# Algorithm

Visual Basic:

Javascript:

All the modules will be installed using the “package.json” dependencies object (**npm Install**).

**Script will check the input argument which is passed as “-I” parameter in the command prompt.**

**Script will check the all the other optional Arguments and update the fallback.**

**=====================================================================================**

**-I ------ Input Path argument not passed in the command line, script won’t execute.**

**-v ------ ePub Version argument not passed, script will take 3.0 as default.**

**-t ------ ePub Title argument not passed, folder name will be updated as title.**

**-ID ------ ePub ISBN argument not passed, “9999999999999” will be updated as ISBN.**

**-l ------ Language argument not passed, “en-US” as default will update.**

**-c ------ Creator Name argument not passed, appropriate tag won’t be updated in metadata.**

**-p ------ Publisher argument not passed, appropriate tag won’t be updated in metadata.**

**-r ------ Copyright Info argument not passed, appropriate tag won’t be updated in metadata.**

**-a ------ Audio Book argument not passed, default value “1”, media-overlay tag will add.**

**-o ------ Orientation argument not passed, default value “both”.**

**-g ------ guide Required argument not passed, default false.**

**-h ------ help Help**

**=====================================================================================**

**Once Arguments are checked and script will start its execution by taking “Input Path”.**

**Function Traverse\_TitleWise\_Folder():**

1. **Pass “Input Path” into the function.**
2. **Checking the current input Path exists or not. If it is not exist, stop its execution.**
3. **If it exists, go to 4.**
4. **Read input Directory.**
5. **Loop the Directory, based on the folder/files length within the input folder (immediate children folder).**
6. **Join the Input path with current folder in the loop.**
7. **Check the current files/folder is directory.**
8. **If it is directory go to 9.**
9. **If the founded directory name is “OEBPS”, go to 11. Else, go to 10.**
10. **Go to 1, pass current directory as input Path.**
11. **Pass Current Directory Name & Current Directory Folder into the “Traverse\_OEBPS\_Folders() Function”.**
12. **After the above function executed, run the spineUpdate() Function.**
13. **Then, check the ePub title argument passed (-t), if it is not passed then check the folder name of the title and assign the folder name as title name.**
14. **Then, run the metaDataUpdate() Function with pass the value of ePub title.**
15. **Update the opf Content and write the contents in it.**
16. **If All the files are reads, ends the Loop and Function.**

**Function Traverse\_OEBPS\_Folders():**

1. Read all the children folder/files of the OEBPS folder.
2. Loop through all the sub folder/files of the OEBPS folder, then go to 3.
3. Check the folder/files, if current object is folder then go to 4, else 5.
4. Go to Step 2.
5. Check current file is “cover.jpg” or “frontcover.jpg”, if yes it will assign “true” to the “**coverImageFound**” variable.
6. Then send “FileName, FileFullPath,OEBPSFolder” to **FileChecking() Function**.

**Function** FileChecking()**:**

1. Increment the value of “I”.
2. If Files present in the root folder of “OEBPSFolder”, then update the href value as “FileName”, else update the href value as “rest of the path (/filefolder/filename)”.
3. If current file has the extension as mentioned below, “.ncx”, “.css”, “.mp3”, “.otf”, “.ttf”. – Go to Step 7
4. If current file has the extension “.jpg” or “.png” – Go to Step 8
5. If current file has the extension “.html” or “.xhtml” – Go to Step 9, 10 & 11
6. If current file has the extension “.smil” – Go to Step 12 & 13
7. Based on the extension, script will pass the values to itemTag() function.
   1. First variable of the function is ID – File Format + ‘-‘ + i
   2. Second Variable – href
   3. Third Variable – Media Type – **extension+xml** for xml files, **application/extension** for image and fonts files, **text/extension** for css files.
   4. Fourth Variable – Media Overlay – ‘’ - Empty
   5. Fifth Variable – Media Overlay Boolean Value – ‘false’
   6. Sixth Variable - Properties Boolean value – ‘false’
8. If current image file name is “cover.jpg” or “cover.png”, id of the file in OPF will be pass to “cover-image” in the itemTag() function. Rest of the image files are followed the same process which are mentioned in the Step 7.
9. If current html/xhtml file name is “nav” then script will change the properties Boolean value as “true” and pass the variables to the itemTag() function. Rest of the field are same as mentioned on Step 7.
10. If current html/xhtml file name is “cover” or “backcover”, then script will pass OEBPS Folder and current file to the **Finding\_Page\_Wise\_Smil\_File () Function** and if smil File and HTML File name are same, then go to Step 14. Else go to Step 15
11. If current HTML/XHTML file name are page wise html, then script will OEBPS Folder and current file into the **Finding\_Page\_Wise\_Smil\_File() Function** and also pass current file, href, OEBPSFolder,extension to **htmlPageID() Function**.
12. If current Smil file name are “cover’ or “backcover”, then script will pass id, href,mediatype to the **itemTag() Function**.
13. Other smil files are pass file, href & OEBPSFolder variable to **smilPageID() Function**.
14. Change the Boolean value of “mediaOverlayFound” as true and pass to **itemTag() Function**.
15. Change the Boolean value of “mediaOverlayFound” as false and pass to **itemTag() Function**.

Function Finding\_Page\_Wise\_Smil\_File():

1. Loop through files within the OEBPS Folder and check the smil files. And check the filename of smil and current html file are equals, then update true to smilFileFound variable. If current HTML file doesn’t have smil file then smilFileFound variable will be false.

Function htmlPageID() & smilPageID():

1. Replace the html file name and take only page numbers from the filename of HTML.
2. If Page number less than 10, then append “page00” before the pageNumber.
3. If Page number less than 100, then append “pge0” before the pageNumber.
4. Greater than 100 will keep as it is.
5. If already 0 present in the filename “script will write a error Log”.
6. Then pass the appropriate variables to the **itemTag() function**.
7. Also, write the pageID in the Array for Spine section Update.

Function itemTag():

1. If MediaOverlayFound variable is true and propertiesFound variable is false. Script will update manifest as “'<item id="' + **id** + '" href="' + **href** + '" media-type="' + **mediaType** + '" media-overlay="' + **mediaOverlay** + '"/>'”
2. If PropertiesFound variable is true, <item id="' + **id** + '" href="' + **href** + '" media-type="' + **mediaType** + '" properties="' + **'nav'** + '"/>'
3. If both are false, <item id="' + **id** + '" href="' + **href** + '" media-type="' + **mediaType** + '"/>

**Function spineUpdate():**

1. If coverfileFound is true, then cover id will be updated as first entry.
2. Else if both coverfileFound & backCoverFileFound is false, then loop through each element within the htmlIDArray and update one by one in ascending order.
3. If backcoverFileFound is true, update the entry at the last of the spine section.

**Function metaDataUpdate():**

1. Update the default elements in the metadata tag of OPF file.
2. ePub title will be pass through –t argument. IF it is not passed, then script by default update the **Title folder name as title**.
3. ePub Creator will be pass through –c argument. IF it is not passed, then script won’t update this tag in the OPF.
4. Rights will be pass through –r argument. IF it is not passed, then script won’t update this tag in the OPF.
5. Publisher will be pass through –p argument. IF it is not passed, then script won’t update this tag in the OPF.
6. ePub Language will be pass through –l argument. IF it is not passed, then script by default update the “**en-US**” as title.
7. ePub Orientation will be pass through –o argument. If it is not passed, then script by default update “auto”.
8. If –ab argument is 1, then script will update media:active-class in OPF. If it is empty or 0, then script won’t update this tag.