HOA SEN UNIVERSITY

www.hoasen.edu.vn

JAVASCRIPT

Phạm Thị Kim Ngôn

ngon.phamthikim@hoasen.edu.vn

Content

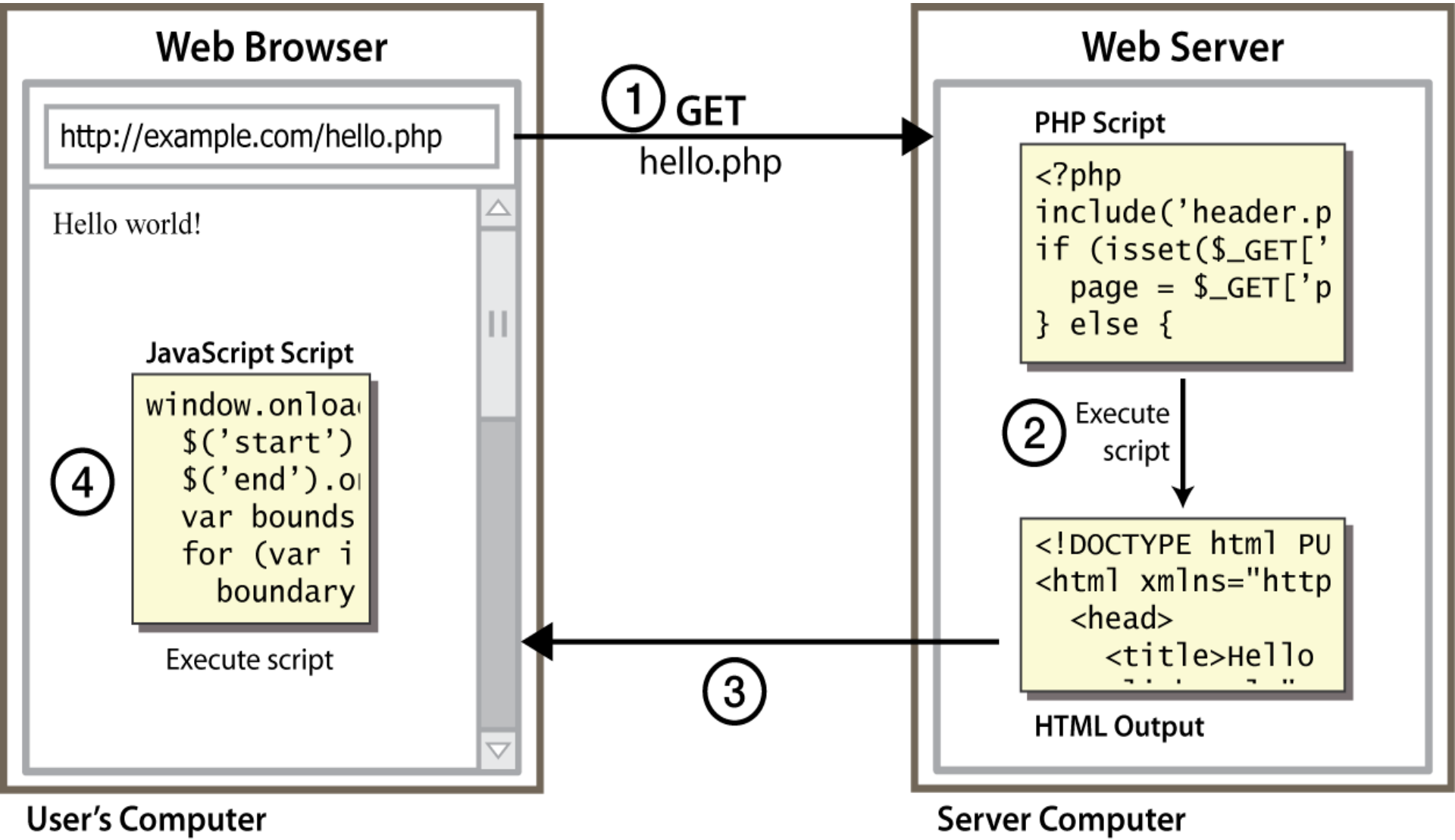
- Introduce Javascript
- JavaScript and HTML page
- Comments and Statements
- Operators
- Conditional Statements
- Looping
- Events
- OOP
- Document Object Model (DOM)

JavaScript
A Client-side Scripting Language

HOA SEN UNIVERSITY

www.hoasen.edu.vn

Client Side Scripting



Why use client-side programming?

PHP already allows us to create dynamic web pages. Why also use client-side scripting?

- client-side scripting (JavaScript) benefits:
  - **usability**: can modify a page without having to post back to the server (faster UI)
  - **efficiency**: can make small, quick changes to page without waiting for server
  - **event-driven**: can respond to user actions like clicks and key presses

Why use client-side programming?

- server-side programming (PHP) benefits:
  - **security**: has access to server's private data; client can't see source code
  - **compatibility**: not subject to browser compatibility issues
  - **power**: can write files, open connections to servers, connect to databases, ...

JavaScript

- JavaScript is designed
  - to add interactivity to HTML pages
- JavaScript
  - consists of lines of interpretable computer code
  - gives HTML designers a programming tool
  - is usually embedded directly into HTML pages.
  - allows to put dynamic text into an HTML page
- Java and JavaScript are two completely different languages in both concept and design
- JavaScript's official name is ECMAScript.

JavaScript

- JavaScript is used in millions of web pages
  - to improve the design
  - to validate forms
  - to detect browsers
  - to create cookies
- JavaScript can react to events and can be used to validate data and to create cookies
- Is the most popular scripting language in all major browsers e.g.
  - Internet Explorer
  - Mozilla
  - Firefox
  - Netscape
  - Opera

JavaScript

- JavaScript is used in millions of web pages
  - to improve the design
  - to validate forms
  - to detect browsers
  - to create cookies
- JavaScript can react to events and can be used to validate data and to create cookies
- Is the most popular scripting language in all major browsers e.g.
  - Internet Explorer
  - Mozilla
  - Firefox
  - Netscape
  - Opera

JavaScript and HTML page

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

Tells where the JavaScript starts

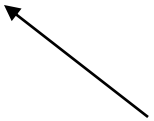
Commands for writing output to a page

Tells where the JavaScript ends

This code produce the output on an HTML page:
Hello World!

JavaScript and HTML page

```
<html>
    <head>
    <script src="xyz.js"> </script>
    </head>
<body>
</body>
</html>
```



A separate file

Statements and Comments

- JavaScript statements
  - are codes to be executed by the browser
  - tells the browser what to do
  - commands to the browser
  - add semicolons at the end
  - can be grouped together into blocks using curly brackets
  - try…catch statement allows to test a block of code for errors
- JavaScript comments make the code more readable
  - Single line comments start with //
  - Multi line comments start with /* and end with */

JavaScript Variables

- JavaScript Variables
  - are containers for storing information e.g. x=15; length=60.10;
  - hold values or expressions
  - can hold a text value like in name="multimedia"
  - **var statement** can declare JavaScript variables: **var x**; **var name**;
- Variable names
  - are case sensitive i.e. "myVar" is not the same as "myvar"
  - must begin with a letter or the underscore character

JavaScript Operators

- Arithmetic Operators:
  - perform arithmetic operations between the values of the variables
  - Addition (+) , Subtraction (-),
  - Multiplication (*), Division (/), Modulus (%),
  - Increment (+ +), Decrement (- -)
- Assignment Operators:
  - assign values to variables
  - =, + =, - =, * =, / =, % =
- Comparison Operators:
  - determines equality or difference between variables or values
  - Equal to (= =), Exactly equal to (= = =),
  - Not equal (!=), Greater than (>), Less than (<),
  - Greater than or equal to (>=), Less than or equal to (<=)
- Logical Operators:
  - impose the logic between variables or values
  - AND (&&), OR (| |), NOT (!)
- Conditional Operator:
  - assign value to a variable based on some conditions
  - ?:

JavaScript Conditional Statements

- **if statement -** to execute some code only if a specified condition is true
- **if...else statement -** to execute some code if the condition is true and another code if the condition is false
- **if...else if....else statement -** to select one of many blocks of code to be executed
- **switch statement -** to select one of many blocks of code to be executed

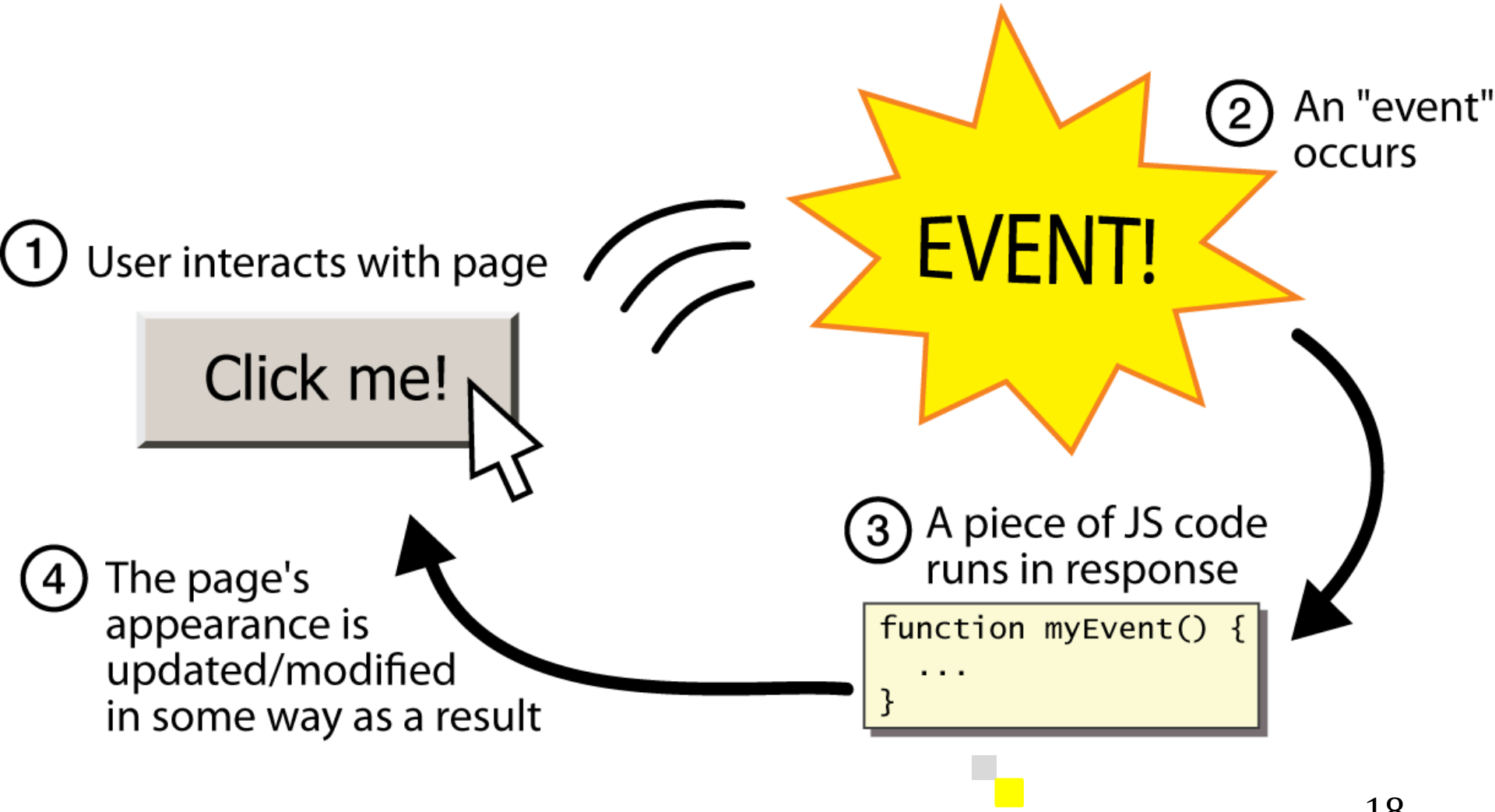
JavaScript Looping

- JavaScript looping
  - Executes the same block of codes
  - Executes a specified number of times
  - Execution can be controlled by some control logic
  - uses **for**, **while**, **do….while** statements
  - uses **for…in** to iterate through the elements of an array
- **Break** breaks the loop and follows the code after the loop
- **Continue** breaks the loop and continues with next value.

JavaScript Functions and Events

- JavaScript Functions
  - Can be called with the function name
  - Can also be executed by an event
  - Can have parameters and return statement
- Events
  - are actions that can be detected e.g. OnMouseOver, onMouseOut etc.
  - are normally associated with functions
  - <input type="text" size="30" id="email" onChange="checkEmail()">

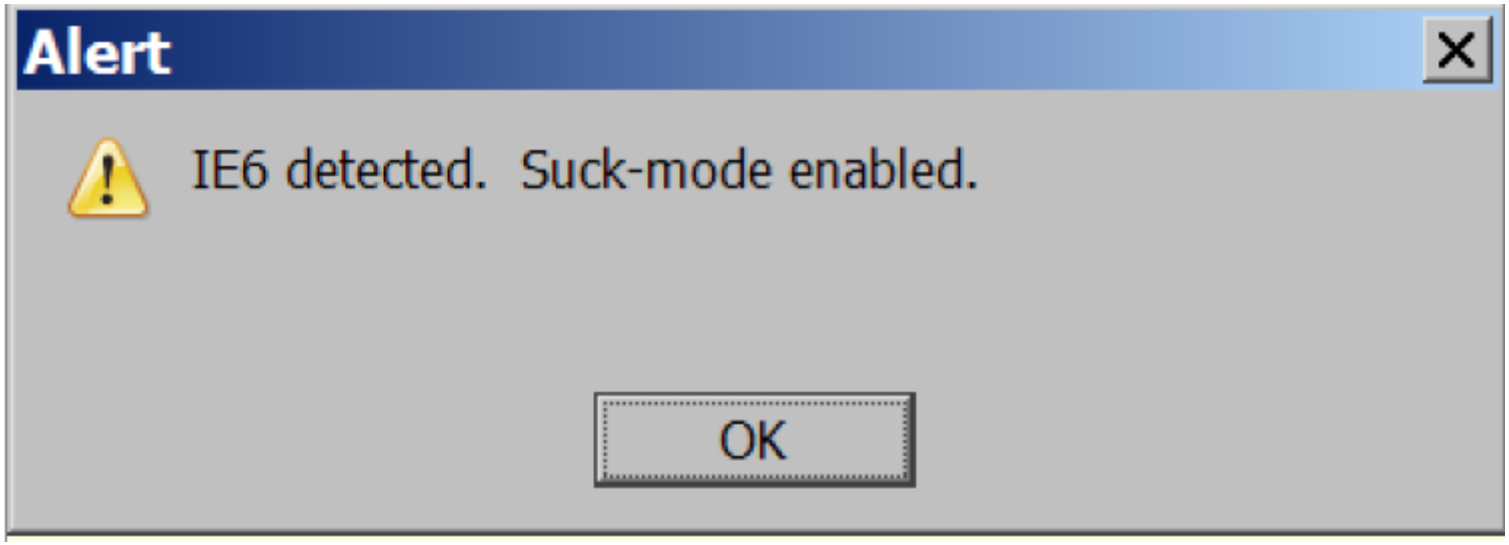
Event-driven programming



HOA SEN UNIVERSITY

A JavaScript statement: `alert`

```
alert("IE6 detected. Suck-mode enabled.");
```
JS



- a JS command that pops up a dialog box with a message

Event-driven programming

- you are used to programs start with a main method (or implicit main like in PHP)
- JavaScript programs instead wait for user actions called events and respond to them
- event-driven programming: writing programs driven by user events

JavaScript Functions and Events

- JavaScript Functions
  - Can be called with the function name
  - Can also be executed by an event
  - Can have parameters and return statement
- Events
  - are actions that can be detected e.g. OnMouseOver, onMouseOut etc.
  - are normally associated with functions
  - <input type="text" size="30" id="email" onChange="checkEmail()">

Example - JavaScript functions

```JS
function name() {
statement ;
statement ;
...
statement ;
}
```

```JS
function myFunction() {
        alert("Hello!");
        alert("How are you?");
}
```

- the above could be the contents of example.js linked to our HTML page
- statements placed into functions can be evaluated in response to user events

Example - Event handlers

```HTML
<element attributes onclick="function();">...
```

```HTML
<button onclick="myFunction();">Click me!</button>
```

- JavaScript functions can be set as event handlers
  - when you interact with the element, the function will execute
- onclick is just one of many event HTML attributes we'll use
- but popping up an alert window is disruptive and annoying
  - A better user experience would be to have the message appear on the page...

JavaScript: Events

- Javascript actions may be triggered from events, e.g. changes on form fields or a submit button being clicked:
  - onfocus = Form field gets focus (validation)
  - onblur= Form field looses focus (validation)
  - onchange= Content of a field changes (validation)
  - onselect= Text is selected
  - onmouseover= Mouse moves over a link (animated buttons)
  - onmouseout= Mouse moves out of a link (animated …)
  - onclick= Mouse clicks an object
  - onload= Page is finished loading (initial actions, info,)
  - onSubmit= Submit button is clicked (validation etc.)

JavaScript Popup boxes

- JavaScript can create:
  - Alert box: to make sure information comes through to the user.
  - Confirm box: to verify or accept something
  - Prompt box: the user to input a value before entering a page

JavaScript and OOP

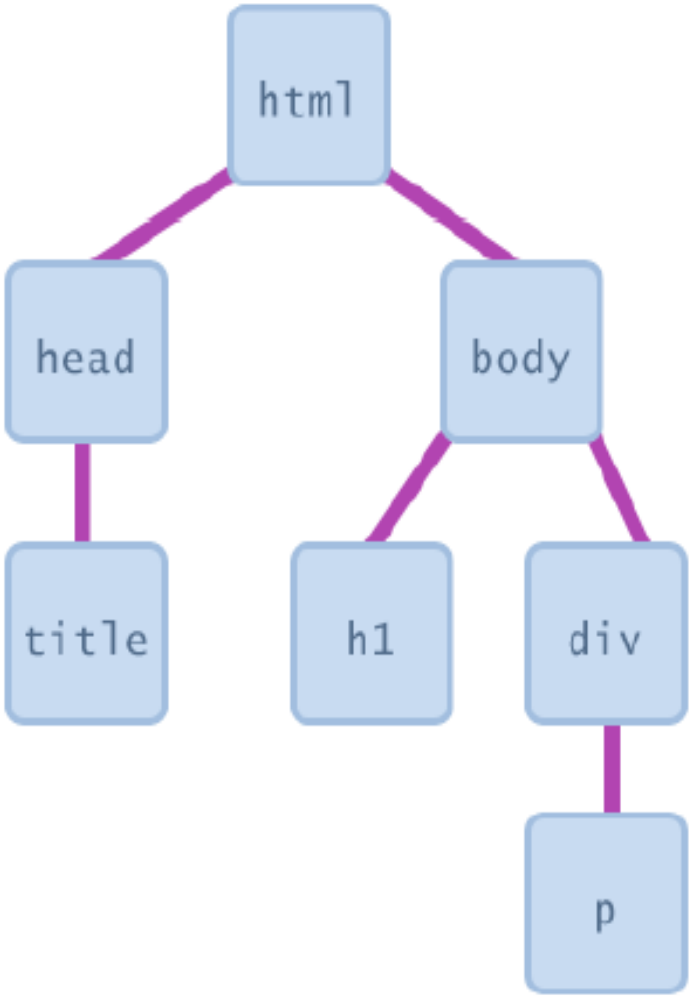
- JavaScript
  - is an Object Oriented Programming language
  - contains built-in JavaScript objects
    - String
    - Date
    - Array
    - Boolean
    - Math
    - RegExp
    - Window
    - Navigator
    - Screen
    - Location
    - History etc.
  - also allows to define new objects
  - objects contain Properties and Methods
  - objects can be used as variable types

JavaScript: DOM

- To access the data in the HTML page
  - needs some data structures to access the HTML page.
- Many browser implement an interface to what is called the Document Object Model (DOM)
  - It allows to output the document in the changed form to the browser.
- DOM is a representation of the document in an object form, accessible from JavaScript programs

Document Object Model (DOM)

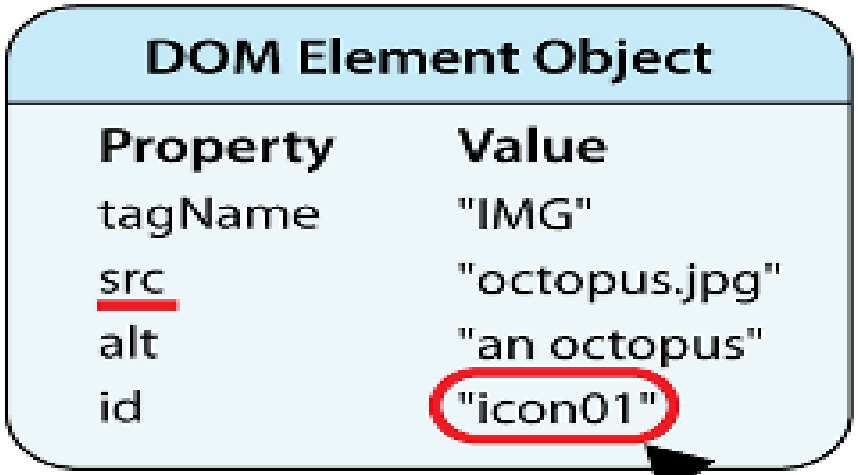
- most JS code manipulates elements on an HTML page
- we can examine elements' stat
  - e.g. see whether a box is checked
- we can change state
  - e.g. insert some new text into a div
- we can change styles
  - e.g. make a paragraph red



DOM element objects

HTML

```
<p>
  Look at this octopus:
  <img src="octopus.jpg" alt="an octopus" id="icon01" />
  Cute, huh?
</p>
```



JavaScript

```
var icon = document.getElementById("icon01");
icon.src = "kitty.gif";
```

HOA SEN UNIVERSITY

Accessing elements: document.getElementById

```JS
var name = document.getElementById("id");
```

```HTML
<button onclick="changeText();">Click me!</button>
<span id="output">replace me</span>
<input id="textbox" type="text" />
```

```JS
function changeText() {
        var span = document.getElementById("output");
        var textBox = document.getElementById("textbox");

         textbox.style.color = "red";

}
```

Accessing elements: document.getElementById

- document.getElementById returns the DOM object for an element with a given id
- can change the text inside most elements by setting the innerHTML property
- can change the text in form controls by setting the value property

HOA SEN UNIVERSITY

Changing element style: `element.style`

Attribute	Property or style object
color	color
padding	padding
background-color	backgroundColor
border-top-width	borderTopWidth
Font size	fontSize
Font famiy	fontFamily