Webpage_Mongo

View Source

class WebpageMongo: View Source

This class is designed to take user inputs from a webpage and return the relevant data stored within mongodb

WebpageMongo(user) View Source

def Single_Day_Inquriy_Obs(self, date): View Source

This function will take a date from a user and return the High and Low temperature that occurred that day. It will also return the Rainfall and Snow fall that occurred that day, all using the units saved by the user in their preferences. This method takes argument date format "Jan 01 2021".

def Single_Day_Inquiry_Hourly_Obs(self, date): View Source

This function will take a date and time from a user and return the High and Low temperature that occurred that day and time. It will also return the Precipitation that occurred at that time, all using the units saved by the user in their preferences. This method takes argument date format "Jan 01 2021 1 PM".

def Interval_Inquiry_Hourly_Obs(self, date): View Source

This function will take a date and time range from a user and return the High and Low temperature that occurred across those days and times. It will also return the Precipitation that occurred across the interval all using the units saved by the user in their preferences. This method takes argument date format "Jan 01 2021 9 PM - Jan 02 2021 12 AM".

def Interval_Inquiry(self, start_date, end_date): View Source

This function will take a date range from a user and return the High and Low temperature that occurred across those days and times. It will also return the Snowfall and Rainfall that occurred across the interval, all using the units saved by the user in their preferences. This method takes argument date format "Jan 01 2021 - Jan 02 2021".

def Current_Forecast(self): View Source

This function returns the most recent forecast data stored in the mongodb database for AccuWeather, Weather.com and the National Weather Service. It returns both the hourly and 10 day forecast (7 day in the case of the National Weather Service) in the form of a list of a list of dictionaries.

def recodedate_num(self, date): View Source

This method takes the name of a month and returns the numerical codification of the month

def recodedate_str(self, date): View Source

This method takes a numerical month and returns the actual name of the month as a string

def time_convert(self, time, ampm): View Source

This method converts times into the format stored in mongodb from weather.gov. It is passed the time to convert and whether the time is in AM or PM. It takes time format 10 and ampm should be specified as either the string "AM" or the string "PM"

def iterate_time(self, time): View Source

This method is passed a time to increase by an hour. For instance 1 AM becomes 2 AM, 12 PM becomes 1 AM.

def detinqtype(
self,
HRorDAY,
range_choice,
start_date,
end_date,
start_time,
end_time,
start_am_pm,
end_am_pm
): View Source

This method is passed information regarding an inquiry that a user wishes to make into past weather observations and directs the user to the appropriate method for their inquiry and returns the data. The HRorDay variable should be either "Daily" if the user wants to see data for entire days or "Hourly" if the user wants to inquire into hourly data, this should be passed as a string. Range_choice should be passed as a string as well specifying if the user is

inquiring into either a "Single day" or "Range of days". The start and end date should both be passed as datetime values which specify the start and end date of an inquiry, if a single day is specified the start_date will be used as the day of inquiry, similarly for start_time and start_am_pm. Start_time and end_time should both be passed as numerical values ranging from 01 to 12 specifying the time. Start_am_pm and end_am_pm should both be passed as strings specifying if the time the user desires is in "AM" or "PM"

def Forecast_Inquiry_M2(self, date, time, am_pm, service_prov, timeinterval): View Source

this method should be passsed the parameters of a user inquiry into prior forecast data that is saved into a mongodb database. It will then return the forecast data that is saved in mongodb as a list of dictionaries. The date variable should be passed as a string format "Jan 01 2021", time should be passed as a string format "9" and am_pm should be passed as a string format "AM". The service_prov variable should be passed as a string specifying the weather service provider that the user wishes to inquire into. The timeinterval variable should be passed as a string specifying whether the user wants to return the 7/10 day forecast or the forecast for the next 24 hours.

def detdatabase(self, service_prov, timeinterval): View Source

This function takes a weather service provider and a forecast type and returns the collection in mongodb that the data is stored within. The service_prov variable should be passed as a string specifying the weather service provider that the user wishes to inquire into. The timeinterval variable should be passed as a string specifying whether the user wants to return the 7/10 day forecast or the forecast for the next 24 hours.