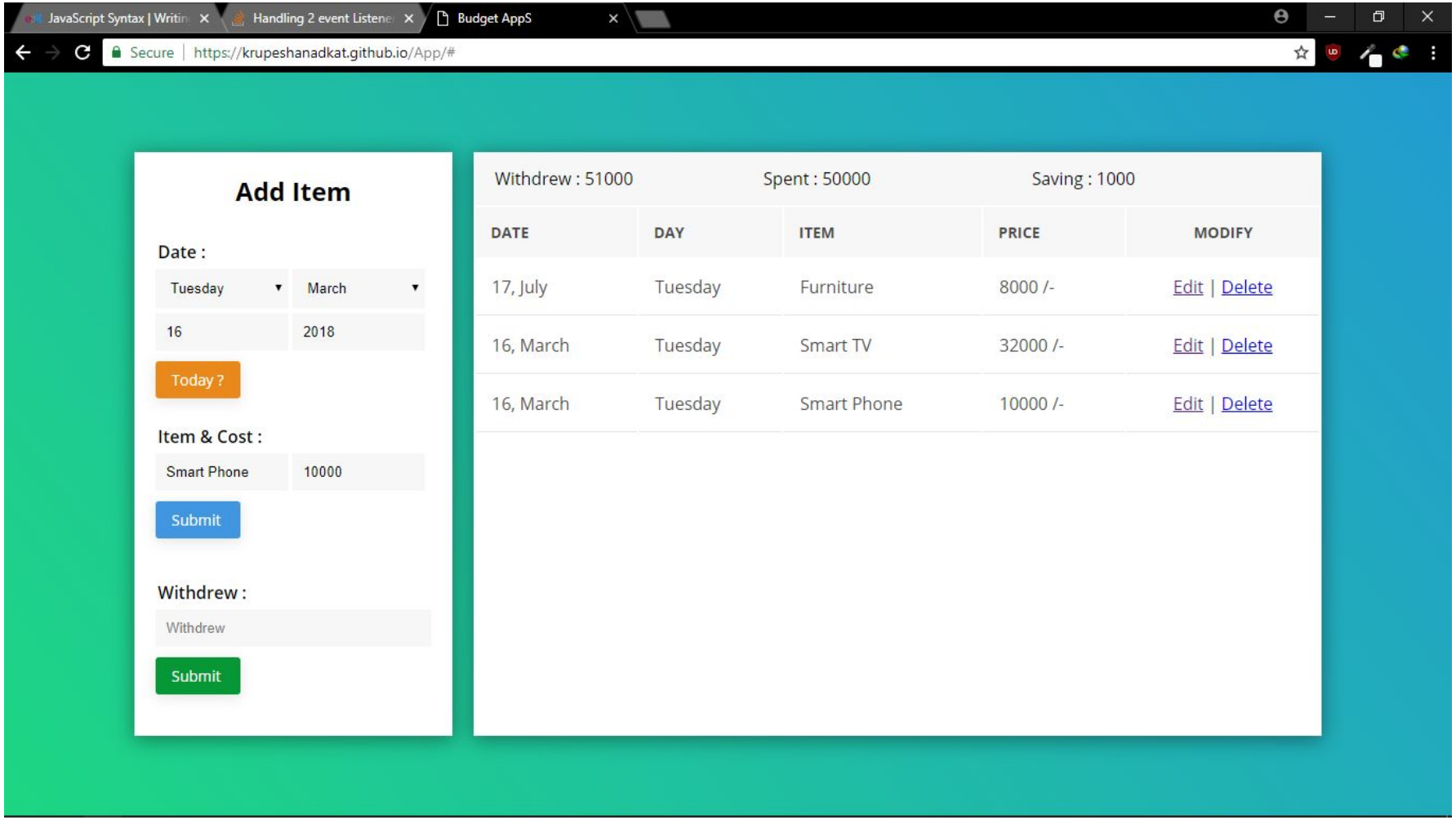
WEB TECHNOLOGIES

JavaScript - Basics

Prof.Vinay Joshi and Dr.Sarasvathi V
Department of Computer Science and Engineering

JavaScript - Basics

Introduction to JavaScript



Introduction to JavaScript

- Client Side Scripting Language

- Originally, LiveScript in NetScape Browser

- JavaScript programs are run by an interpreter built into the user's web browser

- Now the language has evolved with additional Server Side Scripting capabilities (like in Node.JS)

Introduction to JavaScript…(cntd.)

- Client renders (displays) the response received from server (mix of HTML and JavaScript)
- Browser displays HTML
- And runs any JavaScript code within the HTML
- dynamic programming language that can add interactivity to a website.

Introduction to JavaScript…(cntd.)

Pros and Cons of JavaScript

- Pros
  - Allows more dynamic HTML pages, even complete web applications
- Cons
  - Requires a JavaScript-enabled browser
  - Requires a client who trusts the server enough to run the code the server provides
- JavaScript has some protection in place but can still cause security problems for clients

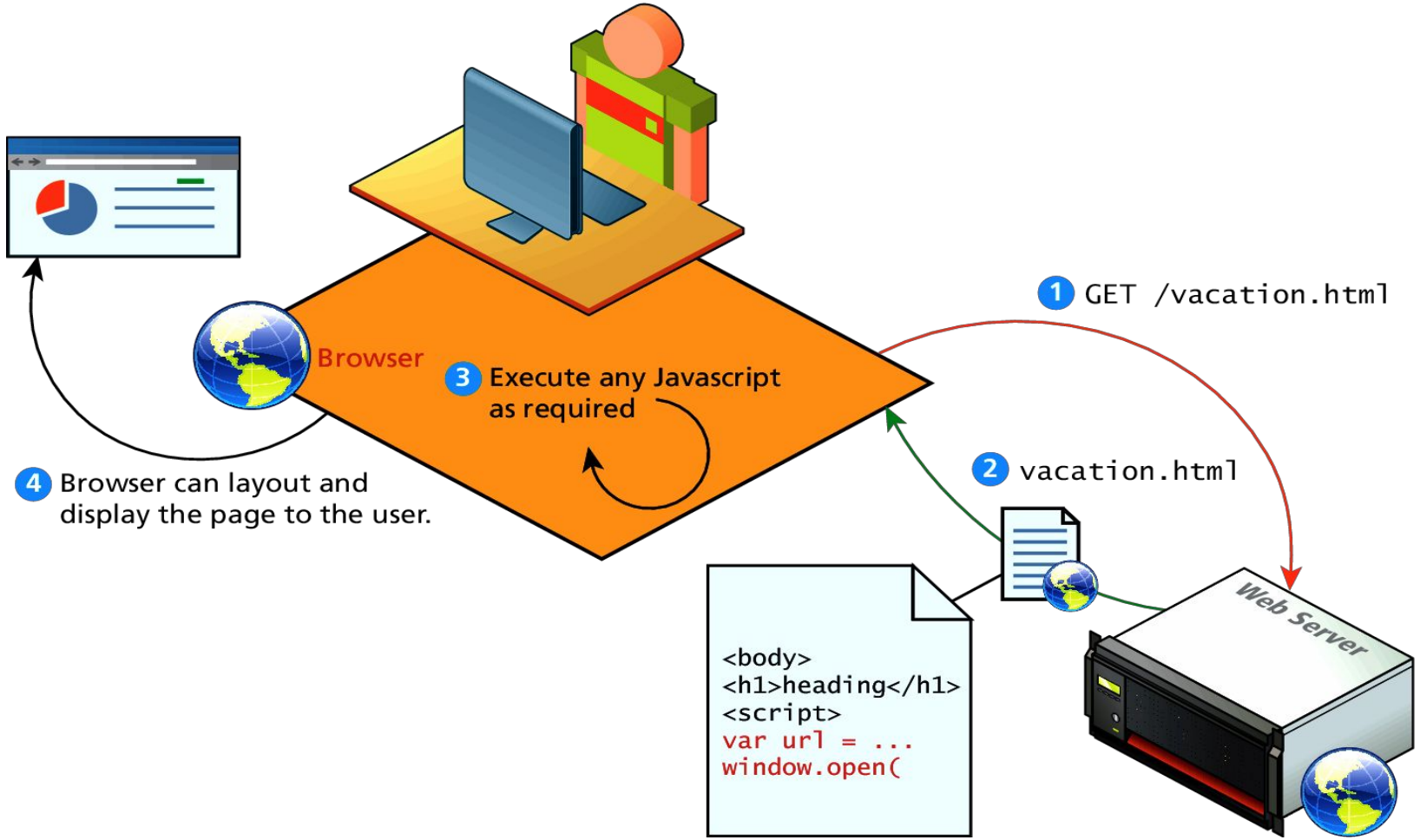
JavaScript - Basics

1. **HTML** to define the content of web pages
2. **CSS** to specify the layout of web pages and give a good look.
3. **JavaScript** to program the behaviour of web pages

- Web pages are not the only place where JavaScript is used. Many desktop and server programs use JavaScript.
- Node.js is the best known.
- Some databases, like MongoDB and CouchDB, also use JavaScript as their programming language.

JavaScript - Basics

Client side scripting



JavaScript Code

JavaScript can be inserted into documents by using the <SCRIPT> tag

```
<html>
    <head>
        <title>Hello World in JavaScript</title>
    </head>
    <body>
        <script type="text/javascript">
            document.write("Hello World!");
        </script>
    </body>
</html>
```

JavaScript Code

Where to Put your Scripts

- You can have any number of script tags at any position
- Scripts can be placed in the HEAD or in the BODY
  - In the HEAD, scripts are run before the page is displayed
  - In the BODY, scripts are run as the page is displayed
  - In the HEAD is the right place to define functions and variables that are used by scripts within the BODY

JavaScript Code – External Scripts

- Scripts can also be loaded from an external file
- This is useful if you have a complicated script or set of subroutines that are used in several different documents

```
<script src="myscript.js"></script>
```

- The myscript.js should not include the script tags

JavaScript Code - Debugging

PES UNIVERSITY
CELEBRATING 50 YEARS

Debugging JavaScript Errors

- When you're learning or using JavaScript, it's important to be able to track typing or other coding errors.

Browser	How to access JavaScript error messages
Apple Safari	Select Safari → Preferences → Advanced → "Show Developer menu in menu bar." You may prefer to use the Firebug Lite JavaScript module, which many people find easier to use.
Google Chrome	Press Ctrl-Shift-J on a PC, or Command-Shift-J on a Mac.
Mozilla Firefox	Press Ctrl-Shift-J on a PC, or Command-Shift-J on a Mac.
Microsoft Internet Explorer & Edge	Press F12 to call up the DevTools Console.
Opera	Select Tools → Advanced → Error Console.

JavaScript Code - Comments

Comments in JavaScript

- Single line comment : //
- Multiline comments : /* */

JavaScript Code – Using Semicolon

- JavaScript generally automatically inserts semicolons at the end of line

```
x += 10 => x += 10;
```

- However, when you wish to place more than one statement on a line, you must separate them with semicolons, like this:

```
x += 10; y -= 5; z = 0
```

- When a statement spans across multiple lines, JavaScript will not raise error if the next line has a valid symbol/literal/token

```
return a
+ b
```

JavaScript Code – Variable Declaration

- The first character of a variable name can be only a-z, A-Z, $, or _ .

-- Then followed by only the letters a-z, A-Z, 0-9, the $ symbol, and the underscore _.

- Variable names are case-sensitive. Count, count and COUNT are three different variables
- Variable can be declared using
  - **let (block scope)**
  - **var (function or global scope)**
  - **Const(block scope)**
  - **use without declaring (global scope)**

KEYWORD	SCOPE	CAN BE REASSIGNED	CAN BE REDECLARED
var	Function	Yes	Yes
let	block	Yes	No
const	block	No	No

JavaScript - Basics

JavaScript Code – Datatypes

```
function example() {
  if (true) {
    var varVariable = "I'm a var"; // Function-scoped
    let letVariable = "I'm a let"; // Block-scoped
  }

  console.log(varVariable); // Outputs "I'm a var"
  console.log(letVariable); // Throws ReferenceError
}

example();

// Re-declaration example
var redeclareVar = "Original var";
var redeclareVar = "Re-declared var"; // No error

let redeclareLet = "Original let";
// let redeclareLet = "Attempt to redeclare let"; // SyntaxError
```

JavaScript Code – Datatypes

- JS is loosely typed or dynamic typed
- **Primitive Datatypes**
  - number
  - string
  - boolean
  - null
  - undefined
- **Non-Primitive Datatypes (used with new keyword)**
  - Object - Boolean
  - Number
  - String
  - Date
  - Array

JavaScript – Objects

- In JavaScript, almost "everything" is an object.

- Booleans can be objects (if defined with the new keyword)
- Numbers can be objects (if defined with the new keyword)
- Strings can be objects (if defined with the new keyword)
- Dates are always objects
- Math are always objects
- Regular expressions are always objects
- Arrays are always objects
- Functions are always objects
- Objects are always objects
- All JavaScript values, except primitives, are objects.

JavaScript Code – Operators and Constructs

JavaScript has most of the operators we're used to from C/Java

- Arithmetic (+, - , *, /, %) .
- Assignment (=, +=, -=, *=/=, %=, ++, --)
- Logical (&&, ||, !)
- Comparison (<, >, <=, >=, ==, ===,!=,!==)

Note: + also does concatenation if one of the operands is string

- Constructs: if, else, while, for, switch, case

JavaScript Code – Data types

```
let length = 16;                    // Number
let lastName = "Johnson";           // String
let x = 16 + "Volvo";
```

Output: 16Volve

When adding a number and a string, JavaScript will treat the number as a string.

```
let x = "Volvo" + 16;
```

Output: Volve16

```
let x = 16 + 4 + "Volvo";
```

Output: 20Volve

```
let x = "Volvo" + 16 + 4;
```

Output: Volve164

- In the first example, JavaScript treats 16 and 4 as numbers, until it reaches "Volvo".
- In the second example, since the first operand is a string, all operands are treated as strings.

JavaScript - Basics

JS concatenation and Addition (+)

- The + operator can also be used to add (concatenate) strings.
- var txt1 = ”A";
  var txt2 = ”SECTION";
  var txt3 = txt1 + " " + txt2; // A SECTION

- var txt1 = "What a very ";
  txt1 += "nice day"; // What a very nice day

- var x = 5 + 5; // 10
  var y = "5" + 5; //55
  var z = "Hello" + 5; // Hello5

Loops

- JavaScript supports different kinds of loops:

1. **for** - loops through a block of code a number of times
2. **for/in** - loops through the properties of an object
3. **while** - loops through a block of code while a specified condition is true
4. **do/while** - also loops through a block of code while a specified condition is true

For/in loop:

For/in loop:

The JavaScript for/in statement loops through the properties of an object.

```
var person = {fname:"John", lname:"Doe", age:25};
var text = "";
var x;
for (x in person) {
    text += person[x];
}
```

JavaScript - Basics

Equality

```
1  console.log(1 == 1);
2  // expected output: true
3
4  console.log('hello' == 'hello');
5  // expected output: true
6
7  console.log('1' ==  1);
8  // expected output: true
9
10 console.log(0 == false);
11 // expected output: true
12
```

Strict Equality

```
1  console.log(1 === 1);
2  // expected output: true
3
4  console.log('hello' === 'hello');
5  // expected output: true
6
7  console.log('1' ===  1);
8  // expected output: false
9
10 console.log(0 === false);
11 // expected output: false
```

THANK YOU

Prof.Vinay Joshi and Dr.Sarasvathi V
Department of Computer Science and Engineering