Lab Notes: Week 9

For future weeks, most of the lab sheets will be self contained, and will not require you to refer to additional notes. However, this week, I want to briefly to the idea of functions and Objects in JavaScript.

Functions in JavaScript

JavaScript embraces functional programming. As such, functions are a core part of the language. Below, I want to briefly discuss some of the key concepts in JavaScript functions.

Functions as Variables

In JavaScript, functions are first class objects. This means that they can be passed around like any other object. For example, we can assign a function to a variable:

```
const myFunction = function () {
  console.log("Hello World");
};
```

Functions as Arguments

We can also pass a function as an argument to another function, this is commonly referred to as a callback function:

```
const myCallbackFunction = function (callback) {
  callback();
};

const helloWorld = function () {
  console.log("Hello World");
};

myFunction(helloWorld); // will output hello world; however, this is
    invoked by myFunction.
```

Finally, we can shorten the above code by using an arrow function:

```
const myFunction = (callback) => {
  callback();
};

myFunction(() => {
```

```
    console.log("Hello World"); // will output hello world; however, this
        is invoked by myFunction.
});
```

JavaScript Objects

JavaScript objects differ from objects in other languages. We can think of JavaScript objects as a collection of key-value pairs. For example, we can create an object that represents a person:

```
const person = {
  name: "John",
  age: 30,
  address: {
    street: "123 Fake Street",
    city: "Belfast",
  },
};
```

We can access the properties of an object using dot notation:

```
console.log(person.name); // will output John
```

Object are extremely useful for representing data. For example, we could create an array of people:

```

const people = [
  {
    name: "John",
    age: 30,
    address: {
      street: "123 Fake Street",
      city: "Belfast",
    },
  },
  {
    name: "Jane",
    age: 25,
    address: {
      street: "123 Fake Street",
      city: "Belfast",
    },
  },
];
```

We can then loop through the array and access the properties of each person:

```

people.forEach((person) => {
```

```
  console.log(person.name);
});
```

Notice how the `forEach` method takes a callback function as an argument. This is a common pattern in JavaScript, discussed above.