# INFO 151
# Web Systems and Services

## Week 6 (Lab)

### Practical JavaScript Exercises

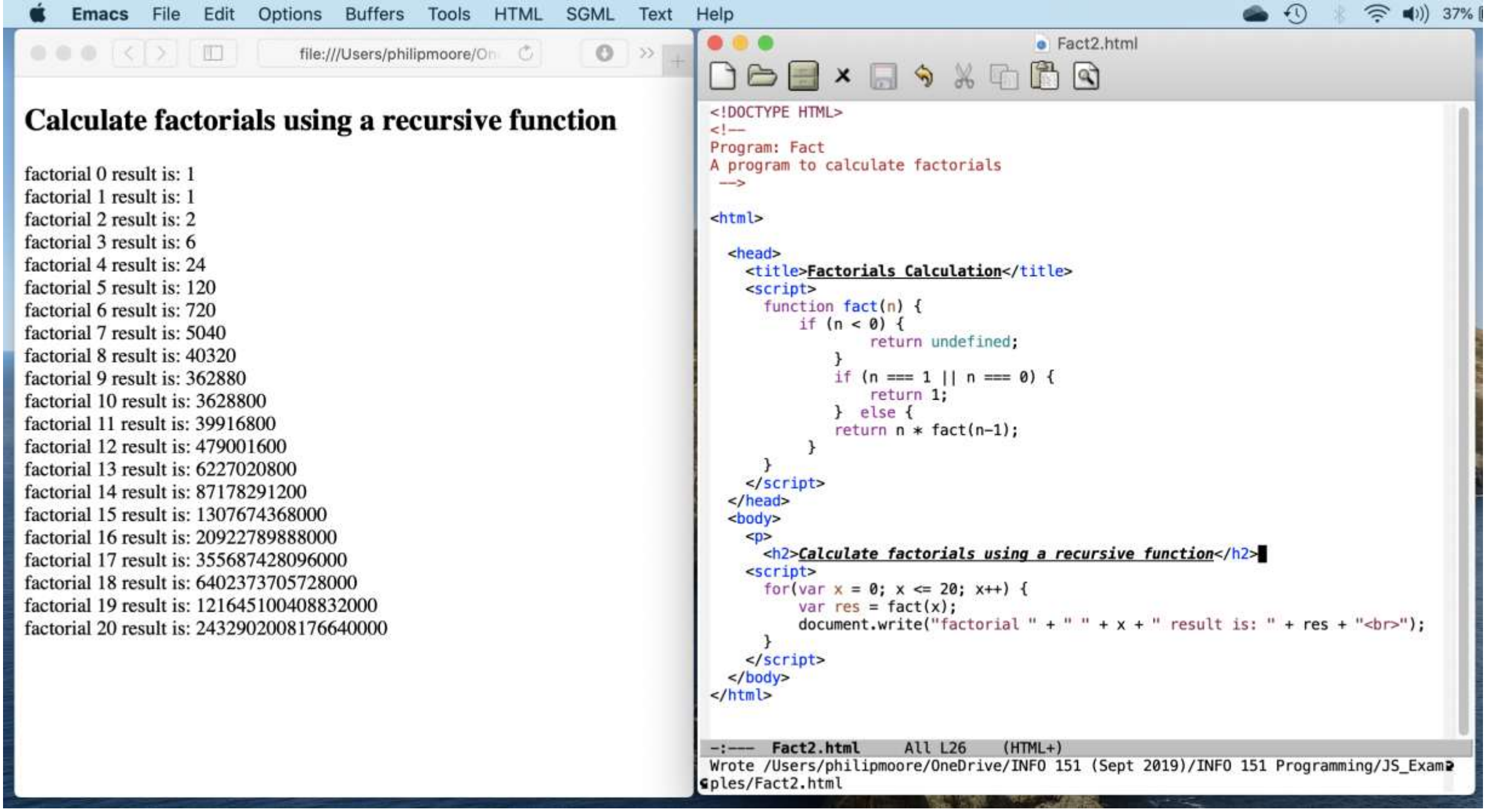
Dr Philip Moore

Dr Zhili Zhao

# Example JavaScript Program

# A Simple Factorials (pure JavaScript) Example

- From the following slide we can see:
  - The JavaScript script embedded in the html file
  - A comment
  - The use of a function (located in the <head>)
  - The calling declaration in the script located in the <body>
  - The use of sequential lines of program code
  - The use of a for loop (iteration)
  - The use of an if statement (conditional)
  - How equality and comparison operators' function
  - The creation of a concatenated string
- The output shows the results in a web browser


Emacs File Edit Options Buffers Tools HTML SGML Text Help
37%
file:///Users/philipmoore/On
Calculate factorials using a recursive function
factorial 0 result is: 1
factorial 1 result is: 1
factorial 2 result is: 2
factorial 3 result is: 6
factorial 4 result is: 24
factorial 5 result is: 120
factorial 6 result is: 720
factorial 7 result is: 5040
factorial 8 result is: 40320
factorial 9 result is: 362880
factorial 10 result is: 3628800
factorial 11 result is: 39916800
factorial 12 result is: 479001600
factorial 13 result is: 6227020800
factorial 14 result is: 87178291200
factorial 15 result is: 1307674368000
factorial 16 result is: 20922789888000
factorial 17 result is: 355687428096000
factorial 18 result is: 6402373705728000
factorial 19 result is: 121645100408832000
factorial 20 result is: 2432902008176640000
Fact2.html
<!DOCTYPE HTML>
<!--
Program: Fact
A program to calculate factorials
-->
<html>
<head>
<title>Factorials Calculation</title>
<script>
function fact(n) {
if (n < 0) {
return undefined;
}
if (n === 1 || n === 0) {
return 1;
} else {
return n * fact(n-1);
}
}
</script>
</head>
<body>
<p>
<h2>Calculate factorials using a recursive function</h2>
<script>
for(var x = 0; x <= 20; x++) {
var res = fact(x);
document.write("factorial " + " " + x + " result is: " + res + "<br>");
}
</script>
</body>
</html>
-:--- Fact2.html All L26 (HTML+)
Wrote /Users/philipmoore/OneDrive/INFO 151 (Sept 2019)/INFO 151 Programming/JS_Examples/Fact2.html

# Exercise #1

# do…while Loop Example

- Write a **do…while** loop JavaScript program to create the matrix style times table with the output in a web browser as shown in this slide
- Use document.write(“ … ”);

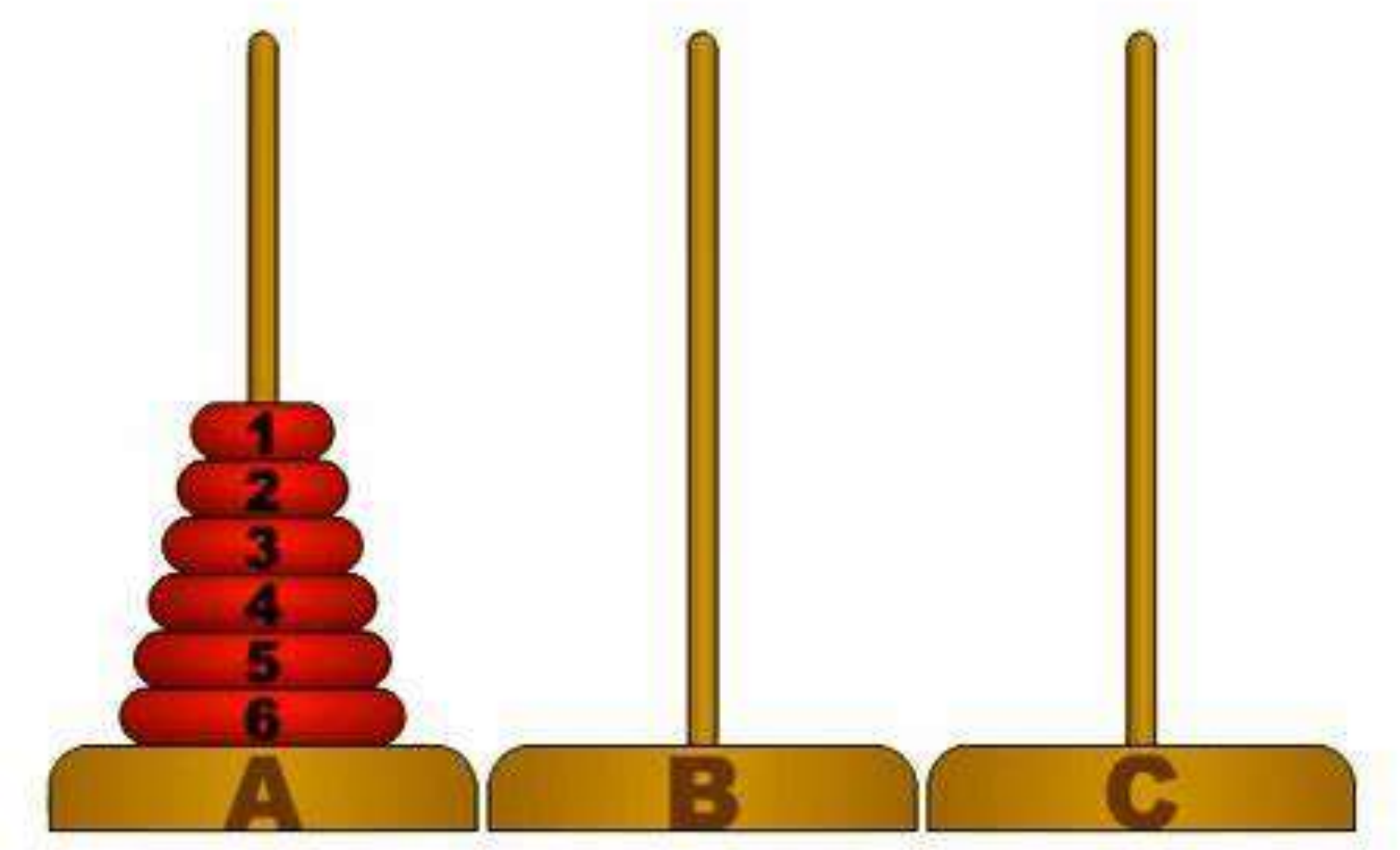
1 2 3 4 5 6 7 8 9 10 11 12
2 4 6 8 10 12 14 16 18 20 22 24
3 6 9 12 15 18 21 24 27 30 33 36
4 8 12 16 20 24 28 32 36 40 44 48
5 10 15 20 25 30 35 40 45 50 55 60
6 12 18 24 30 36 42 48 54 60 66 72
7 14 21 28 35 42 49 56 63 70 77 84
8 16 24 32 40 48 56 64 72 80 88 96
9 18 27 36 45 54 63 72 81 90 99 108
10 20 30 40 50 60 70 80 90 100 110 120
11 22 33 44 55 66 77 88 99 110 121 132
12 24 36 48 60 72 84 96 108 120 132 144

# Exercise #2

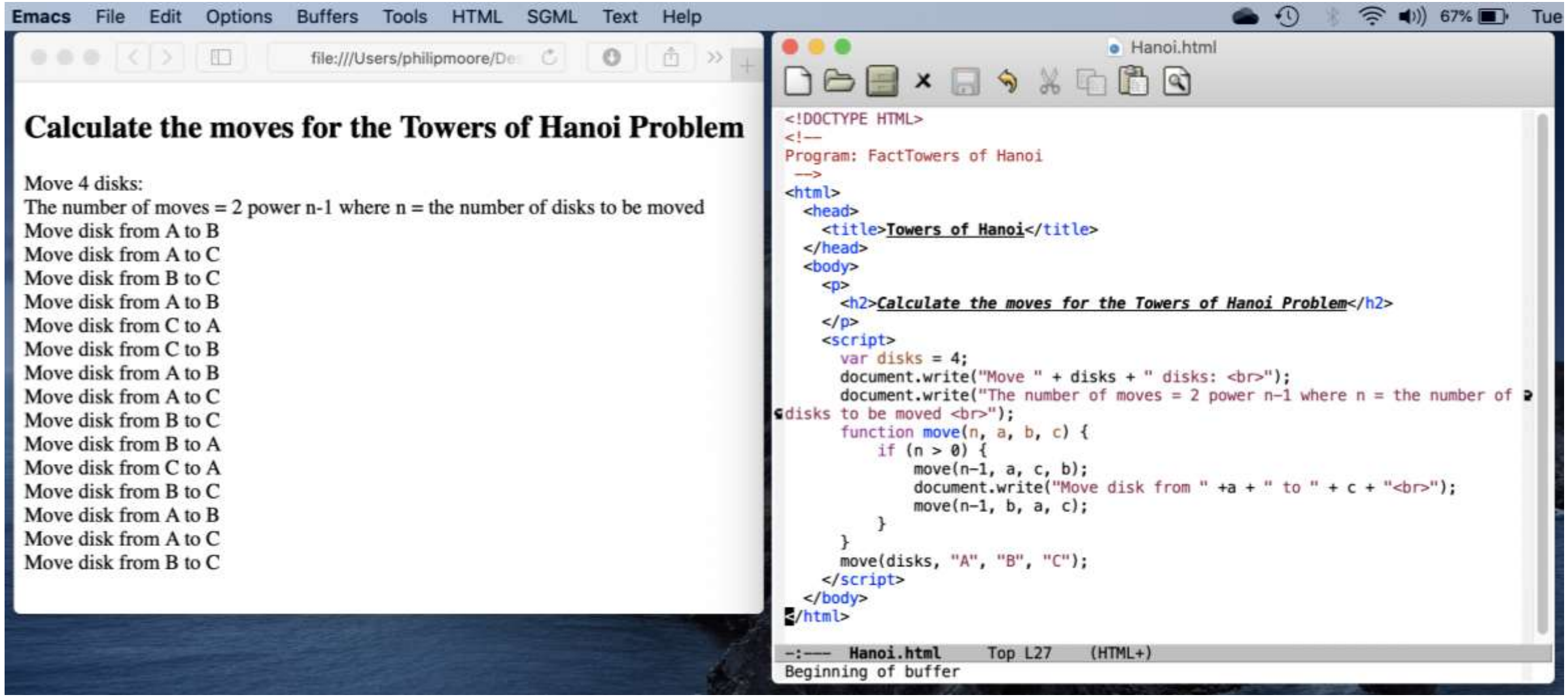
# A Solution to the Towers Of Hanoi Problem

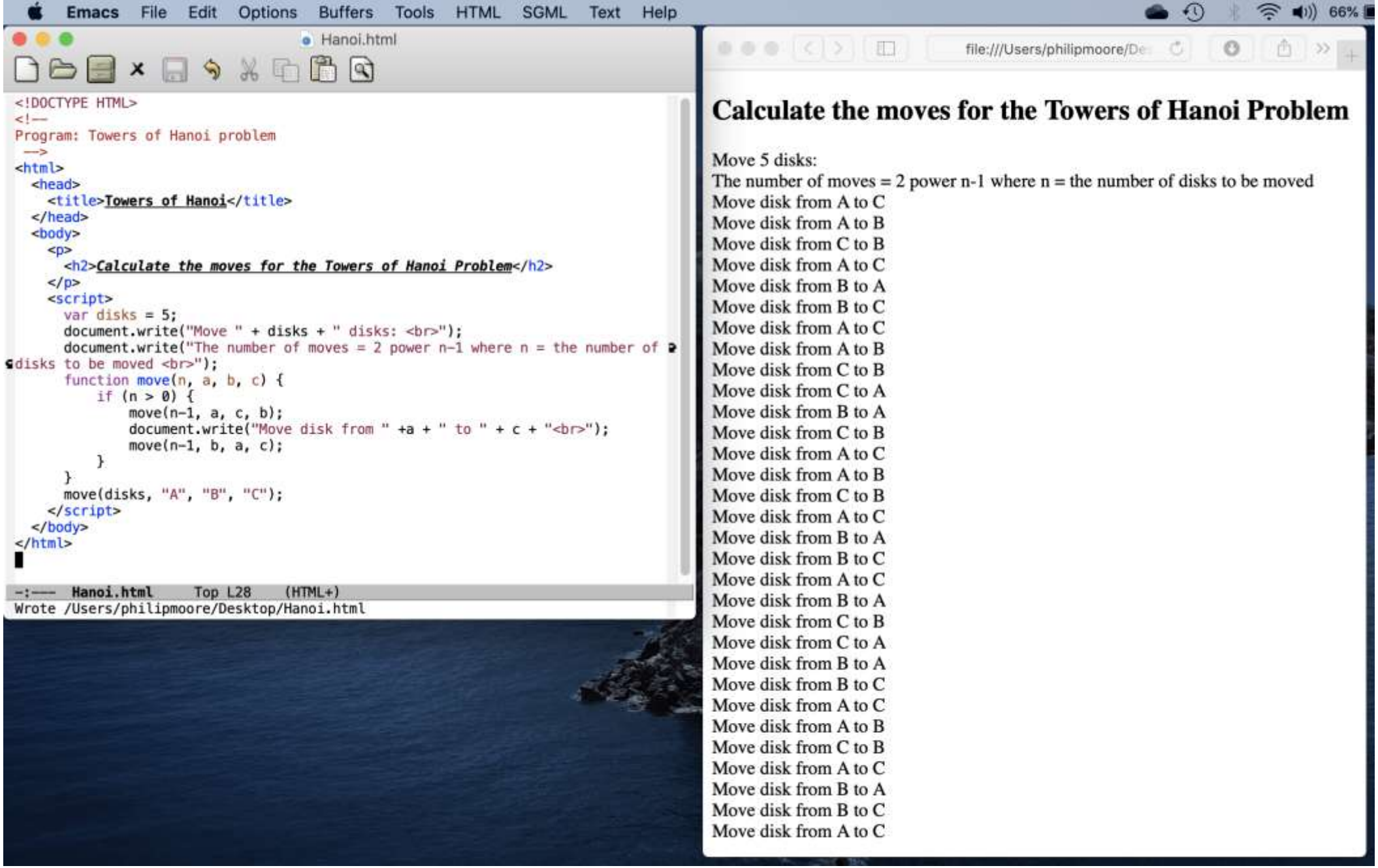
- From the following slide we can see:
  - A solution to the towers of Hanoi problem
  - The program is in pure JavaScript
  - The output is shown in a web browser
- The problem is:
  - To move all the disks from post **A** to post **C** using post **B** without a larger disk being places over a smaller disk

# The Towers Of Hanoi Problem



# A Solution to the Towers Of Hanoi Problem




Emacs File Edit Options Buffers Tools HTML SGML Text Help
66%
Hanoi.html
<!DOCTYPE HTML>
<!--
Program: Towers of Hanoi problem
-->
<html>
<head>
<title>Towers of Hanoi</title>
</head>
<body>

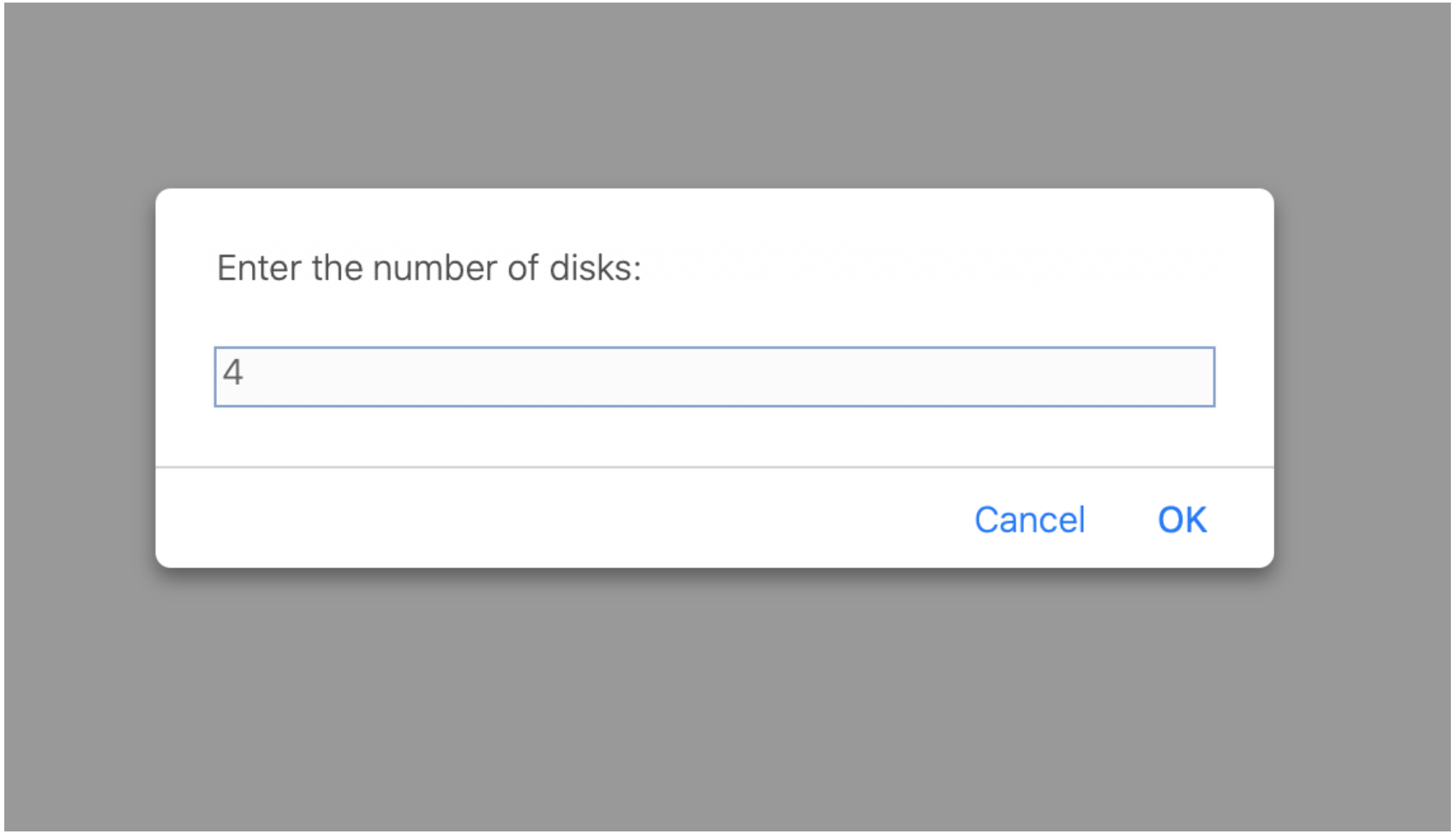
<h2>Calculate the moves for the Towers of Hanoi Problem</h2>

<script>
var disks = 5;
document.write("Move " + disks + " disks: <br>");
document.write("The number of moves = 2 power n-1 where n = the number of disks to be moved <br>");
function move(n, a, b, c) {
if (n > 0) {
move(n-1, a, c, b);
document.write("Move disk from " +a + " to " + c + "<br>");
move(n-1, b, a, c);
}
}
move(disks, "A", "B", "C");
</script>
</body>
</html>
-:--- Hanoi.html Top L28 (HTML+)
Wrote /Users/philipmoore/Desktop/Hanoi.html
file:///Users/philipmoore/De
Calculate the moves for the Towers of Hanoi Problem
Move 5 disks:
The number of moves = 2 power n-1 where n = the number of disks to be moved
Move disk from A to C
Move disk from A to B
Move disk from C to B
Move disk from A to C
Move disk from B to A
Move disk from B to C
Move disk from A to C
Move disk from A to B
Move disk from C to B
Move disk from C to A
Move disk from B to A
Move disk from C to B
Move disk from A to C
Move disk from A to B
Move disk from C to B
Move disk from A to C
Move disk from B to A
Move disk from B to C
Move disk from A to C
Move disk from B to A
Move disk from C to B
Move disk from C to A
Move disk from B to A
Move disk from B to C
Move disk from A to C
Move disk from A to B
Move disk from C to B
Move disk from A to C
Move disk from B to A
Move disk from B to C
Move disk from A to C

# The Task

- Consider the Towers of Hanoi problem:
  - In the JavaScript I have set the number of disks in the script variable
  - You must modify the script to enable users to enter the number of disks in an input field with a button to run the program using the user input value
  - The output to be exactly as shown in the following slides
  - Do not modify the JavaScript functions
  - The user defied input must be assigned to the disks variable and used in the program run

# The Towers Of Hanoi Problem



# The Towers Of Hanoi Problem