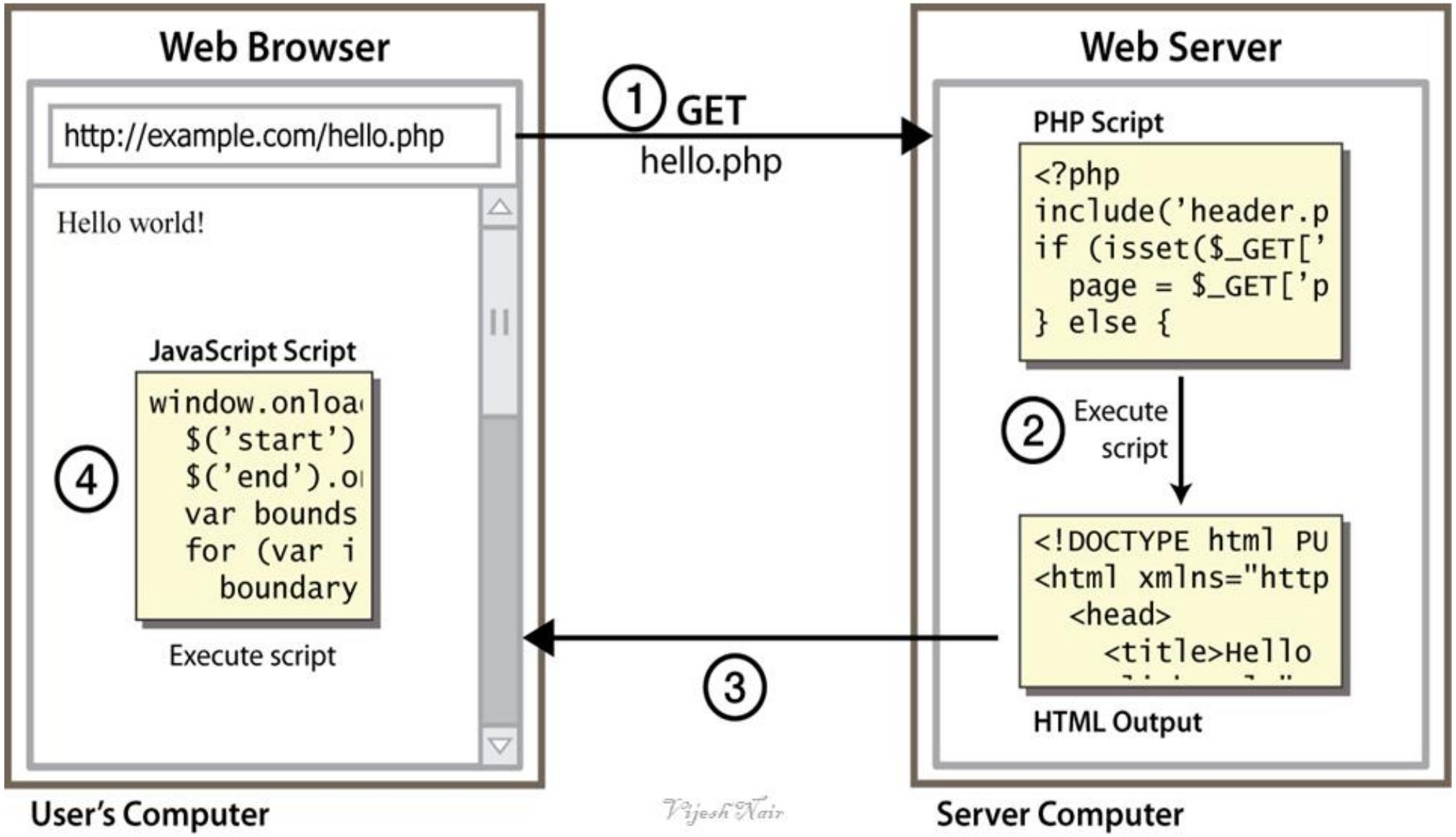
1

Javascript

Vijesh M. Nair
Assistant Professor
Dept. of CSE (AI-ML)

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, ...

What is Javascript?

- a lightweight programming language ("scripting language")
  - used to make web pages interactive
  - insert dynamic text into HTML (ex: user name)
  - **react to events** (ex: page load user click)
  - get information about a user's computer (ex: browser type)
  - perform calculations on user's computer (ex: form validation)

What is Javascript?

- a web standard (but not supported identically by all browsers)
- NOT related to Java other than by name and some syntactic similarities

Javascript vs Java

- interpreted, not compiled
- more relaxed syntax and rules
  - fewer and "looser" data types
  - variables don't need to be declared
  - errors often silent
- key construct is the function rather than the class
  - "first-class" functions are used in many situations
- contained within a web page and integrates with its HTML/CSS content

Javascript vs Java

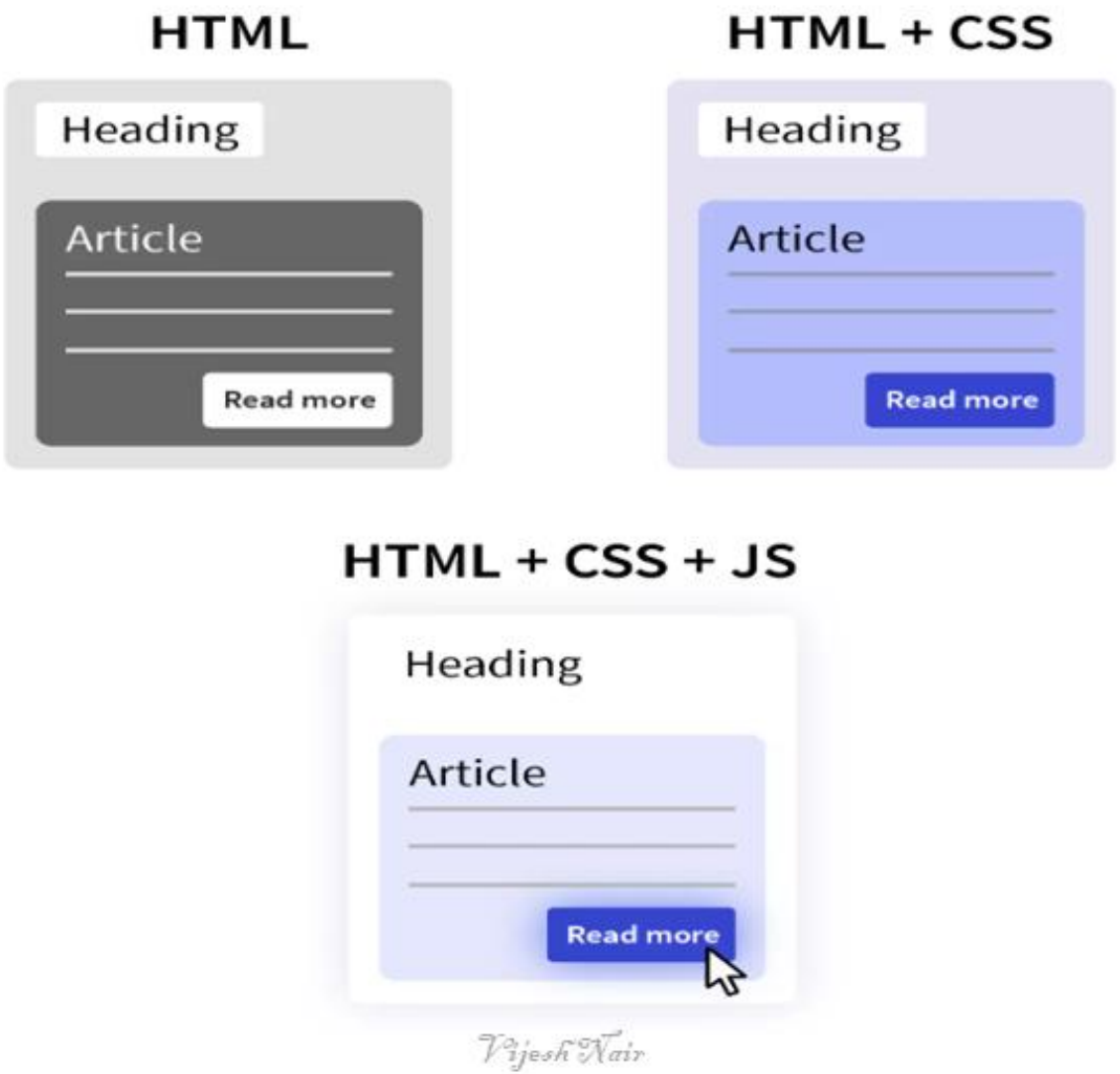
JavaScript vs. PHP

- similarities:
  - both are interpreted, not compiled
  - both are relaxed about syntax, rules, and types
  - both are case-sensitive
  - both have built-in regular expressions for powerful text processing

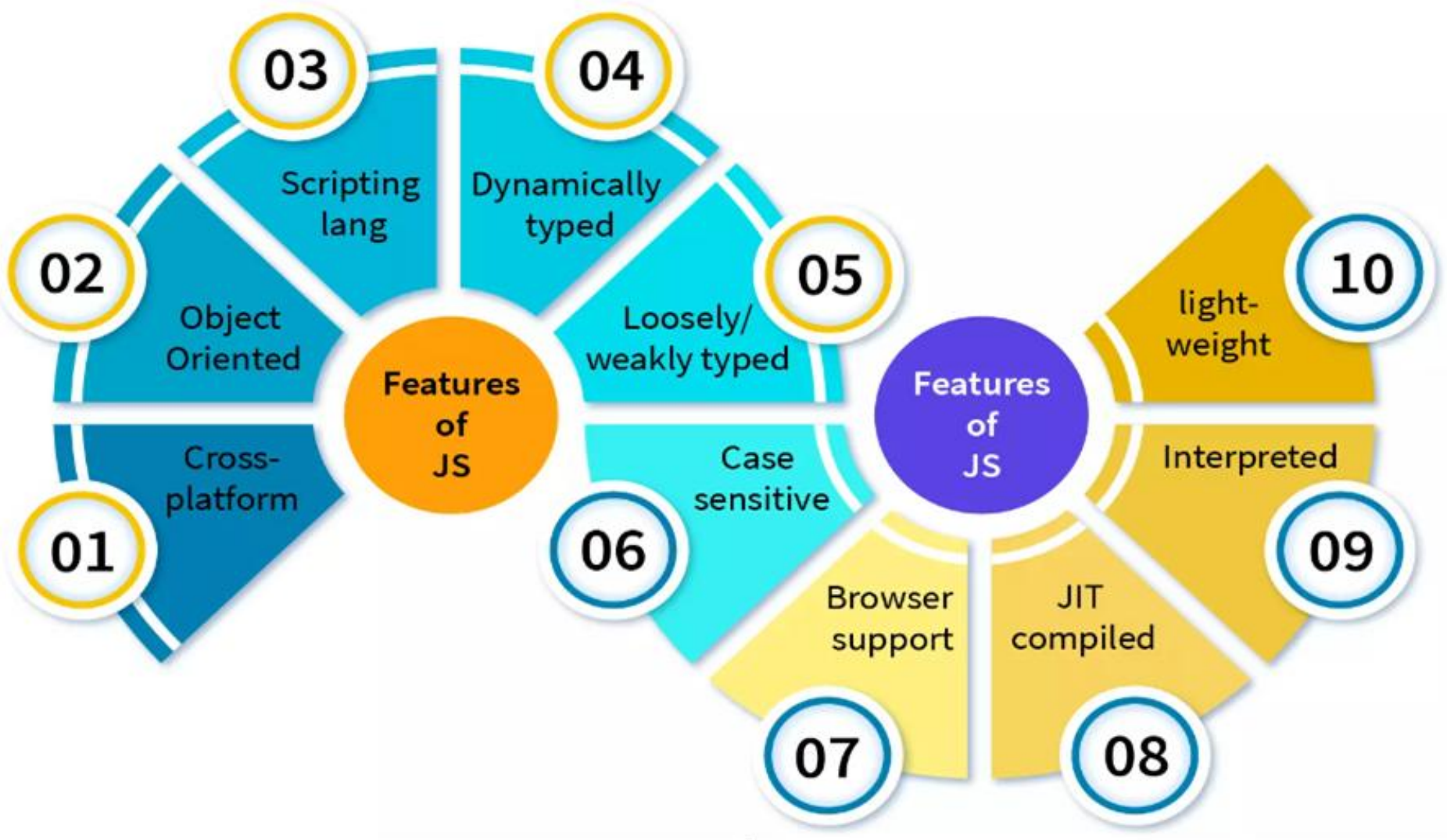
JavaScript vs. PHP

- differences:
  - JS is more object-oriented: noun.verb(), less procedural: verb(noun)
  - JS focuses on user interfaces and interacting with a document; PHP is geared toward HTML output and file/form processing
  - JS code runs on the client's browser; PHP code runs on the web server

HTML + CSS + JavaScript



Features of JavaScript

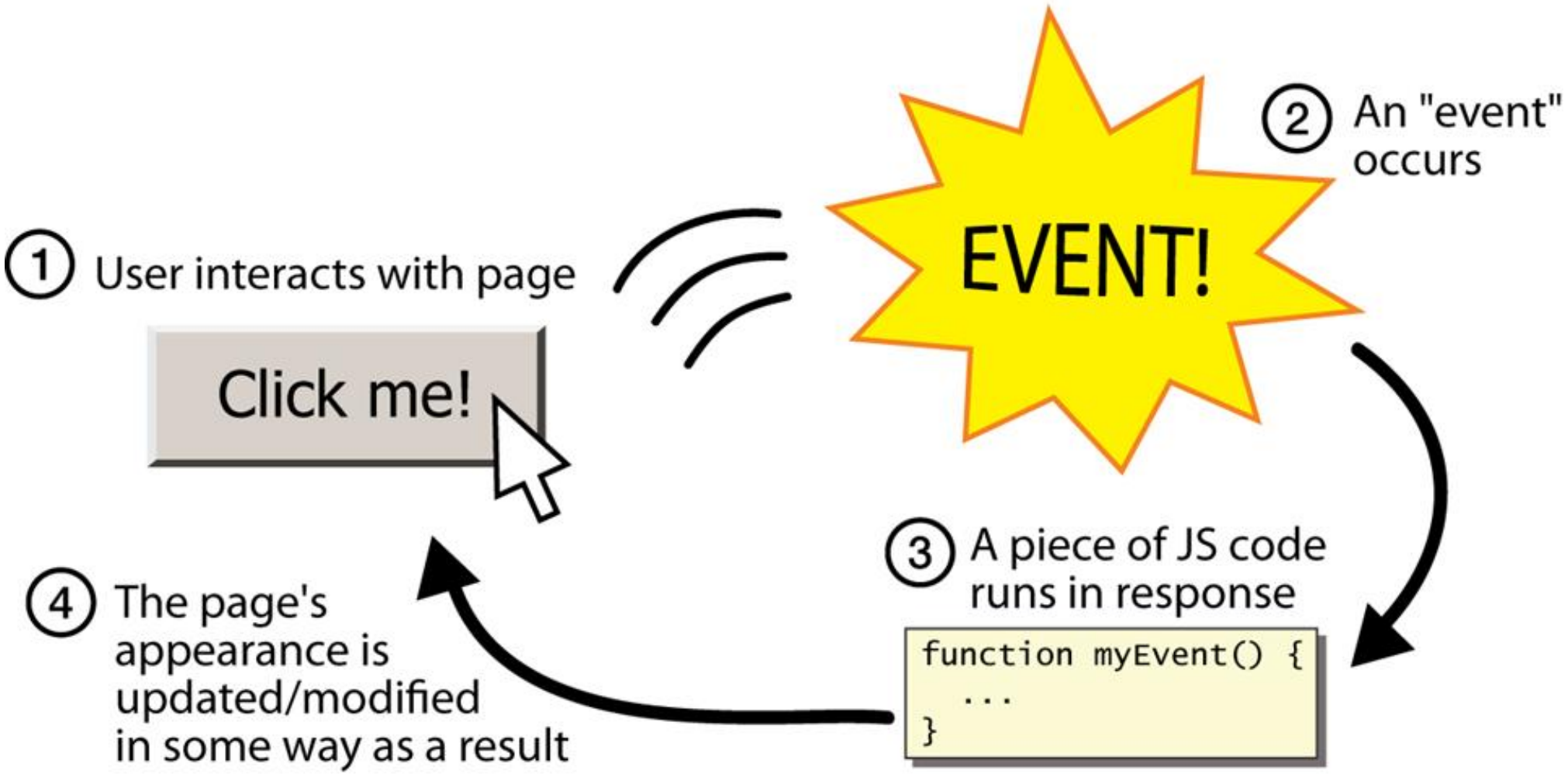


Linking to a JavaScript file: `script`


```HTML
<script src="filename" type="text/javascript"></script>
```

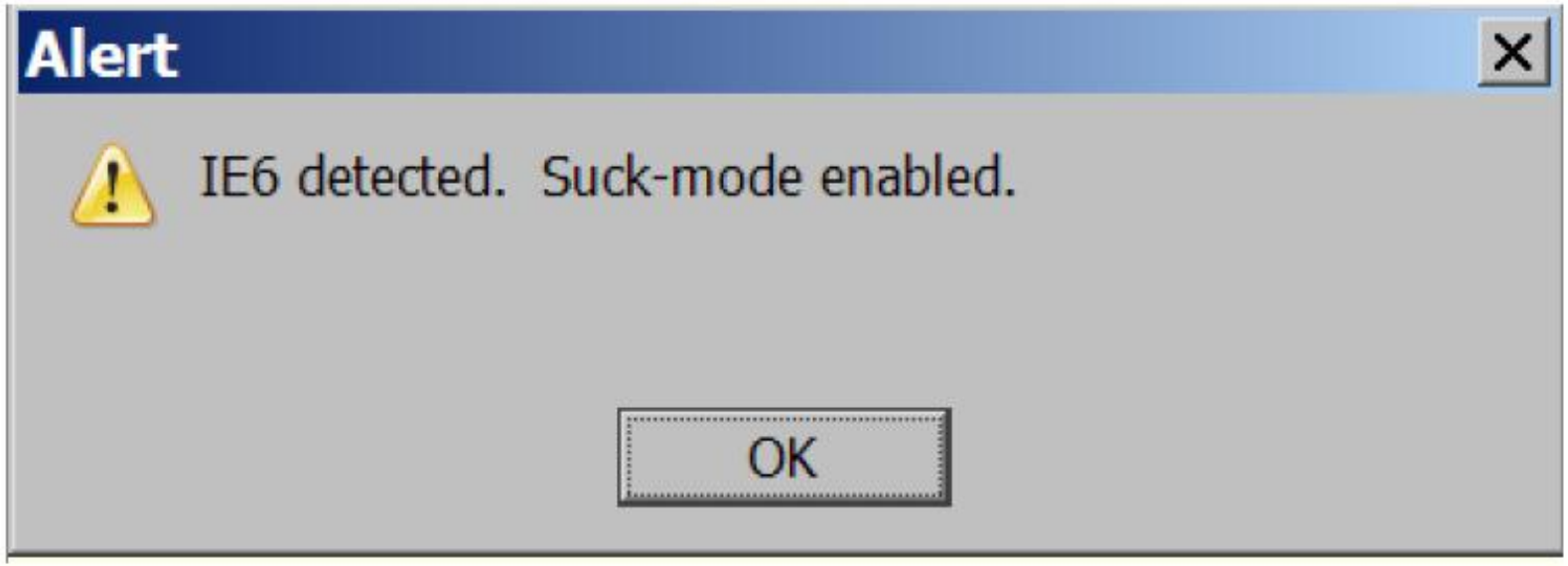
- script tag should be placed in HTML page's head
- script code is stored in a separate .js file
- JS code can be placed directly in the HTML file's body or head (like CSS)
  - but this is bad style (should separate content, presentation, and behavior

Event-driven programming



A JavaScript statement: `alert`


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



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

History of JavaScript

- In September 1995, Brendan Eich, a programmer at Netscape developed a programming language in just 10 days.
- This language was initially called Mocha, then LiveScript.
- In December 1995 Netscape and Sun(the organisation that owned Java) had a license agreement regarding Javascript(then LiveScript) hence it was renamed to Javascript and specifically "Java" script because Java was the most popular language at that time and this would help the marketing of Javascript as well.

History of JavaScript

- Javascript grew rapidly since then and in 1997 ECMA was given the responsibility to create a specification for the language
- Today Javascript is used by millions of developers and is used in major sites such as Google, Facebook, Twitter, etc.

ECMA - the European association for standardizing information and communication systems

What is ECMAScript and how is javascript related to it?

- In 1997 this responsibility was given to ECMA (the European association for standardizing information and communication systems) and a standardized version of Javascript was created, this version is ECMAScript.

- The ECMAScript standard is documented in the ECMA-262 specification.

JavaScript Versions

JavaScript was invented by Brendan Eich in 1995, and became an ECMA standard in 1997.

ECMAScript is the official name of the language.

ECMAScript versions have been abbreviated to ES1, ES2, ES3, ES5, and ES6.

Since 2016, versions are named by year (ECMAScript 2016, 2017, 2018, 2019, 2020).

ES5 vs ES6

Based on	ES5	ES6
Definition	ES5 is the fifth edition of the ECMAScript (a trademarked scripting language specification defined by ECMA International)	ES6 is the sixth edition of the ECMAScript (a trademarked scripting language specification defined by ECMA International).
Release	It was introduced in 2009.	It was introduced in 2015.
Data-types	ES5 supports primitive data types that are string, number, boolean, null, and undefined.	In ES6, there are some additions to JavaScript data types. It introduced a new primitive data type 'symbol' for supporting unique values.
Defining Variables	In ES5, we could only define the variables by using the var keyword.	In ES6, there are two new ways to define variables that are let and const.
Performance	As ES5 is prior to ES6, there is a nonpresence of some features, so it has a lower performance than ES6.	Because of new features and the shorthand storage implementation ES6 has a higher performance than ES5.

ES5 vs ES6

Based on	ES5	ES6
Support	A wide range of communities supports it.	It also has a lot of community support, but it is lesser than ES5.
Object Manipulation	ES5 is time-consuming than ES6.	Due to destructuring and speed operators, object manipulation can be processed more smoothly in ES6.
Arrow Functions	In ES5, both function and return keywords are used to define a function.	An arrow function is a new feature introduced in ES6 by which we don't require the function keyword to define the function.
Loops	In ES5, there is a use of for loop to iterate over elements.	ES6 introduced the concept of for...of loop to perform an iteration over the values of the iterable objects.

Thank You..!