**Natalie Bakes a Cake**

Introduction

* Algorithms, in their simplest forms are processes or sets of rules.
* While we may associate algorithms with computers and technological advancements,
* students interact with algorithms every day from how they get to school each day to how they play a game of checkers.
* To learn about algorithms, we will follow Natalie as she bakes a fancy pink birthday cake for her friend’s birthday.
* Follow the prompts, and think about how the steps reflect a large process or task.
* By the end of the story, you will see how algorithms may be used offline to complete a task or project.

Interactive Fun Question

What is your favorite kind of cake?

Enter: user response

Answer: Yum! That's Delicious. Today we will make a pink birthday cake!

First Story Prompt

Enter 'birthday cake' to take out the ingredients for the pink birthday cake

Enter:

After reviewing the pink birthday cake recipe, Natalie begins to pull out the ingredients from her cupboard and refrigerator.

Let's list the wet and dry ingredients she needs to bake the cake. (Using the dictionary function).

Wet Ingredients include:

{'milk', 'butter', 'eggs'}

Dry Ingredients include:

['sugar', 'vanilla', 'frosting']

Second Story Prompt

Enter 'mixer' to mix the wet and dry ingredients by hand.

Enter:

* Natalie tried to mix the ingredients by hand, but the batter was a little clumpy.
* She decided to use an electric mixer after all to make sure that the batter was smooth.
* Then Natalie poured the batter into a cake pan and popped the cake into the oven.
* Enter 'test the cake' to see if it's cooked through

Third Story Prompt

Enter: ‘test the cake’

Enter:

* It was a good idea to test the cake with a toothpick because even after 35 minutes of baking, the inside was still undercooked.
* Natalie leaves the cake in the oven for another 10 minutes, and when golden brown, takes the cake out and places it on the table.

Fourth Story Prompt

Enter 'frost the cake' to cover the sheet cake with pink buttercream frosting

Enter:

* Patience is important, especially when it comes to waiting for the cake to cool before frosting it.
* Natalie spreads the pink buttercream frosting on thickly, and places it on a cake display.
* The guests begin to arrive and marvel at the birthday cake, and Natalie is ready to celebrate her friend’s birthday! The end!

Fifth Prompt: Make Your Own Story or Return to the Beginning

Enter 'make my own' to make your own story or 'beginning' to follow the story again

Enter:

* First, decide the algorithmic story you want to tell.
* Think about a favorite hobby, a fun vacation, or something you want to teach others.
* You want a story you can tell in 5-10 parts.
* You also want to allow the reader to be able to choose different options or make decisions to make your story interactive and fun to follow.
* Write your 5-10 part using your A-Board, and map out the sequence on your A Board
* When your story is complete, you can share it with family and friends using your A-Board.
* They may have advice on how you can revise or improve your story, which is also called iterative design.
* Iterative design is very important when designing and working with technology,so be open to feedback that may enhance your project.
* Once you have gotten feedback on your story, it's time to start coding your interactive story on Python.
* Remember, you have already completed the hard work by coming up with the story and process.
* Now we can use the code to make your story available virtually

Code Your Story on Python

Enter 'start coding' to write your story in python or 'beginning' to return to the start room.

Enter:

* Once you have downloaded PyCharm, open a new project (.py) file, and put your name, date, and project title on the file.
* Also write 'import time'to give your program some time delays between prompts.
* Remember to use a # before non-coded content so the system doesn't try to read it.
* Next, you will want to create 'objects' for each of the 5-10 process steps in your story.
* They names could be sequential (e.g. Room\_01, Room\_02) or be directly related to the story (room\_drive, room\_arrive, etc).
* Be sure to pick names that make sense and that you will remember.
* To create the objects, use the function 'def object ()':.

* Also make sure to list all of the objects before beginning telling the story or program so that Python can learn them all.
* Next, under each object using the print function --> print() to write out your story.
* Using three quotations on either side as opposed to two will help you keep the formatting that you desire.
* Try to create a dictionary or list of items, like the wet and dry ingredients in 'Natalie Bakes a Cake.
* To do this, name the object = {"object 1","object 2," etc}
* Once you have created all of your objects, you can list them all in sequential order. write time.sleep(5) between each step so that there is a delay between steps
* In the next module, we will make the story a bit more interactive using prompts and conditions.
* Remember, building the algorithm was the hard part.
* Now we can use the code to enhance it, and make the user experience more fun!

Process finished with exit code 0

*#-- Developer: Jessica Gershuny  
#-- Date: May 2019  
# -- Subject: Artificial Intelligence Independent Study  
#-- Program Name: Natalie Bakes a Cake  
# Global Imports***import** random  
**import** time  
**import** random  
**import** time  
**import** textwrap  
  
*#- functions/Defined Objects***def** cls():  
 print(**"\n"**\*100)  
  
**def** start\_room():  
 print(**'''  
Algorithms, in their simplest forms are processes or sets of rules.   
   
While we may associate algorithms with computers and technological advancements,   
   
students interact with algorithms every day from how they get to school each day to how they play a game of checkers.'''**)  
 time.sleep(10)  
  
 print(**'''  
To learn about algorithms, we will follow Natalie as she bakes a fancy pink birthday cake for her friend’s birthday.   
   
Follow the prompts, and think about how the steps reflect a large process or task.   
   
By the end of the story, you will see how algorithms may be used offline to complete a task or project.'''**)  
 time.sleep(10)  
  
**def** age():  
 print(**'''how old are you turning this year'''**)  
 user\_choice = input(**"Enter:"**)  
 **if** user\_choice > 12:  
 number = **'Awesome'  
 else**:  
 **if** user\_choice < 12:  
 number = **'Nice!'  
 return** number  
  
**def** favorite():  
 print(**'''What is your favorite kind of cake?'''**)  
 user\_choice = input(**"Enter:"**)  
 time.sleep(2)  
 print(**'''Yum! That's Delicious. Today we will make a pink birthday cake!'''**)  
  
**def** room\_ingredients():  
 print(**'''  
After reviewing the pink birthday cake recipe, Natalie begins to pull out the ingredients from her cupboard and refrigerator.   
   
Let's list the wet and dry ingredients she needs to bake the cake.'''**)  
 time.sleep(5)  
  
**def** room\_mix():  
 print(**'''  
Natalie tried to mix the ingredients by hand, but the batter was a little clumpy.  
   
She decided to use an electric mixer after all to make sure that the batter was smooth.   
   
Then Natalie poured the batter into a cake pan and popped the cake into the oven.'''**)  
 time.sleep(5)  
  
**def** room\_test():  
 print(**'''  
It was a good idea to test the cake with a toothpick because even after 35 minutes of baking, the inside was still undercooked.  
   
Natalie leaves the cake in the oven for another 10 minutes, and when golden brown, takes the cake out and places it on the table.'''**)  
 time.sleep(5)  
  
**def** room\_frost():  
 print(**'''  
Patience is important, especially when it comes to waiting for the cake to cool before frosting it.   
   
Natalie spreads the pink buttercream frosting on thickly, and places it on a cake display.   
   
The guests begin to arrive and marvel at the birthday cake, and Natalie is ready to celebrate her friend’s birthday! The end!'''**)  
 time.sleep(5)  
  
**def** offline\_activity():  
 print(**'''  
First, decide the algorithmic story you want to tell.   
   
Think about a favorite hobby, a fun vacation, or something you want to teach others.   
   
 You want a story you can tell in 5-10 parts.   
   
You also want to allow the reader to be able to choose different options or make decisions to make your story interactive and fun to follow.'''**)  
  
 time.sleep(10)  
  
 print(**'''Write your 5-10 part using your A-Board, and map out the sequence on your A Board'''**)  
  
 print(**'''  
When your story is complete, you can share it with family and friends using your A-Board.   
   
They may have advice on how you can revise or improve your story, which is also called iterative design.  
   
Iterative design is very important when designing and working with technology,so be open to feedback that may enhance your project.'''**)  
 time.sleep(5)  
  
 print(**'''  
Once you have gotten feedback on your story, it's time to start coding your interactive story on Python.   
   
Remember, you have already completed the hard work by coming up with the story and process.   
   
Now we can use the code to make your story available virtually'''**)  
  
**def** online\_activity():  
 print(**'''  
Once you have downloaded PyCharm, open a new project (.py) file, and put your name, date, and project title on the file.   
  
Also write 'import time'to give your program some time delays between prompts.  
  
Remember to use a # before non-coded content so the system doesn't try to read it. '''**)  
 print(**'''  
Next, you will want to create 'objects' for each of the 5-10 process steps in your story.   
  
They names could be sequential (e.g. Room\_01, Room\_02) or be directly related to the story (room\_drive, room\_arrive, etc).   
  
Be sure to pick names that make sense and that you will remember.  
  
To create the objects, use the function 'def object ()':.  
   
Also make sure to list all of the objects before beginning telling the story or program so that Python can learn them all.'''**)  
 print(**'''  
Next, under each object using the print function --> print('''''') to write out your story.   
  
Using three quotations on either side as opposed to two will help you keep the formatting that you desire. '''**)  
 print(**'''  
Try to create a dictionary or list of items, like the wet and dry ingredients in 'Natalie Bakes a Cake.   
  
To do this, name the object = {"object 1","object 2," etc}'''**)  
  
 print(**'''  
Once you have created all of your objects, you can list them all in sequential order. write time.sleep(5) between each step so that there is a delay between steps'''**)  
  
 print(**'''In the next module, we will make the story a bit more interactive using prompts and conditions.   
   
 Remember, building the algorithm was the hard part.   
   
 Now we can use the code to enhance it, and make the user experience more fun!'''**)  
  
  
*#Dictionary of Ingredients*wet\_ingredients = {**'butter'**,**'eggs'**,**'milk'**}  
dry\_ingredients= [**'sugar'**,**'vanilla'**,**'frosting'**]  
  
  
*#General Program*print(**"Welcome to 'Natalie Bakes a Cake'"**)  
time.sleep(3)  
print(**'''Let's learn about algorithms!'''**)  
*#First Prompt*cls()  
start\_room()  
favorite()  
time.sleep(2)  
start\_room\_options= [**"birthday cake"**]  
user\_choice = **""  
while** user\_choice **not in** start\_room\_options:  
 print(**'''Enter 'birthday cake' to take out the ingredients for the pink birthday cake'''**)  
 user\_choice = str(input(**"Enter:"**))  
**if** user\_choice == start\_room\_options [0]: room\_ingredients()  
  
*#Ingredient Room*room\_ingredients\_options = [**"mixer"**]  
user\_choice=**""**print(**"Wet Ingredients include:"**)  
print((wet\_ingredients))  
time.sleep(2)  
  
print(**"Dry Ingredients include:"**)  
print(dry\_ingredients)  
time.sleep(2)  
  
**while** user\_choice **not in** room\_ingredients\_options:  
 print(**"Enter 'mixer' to mix the wet and dry ingredients by hand."**)  
 user\_choice = str(input(**"Enter:"**))  
**if** user\_choice == room\_ingredients\_options [0]:room\_mix()  
  
*#Time to Test the Cake*room\_mix\_options = [**"test the cake"**]  
user\_choice=**""  
while** user\_choice **not in** room\_mix\_options:  
 print(**'''Enter 'test the cake' to see if it's cooked through '''**)  
 user\_choice = str(input(**"Enter:"**))  
**if** user\_choice == room\_mix\_options [0]:room\_test()  
  
*#Time to Frost the Cake*room\_test\_options = [**"frost the cake"**]  
user\_choice=**""  
while** user\_choice **not in** room\_test\_options:  
 print(**'''Enter 'frost the cake' to cover the sheet cake with pink buttercream frosting'''**)  
 user\_choice = str(input(**"Enter:"**))  
**if** user\_choice == room\_test\_options [0]: room\_frost()  
  
room\_frost\_options = [**"make my own"**,**"beginning"**]  
user\_choice=**""  
while** user\_choice **not in** room\_frost\_options:  
 print(**'''Enter 'make my own' to make your own story or 'beginning' to follow the story again'''**)  
 user\_choice = str(input(**"Enter:"**))  
**if** user\_choice== room\_frost\_options[0]:offline\_activity()  
**elif** user\_choice ==room\_frost\_options[1]: start\_room()  
  
offline\_activity\_options = [**"start coding"**,**"beginning"**]  
user\_choice=**""  
while** user\_choice **not in** offline\_activity\_options:  
 print(**'''Enter 'start coding' to write your story in python or 'beginning' to return to the start room.'''**)  
 user\_choice = str(input(**"Enter:"**))  
**if** user\_choice== offline\_activity\_options [0]:online\_activity()  
**elif** user\_choice == offline\_activity\_options [1]:start\_room()