

# Notes

---

## Class example

**PROBLEM** : Increment a number by one (modified add two numbers)

---

**STEP 1** : start with a minimal HTML5 document

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Increment a number</title>
  <meta charset="utf-8">
</head>
<body>
  <!--all marked up content here-->
<script>
  <!--javascript code here-->
</script>
</body>
</html>
```

---

**STEP 2** : Add the following snippet to the script tag:

```
// define the function plusOne() that takes one argument
// and returns the incremented value
function plusOne(x) {
  return x + 1;
}
/* notice that this is appended to the end of the document */
document.write("plusOne(3) ", plusOne(3), "<br>" );
document.write("plusOne(24) ", plusOne(24), "<br>" );
document.write("plusOne(763) ", plusOne(763), "<br>" );
document.write("plusOne(999) ", plusOne(999), "<br>" );
document.write("<span style='font-weight:bold'>
  Next step: add some marked up content, css, user number input, output e
```

- note the use of **document.write** method to append content to the end of the document

- what is the parameter that we are passing to the method? what can the string(s) contain?
  - is the method *write* available on other element objects?
  - why is the output attend to the end of the document?
  - what happens to the DOM when you use the *write* method?
  - compare the function `plusOne()` to the same function written in Java? What syntax differences can you identify?
  - what would happen if we declared the function after the `document.write()` statements? (explore hoisting)
- 

### STEP 3 : Add

```
29 <script>
30 function plusOne(x) {
31     return x + 1;
32 }
33 uval.onchange = function() {
34     var x = uval.value;
35     document.write("plusOne(" + x + ") ", plusOne(x));
36 }
37 </script>
```

- line 33. Notice we start processing the input only when user changes the value
  - line 34. we get the value from the user
  - Notice the overwritten DOM!
  - Notice the page appearance and calculated value!
- 

### STEP 4 : Stop using `document.write` method; target particular element and add, or change, content within the targeted element

```
uval.onchange = function() {
    var x = uval.value;
    out.innerHTML = "plusOne(" + x + ") " + plusOne(x);
}
```

- compare using `.innerText` and `.innerHTML` and ...

**STEP 5 :** Add the following snippet to the script tag:

```
<input id="userValue" type="number">
<output id="answer"></output>

...
<script>
function plusOne(x) {
    return x + 1;
}
// get value from the user
var x = uval.value;
console.log(x + "typeof " + typeof(x));
var y = uval.valueAsNumber;
console.log(y + "typeof " + typeof(y));
answer.innerHTML = "<p>plusOne(", y, ") ", plusOne(y), "</p>";
```

