A Little Review:
What is the Value of wicked, dude?

- Write your answers on a piece of scratch paper

```
var wicked, dude;
dude = 5;
wicked = 2;
dude = wicked * 5;          [ * means multiply ]
dude = dude + 3;
wicked = dude + 1;
```

- Questions:

1. What values do *dude* and *wicked* contain at the end of this code?

When A Decision Must Be Made:

Conditionals

Computers can be programmed to make decisions – that is, to choose one path to follow from many alternatives. This concept is known as branching. Conditionals are the programming tool used to implement it.

The Reason to Have Conditionals:

- CONCEPT: computer programs execute all statements in the program in order unless the program is instructed to only execute certain statements under certain conditions
- For example:
  - □ if (some expression is true) {

```
if (some expression is true) {
        [do this part of the program]
}
```

Operators:

- CONCEPT: Operators are used to *combine* expressions (logical operators) or to *compare* expressions (comparison operators)
  - □ They are used in combination with values, or variables that contain values – both called operands when using operators - to complete the expression formulae
- Most programming languages have more operators than a pocket calculator
- A very useful logical operator is concatenate, also the + symbol in JavaScript, which connects two strings or variables holding strings together:
  - □ plural = “dog” + “s”
  - □ classname = “FIT” + 100   //only one value has to be a string

Operators

- CONCEPT: Comparison operators are often used in conditional statements to create expressions that evaluate to either “true” or “false”

- The comparison operators JavaScript are:

a < b	True if a is less than b
a > b	True if a is greater than b
a <= b	True if a is less than or equal to b
a >= b	True if a is greater than or equal to b
a == b	True if a is equal to b
a != b	True if a is not equal to b

Operators

- CONCEPT: Logical operators are often used in conditional statements to combine multiple comparison operators together that evaluate to either “true” or “false”

- The logical operators JavaScript are:

&&	Boolean AND
||	Boolean OR
!	Boolean NOT

- Example of both logical and comparison operators:
  - □ a < b && c < d

```
          // True if a is less than b AND c is less than d
```

Basic Conditional

- Use conditionals to test to see if a condition holds:
  - □ if (temp < 32) {

```
if (temp < 32)  {
            state = “frozen”;
            form = “ice”;
            }
```

- General form of basic conditional:

```
if (<T / F expression>) {
      <code statements>;
}
```

- What this means:
  - □ First, the <T / F expression> is evaluated
  - □ If the outcome is true, then the statements that follow inside the curly brackets are performed
  - □ If the outcome is false, then the statements that follow the curly brackets are skipped

General Conditional Statement

- CONCEPT: When one set of statements must be performed for the true conditions and a different set of statements are needed for the false conditions, use the Else statement

- General form

```
if (<T / F expression>) {
      <code statements>;
        }
  else  {
      <code statements>
        }
```

```
if (sky == “clear” &&  temp >70) {
    clothing = “tank top”;
        }
else {
    clothing = “sweats”;
        }
```

“Nested” if-else

- CONCEPT: An advantage of the general conditional is that it can be imbedded within another conditional

```
if  (sky == “clear” &&  temp  > 70)  {
           clothing =“tank top” ;
                    if  (laundry ==“clean”) {
                              clothingColor=“purple” ;
                 }
          }
else      {

          clothing = “sweats” ;

                    if  (ground == “muddy”) {
                              shoes = “boots” ;
                         }
          }
```

Exercise #1

- What does this print?

```
var x;
x=10;
  if (x==1) {
         document.write(“Wassup!”);
            }
  else {
          document.write(“Mariners”);
         }

  document.write( “The End”);
```

Conditionals

- Used when a decision must be made between one or more possibilities (conditions)
- Basic conditional

```
□ if <T/F Statement> {               // tests for one condition: true
            <code statements>;
                        }
```

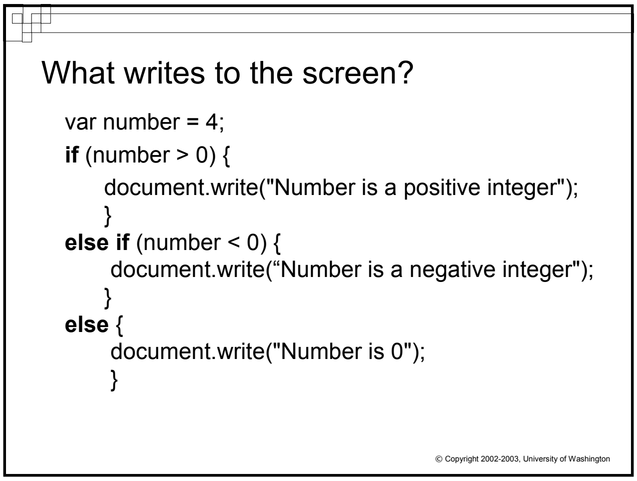
- General conditional

```
□ if (<T/F Statement>) {           // tests for one condition, allows 2
            <code statements>;    // outcomes.  One for True,
       }
  else {                                // the other for False (or otherwise)
            <code statements>;
       }
```

Conditionals

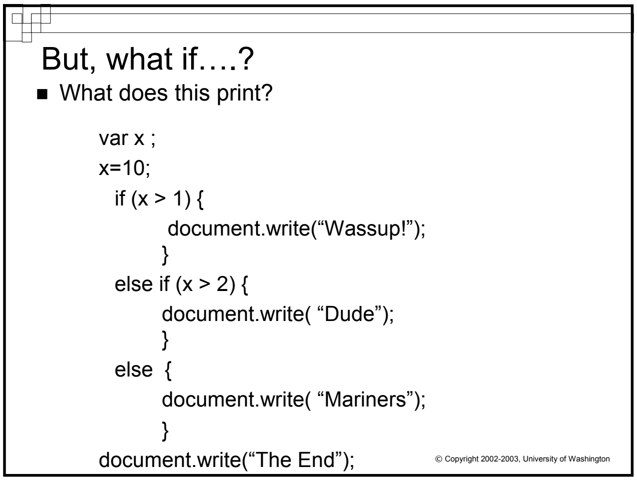
- Multiple conditions to check….

```
□ if (<T/F Statement>) {
            <code statements>;       // tests for multiple conditions
       }

  else if (<T/F Statement>) {
    <code statements>;
       }

  else if (<T/F Statement>) {
    <code statements>;
       }
  ….
  else {                                   // if none of previous are true, do
    <code statements>;                // otherwise
       }
```



What writes to the screen?

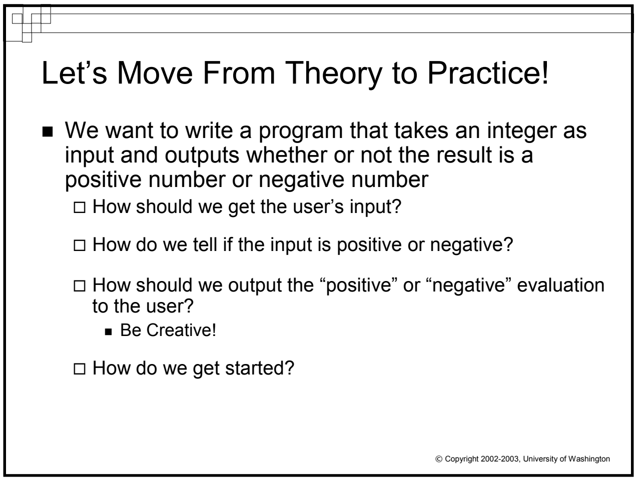
```
var number = 4;
if (number > 0) {
      document.write("Number is a positive integer");
      }
else if (number < 0) {
       document.write("Number is a negative integer");
      }
else {
       document.write("Number is 0");
       }
```



But, what if….?

- What does this print?

```
var x ;
x=10;
  if (x > 1) {
          document.write("Wassup!");
         }
  else if (x > 2) {
         document.write( "Dude");
         }
  else  {
         document.write( "Mariners");
         }
document.write("The End");
```



Let’s Move From Theory to Practice!

- We want to write a program that takes an integer as input and outputs whether or not the result is a positive number or negative number
  - □ How should we get the user’s input?
  - □ How do we tell if the input is positive or negative?
  - □ How should we output the “positive” or “negative” evaluation to the user?
    - Be Creative!
  - □ How do we get started?