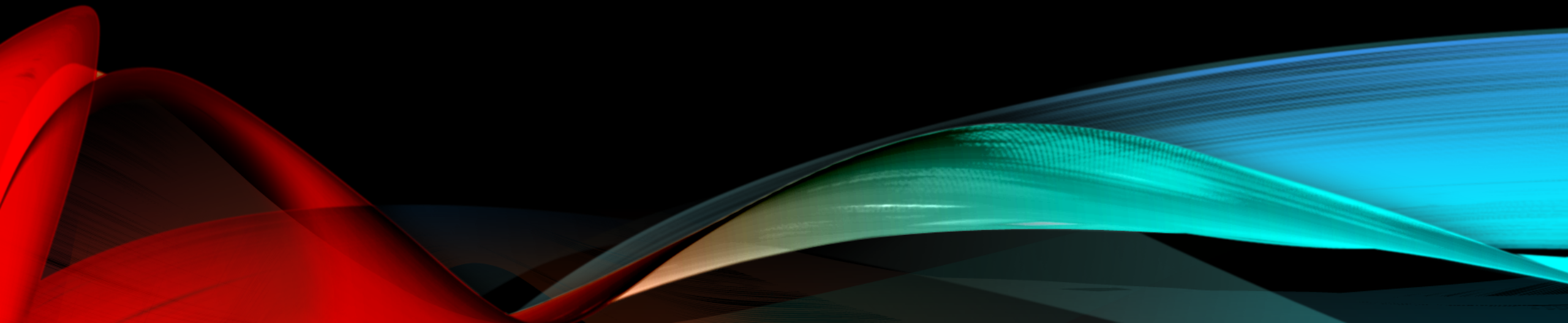
Dr. Charles “Chuck” Bell

Lesson 2: 19 September 2018

CLASS AGENDA

- Bible Study
 - Introduction to the Parables of Jesus
- The Golden Age of British Comedy
 - Spam
- Computer Programming with Python
 - What is Computer Programming?
 - Basics: Input, Output, and Variables
 - How to use the Class Repository
- Homework

BIBLE STUDY: THE PARABLES OF JESUS



WHAT IS A PARABLE?

- Strictly speaking, a parable is a simple story used to illustrate a moral or spiritual lesson.
- The word parable is from the root word “paraballo” or in the Greek “parabole.” This compound word comes from “para” which means “to come along side or compare” and “ballo” which literally means “to throw” or “see” with.
- However, they were often told in a very different manner than what we would expect from a “story”.
- May be an allegory and may include inanimate objects (like trees, plants, or things) or people in various societal positions often with a tension between good and evil.
- But to understand why parables are written the way they are, we have to remember they were originally communicated orally using a specific pattern and to capture the attention of the audience.

THE JEWISH CONTEXT

- Parables are neither new to the New Testament nor are they an invention of Jesus or the disciples.
- In fact, parables exist throughout the Bible.
- Thus, Jesus – the “master story teller” – learned the art of parables from His Jewish heritage.
- Jewish parables were often crafted specifically to “trap” the intellect causing the audience to contemplate the elements of the story.
- Jesus’ parables are completely Jewish in both style and patterns of thought.

A PATTERN FOR JEWISH PARABLES

- **Setting** – What prompts the telling of the parable – life situation or exegetical situation (Jesus' parables tend to be situational).
- **Hermeneutic imagination** – juxtaposition through an analogy (“life is like...”).
- **Mashal** – a brief pattern of images or a structured story.
- **Nimshal** – follow-up instruction or explanation of the story, analogies, or juxtapositions.
- **Audience** – targets specific people to invoke thought, reasoning, and extrapolation of the story elements to their lives. Sometimes includes a suggest response to the story elements - the moral consequences of the story.

ARE PARABLES STORIES OF EVERYDAY LIFE?

- If Jesus' parables represented everyday life in Palestine, life must have been very strange.
 - A woman misplaces a small coin, spends all day searching and when she finds the coin throws an expensive late-night party to celebrate.
 - A woman bakes bread in a tiny clay oven, following a sacred recipe that calls for 60 pounds of flour.
 - A boss pays one-hour workers the same as those who work all day becoming furious when people complain.
 - A formal dinner party becomes packed with "street people".
- Historical analysis shows many of these to be fictional and unrealistic for the times – they were not to be taken literally.

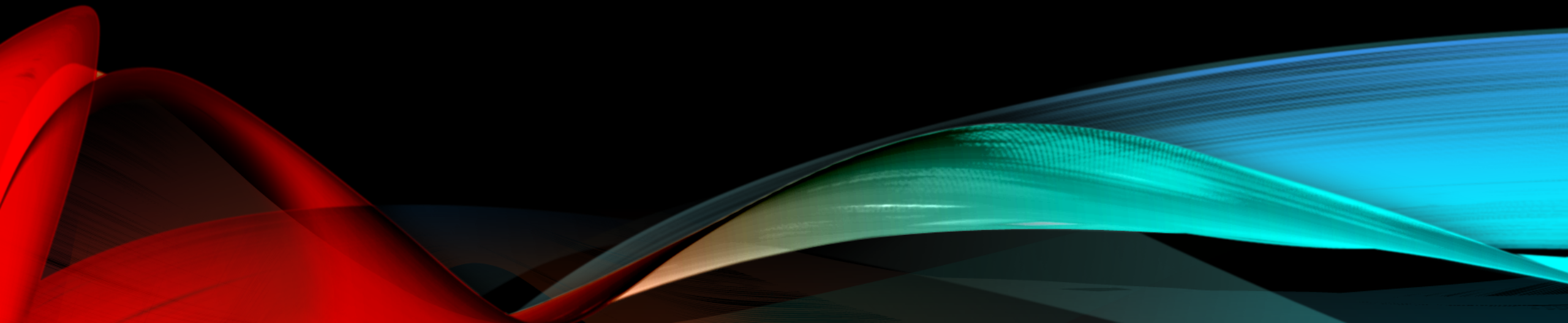
JESUS GAVE HIS PARABLES A PROPHETIC SPIN

- His parables often used if..then parallelism to instruct future moral situations.
 - *If anyone smacks you on the right cheek, then turn to him the other.*
 - *If anyone takes your shirt in a lawsuit, then let him have your topcoat.*
 - *If anyone commandeers you for one mile, then go two.*
- Jesus' parables often have a prophetic message wrapped up in the traditional Jewish form.
- In most parables we find parallel teachings, which not only help us understand, but also help us apply the messages.
- His images call us to thoughtfulness.
- The messages are timeless – not tied to life of the times of Palestine, but also apply to us today.

CONCLUSION

- Jesus is not the only author of parables.
- Parables exist in many areas of the Bible – even the Old Testament.
- Parables may be one-liners or lengthy stories.
- The stories, expertly told, are timeless in their allegories.
- They are also timeless in their moral guidance.
- The parables of Jesus speak to us prophetically and have meaning for every Christian of every age.

THE GOLDEN AGE OF BRITISH COMEDY



And now for something completely different...

- URL: <https://www.youtube.com/watch?v=anwy2MPT5RE>

The logo for Monty Python's Flying Circus, featuring the text "MONTY PYTHON'S FLYING CIRCUS" in a stylized, blocky font. The text is yellow with a red outline and is set against a black background. The logo is tilted slightly to the right.The logo for Monty Python's Flying Circus, featuring the text "MONTY PYTHON'S FLYING CIRCUS" in a stylized, blocky font. The text is yellow with a red outline and is set against a black background. The logo is tilted slightly to the right.

THE SPAM SONG LYRICS

Lovely Spam! Wonderful Spam!
Lovely Spam! Wonderful Spam

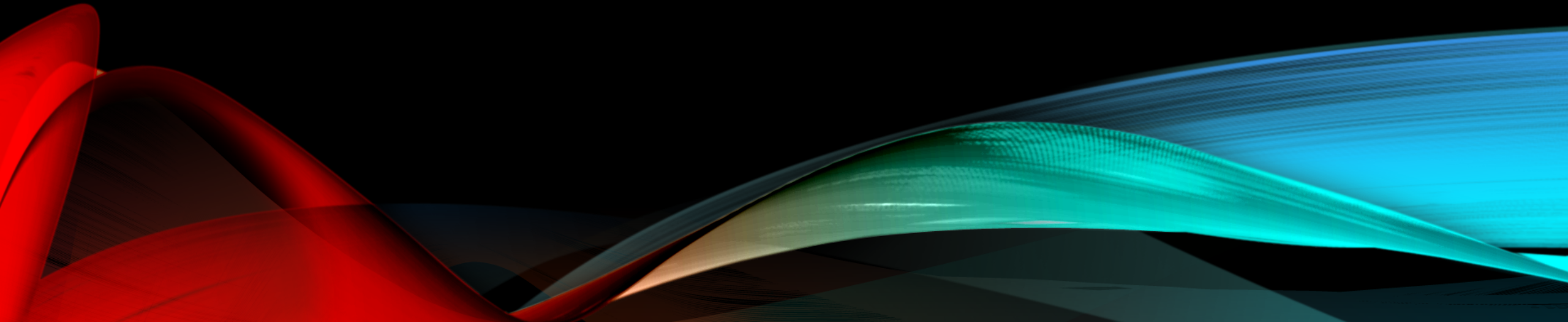
Spa-a-a-a-a-a-a-am
Spa-a-a-a-a-a-a-am
Spa-a-a-a-a-a-a-am
Spa-a-a-a-a-a-a-am

Lovely Spam! (Lovely Spam!)
Lovely Spam! (Lovely Spam!)
Lovely Spam!

Spam, Spam, Spam, Spam!

COMPUTER PROGRAMMING

Introduction to computer programming



WHAT IS PROGRAMMING? (1 OF 2)

- *Programming* means writing instructions to tell the computer to do a certain task.
- Programming is a creative and rewarding activity. Programmers enjoy solving problems and seeing tangible results.
- People write programs for many reasons, from analyzing multi-dimensional data sets to creating an interactive web page to controlling machinery.
- Computers are all around us daily: laptops and cell phones are obvious ones, but computers are also found in cars, home appliances, and entertainment / personal devices.
- If you have ever successfully set a ringtone on your phone, or entered an address in a GPS, or set up a game system to play, then you've had your first experience in programming. You've completed a sequence of steps to tell the computer to do a task.

WHAT IS PROGRAMMING? (2 OF 2)

- If we were to write the instructions to show someone how to set the ringtone on a phone, it would be similar to this:
 1. Go to the phone Settings menu
 2. Choose the Sound menu
 3. Choose the Ringtones menu
 4. Select the ringtone you like
 5. Click OK to save the ringtone that you choose
- The steps above are instructions that we give to our phone, the computer, to perform a task. They are equivalent to a *computer program*.
- When someone calls our phone, the phone (the computer) follows our instructions and plays our chosen ringtone.
- Similar to the “programming” example above, in this class we write instructions that tell the computer to do a specific task, and then we run the program to see the result.



THE BASIC PATTERN

- Most of our programs will use the basic pattern of
 - Get some user input
 - Perform some algorithm on the input
 - Provide results as output

BASIC CONCEPTS OF PROGRAMMING WITH PYTHON

- An interpreter processes python at runtime. No external (separate) compiler is used.
- Python supports object-oriented programming constructs by way of a class.
- Python is a great language for the beginner-level programmers and supports the development of a wide range of applications.
- Python is a scripting language but can be used for a wide range of applications.
- Python is very popular and used throughout the world giving it a huge support base.
- Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- Python code is more clearly defined and visible to the eyes.

IDENTIFIERS

- *Identifiers* are names of various program elements in the code that uniquely identify the elements. They are the names of things like variables or functions to be performed. They're specified by the programmer and should have names that indicate their purpose.
- In Python, identifiers
 - Are made of letters, digits and underscores
 - Must begin with a letter or an underscore
 - Examples: `temperature`, `myPayrate`, `_score2`

KEYWORDS

- *Keywords* are reserved words that have special meaning in the Python language. Because they are reserved, they can not be used as identifiers. Examples of keywords are *if*, *while*, *class*, *import*.

VARIABLES IN PYTHON

- A variable has
 - A name – identifier
 - A data type - int, float, strings, etc. (determined by context implicitly or converted explicitly)
 - `num = 1 + 2`
 - `num = float(25) / 2.5`
 - Storage space sufficient for the type.

INTEGER OPERATORS

- The operations for integers are:
 - + for addition
 - - for subtraction
 - * for multiplication
 - / for integer division: $14/5 = 2$
 - / for floating point division: $14.0/5.0 = 2.8$
 - % for remainder: $14 \% 5 = 4$
- *, /, % take precedence over +, -
 - $x + y * z$ will do $y*z$ first
- Use parentheses to dictate order you want.
 - $(x+y) * z$ will do $x+y$ first.

PYTHON ASSIGNMENT STATEMENTS

- In Python, = is called the *assignment operator* and an *assignment statement* has the form
 <variable> = <expression>
- Here, <variable> would be replaced by an actual variable and <expression> would be replaced by an expression

PYTHON ASSIGNMENT STATEMENT

- **Syntax:** `<variable> = <expression>`
 - Note that variable is on left
- **Semantics:**
 - Compute value of expression
 - Store this as new value of the variable
- **Example:** `Pay = PayRate * Hours`

$$\boxed{400} = \boxed{10} * \boxed{40}$$

Pay Payrate Hours

COMMENTS

- Often we want to put some documentation in our program. These are comments for explanation, but not executed by the computer.
- If we have # anywhere on a line, everything following this on the line is a comment – ignored
- It is always a good practice to document your code!
- Not only to let others know what it does, but to remind yourself as well!

```
#  
# Parables and Pythons  
#  
# Lesson 2: Fahrenheit to Celsius Example  
#  
# This script reads a value from the user in Fahrenheit and converts it to  
# Celsius  
#
```

INPUT

- To get numerical input from the user, we use an assignment statement of the form
 <variable> = input(<prompt>)
- Here, <prompt> would be replaced by a prompt for the user inside quotation marks
 - If there is no prompt, the parentheses are still needed
- Semantics
 - The prompt will be displayed
 - User enters number
 - Value entered is stored as the value of the variable

```
fahrenheit = input("Please enter a temperature in Fahrenheit: ")
```

OUTPUT

- For output we use statements of the form
 - `print(<expression>)`
- Semantics
 - Value of expression is computed
 - This value is displayed
- Several expressions can be printed using parameters wrapped in {}

```
print("{0}F == {1:.2f}C".format(fahrenheit, celsius))
```


HOW TO EXECUTE PYTHON SCRIPTS

- You can use the Python interpreter or the Python IDLE application, but that has limited usability – mostly for testing statements.
- Python scripts are saved in a file and executed in batch mode.
- The script files are named with the .py file extension.
- Example:

```
$ python3 ./fahrenheit_to_celsius.py  
Please enter a temperature in Fahrenheit: 99  
99F == 37.22C
```

EXAMPLE - FAHRENHEIT TO CENTIGRADE

- We want to convert a Fahrenheit temperature to Celsius.
- The formula is $C = (F - 32) \times 5/9$
- We use type float for the temperatures.

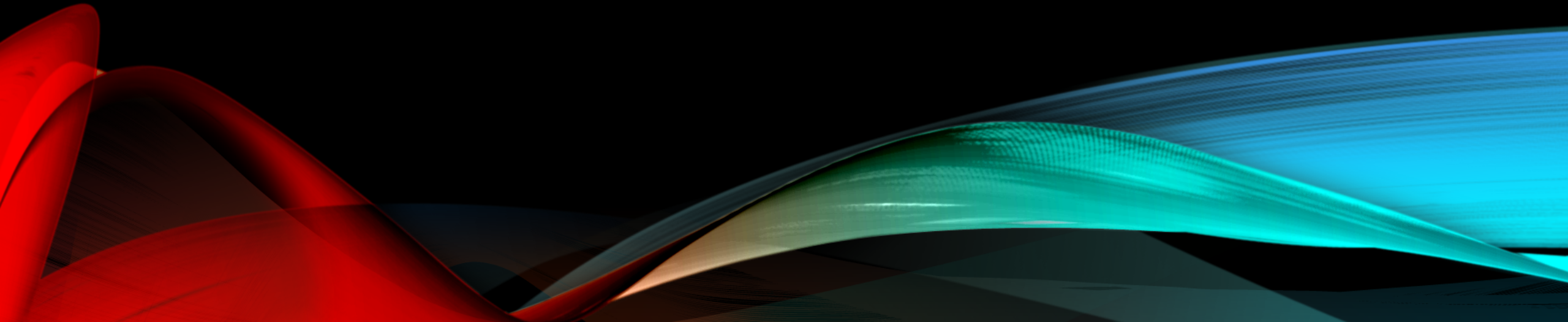
```
#  
# Parables and Pythons  
#  
# Lesson 2: Farenheit to Celsius Example  
#  
# This script reads a value from the user in Fahrenheit and converts it to  
# Celsius  
#  
fahrenheit = input("Please enter a temperature in Fahrenheit: ")  
celsius = (float(fahrenheit) - 32.0) * (5.0/9.0)  
print("{0}F == {1:.2f}C".format(fahrenheit, celsius))
```

- Example execution:

```
$ python3 ./fahrenheit_to_celsius.py  
Please enter a temperature in Fahrenheit: 99  
99F == 37.22C
```

COMPUTER PROGRAMMING

Class Repository: Github





CLASS REPOSITORY

- All of the class files will be saved online in a special repository that you can access from home.
- It uses a service called Github
- URL: <https://github.com/ChuckBell/Parables-and-Pythons>

Using the Class Repository (Github)

- What is Github?
 - Github is a code hosting and sharing website
 - Github, like many code hosting websites, allows for public access to code but private access to updates (can only update by invitation)
 - This allows for interested programmers to take part in furthering development
- Why use code sharing services?
 - Easy to distribute work
 - Easy to improve on the work of others
 - Easy to take help from others
- There are three ways to use Github.
 - Browse to the files and download them individually and/or view the files.
 - Download the files in a .zip file and extract it on your PC.
 - Clone the repository (advanced)

USING GITHUB: DOWNLOAD

ChuckBell / Parables-and-Pythons

Unwatch 1

Star 0

Fork 0

<> Code

Issues 0

Pull requests 0

Projects 0

Wiki

Insights

Settings

This repository contains supporting materials for my class "Parables and Pythons". The class is from my series of technology and the Bible classes for teens and older.

[Manage topics](#)

4 commits

1 branch

0 releases

1 contributor

GPL-3.0

Branch: master

New pull request

Create new file

Upload files

Find file

Clone or download

ChuckBell Updated the readme

homework

Added lesson 1 slides and homework.

slides

Added lesson 1 slides and homework.

LICENSE

Initial commit

README.md

Updated the readme

Clone with HTTPS

Use Git or checkout with SVN using the web URL.

https://github.com/ChuckBell/Parables-and-Pythons.git

Open in Desktop

Download ZIP

CLONING THE REPOSITORY

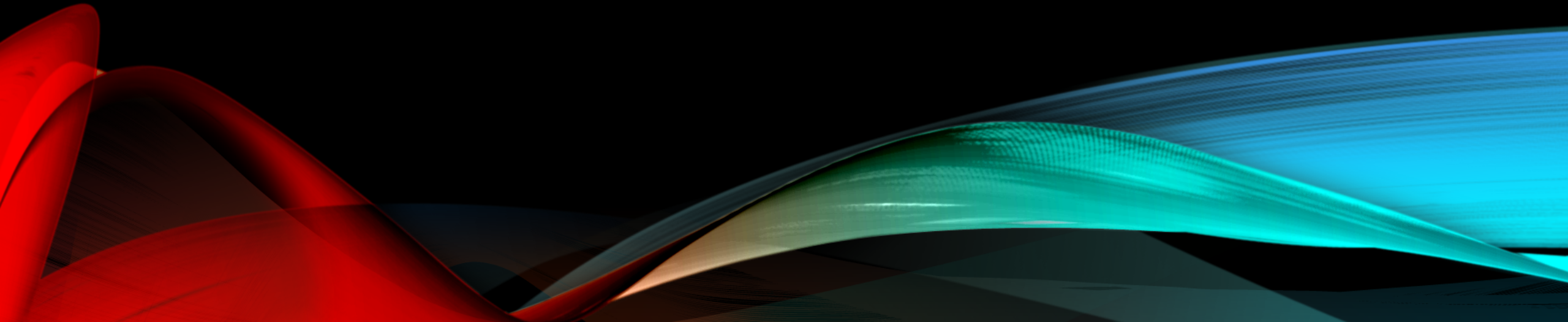
- The best way to use a Github repository is to “clone” it.
- This keeps an active, read-only copy of the repository on your computer.
- Requires you to install “Git”, which can be downloaded from <https://git-scm.com/downloads>
- Concepts
 - Clone/cloning: make a read-only copy
 - Pull: retrieve the latest changes
 - Push: send changes to the repository (restricted)

CLONING THE REPOSITORY

```
MacBook-Pro:~ cbell$ cd Documents/
MacBook-Pro:Documents cbell$ mkdir class
MacBook-Pro:Documents cbell$ cd class
MacBook-Pro:class cbell$ git clone https://github.com/ChuckBell/Parables-
and-Pythons.git
Cloning into 'Parables-and-Pythons'...
remote: Counting objects: 17, done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 17 (delta 3), reused 7 (delta 1), pack-reused 0
Unpacking objects: 100% (17/17), done.
MacBook-Pro:class cbell$ cd Parables-and-Pythons/
MacBook-Pro:Parables-and-Pythons cbell$
```

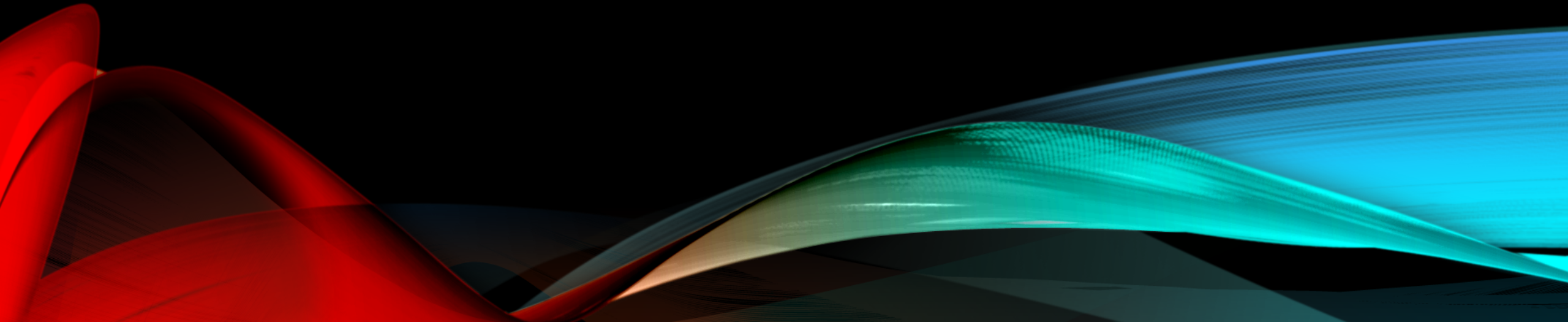
CLONE THE REPOSITORY

Demonstration

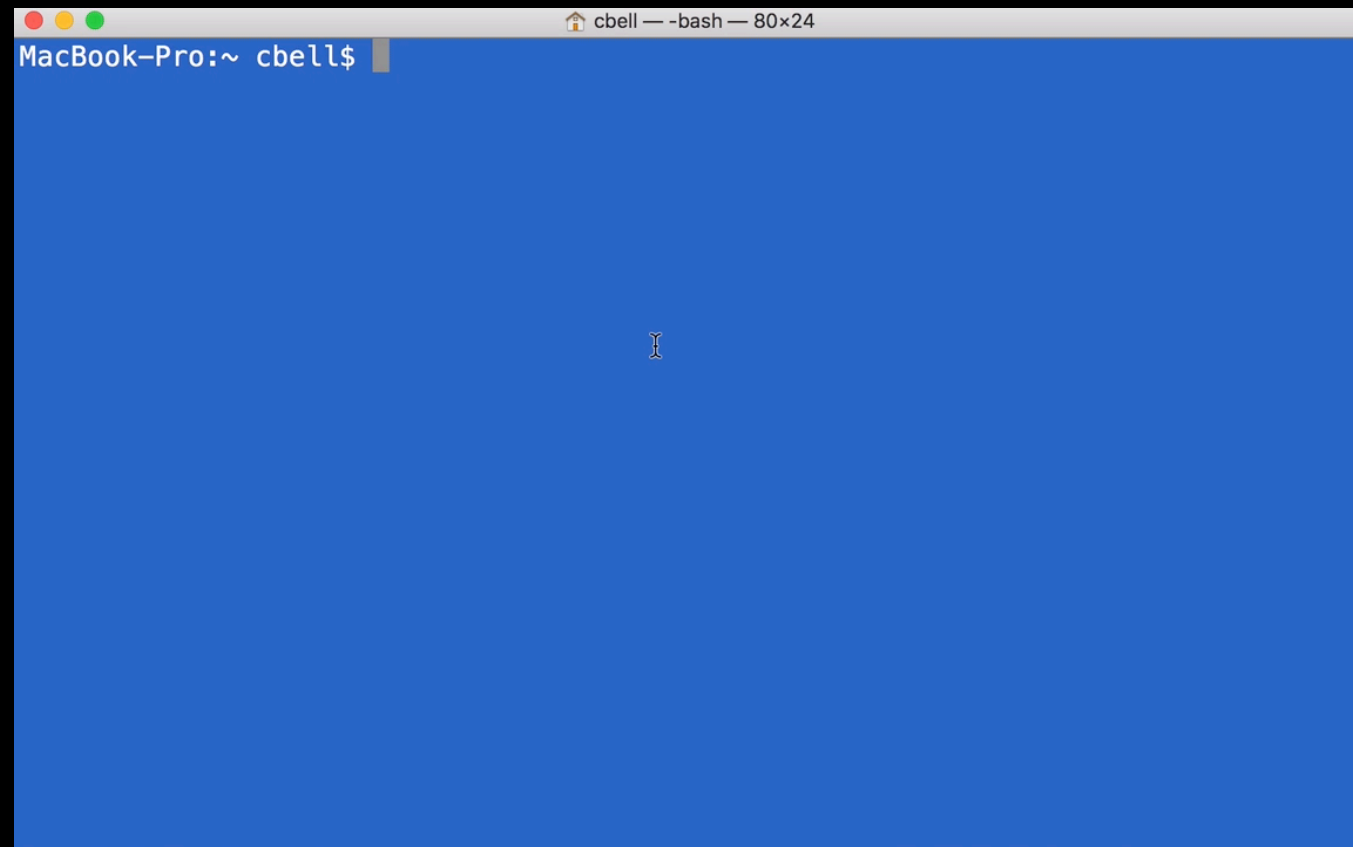


HOMEWORK

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



HOMEWORK #1 - REVIEW



HOMework ASSIGNMENT #2

- (0 points) Download the class files from Github
- (2 points) Create a file named `celsius_to_fahrenheit.py` to convert Celsius to Fahrenheit. Turn in a transcript of at least 4 values converted.
 - Hint: you must reverse the formula.
 - Hint: `python ./celsius_to_fahrenheit.py`.
- (1 point) Install Git on your computer and clone the repository. Turn in the output of the command, “git pull”.

QUESTIONS OR COMMENTS?

