

# CSS for Paged Media (Pt 2 JavaScript)

**Objectives:** Review JavaScript for progressive enhancement. Add a Table of Contents at the beginning of the book so users can easily navigate chapters. Do this dynamically with JavaScript so chapters can be added/deleted and the TOC will update automatically.

**Starter files:** Use your completed files from Part 1.

## Part 2a: Map a Solution in Plain English

Reason through a solution in plain English and develop a list of tasks. Start by listing things you know or things you know should happen.

- Each chapter is identified as a `SECTION` that's a child of `#WRAPPER`.
- The title of each chapter is an `H1` in a chapter `SECTION`.
- The chapter title should display as the Table of Contents text.
- The Table of Contents should display between the book cover (`HEADER`) and first chapter (the first `SECTION` child of `#WRAPPER`).

Next, make a list of specific tasks for JavaScript:

1. Tell the function two things:
  - a. How to identify a chapter and its title text
  - b. Where to place the finished Table of Contents
2. Get all of the chapters.
3. Create an empty `OL` to hold the list of chapter titles.
4. Examine each chapter and:
  - a. Inject an `ID` for each chapter so it can be linked to
  - b. Create the `HREF` value to link to a specific chapter
  - c. Create the text for each Table of Contents entry
  - d. Create a new `LI` element
  - e. Create a new `A` element
  - f. Set the `HREF` of the `A`
  - g. Set the text of the `A`
  - h. Add the `A` to the `LI`
  - i. Add the `LI` to the list
5. Create a `SECTION` to hold the Table of Contents and give it an `ID` so we don't confuse it with a chapter `SECTION`.
6. Create an `H1` for the Table of Content.
7. Add the `H1` to the `SECTION`.
8. Add the `OL` to the `SECTION`.
9. Add the `SECTION` to the page after the `HEADER` (the book cover).

## Part 2b: Code It

### Ready the HTML

1. Edit the HTML to link to a JavaScript file:

```
<script src="js/toc.js" defer></script>
```

2. Create the JS file and folder as per the previous path.

### Code the JavaScript

#### Define the Function and Starter Elements

3. Call a function and pass it:
  - a. The chapter titles (H1)
  - b. The location after which it should be added to the page (HEADER)

```
makeBookmarks('#wrapper > section h1', 'header');
```

4. Define the function with two meaningful parameters:

```
function makeBookmarks(chapterTitles, putBookmarksHere){  
  
};
```

5. Normally, you could use const instead of var to grab elements on the page. However, the user agent we'll use to convert the HTML file to a PDF lacks support for const. So, we'll just use var throughout our code. As the first statement inside the function, grab all the chapter titles.

```
var CHTITLES = document.querySelectorAll(chapterTitles);
```

6. Next, create an OL for later use:

```
var OL = document.createElement('ol');
```

#### Create a Bookmark Link for Each Chapter Title

7. Set up a loop to iterate through all chapter titles:

```
for(i = 0; i < CHTITLES.length; i++){  
  
}
```

8. Within the loop, create a variable to store a text string that will be used as the ID for the current chapter:

```
var newId = 'ch' + (i+1);
```

9. Set the ID attribute of the current chapter:

```
CHTITLES[i].setAttribute('id', newId);
```

10. Create a string to use as the link's HREF:

```
var linkHref = '#' + CHTITLES[i].getAttribute('id');
```

11. Get the text of the current chapter title to use as the text for the link:

```
var linkText = CHTITLES[i].textContent;
```

12. Create two new elements – an LI and an A:

```
var LI = document.createElement('li');  
var A = document.createElement('a');
```

13. Set the new anchor element's HREF attribute and text:

```
A.setAttribute('href', linkHref);  
A.textContent = linkText;
```

14. Add the A (now with an HREF and text) to the LI:

```
LI.appendChild(A);
```

15. Add the LI to the OL:

```
OL.appendChild(LI);
```

16. This completes the FOR loop. The remaining code should be coded outside (after) the loop but still inside the function.

#### Finalize the Table of Contents and Add It to The Page

17. Create a new SECTION and give it an ID to differentiate it from the SECTIONs used as chapters:

```
var SECTION = document.createElement('section');  
SECTION.setAttribute('id', 'toc');
```

18. Create a heading for the Table of Contents:

```
var H1 = document.createElement('h1');  
H1.textContent = 'Table of Contents';
```

19. Get the element *after which* to insert the Table of Contents. At the same time, get that element's parent:

```
var HEADER = document.querySelector(putBookmarksHere);  
var WRAPPER = document.querySelector('#wrapper');
```

20. Unfortunately, JavaScript doesn't allow us to insert an element after another element. It allows us only to insert an element before another element. (JavaScript has `insertBefore()`, but there's no such thing as `insertAfter()`.) Therefore, we need to use `insertBefore()` to do the following:
- Grab the next sibling of the HEADER
  - Insert the Table of Contents *before* that sibling

You may want to research/read about how to use `insertBefore()`:

```
SECTION.appendChild(H1);  
SECTION.appendChild(OL);  
WRAPPER.insertBefore(SECTION, HEADER.nextSibling);
```

21. Save and refresh. It should work! There's now a Table of Contents before the first chapter. Unfortunately, CSS has injected *Chapter 1:* in it! There's also a dotted underline below the Table of Contents. We need to edit the CSS to exclude the Table of Contents from those styles.

## Part 2c: Fix the CSS

1. Create a new comment and rule at the bottom of the CSS to remove the dotted underline from the Table of Contents. (We need `#WRAPPER` in the selector to raise the rule's specificity.)

```
/** Table of contents */  
#wrapper > #toc {  
| border-bottom: none;  
| }  
}
```

2. Wonderful! No more dotted underline. Now let's exclude the Table of Contents from the chapter numbering. Edit the 2<sup>nd</sup> rule in the Chapter Numbering section to exclude it:

```
#wrapper > section:not(#toc) h1::before {  
| content: "Chapter " counter(chapter-num) " : ";  
| counter-increment: chapter-num;  
| }  
}
```

## Summary

The JS is complete and the CSS has been tweaked to accommodate the dynamic Table of Contents.