Lab 3 Javascript Function and Anonymous Function

Anonymous Function Definition

```
function() {
  alert("Hello World!");
}
```

The above definition would not be triggered unless something calls it.

Anonymous Function Self Execution

```
function() {
  alert("Hello World!");
}()
```

How the following work is that, the addEventListener will register a function definition to the event symbol.

```
<script>
document.getElementById("myBtn").addEventListener("click", function() {
  alert("Hello World!");
});
</script>
```

Now a more intuitive way is to interpret the above example in the following. In the example below, we split and move the function definition outside, thus, the “click” event is registered with a function name or let us say a function symbol.

```
<script>
document.getElementById("myBtn").addEventListener("click", myFunction);

function myFunction() {
  alert ("Hello World!");
}
</script>
```

The above example will not issue an error for the reason that, when the HTML parser parses the line of the event registration, it wasn’t executing the **myFunction** function right away until you click the button itself and it will look for the function definition in the memory.

If the parser parses the scriptlet in the html and execute them right away and order of the function definition would matter.

Another way of registration, this is a manifest of this symbolic programming language’s feature.

```
<script>
document.getElementById("myBtn").onclick = displayDate;

function displayDate() {
  document.getElementById("demo").innerHTML = Date();
}
</script>
```

Takeaway

*** HTML DOM Model (skipped here)**

*** CSS Box Model**

*** Javascript Event Registration**