

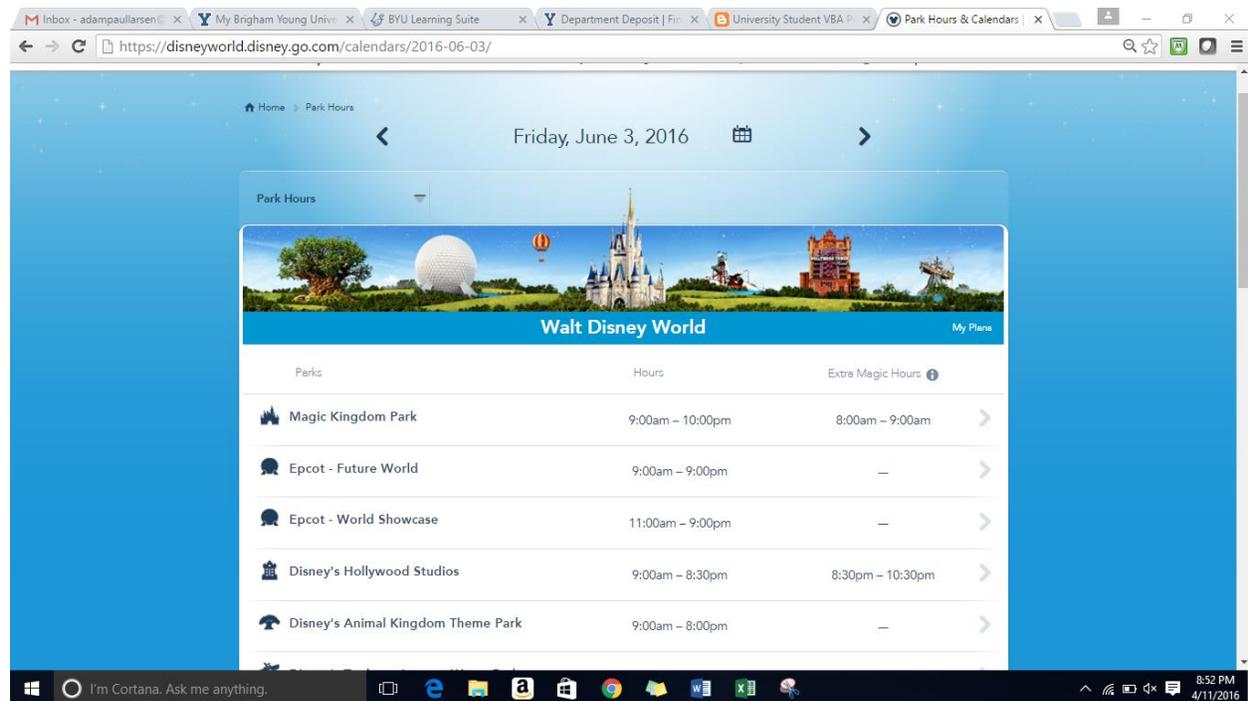
Cell B2 is for the user to enter the first day of their visit, and then they enter the total # of days in cell B3. Cells D2:E3 provide a key as to which parks have Extra Magic Hours for resort guests on each day. Extra Magic Hours allow exclusive access to the designated park for resort guests either before or after the normal park hours (generally 1 hour in the mornings and 2-3 hours in the evenings). Knowing when the Extra Magic Hours can be beneficial for both resort guests (shows where they can maximize their time) and non-resort guests (shows days when they may want to avoid a park as resort guests will already be there or will be gathering to stay late).

The remaining fields shown on the screen shot above are the headers for the data that will come in. All of the dates for the user’s visit will be listed under the Date field. The corresponding park hours will then be listed in each row under the correct park. Epcot has been split into two columns because there are two sections of the park that have different hours.

Once the data has been updated for the 1st day of the visit and the number of days, the user goes to the “Disney World Planning” tab in the ribbon and selects “Park Hours”.



The sub procedure will first delete any previous hours or closures data. Then I’ve set it up so that it uses the agent to go to Disney’s site for Walt Disney World Hours. Since the site only lists hours one day at a time, this project is helpful so that you can see all of the times together for your whole stay.



The sub procedure stores the first day of the visit as a string variable which is then concatenated onto <https://disneyworld.disney.go.com/calendars/>. This makes it so that it will initially open up to the first day that we need hours. The number of days is stored as a variable that is used to determine how many times the sub procedure will need to loop through, pulling in each day's hours. Each time that it loops through a page to pull in the park hours, it first finds the date to put in the first column of each row.

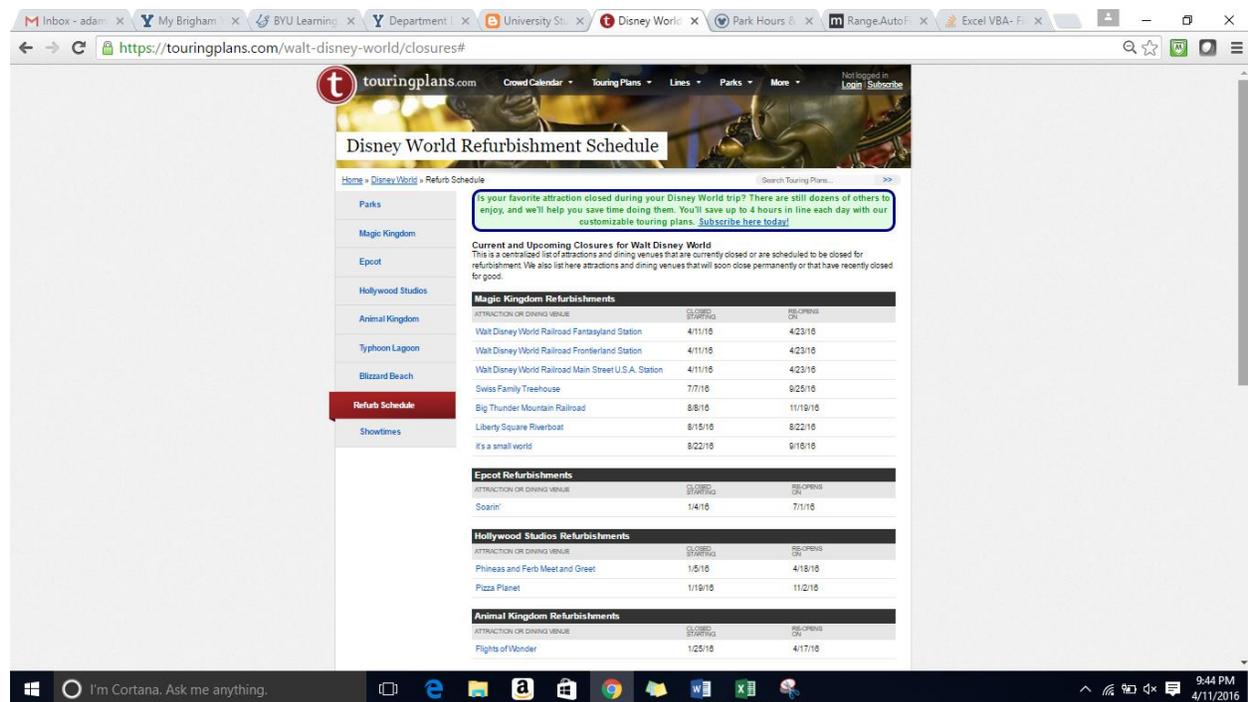
Within the Do Loop is a For Loop that loops five times, once for each park that it is pulling in hours for. Each time it moves to find the text where the hours are located in the html of the web page.

```
<div class="name">
    <span class="iconContainer"><span class="icon icon-places-m disneyland-default"></span></span>
    <span class="parkName">Magic Kingdom Park</span>
</div>
<div class="hours range"><p>9:00am &ndash; 10:00pm</p></div>
<div class="magicHours"><p>8:00am &ndash; 9:00am</p></div>
<div class="icon-interactive-m next-default inactive"></div>
```

First it moves to the block after hours range where it stores the range as variable. I put in a if statement that will replace "–" with "-" if it doesn't originally come in as a dash. Then it moves to find if there are Extra Magic Hours. It gets the text at that spot which will either indicate that there aren't any "noHours", or it will give the hours. I then search the text for "am" and highlight it yellow if "am" is found. Then it goes back to search for "pm" and will highlight it blue if "pm" is found. An evening Extra Magic Hour may go until 12am, but it will still highlight blue because it will find the "pm" after the "am".

After the For Loop goes through to add the hours for each park, I convert the visit date into a date so that I can add 1 to make it go to the next date. Then I convert it back to a string to once again concatenate it onto the base URL so that I can go to the next page for the next date and move through the loop again.

I created a second sub procedure to then bring in the closures for each date. When the procedure is called through the Closures button, the agent pulls up <https://touringplans.com/walt-disney-world/closures>.



This site lists all of the current and upcoming closures that have been announced. The challenge in moving through this list is that I don't know how many closures will apply and further down the page are details that I didn't want to bring in. Each time I would be moving to "<a href", but I wanted to stop after the Animal Kingdom closures. To make this happen I start off the sub procedure by moving down to the Typhoon Lagoon closures which immediately follows Animal Kingdom.

```

<caption>Animal Kingdom Refurbishments</caption>
<thead>
  <th width="348">Attraction or Dining Venue</th>
  <th>Closed<br>Starting</th>
  <th>Re-Opens<br>On</th>
</thead>
<tr>
  <td>
    <a href="/animal-kingdom/attractions/flights-of-wonder">Flights of Wonder</a>
  </td>
  <td>1/25/16</td>
  <td>4/17/16</td>
</tr>
</table>
<br>
<table>
<caption>Typhoon Lagoon Refurbishments</caption>
<thead>
  <th width="348">Attraction or Dining Venue</th>
  <th>Closed<br>Starting</th>
  <th>Re-Opens<br>On</th>
</thead>
</table>

```

I then store that position as z. Then I start a Do Loop that will continue to go while a.position is less than z. This made it so that the sub procedure exits the loop once I've gone as far as I want to go.

I set up a variable for the visit date which starts by being assigned to cell A8 which is where the first visit date will be. Then I assign variables for the park attraction, close date, and open date from the first attraction listed.

```

<td>
  <a href="/magic-kingdom/attractions/walt-disney-world-railroad-fantasyland-station">Walt Disney World Railroad Fantasyland Station</a>
</td>
<td>4/11/16</td>
<td>4/23/16</td>

```

I first move past "a href" to start right after ">". I pull the name in as the attraction. Then I move down to after the first "<td>" to save that as the close date (this is the first date the attraction will be closed). Then I move to next "<td>" which I save as the open date (this is the date the attraction will reopen).

Once I have the information stored as the correct variables, I move into a For Loop that loops through the same number of times as the number of days the user will be visiting. At the end of each time through the For Loop I add 1 one to the visit date so that it will check against all of the dates the user will be visiting. Each time it goes through an If Statement to check if the visit date is greater than or equal to the close date, but less than the open date. If it is, it will add a heading that says "Closed: " concatenated with the name of the attraction. It will also place a bold X under the heading in the appropriate row, highlighted in red.

DATE	EPCOT					Closed: Soarin'
	MAGIC KINGDOM	EPCOT - FUTURE WORLD	EPCOT - WORLD SHOWCASE	HOLLYWOOD STUDIOS	ANIMAL KINGDOM	
6/2/2016	9:00am – 10:00pm	9:00am – 9:00pm	11:00am – 9:00pm	9:00am – 8:30pm	9:00am – 8:00pm	X

If the visit date doesn't fall within the closed and open dates, then nothing will be added. Looping through each day makes it so that if the ride opens or closes during your visit you will be able to see which days it is specifically closed.

DATE	MAGIC KINGDOM	EPCOT		HOLLYWOOD STUDIOS	ANIMAL KINGDOM	Closed: Swiss Family Treehouse	Closed: Big Thunder Mountain Railroad
		EPCOT - FUTURE WORLD	EPCOT - WORLD SHOWCASE				
8/6/2016 9:00am - 10:00pm		9:00am - 9:00pm	11:00am - 9:00pm	9:00am - 8:30pm	9:00am - 5:00pm	X	
8/7/2016 9:00am - 10:00pm		9:00am - 9:00pm	11:00am - 9:00pm	9:00am - 8:30pm	9:00am - 5:00pm	X	
8/8/2016 9:00am - 10:00pm		9:00am - 9:00pm	11:00am - 9:00pm	9:00am - 8:30pm	9:00am - 5:00pm	X	X
8/9/2016 9:00am - 10:00pm		9:00am - 9:00pm	11:00am - 9:00pm	9:00am - 8:30pm	9:00am - 5:00pm	X	X

Each time through the loop the information is put in column "i" which starts out with a value of 7. Before starting the loop over, the sub procedure first checks to see if there is currently a heading for a closure in column "i". If there is then it will add 1 to i, otherwise i will remain the same the next time through the loop. This ensures that it will only move over a column each time a closure has been listed.

Finally, at the end of the procedure I included a command so that the columns for the closures get auto fitted. This makes it look nicer so that the headings don't get cut off by the next column if they are really long.

LEARNING AND CONCEPTUAL DIFFICULTIES

Some of the difficulties I faced were already discussed above, but below are a few additional difficulties I came across during this project and how I resolved them.

- Listing Extra Magic Hours
 - I originally planned on specifically listing out what the Extra Magic Hours would be for a given park. However, I found that it was difficult to add them within the two dimensional grid that I built out. I considered adding a second line for each date that would be blank for each park that didn't have Extra Magic Hours and only include the hours for the park(s) that did have Extra Magic Hours. In the end I felt that this would clutter the data too much and not look as clean.
 - I figured that the best alternative was to mark which park had Extra Magic Hours and whether it was in the morning or evening. This way the user could see where they were and have a general idea what they would be.
- Moving to the next page
 - My first thought had been that after going through the first page of hours on the website I would be able to advance the page to the next day. I tried finding a place on the page where I could give the submit command, but was unable to find anywhere that it would work.
 - My workaround was to add a value to the day and then concatenate it onto base URL as a string and this seemed to work just as well.
- Speed of retrieving the park hours
 - Once I had finished the code for retrieving the park hours I adjusted it so that the web page would not be visible. However, I found that it was taking a very long time to pull in the data for each date as it executed the sub procedure. There were even times that it would eventually fail.
 - I realized that if I changed the a.openpage argument to be False instead of True, it went much faster as it used my default browser rather than IE.

- With the closures there wasn't any noticeable difference in speed, and my default browser caused some formatting issues in the attraction name, so I kept it as True on that procedure.

Overall, going through this project helped me become much more familiar with using the agent to pull in data. I learned that I had to get creative in order to portray the data that I wanted to show as well as in how to loop through the data on the page. Specifically, I was really stuck at first on how to only loop through the data that I wanted on the closures page. Coming up with the solution to first find out how far I wanted to go down and then using that as my out for the loop was very helpful.

I felt that this project was a good illustration to me of the value that automating Excel can have. It is great to be able to pull in data from two sources to match them together in a way that will be beneficial. This program will be beneficial to me as I plan my own trips to Disney World and hopefully others will find it useful as well.

ASSISTANCE

Originally I was planning on building a table on a second sheet to list the closures, but based on feedback from Professor Allen I decided to list it on the same sheet as the hours so that it matched up with the dates. When I first listed the closures, I listed them on each row. However, I had another student that suggested I put the name as a heading and then mark each row where it is closed. I liked this format better so I implemented it and then decided to make some additional updates to it (highlighting, bolding, and auto fitting the columns). Other than incorporating these suggestions, I did everything else on my own.