***//function called when document ready***

***$***(***document***).ready(function() {  
***//function called when element “country” changes***

***$***("#country").change(function() {  
//declares variable from the “country” element selected value  
 let country = ***$***(this).val();

***//initiatives cityList variable***  
 let cityList = "";

***//based on “country” variable, sets “cityList” variable as path to the***

***//provided html files with city <options> for dropbox***  
 if (country==="England") {  
 cityList = "./uk\_cities/england-cities.html";  
 }  
 else if (country==="Northern Ireland") {  
 cityList = "./uk\_cities/nireland-cities.html"  
 }  
 else if (country==="Scotland") {  
 cityList = "./uk\_cities/scotland-cities.html"  
 }  
 else if (country==="Wales") {  
 cityList = "./uk\_cities/wales-cities.html"  
 }

//based on path to html files

//loads these into the “city” <select> element  
 ***$***("#city").load(cityList);  
  
 })  
//when user selects a city, this functions is called  
 ***$***("#city").change(function() {

//”city” variable is declared with users selection  
 let city = ***$***(this).val();

//”APPID” variable is declared my API key value  
 let APPID = "928482e19bbfa4b1442601b2c23e9952"

//”URL” variable is concantenated from Open Weather Map endpoint,target city (stored in the “city” variable) and my api key  
 let URL = "https://api.openweathermap.org/data/2.5/weather?q="  
 + city + ",uk&APPID=" + APPID

//jQuery getJSON method is called using the “URL” variable  
 ***$***.getJSON(URL, function(response) {  
//values from the response are assigned to appropriately named variable

//with minimal logic translating kelvin into fahrenheit/celsius, and m/sec into mph and kmph  
 let cityName = response.name;  
 let weatherDate = GetFormattedDate(new ***Date***(response.dt\*1000));  
 let weatherConditions = response.weather[0].main;  
 let kelvinTemperature = parseFloat(response.main.temp);  
 let celsiusTemperature = parseInt(kelvinTemperature - 273.15);  
 let fahrenheitTemperature = parseInt((celsiusTemperature\*1.8)+32);  
 let windSpeedKmph = parseFloat(response.wind.speed \* 3.6);  
 let windSpeedMph = parseFloat(response.wind.speed \* 2.237);  
 let windDegrees = parseInt(response.wind.deg);  
//function to convert wind direction from 360 degree metereological scale

//into compass bearings

let windDirection = windDirectionConverter(windDegrees);

//a variable is declared and value set to Open Weather Maps weather icons, and is concatenated with icon code from api response body  
 let weatherIcon = '<img id="icon" src="http://openweathermap.org/img/wn/' + response.weather[0].icon  
 + '@2x.png" alt="weatherIcon" width="150px" height="100%"/>'

//a variable is set to the Boolean value of a alertChecker function  
 let extremeWeatherAlert = alertChecker(celsiusTemperature, windSpeedMph);  
//a function is called to display values of previous declared variables

weatherDisplay(cityName,weatherDate,weatherConditions,celsiusTemperature,fahrenheitTemperature,  
 windSpeedKmph, windSpeedMph,windDegrees,windDirection,weatherIcon, extremeWeatherAlert);  
 })  
 })  
  
})  
//function to build html concatenating it with arguments passed to it by the previous functions, and store it as sTxt variable

function weatherDisplay(city, date, conditions, celsius, fahrenheit, kmph, mph, degrees, direction, icon, alert){  
 let sTxt = "<h2>" + city + ", United Kingdom<br>" + date + "</h2><br>" + icon  
 + "<br> Temperature: " + celsius + "&deg; C / " + fahrenheit + "&deg; F <br>"  
 + "Conditions: " + conditions + "<br> Wind speed: " + mph.toFixed(2) + " mph , "  
 + kmph.toFixed(2) + " kmph ,<br>Wind direction: "+ direction;

//if “alertChecker” returns “true”, additional HTML code for EXTREME WEATHER ALERT is appended to “sTxt” variable  
 if (alertChecker()) {  
 sTxt += `<div id="red\_alert"><br> <p id="alert"><h1>EXTREME WEATHER WARNING</h1></p></div>`;  
 }  
//previous contents (if any) are cleared from the “weather” <div> element  
 ***$***("#weather").html("");

//html code from “sTxt” variable is assigned to “weather” <div> element  
 ***$***("#weather").append(sTxt);  
  
}  
//checks whether air temperature or wind speed fall into EXTREME RANGE  
function alertChecker(celsius, mph) {  
 if (celsius < 5 || celsius > 35 || mph > 50) {  
 return true;  
 }  
  
}  
  
//converts degrees of bearing into traditional compass values  
function windDirectionConverter(degrees) {  
 if (degrees > 0 && degrees < 25) {  
 return "NORTH";  
 } else if (degrees > 25 && degrees < 65) {  
 return "NORTH EAST";  
 } else if (degrees > 65 && degrees < 115) {  
 return "EAST";  
 } else if (degrees > 115 && degrees < 155) {  
 return "SOUTH EAST";  
 } else if (degrees > 155 && degrees < 205) {  
 return "SOUTH";  
 } else if (degrees > 205 && degrees < 245) {  
 return "SOUTH WEST";  
 } else if (degrees > 245 && degrees < 295) {  
 return "WEST";  
 } else if (degrees > 295 && degrees < 335) {  
 return "NORTH WEST";  
 } else if (degrees > 335 && degrees <= 360) {  
 return "NORTH";  
 }  
}

//function to convert javascript Date object into MM/DD/YY format

//by extracting day, month and year values, adjusting month numbering from zero-indexed value to one-indexed value and then CONCATENATING these into string and return the string. © <https://dzone.com/articles/javascript-convert-date>  
function GetFormattedDate(todayTime) {

let month = todayTime.getMonth() + 1;  
 let day = todayTime.getDate();  
 let year = todayTime.getFullYear();  
 return day + "/" + month + "/" + year;  
}