

	Mastery	Approaching Mastery	Progressing	Emerging	Incomplete
Retrieve basic information with API call (10 points)	✓ Deliverable retrieves the following information from the API call: <ul style="list-style-type: none"> Latitude and longitude Maximum temperature Percent humidity Percent cloudiness Wind speed Weather description ✓ Add data listed above to the DataFrame	✓ Deliverable retrieves 4-5 of the following from the API call. <ul style="list-style-type: none"> Latitude and longitude Maximum temperature Percent humidity Percent cloudiness Wind speed Weather description ✓ Add data listed above to the DataFrame	✓ Deliverable retrieves 2-3 of the following from the API call. <ul style="list-style-type: none"> Latitude and longitude Maximum temperature Percent humidity Percent cloudiness Wind speed Weather description ✓ Add data listed above to the DataFrame	✓ Deliverable retrieves 1 of the following from the API call. <ul style="list-style-type: none"> Latitude and longitude Maximum temperature Percent humidity Percent cloudiness Wind speed Weather description ✓ Add data listed above to the DataFrame	
	Note: if you completed the lessons in this module, you will have already completed “Retrieve basic information with API call” at the Mastery level.				
Try-except block (20 points)	✓ Deliverable retrieves both of the following: <ul style="list-style-type: none"> The amount of rainfall in inches for the last three hours from the API using a try-except block. In the try-except block, if there is no rain then the amount of rain should be 0 inches. The amount of snow in inches for the last three hours from the API using a try-except block. In the try-except block, if there is no snow then the amount should be 0 inches. ✓ Add the amount of rain	✓ Deliverable retrieves both of the following: <ul style="list-style-type: none"> The amount of rainfall in inches for the last three hours from the API using a try-except block. In the try-except block, if there is no rain then the amount of rain should be 0 inches. The amount of snow in inches for the last three hours from the API using a try-except block. In the try-except block, if there is no snow then the amount should be 0 inches. ✓ Add the amount of rain or snow, whichever is retrieved to	✓ Deliverable retrieves either: <ul style="list-style-type: none"> The amount of rainfall in inches for the last three hours from the API using a try-except block. In the try-except block, if there is no rain then the amount of rain should be 0 inches. The amount of snow in inches for the last three hours from the API using a try-except block. In the try-except block, if there is no snow then the amount should be 0 inches. ✓ Add the amount of rain or snow, whichever is retrieved to the DataFrame. ✓ Save DataFrame as a CSV.	✓ Deliverable retrieves either: <ul style="list-style-type: none"> The amount of rainfall in inches for the last three hours from the API using a try-except block. In the try-except block, if there is no rain then the amount of rain should be 0 inches. The amount of snow in inches for the last three hours from the API using a try-except block. In the try-except block, if there is no snow then the amount should be 0 inches. ✓ Save DataFrame as a CSV.	No submission was received -OR- Submission was empty or blank -OR- Submission contains evidence of academic dishonesty

	<p>or snow, whichever is retrieved to the DataFrame.</p> <ul style="list-style-type: none"> ✓ Save DataFrame as a CSV. ✓ Use Pandas to correctly answer this question: How many cities have recorded rainfall or snow? 	<p>the DataFrame.</p> <ul style="list-style-type: none"> ✓ Save DataFrame as a CSV. 			
<p>Have Customers Narrow Their Travel Searches Based on Temperature and Precipitation using if/elif/else statements</p> <p>(20 points)</p>	<ul style="list-style-type: none"> ✓ Writes code to prompt the customer to get the following information <ul style="list-style-type: none"> • The minimum temperature preference. • The maximum temperature preference. • If they want it to be raining or not. • If they want it to be snowing or not. ✓ Uses if/elif/else statements to filter the <code>city_data_df</code> DataFrame based on the minimum and maximum temperature, and whether it is either raining and snowing or not to get the cities that meet the customer criteria. 	<ul style="list-style-type: none"> ✓ Writes code to prompt the customer to get the following information <ul style="list-style-type: none"> • The minimum temperature preference. • The maximum temperature preference. And: • Whether they would like it to be raining or snowing or not, depending on if they retrieved the rainfall or snowfall in the API. ✓ Uses if/elif/else statements to filter the <code>city_data_df</code> DataFrame based on the minimum and maximum temperature, and whether it is either raining or snowing (depending on what they retrieved in the API) or not to get the cities that meet the customer criteria. 	<ul style="list-style-type: none"> ✓ Writes code to prompt the customer to get the following information <ul style="list-style-type: none"> • The minimum temperature preference. • The maximum temperature preference. And: • Whether they would like it to be raining or snowing or not, depending on if they retrieved the rainfall or snowfall in the API. ✓ Uses if/elif/else statements to filter the <code>city_data_df</code> DataFrame based on the minimum and maximum temperature preferences to get the cities that meet the customer criteria. 	<ul style="list-style-type: none"> ✓ Writes code to prompt the customer to get the following information <ul style="list-style-type: none"> • The minimum temperature preference. • The maximum temperature preference. ✓ Uses if/elif/else statements to filter the <code>city_data_df</code> DataFrame based on the minimum and maximum temperature preferences to get the cities that meet the customer criteria. 	
<p>Create a New Dataframe with Hotel Information Using Google API Places and try/except</p> <p>(10 points)</p>	<ul style="list-style-type: none"> ✓ Creates a new DataFrame with the following columns: <ul style="list-style-type: none"> • City • Country • Max Temp • Current Description • Lat • Lng ✓ Code is written to search for a hotel using a Google API and the JSON data is retrieved. 	<ul style="list-style-type: none"> ✓ Creates a new DataFrame with the following columns: <ul style="list-style-type: none"> • City • Country • Max Temp • Current Description • Lat • Lng ✓ Code is written to search for a hotel using a Google API and the JSON data is retrieved. ✓ A try/except block is used 	<ul style="list-style-type: none"> ✓ Creates a new DataFrame with the following columns: <ul style="list-style-type: none"> • City • Country • Max Temp • Current Description • Lat • Lng ✓ Code is written to search for a hotel using a Google API and the JSON data is retrieved. ✓ Save and upload the DataFrame to a CSV file 	<ul style="list-style-type: none"> ✓ Creates a new DataFrame with the following columns: <ul style="list-style-type: none"> • City • Country • Max Temp • Current Description • Lat • Lng ✓ Code is written to search for a hotel using a Google API but the JSON data is not retrieved. 	

	<ul style="list-style-type: none">✓ A try/except block is used to add the hotel to the new DataFrame, and the IndexError is resolved.✓ Save and upload the DataFrame to a CSV file.	<p>to add the hotel to the new DataFrame, but there is an IndexError.</p> <ul style="list-style-type: none">✓ Save and upload the DataFrame to a CSV file			
	<p>Note: if you completed the lessons in this module, you will have already completed “Create a New Dataframe with Hotel Information Using Google API Places and try/except” at the Mastery level.</p>				
<p>Create a pop-up marker with city and weather data and hotel name</p> <p>(10 points)</p>	<ul style="list-style-type: none">✓ Creates a marker layer map with a pop-up marker for each city that includes:<ul style="list-style-type: none">• Hotel name• City• Country• Current weather description with the maximum temperature✓ Save and upload the new marker layer map as PNG	<ul style="list-style-type: none">✓ Creates a marker layer map with a pop-up marker for each city that includes:<ul style="list-style-type: none">• Hotel Name• City• Country• Maximum temperature only✓ Save and upload the new marker layer map as PNG	<ul style="list-style-type: none">✓ Creates a marker layer map with a pop-up marker for each city that includes:<ul style="list-style-type: none">• City• Country• Maximum temperature only✓ Save and upload the new marker layer map as PNG	<ul style="list-style-type: none">✓ Creates a marker layer map with a pop-up marker for each city that includes:<ul style="list-style-type: none">• City• Maximum temperature only✓ Save and upload the new marker layer map as PNG	
<p>Note: if you completed the lessons in this module, you will have already completed “Create a pop-up marker with city and weather data and hotel name” at the Approaching Mastery level.</p>					
<p>Create a directions layer map to travel between cities.</p> <p>(20 points)</p>	<ul style="list-style-type: none">✓ Writes code to filter the vacation DataFrame for four cities to travel to.✓ Writes code to get the latitude and longitude pairs for each city to visit.✓ An itinerary map between the cities is created.✓ Save and upload the directions layer map as PNG.	<ul style="list-style-type: none">✓ Writes code to filter the vacation DataFrame for three cities to travel to.✓ Writes code to get the latitude and longitude pairs for each city to visit.✓ An itinerary map between the cities is created.✓ Save and upload the directions layer map as PNG.	<ul style="list-style-type: none">✓ Writes code to filter the vacation DataFrame for two cities to travel to.✓ Writes code to get the latitude and longitude pairs for each city to visit.✓ An itinerary map between the cities is created.✓ Save and upload the directions layer map as PNG.	<ul style="list-style-type: none">✓ Writes code to filter the vacation DataFrame for two cities to travel to.✓ Writes code to get the latitude and longitude pairs for each city to visit.Ø An itinerary map between the cities is not created.	

Create a pop-up marker for each city on the itinerary. (10 points)	<p>✓ Creates a new DataFrame that contains the cities in the itinerary.</p> <p>✓ Creates a marker layer map with a pop-up marker for the cities in the directions layer map which has the following information:</p> <ul style="list-style-type: none">• Hotel name• City• Country• Current weather description with the maximum temperature <p>✓ Save and upload the new marker layer map for the four cities as PNG</p>	<p>✓ Creates a new DataFrame that contains the cities in the itinerary.</p> <p>✓ Creates a marker layer map with a pop-up marker for the cities in the directions layer map which has the following information:</p> <ul style="list-style-type: none">• Hotel Name• City• Country• Maximum temperature only <p>✓ Save and upload the new marker layer map for any number of cities as PNG</p>	<p>✓ Creates a new DataFrame that contains the cities in the itinerary.</p> <p>✓ Creates a marker layer map with a pop-up marker for the cities in the directions layer map which has the following information:</p> <ul style="list-style-type: none">• City• Country• Maximum temperature only <p>✓ Save and upload the new marker layer map for any number of cities as PNG</p>	<p>✓ Creates a new DataFrame that contains the cities in the itinerary.</p> <p>✓ Creates a marker layer map with a pop-up marker for the cities in the directions layer map which has the following information:</p> <ul style="list-style-type: none">• City• Country• Maximum temperature only <p>✓ A marker layer map is not saved nor uploaded.</p>	
---	--	---	--	---	--

