Requirements

Complete each of the following requirements.

Step 1. 111/p3/LabFunctions-04.js: Lab 4

You should have completed Week 4 Lab and uploaded LabFunctions-04.js to your 111/p3 folder on the UO Pages server.

Step 2. 111/debug/arraysObjects.js and 111/debug/arraysObjects.html: Lab 5

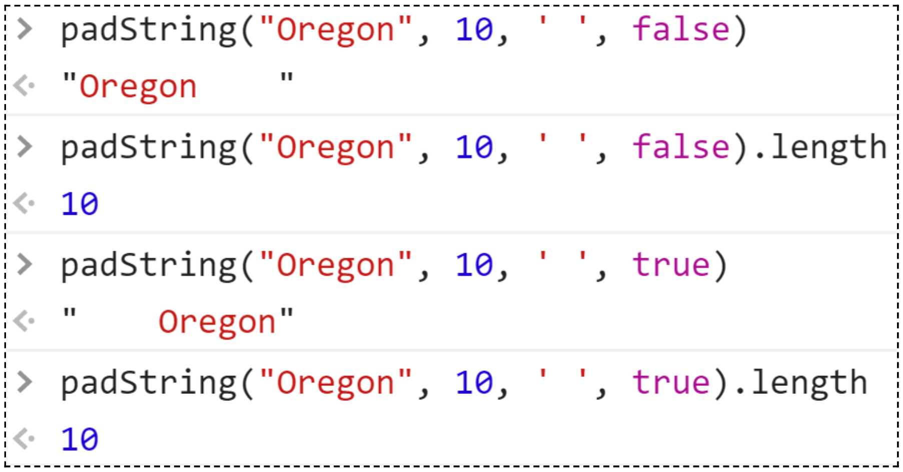
You should have completed Week 5 Lab and uploaded arraysObjects.js and arraysObjects.html to your 111/debug folder on the UO Pages server.

Step 3. 111/p3/catalog.js: padString function

Create a function called ***padString(data, maxLength, padCharacter, padLeft)*** that returns ***data*** padded with ***padCharacter***, either to the left (before ***data***) or the right (after ***data***) based on ***padLeft***, up to ***maxLength***. The function must be saved in a file called ***catalog.js***.

- ***data***: Data to be padded, consider using *String()* to assist in performing length calculations if data is not a string
- ***maxLength***: The numerical maximum length of padding and data combined, and no padding if the length of the data equals ***maxLength***
- ***padCharacter***: The string padding character (e.g. space)
- ***padLeft***: Boolean true or false, where true means pad to the left of data, and false means pad to the right of data

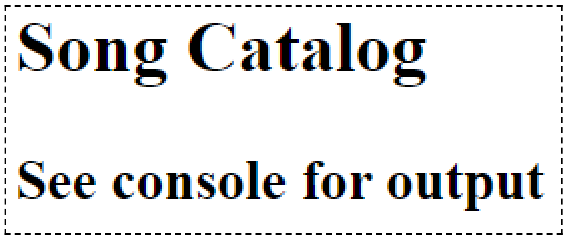
Use the Chrome Console to test the ***padString()*** function. For example:



```
> padString("Oregon", 10, ' ', false)
<· "Oregon    "
> padString("Oregon", 10, ' ', false).length
<· 10
> padString("Oregon", 10, ' ', true)
<· "    Oregon"
> padString("Oregon", 10, ' ', true).length
<· 10
```

Step 4. 111/p3/catalog.html: Catalog web page

Create an HTML file ***catalog.html*** in your 111/p3 folder. Code the HTML based on your hello-world.html file, changing the *title* to "Song Catalog" and removing all HTML within the *body* tags. Add an *h2* tag with "Song Catalog" and an *h3* tag with "See console for output." The required HTML output is shown below.



Song Catalog

See console for output

Step 5. 111/p3/catalog.html: Reference catalog.js external JavaScript file

Update ***catalog.html*** to use a ***script*** tag to reference the ***catalog.js*** external JavaScript file.

Step 6. 111/p3/catalog.js: Array of song objects

Create an array of song objects using the object literal syntax. Your array must have five songs. You can use any music billboard/chart that lists top songs to find the required information. Eventually, this array of song objects will be used to populate your song catalog. Each song object must have the following properties:

- ***title***: song title
- ***artist***: artist
- ***position***: numerical order on the chart
- ***weeksOnChart***: number of weeks on the chart

Below is an example of one of the objects. Remember, you must create five objects, and assign them to an array. The array should be declared using ***const***. Use the variable name ***songs*** for the array.

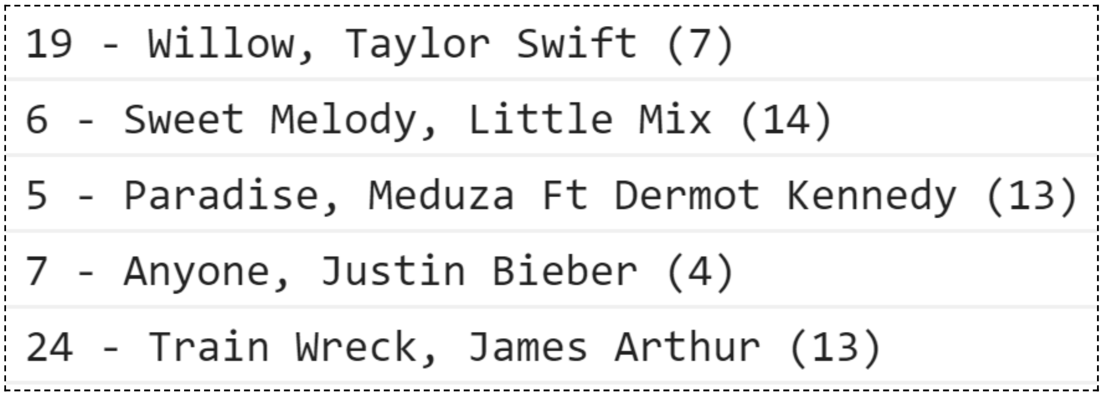
```
{
  title: "Song Title",
  artist: "Artist Name",
  position: 1,
  weeksOnChart: 10
}
```

Step 7. 111/p3/catalog.js: catalogObject custom object

You will be creating a custom object called ***catalogObject*** that will server as a container for all of the song catalog functionality. This object will include catalog properties and methods. We will incrementally add and test functionality. Initially, we'll declare the ***catalogObject***, declare an array property ***_songs*** initialized to an empty array, add method ***addSong(props)*** to add a song to ***_songs***, and method ***listSongs()*** to list all of the songs in ***_songs***:

- ***_songs***: empty array property
- ***addSong(props)***: Add a song to ***_songs***, where method parameter ***props***, short for properties, is a song object. The ***addSong()*** method must use the ***String*** [***trim()***](https://www.w3schools.com/jsref/jsref_trim_string.asp) ***(https://www.w3schools.com/jsref/jsref_trim_string.asp)*** method to ensure the song ***title*** has no leading or trailing white space.
- ***listSongs()***: Iterates through _songs array and lists out the songs. You must use a **template literal** **(https://css-tricks.com/template-literals/)** to output the songs to the Console. Template literals make formatting multi-variable output easier.

```
console.log(`${position} - ${title}, ${artist} (${weeks})`);
```



```
19 - Willow, Taylor Swift (7)
6 - Sweet Melody, Little Mix (14)
5 - Paradise, Meduza Ft Dermot Kennedy (13)
7 - Anyone, Justin Bieber (4)
24 - Train Wreck, James Arthur (13)
```

Use the following steps to test catalogObject custom object

- Create an instance of ***catalogObject***:

```
const catalog = Object.create(catalogObject);
```

- Load the catalog of songs by creating a ***for*** loop that iterates through the ***songs*** array, calling ***catalog.addSong()*** for each song in the ***songs*** array.
- Call ***catalog.listSongs()*** to list each song in the ***songs*** array
- Test your code changes by using the browser refresh button to reload ***catalog.html***

Step 8. 111/p3/catalog.js: Add padString() to catalogObject, and update addSong() and listSongs() to support padded output

The output in the Console isn't very easy to read, so format the output so that each column is padded.

- Move the ***padString()*** function into the ***catalogObject***
- Add the following properties to ***catalogObject*** to keep track of the maximum length of each column as a song is added to the catalog.
  - ***_maxTitleLength***: Maximum ***title*** property length, initialized to the length of column header **"Title"**
  - ***_maxArtistLength***: Maximum ***artist*** property length, initialized to the length of column header **"Artist"**
  - ***_maxPositionLength***: Maximum ***position*** property length, initialized to the length of column header **"Position"**
  - ***_maxWeeksOnChartLength***: Maximum ***weeksOnChart*** property length, initialized to the length of column header **"Weeks On Chart"**
- Update ***addSong()*** to update each of the maximum length properties as a song is added
- Update ***listSongs()*** to use ***padString()*** and each of the maximum length properties to format the catalog Console output, and to include column headers, also padded

Position	Title	Artist	Weeks On Chart
19	Willow	Taylor Swift	Weeks On Chart
6	Sweet Melody	Little Mix	Weeks On Chart
5	Paradise	Meduza Ft Dermot Kennedy	Weeks On Chart
7	Anyone	Justin Bieber	Weeks On Chart
24	Train Wreck	James Arthur	Weeks On Chart

Extra Credit Step 9. **[10 pts extra credit] 111/p3/catalog.js: Add getSongByIndex()**

Add method ***getSongByIndex(index)*** to return a song object from ***catalogObject _songs*** given an ***index*** value, or an empty object if the ***index*** value is invalid

Extra Credit Step 10. **[10 pts extra credit] 111/p3/catalog.js: Add getSongByTitle()**

Add method ***getSongByTitle(title)*** to return a song object from ***catalogObject _songs*** given an ***title*** value, or an empty object if the ***title*** value is not found in ***_songs*** array

Extra Credit Step 11. **[10 pts extra credit] 111/p3/catalog.js: Update listSongs() to return catalog in sorted order by position**

Modify ***listSongs()*** to ***listSongs(sorted = false)***, and display the catalog of songs in sorted order based on the song ***position*** property. Note that the **(sorted = false)** supplies a default value of false for the ***sorted*** method parameter, so you'll need to use ***sorted*** to determine if sorting should be applied or not.

Step 12. **Upload and Test**

Remember you must upload all of the items for Project 3 to your UO Pages account using an FTP program, and verify that the files successfully uploaded. You will only be graded on what you've uploaded.

Your **111/p3** folder will contain the following files for this project:

- **[5 pts] LabFunctions-04.js**
- **[90 pts] catalog.js** and **catalog.html**

Your **111/debug** folder will contain the following files for this project:

- **[5 pts] arraysObjects.js** and **arraysObjects.html**

Do not change any item that you've uploaded once the due date deadline has passed, as we use the time/date stamp of the files to determine if you uploaded the files before the deadline.

[redacted]