

# Some Programmer's “Tricks”



Dr. Stephen Blythe  
CSC I 0022, Fall 2019

# External Scripts

- Most JavaScript Programmers place code in other file(s)
  - script tag(s) are used to specify the other files
  - use the src option (attribute)

- For example:

```
<script src=otherfile.js type=text/javascript>  
</script>
```

- Note that you have to actually create otherfile.js
  - of course, you can name the file anything you like!
- You can have as many script tags as you like ...
  - ... but, for now, keep using only one.



# Incrementing

- Consider the following code sequence:

```
var n=17;  
n=n+1;  
document.write("n is now:"+n);
```

- Is this valid JavaScript?
- why / why not?

# Incrementing ...

- Not only is `n=n+1;` valid, there are shorthands for it!
  - the `+=` statement/operator:
    - same as `=` (assignment), except the RHS is added to the LHS
      1. evaluate the right hand side (ignoring left hand side)
      2. add result of step 1 to the variable on the left hand side
  - examples:
    - `n += 1;`                    `// adds 1 to n`
    - `n += 7;`                    `// adds 7 to n`
    - `n += 3*6 + ab;`    `// adds 3*6+ab to n`



# Decrementing ...

- the `--` statement/operator:
  - same as `=` (assignment), except the RHS is subtracted from the LHS
    1. evaluate the right hand side (ignoring left hand side)
    2. subtract result of step 1 from the variable on the left hand side
- examples:
  - `n -= 1;`                    `// subtracts 1 from n`
  - `n -= 7;`                    `// subtracts 7 from n`
  - `n -= 3*6 + ab;`   `// subtracts 3*6+ab from n`

# Similar Ideas

- the `*=` statement/operator:
  - the LHS variable is multiplied by the RHS
- the `/=` statement/operator:
  - the LHS variable is divided by the RHS
- the `%=` statement/operator:
  - the LHS variable is set to the remainder of LHS/RHS



# Back to Incrementing

other ways to write `n=n+1;`

- `n++;`
- `++n;` //we'll discuss the difference later

also, `n=n-1;` can be written

- `n--;`
- `--n;`

# Notes on Types in JavaScript

- What is the difference between “3.14159” and 3.14159?
  - the first is a string (a sequence of characters)
  - the second is a number (in this case close to pi)
  - radically different things!!
- Incrementing variables holding numbers makes sense.
  - Incrementing a variable holding a string does not!
- Need to make a number out of a string?

```
var myStr = "3.14159";      // trouble - arithmetic may not work!  
  
var myNum = Number(myStr); // myNum holds the numeric equivalent!
```