Adventure Project

Welcome! The Adventure Project is a Python programming introduction course with the outcome being the start of an Adventure Game.

This Python programming introduction is IDE (Integraded Development Environment) independent. All of the lessons can be completed with any python editor. Understanding the foundations of programming is critical for future programming interests and problem-solving in general. This project does not focus on term and definition memorization. It does focus on interest driving learning and my underlying goal is to sparking curiosity in programming! Each section introduces new concepts and gives learners a chance to dive deeper into the content while maintaining focus on completing the framework for an Adventure Game.

What do you need to know in order to complete the Adventure Project?

The Adventure Project starts right after 'hello world'! Students should have the knowledge of and be comfortable with opening a python IDE (i.e., RepL.it) and type a couple lines of code (i.e., print() functions) to produce the standard 'hello world' script. The Adventure Project learning is divided into sections called 'Labs' with one or more programming concepts explained and demonstrated with code examples. Each section has a specific programming concept and focus for the Adventure game framework. Once all nine Labs are completed the adventure game framework is ready to begin coding the adventure!

Lab Assignments

Each Lab is estimated to take between 20 and 30 minutes to complete, with a total time of 3 hours to complete all nine labs. Once familure with the programming concepts the time is easily cut in half to recreate. If this if your first time programming, it is highly recommended to focus on the specific learning objective and complete all Labs in order. This project is meant to be easy to navigate and quick to complete! (3 hours is really quick :)

Learning Breakdown

Each Lab includes a 'before you start' section. This section lists the items you need in order to successfully complete the lab. This includes links to resources, lab names, and general information about the lab. Next is the Read.Me file. If you have access to the Adventure Project RepL's, the Read.Me file for each lab is where all of the good stuff is located! This includes the step-by-step instructions, specific python content covered, lab assignment point values, and a rubric to guide you to successful completion. All of that information is located here as well.

Lastly, each Adventure lab includes discussions on specific python content and programming objectives. These are code explanations and examples to help understand the what and how. At the end of most Labs the entire code solution is given. Use the solution code more as a guide to what the lab is trying to accomplishto and less of here is the anwser and you are done! Your solution 'should' be similar but does not have to be exactly like mine. :)

-Enjoy!

Adventure Labs

- Adventure 1-1 - Comments, print(), Variables, Casting
- Adventure 1-2 - User Input()
- Adventure 1-3 - functions 1
- Adventure 1-4 - functions 2
- Adventure 1-5 - functions 3
- Adventure 1-6 - Colors
- Adventure 1-7 - User Input
- Adventure 1-8 - ASCII Art
- Adventure 1-9 - If & elif Statements

Extras

- Adventure Points and Rubrics per Lab
- Resources

Adventure Project

Adventure 1-1 Comments, print(), variables, and casting

```
- Before you start
- Read.Me file
```

Steps:

```
1. Comments
2. Splash Screen
3. Welcome Message
4. Creating Variables
5. Display User Information

- Solution
```

Before you start:

- Basic python rules to keep in mind.

- Know how to insert comments in python.

- Understand the how to use the print() function

- Knowledge of python numbers and operators

- Create a new Repl

- Rename your Repl as 'yourLastNameAdv1-1'

- Share your Repl with your instructor!

- Read the Read.Me file!

- Complete all steps

Read.ME

Welcome to the adventure game. We are starting by creating a working environment that is ready for our amazing adventure. To set this up we are using comments to explain what we are trying to accomplish with this Lab. Our first message to the user is a splash screen. Splash screens are used for many reasons and we are using it to let the user know the name of our game. Next is our welcome message and to let the user know we are about to start the game.

After we display our messages to the user, we will create several variables and then display the data to the console.

In [ ]:
```
####################################################
##################      Adventure 1-1    ###########
####################################################
####################################################
#                                                  #
# Directions:                                      #
#       - Create a new repl.                       #
#       - Change repl name to: yourLastNameAdv1-1  #
#       - SHARE your repl is shared with me!       #
#       - Read the Read.Me file!                   #
#       - Complete the steps.                      #
#                                                  #
#  ----------------------------------------------- #
# Steps:                                           #
#        1. Enter Documentation Comments           #
#               a. Name                            #
#               b. Date                            #
#               c. Description                     #
#        2. Create a Splash Screen                 #
#        3. Create a Welcome message               #
#        4. Create four (4) Variables              #
#               a. plrName                         #
#               b. plrTitle                        #
#               c. plrLevel                        #
#               d. plrHP                           #
#        5. Assign data to four (4) Variables      #
#               a. Name = "Dave"                   #
#               b. Title = "Warrior"               #
#               c. Level = 7                       #
#               d. HP = 123                        #
#        6. Display variable data                  #
#               a. Print two (2) result messages   #
#               b. Include 2 variables within each message#
#               c. Cast int data type to str data type #
#                                                  #
#  ----------------------------------------------- #
# Notes:                                           #
#          Read all Instructions!                  #
#          Errors are expected. Start slow. Fix one#
#               Error at a time.                   #
#          ValueErrors are typically solved by casting #
#                                                  #
#  ----------------------------------------------- #
# Python Content:                                  #
#          Comments (inline & multi-line)          #
#          print() Function                        #
#          String Variables                        #
#          Interger Variables                      #
#          New Line Escape Character '\n'          #
#          Create string variable                  #
#          Create integer variable                 #
#          Cast variables: print("today is" + int(day)) #
#  ----------------------------------------------- #
```

1-1.1. Comments

Comments: Starting at line 1. Add comments to give basic information about your app. Include the following information:

- Name
- Date
- Description

In [ ]:
```
###########################
#                         #
# 1a. Name comment        #
# 1b. Date comment        #
# 1c. Description comment #
#                         #
###########################
#  Ernie Drumm, 9/1/2020
#  Description: Adventure 1-1
#               Comments
#               print() functions
#               Variables
#               Sring data type
#               Numeric data type
#               Casting data types!
#               New line escape character \n
```

1-1.2. Splash Screen

Create a Splash screen to let the user know the name of our app. This helps the user remember the name of the app. Lastly, use the new-line escape character '\n' to print several blank lines.

In [ ]:
```
###########################
#                         #
# 2. Create a Splash Screen  #
#                         #
###########################
print(">>>>>> Awesome Adventure! <<<<<<")
print("\n" * 5)
```

1-1.3. Welcome Message

Next, is the welcome message. This message is giving the user a welcome to the game and getting the excited to play the game. Why are we doing this? It may not make sense yet, but we are we are building the frame work for a larger part of the game. It starts with a simple message.

In [ ]:
```
###########################
#                         #
# 3. Create a Welcome Screen #
#                         #
###########################
print("Welcome to the Awesome Adventure!\n")
print("\n" * 3)
```

1-1.4. & 5. Variables and Data

Create four variables and assign data.

```
- Player Name with the value of "Dave"
- Player Title with the value of "Warrior"
- Player Level with the value of 7
- Player Hit Points with the value of 123
```

In [ ]:
```
#####################################
#                                   #
# 4. Create Player variables        #
# 5. Assign data to Player variables #
#                                   #
#####################################
plrName = "Dave"
plrTitle = "Warrior"
plrLevel = 7
plrHP = 123
```

1-1.6. Display User Information

```
- Display all the player data.
- Friendly formatted!
- Include:
        -Player Name
        -Player title
        -Player Level
        -Player Hit Points
```

In [ ]:
```
# 6a. Print result messages
# 6b. Include 2 variables within a message
# 6c. Cast int data type to str data type
print("Welcome,", plrTitle, plrName )
print()
print("You are level " + str(plrLevel) + " with " + str(plrHP) + " hit points.")
print()
print("Looking forward to our adventure together" , plrName + "!\n")
```

Caution: In general, TypeError happen when using the '+' concatenator in a print() statement and a variable happens to be a integer or float data type. This is easy to solve by casting the data type to a string type. For example, below we set the 'myInteger' variable to an integer data type 'myInteger = 10'. The first print() statement is successful because we cast the variable to a string 'str(myInteger)'. The second print() statement causes a TypeError because a print() functions can only concatenate a string and 'myInteger' variable is an integer data type.

In [ ]:
```
myInteger = 10

print("Successfully casting an integer to string = " + str(myInteger))
print("Unsuccessful!" + myInteger)
```

Solution to Adventure Lab 1-1

In [ ]:
```
###########################
#                         #
# 1a. Name comment        #
# 1b. Date comment        #
# 1c. Description comment #
#                         #
###########################
#  Ernie Drumm, 9/1/2020
#  Description: Adventure 1-1
#               Comments
#               print() functions
#               Variables
#               Sring data type
#               Numeric data type
#               Casting data types!
#               New line escape character \n

###########################
#                         #
# 2. Create a Splash Screen  #
#                         #
###########################
print(">>>>>> Awesome Adventure! <<<<<<")
print("\n" * 4)

###########################
#                         #
# 3. Create a Welcome Screen #
#                         #
###########################
print("Welcome to the Awesome Adventure!\n")
print("\n" * 2)

#####################################
#                                   #
# 4a. Create Player Name variable   #
# 5a. Assign "Dave" to Player Name  #
#                                   #
#####################################
plrName = "Dave"

#####################################
#                                   #
# 4b. Create Player Title variable   #
# 5b. Assign "Player" to Player Title #
#                                   #
#####################################
plrTitle = "Warrior"

#####################################
#                                   #
# 4c. Create Player Level variable  #
# 5c. Assign 7 to Player Level      #
#                                   #
#####################################
plrLevel = 7

#####################################
#                                   #
# 4d. Create Player HP variable     #
# 5d. Assign 123 to Player HP       #
#                                   #
#####################################
plrHP = 123

#####################################
#                                   #
# 6a. Result messages               #
#                                   #
#####################################
# 6b. Included variables within message
print("Welcome,", plrTitle, plrName )
print()

# 6c. Cast int data type to str data type
print("You are level " + str(plrLevel) + " with " + str(plrHP) + " hit points.")
print()

print("Looking forward to our adventure together" , plrName + "!\n")
```

Top

Adventure Lab 1-2

Adventure 1-2 User Input

```
- Before you start
- Read.Me file
```

Steps

```
1. Update Comments
2. Update Splash Screen
3. Update Welcome Message
4. Use input() function to gather user information
5. Display user information

- Solution
```

Before you start

- Fork your Repl yourLastNameAdv1-1

- Rename new Repl to yourLastNameAdv1-2'

- Share your Repl with your instructor.

- Read the document on input() functions

- Read the Read.Me file!

- Complete all steps.

Read.Me file

In [ ]:
```
####################################################
##################      Adventure 1-2    ###########
##################       User Input      ###########
####################################################
#                                                  #
# Directions:                                      #
#       - Fork yourLastNameAdv1-1                  #
#       - Save repl as: yourLastNameAdv1-2         #
#       - SHARE your repl is shared with instructor! #
#       - Read the Read.Me file!                   #
#       - Complete all steps.                      #
#                                                  #
#  ----------------------------------------------- #
# Steps:                                           #
#        1. Update Comments                        #
#        2. Update Splash Screen                   #
#               a. Add 'Adventure' version number  #
#        3. Update Welcome Message                 #
#               a. print message letting user know they #
#                  are going to be asked for information #
#        4. Use input() statements to gather user data #
#               a. Create four variables for user input #
#                  plrName, plrTitle, plrLevel, and plrHP#
#               b. Use input() statements to gather input #
#                  from user.                      #
#        5. Display User Information               #
#               a. Print two (2) result messages   #
#               b. Include 2 variables within each message#
#               c. Cast int data type to str data type #
#                                                  #
#  ----------------------------------------------- #
# New Python Content:                              #
#          Casting within print() functions        #
#          More String Variables                   #
#          input() Function                        #
#                                                  #
#  ----------------------------------------------- #
```

1-2.1. Update Comments

Every time you fork or make a copy of your Adventure project you need to update comments and version number. This will help you keep track or your progress and assist the instructor in grading your work!

In [ ]:
```
#  Ernie Drumm, 9/2/2020
#  Description: Adventure 1-2
#               print() function
#               More Variables
#               Sring data type
#               Numeric data type
#               Casting
```

1-2.2 Update Splash Screen

This update should include the Adventure version number. Continue to add friendly formatting like the new line escape character '\n' to force new lines in the console display.

In [ ]:
```
##################################################
#                                                #
# 2. Update Splash Screen                        #
# 2a. Add 'Adventure' version number             #
#                                                #
##################################################
print(">>>>>>>>>    Awesome Adventure!   <<<<<<<<<")
print("\n" * 2)
print("---> Adventure v1-2")
print("\n" * 3)
```

1-2.3 Update Welcome Message

Add another print statement to let the user know we need them to answer several questions. Continue to add friendly formatting like the new line escape characters '\n' to force new lines in the console display.

In [ ]:
```
###########################
#                         #
# 3a. Update Welcome message #
#                         #
###########################
print("Welcome to the Awesome Adventure!")
print("\n" * 2)
print("Before we start we need to ask you several questions.")
print("\n" * 2)
```

1-2.4. Use input() statements to gather user data

Python uses the input() function to get input from the console window. For each of the player variables we created ealier, (plrName, plrTitle, plrLevel, and plrHP) we will use the input() statement to ask the user to enter their own data.

Each input() function needs to include a question. This is the prompt for the user to see and answer.

In [ ]:
```
##################################################
#                                                #
# 4a. Create four variables for user input       #
#     plrName, plrTitle, plrLevel, and plrHP     #
# 4b. Use input() statements to gather input     #
#                                                #
##################################################
plrName = input("What is your first name? ")
plrTitle = input("What is your title: ")
plrLevel = input("What is your level? ")
plrHP = input("What is your HP? ")
```

1-2.5 Display user information

Lastly, display the information the user entered.

In [ ]:
```
###########################
#                         #
# 5. Display User Information #
#                         #
###########################
# 5a. Print two (2) result messages
# 5b. Include 2 variables within each message
print("Welcome,", plrTitle, plrName )
print()

# 5c. Cast int data type to str data type
print("You are level " + str(plrLevel) + " with " + str(plrHP) + " hit points.")
print()

print("Looking forward to our adventure together" , plrName + "!\n")
```

Solution to Adventure 1-2

In [ ]:
```
###########################
# 1. Update comments:     #
#      Name, Date, and    #
#      Description        #
#                         #
###########################
#  Ernie Drumm, 9/2/2020
#  Description: Adventure 1-2
#               print() function
#               Variables
#               Sring data type
#               Numeric data type

##################################################
#                                                #
# 2. Update Splash Screen                        #
# 2a. Add 'Adventure' version number             #
#                                                #
##################################################
print(">>>>>>>>>    Awesome Adventure!   <<<<<<<<<")
print("\n" * 2)
print("---> Adventure v1-2")
print("\n" * 3)

###########################
#                         #
# 3a. Update Welcome message #
#                         #
###########################
print("Welcome to the Awesome Adventure!")
print("\n" * 2)
print("Before we start we need to ask you several questions!")
print("\n" * 2)

##################################################
#                                                #
# 4a. Create four variables for user input       #
#     plrName, plrTitle, plrLevel, and plrHP     #
# 4b. Use input() statements to gather input     #
#                                                #
##################################################
plrName = input("What is your first name? ")
plrTitle = input("What is your title: ")
plrLevel = input("What is your level? ")
plrHP = input("What is your HP? ")

###########################
#                         #
# 5. Display User Information #
#                         #
###########################
# 5a. Print two (2) result messages
# 5b. Include 2 variables within each message
print("Welcome,", plrTitle, plrName )
print()

# 5c. Cast int data type to str data type
print("You are level " + str(plrLevel) + " with " + str(plrHP) + " hit points.")
print()

print("Looking forward to our adventure together" , plrName + "!\n")
```

Top

Adventure Lab 1-3

Adventure 1-3 (functions 1)

```
- Before you start
- Read.Me
```

Steps

```
1. Update Comments
2. Update version number
3. Create function for Splash Screen
      a. Create pSplash function
      b. Call pSplash function
4. Create functions for printing messages:
      a. Create pWelcome function
      b. Call pWelcome function
5. Use input() function to gather user information
6. Display user information

- Solution
```

Before you start

- Fork your Repl yourLastNameAdv1-2

- Rename new Repl to yourLastNameAdv1-3'

- Share your Repl with your instructor.

- Read the document on functions.

- Read the Read.Me file!

- Complete all steps.

Read.Me file

```
#################################################
##################     Adventure 1-3     ########
##################      Functions 1      ########
#################################################
#                                               #
# Directions:                                   #
#       - Fork yourLastNameAdv1-2               #
#       - Save repl as: yourLastNameAdv1-3      #
#       - SHARE your repl is shared with instructor! #
#       - Read the Read.Me file!                #
#       - Complete all steps.                   #
#                                               #
#  ---------------------------------------------#
# Steps:                                        #
#         1. Update Comments                    #
#         2. Update Splash Screen               #
#              a. Update version number         #
#         3. Create Function for Splash Screen  #
#              a. Create pSplash function       #
#              b. Call pSplash function         #
#         4. Create Function for Welcome Message #
#              a. Create pWelcome function      #
#              b. Call pWelcome function        #
#         5. Use input() statements to gather user data #
#         6. Display User Information           #
#                                               #
#  ---------------------------------------------#
# New Python Content:                           #
#              Functions                        #
#                                               #
#  ---------------------------------------------#
```

1-3.2a Create pSplash function:

We are creating a function for the Splash Screen for several reasons. Mainly, so we can reuse the code whenever we want and also to move the code out of the main.py file (eventually :).

```
#########################
#                       #
# 3a. Splash Function   #
#                       #
#########################
def pSplash():
  print(">>>>>>>>>>   Awesome Adventure!   <<<<<<<<<<")
  print("\n" * 2)
  print("---> Adventure v1-3")  #2a. Update Adventure version
  print("\n" * 3)
```

Caution: Be aware of Indentation Errors. If the print() statements below the 'def pSplash():' statement is not indented correctly python will throw an IndentationError. See the myExample() function below where I did not indent correctly.

```
def myExample():
print("Some cool text here")
print("\n" * 2)
print("more text")
```

1-3.4 Create pWelcome function:

This is the same process as above. Notice I have included additional print statements. I wanted to notify the user of the information they need to provide and a funny reason for why it is needed.

```
###############################
#                             #
# 4a. Create pWelcome function #
#                             #
###############################
def pWelcome():
  print("Welcome to the Awesome Adventure!")
  print("\n" * 2)
  print("Before we start we need to ask you several questions!")
  print("\n" * 2)
  print("We need your name, title, level, and hit points!")
  print()
  print("This is REQUIRED to verify if you are skilled for this adventure!\n")
```

1-3.3b. Call pSplash function:

Calling functions is simple. All that is needed it the name of the function followed by empty parentheses. These functions have no arguments, so there is nothing to include in the parentheses.

```
############################
#                          #
# 3b. Call pSplash function #
#                          #
############################
pSplash()
```

1-3.4b. Call pWelcome function:

```
#############################
#                           #
# 4b. Call pWelcome function #
#                           #
#############################
pWelcome()
```

Solution to Adventure 1-3

```
# 1. Update Comments
#  Ernie Drumm, 9/3/2020
#  Description:
#                 -Adventure 1-3
#                 -Create functions
#                 -Call functions

#########################
#                       #
# 3a. Splash Function   #
#                       #
#########################
def pSplash():
  print(">>>>>>>>>>   Awesome Adventure!   <<<<<<<<<<")
  print("\n" * 2)
  print("---> Adventure v1-3")  #2a. Update Adventure version
  print("\n" * 3)

###############################
#                             #
# 4a. Create pWelcome function #
#                             #
###############################
def pWelcome():
  print("Welcome to the Awesome Adventure!")
  print("\n" * 2)
  print("Before we start we need to ask you several questions!")
  print("\n" * 2)
  print("We need your name, title, level, and hit points!")
  print()
  print("This is REQUIRED to VERIFY you are skilled for this adventure!\n")

############################
#                          #
# 3b. Call pSplash function #
#                          #
############################
pSplash()

#############################
#                           #
# 4b. Call pWelcome function #
#                           #
#############################
pWelcome()

###############################################
#                                             #
# 5. Use input() to gather user data          #
#                                             #
###############################################
plrName = input("What is your first name? ")
plrTitle = input("What is your title: ")
plrLevel = input("What is your level? ")
plrHP = input("What is your HP? ")

##############################
#                            #
# 6. Display User Information #
#                            #
##############################
print("Welcome,", plrTitle, plrName )
print()
print("Your are level", plrLevel, "with", plrHP, "hit points.")
print()
print("Looking forward to our adventure together" , plrName + "!\n")
```

Top

Adventure Lab 1-4

Adventure 1-4 (functions 2)

- Before you start
- Read.Me

Steps

1. Update Comments
2. Update Version Number
3. Create functions printing results
    a. Create pResults function
    b. Pass arguments into pResults function
    c. Use arguments in pResults function
    d. Call pResults function with four arguments
4. Use input() function to gather user information
5. Display user information

- Solution

Before you start

- Fork repl and save as 'Adventure1-4'

- Share your repl

- Read the document on functions.

- Read the Read.Me file

- Complete all steps

Read.Me file

```
#################################################
##################     Adventure 1-4     ########
##################      Functions 2      ########
#################################################
#                                               #
# Directions:                                   #
#       - Fork yourLastNameAdv1-3               #
#       - Save repl as: yourLastNameAdv1-4      #
#       - SHARE your repl is shared with instructor! #
#       - Read the Read.Me file!                #
#       - Complete all steps.                   #
#                                               #
#  ---------------------------------------------#
# Steps:                                        #
#         1. Update Comments                    #
#         2. Update Version Number              #
#         3. Create function for printing results #
#              a. Create pResults function      #
#              b. Pass arguments into pResults function #
#              c. Use arguments in pResults function #
#              d. Call pResults function with four #
#                   arguments                   #
#                                               #
#  ---------------------------------------------#
# New Python Content:                           #
#              Passing Arguments into functions #
#              Using Argument data within functions #
#                                               #
#  ---------------------------------------------#
```

1-4.3a-c. Create pResults function for printing results.

```
##############################################
#                                            #
# 3a. Create pResults function               #
# 3b. Pass arguments into pResults function  #
#        plrName, plrTitle, plrLevel, plrHP  #
#                                            #
##############################################
def pResults(plrName, plrTitle, plrLevel, plrHP):
  print("Welcome,", plrTitle, plrName) #3c. Use arguments
  print()
  print("Your are level", plrLevel, "with", plrHP, "hit points.") #3c.
  print()
  print("Looking forward to our adventure together" , plrName + "!\n") #3c.
```

1-4.3d. Call pResults function with four arguments

```
#################################################
#                                               #
# 3d. Call the pResults function with four      #
#       arguments                               #
#                                               #
#     - Pass in four pieces of information      #
#         we received from user input as        #
#         arguments.                            #
#                                               #
#         plrName, plrTitle, plrLevel, plrHP    #
#                                               #
#################################################
pResults(plrName, plrTitle, plrLevel, plrHP)
```

Solution to Adventure 1-4

```
# 1. Update Comments
#  Ernie Drumm, 9/4/2020
#  Description:
#                 -Adventure 1-4
#                 -Create functions
#                 -Call functions
#                 -Add arguments to functions
#                 -return results from functions

# pSplash Function
def pSplash():
  print(">>>>>>>>>>   Awesome Adventure!   <<<<<<<<<<")
  print("\n" * 2)
  print("---> Adventure v1-4")  #2. Update Adventure version
  print("\n" * 3)

# pWelcome function
def pWelcome():
  print("Welcome to the Awesome Adventure!")
  print("\n" * 2)
  print("Before we start we need to ask you several questions!")
  print("\n" * 2)
  print("We need your name, title, level, and hit points!")
  print()
  print("This is REQUIRED to VERIFY you are skilled for this adventure!\n")

pSplash()   # Call pSplash Function
pWelcome()  # Call pWelcome Function

# Use input() to gather user data
plrName = input("What is your first name? ")
plrTitle = input("What is your title: ")
plrLevel = input("What is your level? ")
plrHP = input("What is your HP? ")

##############################################
#                                            #
# 3a. Create pResults function               #
# 3b. Pass arguments into pResults function  #
#        plrName, plrTitle, plrLevel, plrHP  #
#                                            #
##############################################
def pResults(plrName, plrTitle, plrLevel, plrHP):
  print("Welcome,", plrTitle, plrName) #3c. Use arguments
  print()
  print("Your are level", plrLevel, "with", plrHP, "hit points.") #3c.
  print()
  print("Looking forward to our adventure together" , plrName + "!\n") #3c.

#################################################
#                                               #
# 3d. Call the pResults function with four      #
#       arguments                               #
#                                               #
#     - Pass in four pieces of information      #
#         we received from user input as        #
#         arguments.                            #
#                                               #
#         plrName, plrTitle, plrLevel, plrHP    #
#                                               #
#################################################
pResults(plrName, plrTitle, plrLevel, plrHP)
```

Top

Adventure Lab 1-4

Adventure 1-5 (functions 3)

- Before you start
- Read.Me

Steps

1. Update Comments
2. Update Version number
3. messages.py file.
    a. create messages.py file
4. Move pSplash() function
    a. Cut and paste pSplash() function into messages.py file
    b. add import statement for pSplash() function in main.py file
    c. Call pSplash() function from main.py
3. Move pWelcome() function
    a. Cut and paste pWelcome() function into messages.py file
    b. add import statement for pWelcome() functin in main.py file
    c. Call pSplash() function from main.py
3. Move pResults() function
    a. Cut and paste pResults() function into messages.py file
    b. add import statement for pResults() function in main.py file
    c. Call pSplash() function from main.py

Solution files

- main.py file
- messages.py file

Before you start

- Fork repl and save as 'Adventure1-5'

- Share your repl with Instructor

- Read the document on functions.

- Read the Read.Me file

- Complete all steps

Read.Me file

```
#################################################
##################     Adventure 1-5     ########
##################      Functions 3      ########
#################################################
#                                               #
# Directions:                                   #
#       - Fork yourLastNameAdv1-4               #
#       - Save repl as: yourLastNameAdv1-5      #
#       - SHARE your repl with instructor!      #
#       - Read the Read.Me file!                #
#       - Complete all steps.                   #
#                                               #
#  ---------------------------------------------#
# Steps:                                        #
#         1. Update Comments                    #
#         2. Update Version Number              #
#         3. messages.py file.                  #
#              a. create messages.py file       #
#         4. Move pSplash() function            #
#              a. Cut and paste pSplash() function into #
#                   messages.py file            #
#              b. add import statement for pSplash() #
#                   function in main.py file    #
#              c. Call pSplash() function from main.py #
#         5. Move pWelcome() function           #
#              a. Cut and paste pWelcome() function #
#                   into messages.py file       #
#              b. add import statement for pWelcome() #
#                   function in main.py file    #
#              c. Call pWelcome() function from main.py #
#         6. Move pResults() function           #
#              a. Cut and paste pResults() function #
#                   into messages.py file.      #
#              b. add import statement for pResults() #
#                   function in main.py file.   #
#              c. Call pResults() function from main.py #
#                   include four arguments.     #
#                                               #
#  ---------------------------------------------#
# New Python Content:                           #
#              Passing Arguments into functions #
#              Using Argument data within functions #
#                                               #
#  ---------------------------------------------#
```

main.py file

```
# 1. Update Comments
#  Ernie Drumm, 9/5/2020
#  Description:
#                 -Adventure 1-5
#                 -Using external files for functions

#############################################
#                                           #
# 4b. add import statement for pSplash()    #
# 5b. add import statement for pWelcome()   #
# 6b. add import statement for pResults()   #
#                                           #
#############################################
from messages import pSplash, pWelcome, pResults

pSplash()    # 4c. Call pSplash Function
pWelcome()   # 5c. Call Welcome Function

# Use input() to gather user data
plrName = input("What is your first name? ")
plrTitle = input("What is your title: ")
plrLevel = input("What is your level? ")
plrHP = input("What is your HP? ")

##################################
#                                #
# 6c. Call the pResults function #
#     with four arguments        #
##################################
pResults(plrName, plrTitle, plrLevel, plrHP)
```

messages.py file

```
##############################################
#                                            #
#   3a. Create file called Messages.py       #
#       This is the file.                    #
#                                            #
##############################################

# 4a. Splash Function
def pSplash():
  print(">>>>>>>>>>   Awesome Adventure!   <<<<<<<<<<")
  print("\n" * 2)
  print("---> Adventure v1-5")  #2. Update Adventure version
  print("\n" * 3)

# 5a. Welcome Function
def pWelcome():
  print("Welcome to the Awesome Adventure!")
  print("\n" * 2)
  print("Before we start we need to ask you several questions!")
  print("\n" * 2)
  print("We need your name, title, level, and hit points!")
  print()
  print("This is REQUIRED to VERIFY you are skilled for this adventure!\n")

##############################################
#                                            #
# 6a. Welcome Function that accepts four     #
#       arguments.                           #
#                                            #
##############################################
def pResults(plrName, plrTitle, plrLevel, plrHP):
  print("Welcome,", plrTitle, plrName )
  print()
  print("Your are level", plrLevel, "with", plrHP, "hit points.")
  print()
  print("Looking forward to our adventure together" , plrName + "!\n")
```

Top

Adventure Lab 1-4

Adventure 1-6 (Colors)

- Before you start
- Read.Me

Steps:

- 1. Update comments
- 2. Update Adventure Version # with a variable.
    a. Create variable 'AdventureVersion'
       and assign value (i.e.,v1-6)
    b. Pass variable to pWelcome() function
- 3. Use escape characters to change text color
    a. in messages.py create functions to
       change color of a string message.
    b. Import color functions in main.py
    c. Change colors in pSplash function
    d. Change colors in pWelcome function
    e. Change colors in pResults function

Solution files

- main.py file
- messages.py file

Before you start

- Fork repl and save as 'Adventure1-6'

- Share repl with instructor

- Read the document on Colors.

- Read the Read.Me file

- Complete all steps

Read.Me

```
#################################################
##################     Adventure 1-6     ########
##################        Colors         ########
#################################################
#                                               #
# Directions:                                   #
#       - Fork yourLastNameAdv1-5               #
#       - Save repl as: yourLastNameAdv1-6      #
#       - SHARE your repl with instructor!      #
#       - Read the Read.Me file!                #
#       - Complete all steps.                   #
#                                               #
#  ---------------------------------------------#
# Steps:                                        #
#         1. Update Comments                    #
#         2. Update Adventure Version # with a variable. #
#              a. Create variable 'AdventureVersion' #
#                 and assign value (i.e.,v1-6)  #
#              b. Pass variable to pWelcome() function #
#         3. Use escape characters to change text color #
#              a. in messages.py create functions to #
#                 change color of a string message. #
#              b. Import color functions in main.py #
#              c. Change colors in pSplash function #
#              d. Change colors in pWelcome function #
#              e. Change colors in pResults function #
#                                               #
#  ---------------------------------------------#
# New Python Content:                           #
#              Variable for version control     #
#              Color escape codes to chage text colors #
#                                               #
#  ---------------------------------------------#

# GREEN
def txtGreen(message):
  print("\033[0;32;48m", message, "\u001b[0m", end= "")

# Bright Blue
def txtBrightBlue(message):
  print("\033[1;34;48m", message, "\u001b[0m", end= "")

# CYAN
def txtCyan(message):
  print("\033[0;36;48m", message, "\u001b[0m", end= "")

# RED
def txtBrightRed(message):
  print("\033[1;31;48m", message, "\u001b[0m", end= "")

# Purple
def txtPurple(message):
  print("\033[1;35;48m", message, "\u001b[0m", end="")

# Example of how to call each function and pass in an
#   argument 'message' to change the color of the text.

# Text message
message = "The quick brown fox jumped"
myTxt1 ="over"
myTxt2 ="the"
myTxt3 ="lazy"
myTxt4 ="dog"

txtGreen(message)
txtBrightBlue(myTxt1)
txtCyan(myTxt2)
txtBrightRed(myTxt3)
txtPurple(myTxt4)

print("\n\n\033[1;35;48mAspire\u001b[0m to \033[1;35;48minspire\u001b[0m before we \033[1;35;48mexpire\u001b[0m.")
```

The quick brown fox jumped over the lazy dog

Aspire to inspire before we expire.

main.py file

```
#################################################
#  Ernie Drumm, 9/7/2020                        #
#  Description:                                 #
#              - Adventure 1-6 (Update Variable! :) #
#              - Read the Read.Me file!         #
#                                               #
#################################################
# 3b. Import color functions in main.py
from messages import pSplash, pWelcome, pResults, txtGreen, txtBrightBlue, txtCyan, txtRed

AdventureVersion = "v1.6"  # 2a. Create variable to Update Version.

# Call Splash screen function
pSplash(AdventureVersion)

# Call Welcome message function
pWelcome(AdventureVersion)  # 2b. pass version variable into function

# Get User Input
plrName = input("What is your first name? ")
plrTitle = input("What is your title: ")
plrLevel = input("What is your level? ")
plrHP = input("What is your HP? ")

# Call pResults function
pResults(plrName, plrTitle, plrLevel, plrHP, AdventureVersion)
```

messages.py

```
In [ ]: ##########################################################
        #   3a. In Messages.py create functions to change         #
        #       color of a string message.                       #
        ##########################################################
        # GREEN
        def txtGreen(message):
          print("\033[0;32;48m", message, "\u001b[0m")

        # Bright Blue
        def txtBrightBlue(message):
          print("\033[1;34;48m", message, "\u001b[0m")

        # CYAN
        def txtCyan(message):
          print("\033[0;36;48m", message, "\u001b[0m")

        # RED
        def txtRed(message):
          print("\033[1;31;48m", message, "\u001b[0m", end=" ")

        # Purple
        def txtPurple(message):
          print("\033[1;35;48m", message, "\u001b[0m", end=" ")


        # Splash Function
        def pSplash(AdventureVersion):
          print("\n" * 10)
          message = """
          >>>>>>>>>>   Awesome Adventure!   <<<<<<<<<<
          """
          txtGreen(message) # 3c. Change color to green
          print("\n" * 3)

          print("\033[1;35;48m------------> \033[1;36;48mAdventure " + AdventureVersion + "!\u001b[0m") #3c.
          print("\n" * 2)


        # Welcome Function
        def pWelcome(AdventureVersion):
          print("The adventure starts by gathering some basic information.")
          print()
          print("We need to know your name, title, level, and hit points!")
          print()
          print("This is \033[1;31;48mREQUIRED\u001b[0m to ensure you are ready for this adventure!\n") #3d. Change color


        # Results function
        def pResults(plrName, plrTitle, plrLevel, plrHP, AdventureVersion):
          print("\n" * 60)  # print 60 blank lines!
          print("\nYou are registering as: ")
          print("A level", plrLevel, plrTitle, " with", plrHP, "hit points!\n")
          print("Impressive!")
          print("You meet the requirements and are allowed to enter!\n")
          print("Looking forward to our adventure together" , plrName + "!\n")
          txtPurple(AdventureVersion) # 3e. Change colors
          print("\bReady?")
```

Top

Adventure Lab 1-4

Adventure 1-7 (Exception Validation)

```
- Before you start
- Read.Me file
```

Steps:

```
- 1. Update comments
- 2. Update Adventure Version variable
- 3. Exception handling
     a. create input.py file
     b. Create validation functions for each variable
     c. Use input validation code to ensure correct data is entered
     d. Call each validation functions
```

Solution files

```
- main.py
- messages.py
- input.py
```

Before you start

- Fork repl and save as 'Adventure1-7'

- Share repl with instructor

- Read the document on Exception handling.

- Read the Read.Me file

- Complete all steps

Read.Me

```
In [ ]: ##################   Adventure 1-7    ##################
        ##################     Exception      ##################
        ##################     Validation     ##################
        ########################################################
        #                                                      #
        # Directions:                                          #
        #       - Fork repl to yourLastNameAdv1-7              #
        #       - Read ALL comments in Read.Me file            #
        #       - Complete the necessary missing code.         #
        #                                                      #
        # Steps to complete:                                   #
        #         1. Update comments                           #
        #         2. Update Adventure Version variable.        #
        #         3. Get & Validate User Input                 #
        #            a. Create input.py file                   #
        #            b. Create Functions for User input variables#
        #                  getplrName() function               #
        #                  getplrTitle() function              #
        #                  getplrLevel() function              #
        #                  getplrHP() function                 #
        #            c. Call each function from the main.py file#
        #                                                      #
        #   ------------------------------------------------   #
        #  Files & Functions:                                  #
        #         main.py  :                                   #
        #         inputs.py  : getplrName                      #
        #                      getplrTitle                     #
        #                      getplrLevel                     #
        #                      getplrHP                        #
        #         Messages.py: txtGreen()                      #
        #                      txtBrightBlue()                 #
        #                      txtCyan()                       #
        #                      txtRed()                        #
        #                      pWelcome                        #
        #                      pResults                        #
        #         Read.Me      INSTRUCTIONS!                   #
        #                                                      #
        #   ------------------------------------------------   #
        # New Python Content:                                  #
        #            Exception Error Handling                  #
        #            While loop                                #
        #            IF Statements                             #
        ########################################################
```

getplrName() validation function

Notice the 'while True:' loop keeps the user in a loop to keep trying to enter correct data. If no 'While True' loop is present, the user only gets one try. The function does not need any arguments because we are asking the user to input their name. The first part of the if statement checks to see if the user entered a blank name. The 'elif' is another way of saying else if check this condition. The condition is using the len() function to ensure the name entered is less than 15 characters. We can also look for bad words to make sure those are not being used!

Lastly, if none of the conditions are met, the else statement is executed and the name the user entered is returned.

```
In [ ]: """----------------------------------------------
        The function getplrName is located in the inputs.py file
          and it is used to ensure NO blank names or NO
          player names are over 15 characters.
        ----------------------------------------------"""
        def getplrName():
          while True:
            aName = input("What is your name? ")
            if aName == "":
              print("NO blank names allowed. Please enter a name!")
            elif len(aName) > 15:
              print("Please enter a name shorter than 10 characters!")
            elif aName.lower() == "badword":
              print("NO bad words allowed!")
            else:
              return(aName)

        plrName = getplrName()
```

getplrHP() validation function

Notice the 'while True:' loop keeps the user in a loop to keep trying to enter correct data. The loop continues until the 'return(aHP)' line is executed. If the 'While True' loop is not present, the user only gets one try. The function does not need any arguments because we are asking the user to input their HP.

The first part of this validation uses the 'try:' statement. The 'try:' statement trys to execute the statements below it; the 'except ValueError:' is executed if the user entering anything other than an integer. If the user enters a valid integer the 'if' statement is started to be executed.

The 'if' statement checks to make sure the aHP is more than 1 and less than 300. If the user data passes all conditions the value the user entered is returned.

```
In [ ]: """----------------------------------------------
        The function getplrHP is located in the inputs.py file
          It is used to make sure an integer is entered by the
          user.  It also checks to make sure the HP is a positive
          number under 300. If all this checks out, the function
          returns what the user entered.
        ----------------------------------------------"""
        def getplrHP():
          while True:
            try:
              aHP = int(input("What is your HP? "))
            except ValueError:
              print("You entered invalid hit points. Please try again...")
              continue

            if aHP > 300:
              print("Over 300 is not normal! Please try again..")
            elif aHP < 1:
              print("HP of 1 is being alive.  Please enter a HP from 1 to 300")
            else:
              return(aHP)

        plrHP = getplrHP()
```

main.py

```
In [ ]: ########################################################
        #  Ernie Drumm, 9/1/2020                               #
        #  Description:                                        #
        #                - Adventure 1-7 (Update VERSION! :)   #
        #                - Read the Read.Me file!              #
        #                                                      #
        ########################################################
        from Messages import pSplash, pWelcome, pResults
        from inputs import getplrName, getplrTitle, getplrLevel, getplrHP

        # Call Splash screen
        pSplash()

        AdventureVersion = "v1.7"
        pWelcome(AdventureVersion)

        # Get User Input and validate
        plrName = getplrName()
        plrTitle = getplrTitle()
        plrLevel = getplrLevel()
        plrHP = getplrHP()

        # print results
        pResults(plrName, plrTitle, plrLevel, plrHP, AdventureVersion)
```

messages.py

```
In [ ]: ################################################
        #                                              #
        # Messages.py- This file is for all printing.  #
        #                                              #
        ################################################
        ################################################
        #                                              #
        #  Functions to change the color of a          #
        #       string message.                        #
        #                                              #
        ################################################
        # GREEN
        def txtGreen(message):
          print("\033[0;32;48m", message, "\u001b[0m")
        # Bright Blue
        def txtBrightBlue(message):
          print("\033[1;34;48m", message, "\u001b[0m")
        # CYAN
        def txtCyan(message):
          print("\033[0;36;48m", message, "\u001b[0m")
        # RED
        def txtRed(message):
          print("\033[1;31;48m", message, "\u001b[0m", end=" ")

        def txtNew():
          newColor = "\033[1;33;48m"
          return newColor

        ##########################
        #                        #
        # Splash Function        #
        #                        #
        ##########################
        def pSplash():
          print("\n" * 10)
          # Create a variable called 'message' and
          #  pass in the results of the txtGreen function
          # The txtGreen function augments includes all text within the
          #  double quotes.
          message = """
          >>>>>>>>>>>>>>>>>> Welcome to the Awesome Adventure! <<<<<<<<<<<<<<<<<<
          """
          # Change colors in pSplash function
          txtGreen(message)
          print("\n" * 3)

        ####################################
        #                                  #
        # Welcome Function                 #
        #                                  #
        ####################################
        def pWelcome(AdventureVersion):
          # Change colors in pWelcome function
          print("\033[1;35;48m------------> \033[1;36;48mAdventure " + AdventureVersion + "!\u001b[0m")
          print("\n" * 2)
          print("The adventure starts by gathering some basic information.")
          print()
          print("We need to know your name, title, level, and hit points!")
          print()
          print("This is REQUIRED to ensure you are ready for this adventure!\n")

        ####################################
        #                                  #
        # pResults function                #
        #      (pName, pTitle, pLevel, pHP)#
        #    10/1-Added AdventureVersion   #
        ####################################
        def pResults(plrName, plrTitle, plrLevel, plrHP, AdventureVersion):

          print("\n" * 60)
          print("\nYou are registering as: ")

          print("A level", plrLevel, plrTitle, "with", plrHP, "hit points!\n")
          print("Impressive!\n")

          print("You surly meet the requirements to enter!\n")
          print("Looking forward to hear about your adventure" , plrName , "\b!\n")

          # Change colors in pResults function
          print("Welcome to adventure", end="")
          txtRed(AdventureVersion)
          print("\bReady?")
```

inputs.py

```
In [ ]: ################################################
        #                                              #
        # 4a. Create inputs.py file (i.e., This file)  #
        #                                              #
        ################################################

        ################################################
        # 4b. Create getplrName()                      #
        ################################################
        def getplrName():
          while True:
            aName = input("What is your name? ")
            if aName == "":
              print("NO blank names allowed. Please enter a name!")
            elif len(aName) > 10:
              print("Please enter a name shorter than 10 characters!")
            elif aName.lower() == "badword":
              print("NO bad words allowed!")
            else:
              return(aName)

        ################################################
        # 4c. Create getplrTitle()                     #
        ################################################
        def getplrTitle():
          # Your code goes here...

        ################################################
        # 4d. Create getplrLevel()                     #
        ################################################
        def getplrLevel():
          # Your code goes here...

        ################################################
        # 4e. Create getplrHP()                        #
        ################################################
        def getplrHP():
          # Your code goes here...
```

Top

Adventure Lab 1-4

Adventure 1-8 (ASCII ART)

```
- Adventure 1-8 Directions
- Read.Me (NOT a python file, just instructions :)
- main.py
- Messages.py  *update
- inputs.py
```

Adventure 1-8 Directions:

- Fork repl and save as 'Adventure1-8'

- Read ALL comments including Read.Me file

- Update Adventure Version "#" with a variable

- Include ASCII Art into your app!

Example: Check out the Text to ASCII Art converter website from patorjk.com to get up and running quickly!

Read.Me

```
In [ ]: ########################################################
        ##################        ASCII       ##################
        ##################         ART        ##################
        ########################################################
        #                                                      #
        # Directions:                                          #
        #       - Fork repl to yourLastNameAdv1-8              #
        #       - Read ALL comments including Read.Me file     #
        #       - Complete the necessary missing code.         #
        #                                                      #
        # Steps to complete:                                   #
        #         1. Update Documentation Comments             #
        #         2. Update Adventure Version variable         #
        #                                                      #
        # *new    7. Add ASCII Art as a Splash screen          #
        #            a. in Messages.py Create function called  #
        #               'pAwesome'                             #
        #            b. Create some ASCII art!                 #
        #            c. Past ASCII art into function           #
        #            d. Call 'pAwesome' function from main.py  #
        #            e. Enjoy!                                 #
        #                                                      #
        #   ------------------------------------------------   #
        #  Files & Functions:                                  #
        #         main.py   :                                  #
        #         inputs.py :  getplrName                      #
        #                      getplrTitle                     #
        #                      getplrLevel                     #
        #                      getplrHP                        #
        #         Messages.py: pSplash                         #
        #                      pWelcome                        #
        #                      pResults                        #
        # *new                 pAwesome                        #
        #         Read.Me      INSTRUCTIONS!                   #
        #                                                      #
        ########################################################
        '7a. Create your ASCII function.

            1. In the messages.py file. Create a function for your ASCII art.  For example,
                    -------------------------------------------------------------
                    def pAwesome():
                      # print statements here!
                    -------------------------------------------------------------

        '7b. Create some ASCII art!

            1. One way to convert text to ASCII visit the web site below:

                    http://patorjk.com/software/taag/#p=display&f=Graffiti&t=Type%20Something%20

                    There you can type in words to convert to ASCII art.  You can change the Font and size as well.
            I put in the work 'Amazing', changed the font to 'Star Wars'.

            1. Copy your ASCII art you created to paste in your function.

        '7c. past ASCII art into function

            - TWO ways to ensure your ASCII art shows up correctly.

                    1. Use print statements and align your ASCII art.

                    -------------------------------------------------------------
                    def pAwesome():
                      # print statements here!

                      print("  __    __   _______  __        __          ______")
                      print(" |  |  |  | |   ____||  |      |  |        /  __  \ ")
                      print(" |  |__|  | |  |__   |  |      |  |       |  |  |  | ")
                      print(" |   __   | |   __|  |  |      |  |       |  |  |  | ")
                      print(" |  |  |  | |  |____ |  `----.|  `----.|  `--'  | ")
                      print(" |__|  |__| |_______||_______||_______| \______/  ")
                    -------------------------------------------------------------


                    2. Use the Multi-Line Comment to capture ASCII art.

                    -------------------------------------------------------------
                    def pAwesome():
                      print("""
                       __    __   _______  __        __          ______
                      |  |  |  | |   ____||  |      |  |        /  __  \
                      |  |__|  | |  |__   |  |      |  |       |  |  |  |
                      |   __   | |   __|  |  |      |  |       |  |  |  |
                      |  |  |  | |  |____ |  `----.|  `----.|  `--'  |
                      |__|  |__| |_______||_______||_______| \______/
                      """)
                    -------------------------------------------------------------


        7d. Call function.

            1. In the main.py file, add the 'pAwesome' function to the Messages import statement.

                    -------------------------------------------------------------

                    from Messages import pWelcome, pResults, pAwesome

                    -------------------------------------------------------------



            a. Remove the call to 'pSplash()' function.
                # Call Splash screen
                # pSplash()

            b. Add the call to 'pAwesome()' function and remove the call to 'pSplash'

                    -------------------------------------------------------------

                    from Messages import pWelcome, pResults,pAwesome, pAdventure
                    from inputs import getplrName, getplrTitle, getplrLevel, getplrHP

                    # Call Splash screen
                    #pSplash()
                    pAwesome()

                    -------------------------------------------------------------

        7e. Enjoy!

            Experiment!

            What other places in the app would a nice message be helpful?
```

main.py

```
In [ ]: ########################################################
        #  Ernie Drumm, 9/1/2020                               #
        #  Description:                                        #
        #                - Adventure 1-8 (Update VERSION! :)   #
        #                - Read the Read.Me file!              #
        #                                                      #
        ########################################################
        # 7d. Call Messages.py 'pAwesome' function.
        from Messages import pWelcome, pResults,pAwesome, pAdventure
        from inputs import getplrName, getplrTitle, getplrLevel, getplrHP

        # Call Splash screen
        #pSplash()
        pAwesome()
        pAdventure()

        AdventureVersion = "v1.8"
        pWelcome(AdventureVersion)

        # Get User Input and validate
        plrName = getplrName()
        plrTitle = getplrTitle()
        plrLevel = getplrLevel()
        plrHP = getplrHP()

        # print results
        pResults(plrName, plrTitle, plrLevel, plrHP, AdventureVersion)
```

Messages.py

```
In [ ]: ################################################
        #                                              #
        # Messages.py- This file is for all printing.  #
        #                                              #
        ################################################
        ################################################
        #                                              #
        #  Functions to change the color of a          #
        #       string message.                        #
        #                                              #
        ################################################
        # GREEN
        def txtGreen(message):
            print("\033[0;32;48m", message, "\u001b[0m")

        # Bright Blue
        def txtBrightBlue(message):
            print("\033[1;34;48m", message, "\u001b[0m")

        # CYAN
        def txtCyan(message):
            print("\033[0;36;48m", message, "\u001b[0m")

        # RED
        def txtRed(message):
            print("\033[1;31;48m", message, "\u001b[0m", end=" ")

        def txtNew():
            newColor = "\033[1;33;48m"
            return newColor

        ##########################
        #                        #
        # Splash Function        #
        #                        #
        ##########################
        def pSplash():
            print("\n" * 10)
            message = """
            >>>>>>>>>>>>>>>>>> Welcome to the Awesome Adventure! <<<<<<<<<<<<<<<<<<<
            """
            # Change colors in pSplash function
            txtGreen(message)
            print("\n" * 3)

        ##################################################
        #                                                #
        # Welcome Function                               #
        #                                                #
        ##################################################
        def pWelcome(AdventureVersion):
            # Change colors in pWelcome function
            print("\033[1;35;48m------------> \033[1;36;48mAdventure " + AdventureVersion + "!\u001b[0m")
            print("\n" * 2)
            print("The adventure starts by gathering some basic information.")
            print()
            print("We need to know your name, title, level, and hit points!")
            print()
            print("This is REQUIRED to ensure you are ready for this adventure!\n")

        ######################################
        #                                    #
        # pResults function                  #
        #       (pName, pTitle, pLevel, pHP) #
        #                                    #
        ######################################
        def pResults(plrName, plrTitle, plrLevel, plrHP, AdventureVersion):
            print("\n" * 60)
            print("\nYou are registering as: ")

            print("A level", plrLevel, plrTitle, "with", plrHP, "hit points!\n")
            print("Impressive!\n")

            print("You surly meet the requirements to enter!\n")
            print("Looking forward to hear about your adventure" , plrName , "\b!\n")

            # Change colors in pResults function
            print("Welcome to adventure", end="")
            txtRed(AdventureVersion)
            print("\bReady?")

        def pAwesome():
            print("""
               ___   ____    __    ____  _______     _______.  ______    .___  ___.  _______
              /   \  \   \  /  \  /   / |   ____|   /       | /  __  \   |   \/   | |   ____|
             /  ^  \  \   \/    \/   /  |  |__     |   (----`|  |  |  |  |  \  /  | |  |__
            /  /_\  \  \            /   |   __|     \   \    |  |  |  |  |  |\/|  | |   __|
           /  _____  \  \    /\    /    |  |____.----)   |   |  `--'  |  |  |  |  | |  |____
          /__/     \__\  \__/  \__/     |_______|_______/     \______/   |__|  |__| |_______|

            """)

        def pAdventure():
            print("""
               ___       _______  ____    ____  _______ .__   __. .___________. __    __  .______       _______
              /   \     |       \ \   \  /   / |   ____||  \ |  | |           ||  |  |  | |   _  \     |   ____|
             /  ^  \    |  .--.  | \   \/   /  |  |__   |   \|  | `---|  |----`|  |  |  | |  |_)  |    |  |__
            /  /_\  \   |  |  |  |  \      /   |   __|  |  . `  |     |  |     |  |  |  | |      /     |   __|
           /  _____  \  |  '--'  |   \    /    |  |____ |  |\   |     |  |     |  `--'  | |  |\  \----.|  |____
          /__/     \__\ |_______/     \__/     |_______||__| \__|     |__|      \______/  | _| `._____||_______|


            """)
```

Top

Adventure Lab 1-4

Up to you: Use green boxes sparingly, and only for some specific purpose that the other boxes can't cover. For example, if you have a lot of related content to link to, maybe you decide to use green boxes for related links from each section of a notebook.

HINT: Adventure Lab 1-5 is the last part of functions. After working throught this lab we will see a declutered work space and the ability to add lots more functionality!

Tip: This lab

Adventure 1-9 (if, elif, and else Statements)

```
- Adventure 1-9 Directions
- Read.Me (NOT a python file, just instructions :)
- main.py  *upate
- Messages.py
- inputs.py
```

Top

Our game has a basic story of being in a room and has a decision to make. The user has to decide if they are 1. going to try the door, 2. Read a note, or 3. Sit on the floor.

Before we make the logic we need to ask the user the questions so they know which one they want to choose. We will do this within the start of the game.

1. Make a function called play()
2. within that function make another function called start() and
3. Within the start() function, create several print statements to ask your questions to the user.
4. Create a list for user responses: cmdList=["1","2","3"]
5. Call a function getCMD(cmdList) to check the user input

   a. If a user enters 'quit' then quit the app.

   b. If the user enters 'help' then print some help.
6. Call the function the user requested (i.e., 1, 2, or 3)

Read.Me

```
In [ ]:
##########################################################
#################       if elif       ##################
#################      Statements     ##################
##########################################################
#                                                        #
# Directions:                                            #
#       - Fork repl to yourLastNameAdv1-9                #
#       - Read ALL comments including Read.Me file       #
#       - Complete the necessary missing code.           #
#       - Update Adventure Version "#" with a variable   #
#       - Include ASCII Art into your app!               #
#                                                        #
# *new    9. If statements                               #
#              a. Ask user a question                    #
#              b. Check to see if user wants to 'quit'   #
#              c. Check to see if user needs help        #
#              d. Run a function based on user response: #
#                   function 1, 2, or 3.                 #
#                                                        #
#     ------------------------------------------         #
#                                                        #
#  Files & Functions:                                    #
#         main.py    : play                              #
#                      start                             #
#                      pDoor01                           #
#                      pNote_Cake                        #
#                      pSit                              #
#                      getCMD                            #
#         inputs.py  : getplrName                        #
#                      getplrTitle                       #
#                      getplrLevel                       #
#                      getplrHP                          #
#         Messages.py: txtGreen                          #
#                      txtBrightBlue                     #
#                      txtCyan                           #
#                      txtRed                            #
#                      txtNew                            #
#                      pSplash                           #
#                      pWelcome                          #
#                      pResults                          #
#                      pAwesome                          #
#                      pAdventure                        #
#         Read.Me                                        #
#                                                        #
##########################################################
```

main.py file..

```
In [ ]:
#########################################################
#  Ernie Drumm, 9/1/2020                                #
#  Description:                                         #
#                - Adventure 1-9 (Update VERSION! :)    #
#                - Read the Read.Me file!               #
#                                                       #
#########################################################

from Messages import pWelcome, pResults,pAwesome, pAdventure
from inputs import getplrName, getplrTitle, getplrLevel, getplrHP

# Call Splash screen
pAwesome()
pAdventure()

AdventureVersion = "v1.9"
pWelcome(AdventureVersion)

# Get User Input and validate
plrName = getplrName()
plrTitle = getplrTitle()
plrLevel = getplrLevel()
plrHP = getplrHP()

# print results
pResults(plrName, plrTitle, plrLevel, plrHP, AdventureVersion)

#########################################################
#                                                       #
#     play function                                     #
#                                                       #
#########################################################
def play():

  def start():
    print("\n -------------------")
    print("What do you do? ...")
    print("\n")
    print("1. Try the door\n")
    print("2. Read the hand written note\n")
    print("3. Sit on the floor")
    print()
    print("Enter \'quit\' to end the game or \'help\' for help menu")
    cmdList=["1","2","3"]
    cmd = getCMD(cmdList)

    if cmd == "1":
        pDoor01()
    elif cmd == "2":
        pNote_Cake()
    elif cmd == "3":
        pSit()
  #
  def pDoor01():
    print("\n" * 60)
    print("""
    You walk up to the ...
    and I'll tell you of Great Treasure..."
    """)
  #
  def pNote_Cake():
    print("\n" * 60)
    print("""
    You start reading the note and it
    says... """)
  #
  def pSit():
    print("\n" * 60)
    print("""
    You decide to sit on the ground
    and then ... """)
  #
  def getCMD(cmdList):
    cmd = input(plrName + " -> ")
    if cmd in cmdList:
      return cmd
    elif cmd == "help":
      print("\nEnter your choices as detailed in the game.")
      print(" or enter 'quit' to leave the game.")
      return getCMD(cmdList)
    elif cmd == "quit":
      print("\n-------------------")
      print("Sadly you return to your homeland without fame or fortune...")
      exit()

  print("""
  You find yourself at a small room with a door in front of you.
  There's a hand written note in your hand.
  Otherwise, the room is empty!
  """)
  #
  start()


play()
```

[Top](#)

Adventure Lab 1-4

```
In [ ]:
#########################################################
#  yourName, today's date                               #
#  Description:                                         #
#                - Adventure 1-10                       #
#                - Your code adventure continues here.. #
#########################################################
```

Adventure Lab 1-4

Adventure Lab 1-1

```
In [ ]:
#  --------------------------------------------------    #
# Point Values:                                          #
#        Comments:                                       #
#                   Name                      5          #
#                   Date                      2          #
#                   Description               3          #
#                                                        #
#        Splash Screen: (Minimum Requirements)           #
#                   2 print() functions       2          #
#                   1 splash message          3          #
#                                                        #
#        Welcome Message: (Minimum Requirements)         #
#                   2 print() function        2          #
#                   1 welcome message         3          #
#                                                        #
#        Variables: (Minimum Requirements)               #
#                   1 str variable for Name   5          #
#                   1 str variable for Title  5          #
#                   1 int variable for Level  5          #
#                   1 int variable for HP     5          #
#                                                        #
#        Display Data: (Minimum Requirements)            #
#                   Casting a variable        5          #
#                   Display Name variable     5          #
#                   Display Title variable    5          #
#                   Display Level variable    5          #
#                   Display HP variable       5          #
#                                                        #
#                                           65 Total     #
#                                                        #
#  --------------------------------------------------    #
#  Rubric:                                               #
#                                                        #
#      Comments                                          #
#               0             1-5          6-10          #
#           ------------   ------------   ------------   #
#               No          Some           Lots of       #
#            Comments       Comments       comments      #
#                           Little         Lots of       #
#                            Detail         details      #
#                                                        #
#      Splash Screen                                     #
#              0-1           2-4             5           #
#         ------------   ------------   ------------     #
#          No print()    Only one       2 or more        #
#           functions     print()        print()         #
#            and/or       function       functions       #
#          No message     AND/OR        Nicely           #
#                        Very short      formatted       #
#                         message        messages        #
#                                                        #
#      Welcome Message:                                  #
#              0-1           2-4             5           #
#         ------------   ------------   ------------     #
#          No print()    Only one       2 or more        #
#           functions     print()        print()         #
#            and/or       function       functions       #
#          No message     AND/OR        Nicely           #
#                        Very short      formatted       #
#                         message        messages        #
#                                                        #
#      Variables                                         #
#              0-9          10-17           18-20        #
#         ------------   ------------   ------------     #
#          No Variables  Missing more    All Variables   #
#           OR            than one                       #
#          Mislabeled     variables      Correct Data    #
#           Variable     OR               Types          #
#           Names        Missing data                    #
#           AND/OR       Wrong data      Pleasent data   #
#          Wrong Data     type                           #
#           type                                         #
#                                                        #
#      Display Data                                      #
#              0-9          10-19           20-25        #
#         ------------   ------------   ------------     #
#          No print()    Missing 1 or   2 or more print()
#           functions     more print()   functions.      #
#          Missing        functions.    All variable     #
#           most of      AND/OR          data printed.   #
#           the variable Missing 1 or   Proper use of    #
#           data          more variable  casting.        #
#          AND/OR         data.                          #
#          Very short                                    #
#           message.                                     #
#          No casting                                    #
#                                                        #
##########################################################
```

[Top](#)

Adventure Lab 1-2

```
In [ ]:
#  --------------------------------------------------    #
# Point Values:                                          #
#        Update Comments:                                #
#                   name, date, description   5          #
#                                                        #
#        Update Splash Screen:                           #
#                   print() functions         1          #
#                   Splash message            1          #
#                   Include Version Number    3          #
#                                                        #
#        Update Welcome Message:                         #
#                   rint() functions          1          #
#                   2 welcome messages        4          #
#                                                        #
#        input() functions: (Minimum Requirements)       #
#                   1 str variable for Name   5          #
#                   1 str variable for Title  5          #
#                   1 int variable for Level  5          #
#                   1 int variable for HP     5          #
#                                                        #
#        Display Data: (Minimum Requirements)            #
#                   Casting a variable        5          #
#                   Display Name variable     5          #
#                   Display Title variable    5          #
#                   Display Level variable    5          #
#                   Display HP variable       5          #
#                                                        #
#                                           60 Total     #
#                                                        #
#  --------------------------------------------------    #
#  Rubric:                                               #
#                                                        #
#      Update Comments                                   #
#               0-1           2-3          4-5           #
#           ------------   ------------   ------------   #
#               No          Some           Lots of       #
#            Comments       Comments       comments      #
#               No          Little         Lots of       #
#             Updates        Detail         details      #
#                                           Updated      #
#      Update Splash Screen                              #
#              0-1           2-4             5           #
#         ------------   ------------   ------------     #
#          No print()    Only one       2 or more        #
#           functions     print()        print()         #
#            and/or       function       functions       #
#          No message     AND/OR        Nicely           #
#                        Very short      formatted       #
#                         message or     messages        #
#                        No version                      #
#                         Number                         #
#                                                        #
#      Update Welcome Message:                           #
#              0-1           2-4             5           #
#         ------------   ------------   ------------     #
#          No print()    Only one       2 or more        #
#           functions     print()        print()         #
#            and/or       function       functions       #
#          No message     AND/OR        Nicely           #
#                        Very short      formatted       #
#                         message        messages        #
#                                                        #
#      input() functions:                                #
#              0-9          10-17           18-20        #
#         ------------   ------------   ------------     #
#          No Variables  Missing more    All Variables   #
#           OR            than one        created        #
#          Mislabeled     variables      Correct Data    #
#           Variable     OR               Types          #
#           Names        Missing data                    #
#           AND/OR       Wrong data      Pleasent        #
#          Wrong Data     type             questions     #
#           type         OR              4 input()       #
#           AND/OR       Missing some      statements    #
#          No input()     input()                        #
#           statements    statements                     #
#          No question   Incorrect or                    #
#           in input()    missing                        #
#           statements    questions                      #
#                                                        #
#      Display Data                                      #
#              0-9          10-19           20-25        #
#         ------------   ------------   ------------     #
#          No print()    Missing 1 or   2 or more print()
#           functions     more print()   functions.      #
#          Missing        functions.    All variable     #
#           most of      AND/OR          data printed.   #
#           the variable Missing 1 or   Proper use of    #
#           data          more variable  casting.        #
#          AND/OR         data.                          #
#          Very short                                    #
#           message.                                     #
#          No casting                                    #
#                                                        #
##########################################################
```

[Top](#)

Adventure Lab 1-3

```
In [ ]:
#  --------------------------------------------------    #
# Point Values:                                          #
#        Update Comments:                                #
#                   name, date, description   5          #
#                                                        #
#        Splash Screen function:                         #
#                   Create & Call            20          #
#                                                        #
#        Welcome Message function:                       #
#                   Create & Call            20          #
#                                                        #
#        Update Splash Screen:                           #
#                   Message, Version Number   5          #
#                                                        #
#        Welcome Message:                                #
#                   2 welcome messages        5          #
#                                                        #
#        input() statements:                             #
#                   2 str variables           5          #
#                   2 int variables           5          #
#                                                        #
#        Display Data:                                   #
#                   Casting & Display data   10          #
#                   Friendly formattted      10          #
#                                                        #
#                                           85 Total     #
#                                                        #
#  --------------------------------------------------    #
#  Rubric:                                               #
#                                                        #
#      Update Comments                                   #
#               0-1           2-3          4-5           #
#           ------------   ------------   ------------   #
#               No          Some           Lots of       #
#            Comments       Comments       comments      #
#               No          Little         Lots of       #
#             Updates        Detail         details      #
#                                           Updated      #
#      Splash Screen Function                            #
#              0-1           2-4             5           #
#         ------------   ------------   ------------     #
#          No pSplash()  Spelling       pSplash()        #
#           function      wrong          function works  #
#          AND/OR        AND/OR         Version number   #
#          Not working   Not fully       updated         #
#                         working       Formatted nicely#
#                        OR                              #
#                        No Version                      #
#                         number                         #
#                                                        #
#      Welcome Message Function                          #
#              0-11         12-14         15-20          #
#         ------------   ------------   ------------     #
#         No pWelcome()  Spelling       pWelcome()       #
#          function       wrong          function works  #
#         AND/OR         AND/OR         Message included#
#         Not working    Not fully                       #
#           correctly     working       Formatted nicely#
#                        OR                              #
#                        No message                      #
#                                                        #
#      Update Splash Screen                              #
#              0-1           2-4             5           #
#         ------------   ------------   ------------     #
#          No print()    Only one       2 or more        #
#           functions     print()        print()         #
#            and/or       function       functions       #
#          No message     AND/OR        Nicely           #
#                        Very short      formatted       #
#                         message or     messages        #
#                        No version     Updated Version  #
#                         Number           number        #
#                                                        #
#      Update Welcome Message:                           #
#              0-1           2-4             5           #
#         ------------   ------------   ------------     #
#          No print()    Only one       2 or more        #
#           functions     print()        print()         #
#            and/or       function       functions       #
#          No message     AND/OR        Nicely           #
#                        Very short      formatted       #
#                         message        messages        #
#                                                        #
#      input() functions:                                #
#              0-3           4-7            8-10         #
#         ------------   ------------   ------------     #
#          No Variables  Missing more    All Variables   #
#           OR            than one        created        #
#          Mislabeled     variables      Correct Data    #
#           Variable     OR               Types          #
#           Names        Missing data                    #
#           AND/OR       Wrong data      Pleasent        #
#          Wrong Data     type             questions     #
#           type         OR              4 input()       #
#           AND/OR       Missing some      statements    #
#          No input()     input()                        #
#           statements    statements                     #
#          No question   Incorrect or                    #
#           in input()    missing                        #
#           statements    questions                      #
#                                                        #
#      Display Data                                      #
#              0-3           4-7            8-10         #
#         ------------   ------------   ------------     #
#          No print()    Missing 1 or   2 or more print()
#           functions     more print()   functions.      #
#          Missing        functions.    All variable     #
#           most of      AND/OR          data printed.   #
#           the variable Missing 1 or   Proper use of    #
#           data          more variable  casting.        #
#          AND/OR         data.         Frendly          #
#          Very short    Hard to read    formatted       #
#           message.      formatting                     #
#          No casting                                    #
#                                                        #
##########################################################
```

[Top](#)

Adventure Lab 1-4

[Top](#)

Adventure Lab 1-5

[Top](#)

Adventure Lab 1-6

[Top](#)

Adventure Lab 1-7

[Top](#)

Adventure Lab 1-8

[Top](#)

Adventure Lab 1-9

[Top](#)