/\*\*

\* This program is an event planner based within an html document. When the program is

\* opened, it will display a calendar with the current month active within it. The user has the

\* ability to navigate the months of the year either by going forward or going backward in

\* chronological order of. The user is able to select any date and the web document will

\* navigate to said url and display a new screen.

\*

\* Within the the new screen it will have two different abilities. The first ability is the

\* capability to add an event for that day. The contents of what makes a date is the host of

\* the events name, the name and description of the event, and finally the times which the

\* event will be held. The times of the events are based on 30 minutes segments with

\* either xx:00 start of xx:30 start. When the event is created it will be displayed within the

\* day that is was created on.

\*

\* The second ability to navigate click on any pre-arranged event and RSVP yourself for

\* the event. The prerequisites for the event entail the user’s name and what times

\* the user will be available for the event. When the user RSVPs for the event, they will

\* show up under the attendees column in the event itself.

\*

\*

\*

\*

\* @author Mark VanLandingham, Quinn Meier, Alex Bohlken, Dylan Vondracek

\* @version 1.0

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function drawEvents

\* This method puts all the events in daysEvents onto the page

\*

\* @pre daysEvents- is filled with all events on the page's date

\* @post none

\* @since September 17, 2017

\*

\*/

/\*\*

\* @File date-vm.js

\* This date-vm.js file verifies that the date that is inputted by the user will be a valid

\* input for the event. Valid input implies that the day, month and the year are all valid

\* expressions. Users will be able to distinguish between leap years and the extra within

\* the month of February.

\*

\* @version 1.0

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function validateDate

\* This function takes a date as a parameter. This function which came from stackoverflow,

\* takes said date, and verifies that there is indeed valid information from the date.

\*

\* @pre Date - takes a date as the input of the function

\* @post a composed date via just date, month, and the year of the event.

\* @since September 17, 2017

\*

\*/

//Funciton from PhiLho & Anil Namde of Stack Overflow - <https://stackoverflow.com/questions/276479/javascript-how-to-validate-dates-in-format-mm-dd-yyyy>

/\*\*

\* @Function getEventsForDays

\* This function takes a date as a parameter. This function which came from stackoverflow,

\* takes said date, and verifies that there is indeed valid information from the date.

\*

\* @pre Date - takes a date as the input of the function

\* @post a composed date via just date, month, and the year of the event.

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function getTimes

\* This function takes each time block and formats that information into actual times

\* to be used for the event.

\*

\* @pre timeBlocks is formatted correctly: non-negative numbers separated by

\* commas, ordered from least to greatest

\* @post returns string that says times

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function expandEvent

\* This function takes the event and displays all the information from the event to the

\* html document. It will display all the active events from the active day.

\*

\* @pre event - is an active event from the

\* @post none

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function clearEventFieldst

\* This function clears all the active events from the current day.

\*

\* @pre element - is an active day from the calendar.

\* @post none

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function addUserToEvent

\* This function will add a user to the event. For a user to be created to an even

\* they first need to leave their name and select the time blocks that they will be attending

\* the event for.

\*

\* @pre element - is an active day from the calendar.

\* @post none

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function updateEvent

\* This function sends information from the html document to the airTable database

\* to be updated with the information

\*

\* @pre data - information to be updated at the database..

\* @post none

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function checkUserName

\* This function verifies that what the user provided was valid input.

\*

\* @pre name - the name of the person attending event..

\* @post true if valid name, false if the name is invalid

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function convertToStandardTime

\* This function takes all the existing military time and converts that time into

\* standard time

\*

\* @pre none

\* @post the time slots in standard time.

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function convertToMilitaryTime

\* This function takes all the existing standard time and converts that time into

\* military time

\*

\* @pre none

\* @post the time slots in military time.

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function logEvents

\* This function takes the person and their selected time slots and sends this information to

\* the  Air Table database.

\*

\* @pre none

\* @post none

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function getPeopleArray

\* This function formats the string of event participants. It takes all the user

\* information for an event and converts it.

\*

\* @pre peopleString - Is properly formatted from the database.

\* @post Returns an array of formatted participants’ data

\* @since September 17, 2017

\*

\*/

File server.js

/\*\*

\* @File server.js

\* The server.js gives the ability for us to navigate or database and access different uses

\* from within, This allows us to access different days within the month and navigates to

\* said screen. We will be using Airtable as our database  to store all the information use

\* to store the event. When each date is called it will then go to that day and display any

\* events from said day

\*

\* @version 1.0

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function eachPage

\* Each page is just a singular HTML file that will hold all of the events for said day. It takes

\* the record of the information and the following day as parameters. It will then display

\* this information within the HTML page.

\*

\* @pre records-information based on the events

\* fetchNextPage- gets the next day of the month to make a clickable link.

\* @post no return

\* @version 1.0

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function /date/:date/events

\* This function takes in two parameters. This function verifies that the date is in correct

\* format.Then the function will go to the eventArrs function and obtain all the current

\* events for that day

\*

\* @pre eventsArr has all events on it

\* @post returns array of events on specific date

\* @version 1.0

\* @since September 17, 2017

\*

\*/

//converter.js

/\*\*

\* @Function getPeopleArray

\* This function formats the string of event participants. It takes all the user

\* information for an event and converts it.

\*

\* @pre peopleString - Is properly formatted from the database.

\* @post Returns an array of formatted participants’ data

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function getTimes

\* This function takes each time block and formats that information into actual times

\* to be used for the event.

\*

\* @pre timeBlocks is formatted correctly: non-negative numbers separated by

\* commas,ordered from least to greatest

\* @post returns string that says times

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function blockConversion

\* This function takes given block string and converts that into the 12 hour time.

\*

\* @pre block is a number between 0 and 48

\* @post returns converted string

\* @since September 17, 2017

\*

\*/

/\*\*

\* @File app.js

\* This app.js file navigates between different html documents. It also functions with

\* communicating with the AirTable database in creating an event.

\*

\* @version 1.0

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function navigateToAdminMode

\* This function calls the admin mode html window.

\*

\* @pre none

\* @post none

\* @version 1.0

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function createEvent

\* This function takes the contents of the event and sends that information to the AirTable

\* database.

\*

\* @pre none

\* @post none

\* @version 1.0

\* @since September 17, 2017

\*

\*/

//admin-vm.js

/\*\*

\* @Function createEvent

\* This function takes the contents of the event and sends that information to the AirTable

\* database.

\*

\* @pre none

\* @post none

\* @version 1.0

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function buildCreateElements

\* This function waits until all the information is loaded then will go in and display

\* all the active events.

\*

\* @pre properly formatted event.

\* @post none

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function clearCreateEventElements

\* This function clears all the active event boxes that have been checked

\*

\* @pre none.

\* @post none

\* @since September 17, 2017

\*

\*/

/\*\*

\* @Function buildEventElements

\* This function takes all the times slots that are going to be used for the event, it takes

\* those elements and creates time slot boxes based on how long the event is.

\*

\* @pre none

\* @post none

\* @since September 17, 2017

\*

\*/