

Ch. 11 Further Functions

- Functions can be passed around the same way as any other value.
 - For instance, the `.length` property returns the number of parameters that a function has.
- Immediately Invoked Function Expression (IIFE) can be used to create sections of limited scope within the code. For example,

```
let a = 1;  
let b = 2;
```

```
((()=>{  
    const temp = a;  
    a = b;  
    b = temp;  
})();
```

- Can be used to create a temporary variable that will no longer be in scope once the function has closed.
 - This type of code is great for initializing a program.
 - If the program is going to use strict mode, it is recommended to put the strict sections within an IIFE as well.
- Recursive functions call themselves repeatedly until a condition is met. From what I can tell of the reading, their format more or less follows what I've seen in other languages.
- Promises help to simplify the spaghetti code that can come about as a result of using callbacks. They go through several different stages of life:
 - Pending – the promise has called an asynchronous operation, waiting for the operation to be completed.
 - While waiting, the promise is unsettled.
 - Once the operation is complete, the promise is considered settled. This can then result in one of two outcomes:
 - Resolved – The operation was completed successfully.
 - Rejected – something went wrong during execution.
- Promises tend to conform to the following layout:

```
const promise = new Promise( (resolve, reject) => {  
    // initialization code goes here  
    if (success) {  
        resolve(value);  
    } else {  
        reject(error);  
    }  
});
```

- The following fact pulled directly from the book stood out to me as particularly interesting: “JavaScript is a multi-paradigm language, meaning that it can be used to program in a variety of paradigms (and sometimes a mash-up of them!).”
- Currying a function involves calling a function over and over, slowly providing more of its arguments until all of them have been provided.

Ch. 8 Transforms and Transitions

- Transform is used to
 - Translate,
 - Rotate,
 - Scale, or
 - Skew
 - Any element on your page.
- Transforms don’t work on inline elements, so be sure to add “display: inline-block;” to your .css file for any of the appropriate elements.
- Note that transforming does not reflow the page, so it is only suitable for some applications.
- Different transformations on the same object can be done in a space-separated list, and will be applied in the same order in which they are written.
- For transitions to work, there must be a valid midpoint between the two states.
 - With the exception of visible -> hidden and vice versa.
- Animation is worth looking into more, but seems to be based on having multiple images that scroll through one after the other. Similar to Flash animation.

Getting Started with JSON Web Tokens

- JWTs are both compact and self-contained
- JWTs are useful in a couple different scenarios, including: authentication and information exchange
- JWTs consist of three sections separated by dots
 - Header
 - Typically has two parts – the type of token and the hashing algorithm
 - Payload
 - Contains claims, which are statements about an entity
 - Signature
 - Essentially is just hardening against message tampering
- JWT is widely used and should be used for a number of reasons, including native parsing in most languages, and a very compact size

Develop, Debug, Learn?

- As web developers, we often caught up in the rush of doing the next best thing, instead of working with the things that we already have
- This is the wrong focus – the focus should always be on the end users of the website
 - This is largely visible in old tech, which is often abandoned in the pursuit of “new”
- It is important to learn during development, and to learn during debug