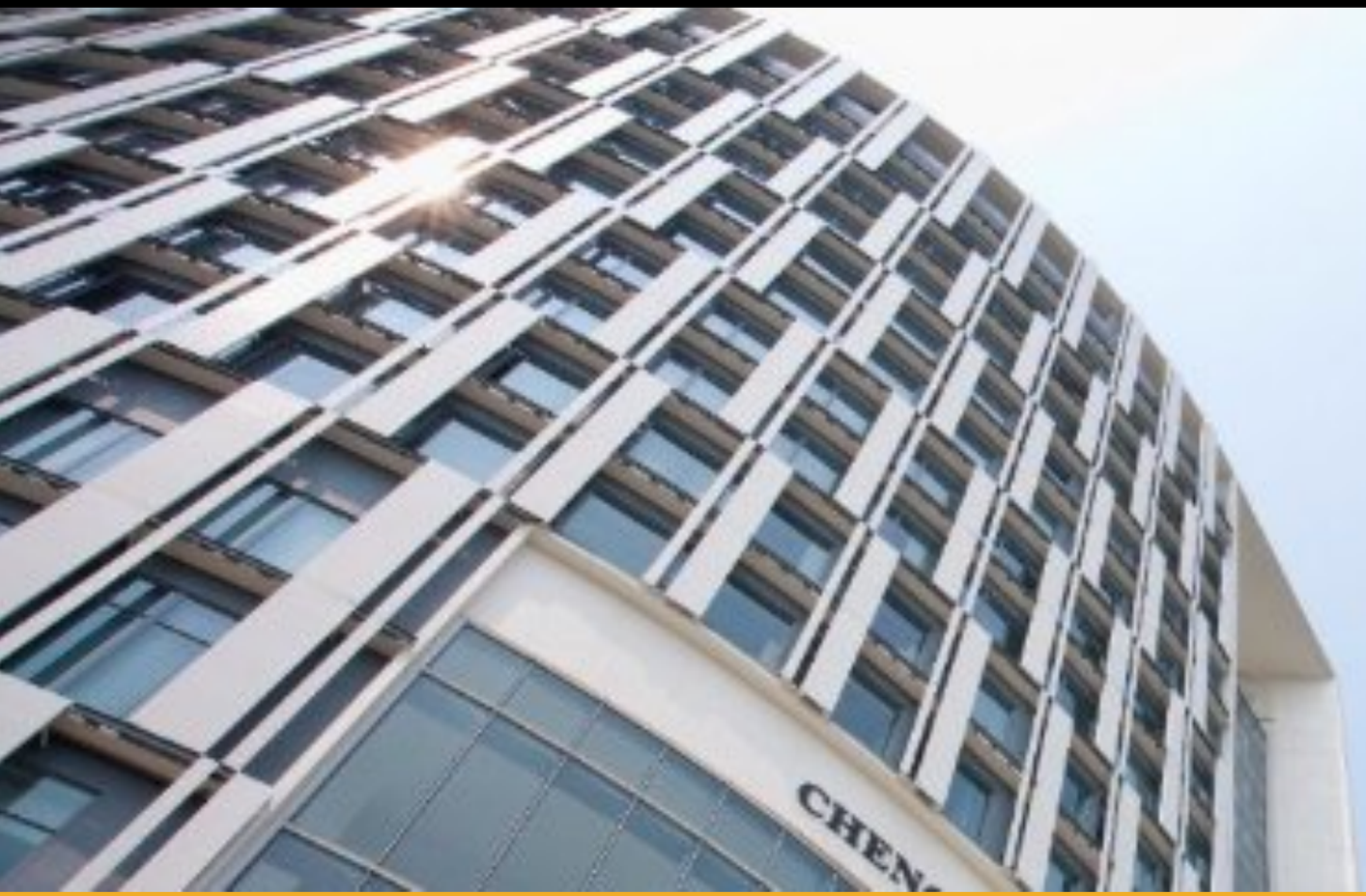


DOM MANIPULATION WITH JAVASCRIPT AND JQUERY

Bernard Suen
Center for Entrepreneurship
Chinese University of Hong Kong

CUHK BUSINESS SCHOOL
Center for Entrepreneurship

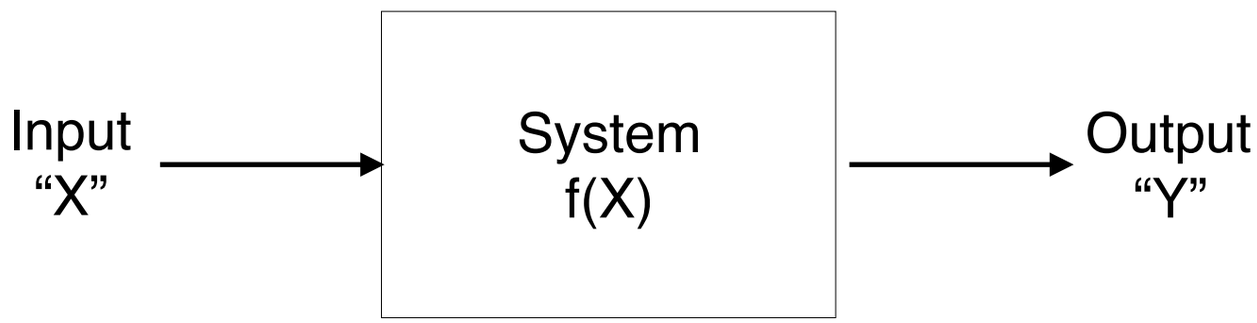
VERB

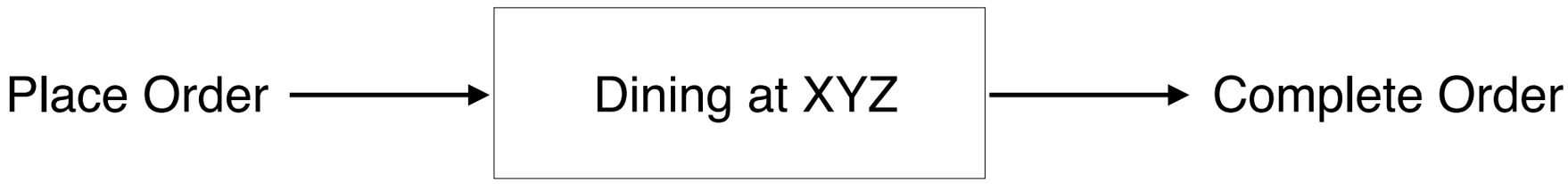
JS

JavaScript = act on a HTML tag, CSS property or respond to an event triggered by user action

What is JavaScript? How does it fit into computational thinking?

$$Y = f(X)$$





Example: Cell manipulation in Excel with some cells controlling inputs while others outputs.

JavaScript provides us with the capabilities to build system and transform data.

Data Types in JavaScript

Declaring a variable and its data type:

- String - e.g. var str_var = “This is a string.”;
- Numeric - e.g. var num_var = 3.2;
- Boolean - e.g. var bol_var = true;

Basic Input/Output Commands

- **Entering a variable - e.g. var x = prompt(“Enter x value”);**
- **Displaying a variable - e.g. alert(“x = “ + x_var);**

JavaScript Functions that Transform Input into Output

Basic Structure of a JavaScript Function

optional parameters

```
<head>
<script>
function function_name(parameter1, parameter 2…) {
```

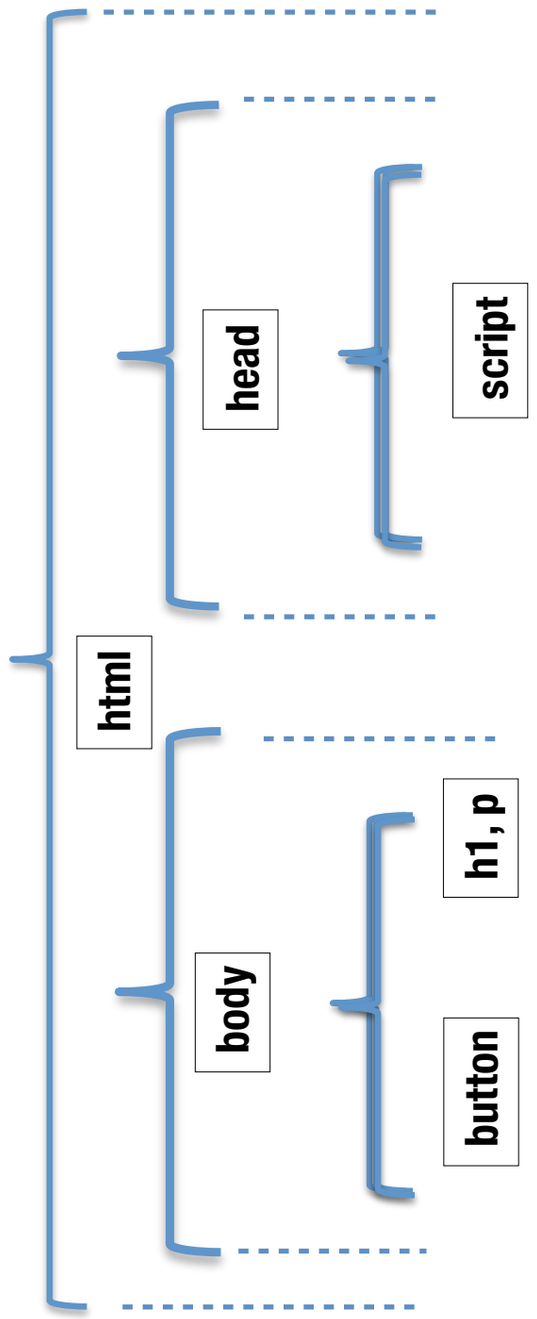
Embed data type variables, input/output commands and logical and mathematical operators in the function to compute and return values.

```
}
</script>
</head>
```

First hands-on exercise on JavaScript:

https://www.javascript.com/try

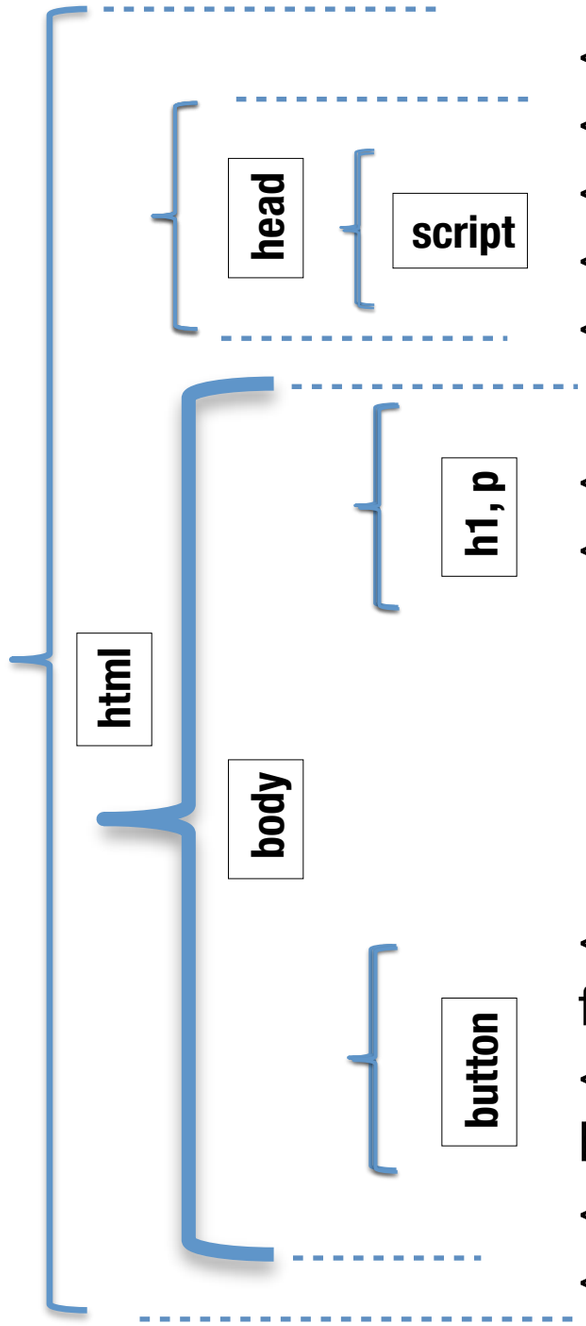
Code View




```
<!DOCTYPE html>
<html>
<head>
<script>
function show_topic() {
    var x = document.getElementById("demo");
    x.style.fontSize = "25px";
    x.style.color = "red";
}
</script>
</head>
<body>
<h1>Demo Page</h1>
<p id="demo">Click the button to change the color and font
style of this line.</p>
<button onclick="show_topic()">Click this button!</button>

</body>
</html>
```

Code View

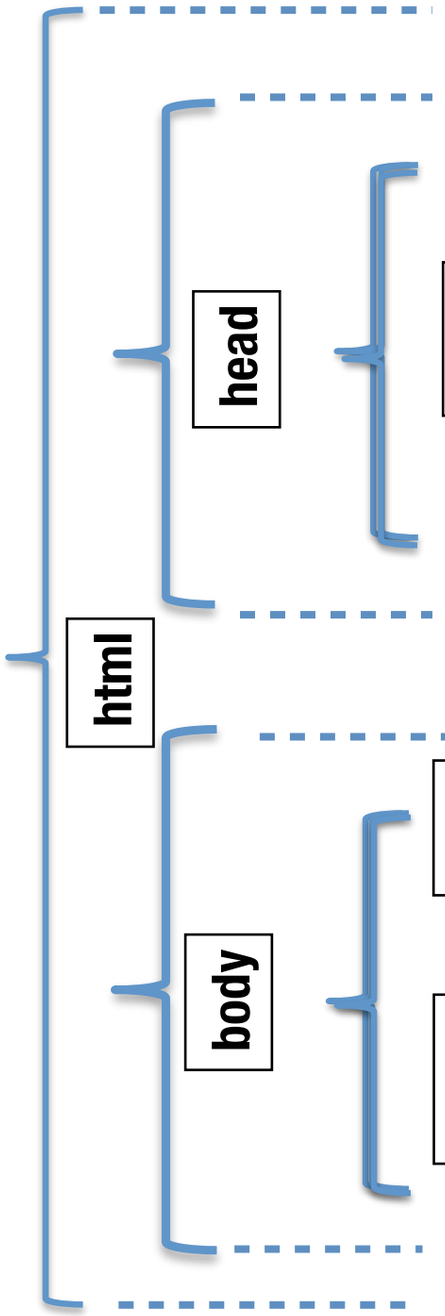



```
<!DOCTYPE html>
<html>
<head>
<script src="script.js"></script>
</head>
<body>

<h1>Demo Page</h1>
<p id="topic">A Paragraph</p>

<p id="demo">Click the button to change the color and
font style of this line.</p>
<button onclick="show_topic()">Click this button!</
button>
</body>
</html>
```

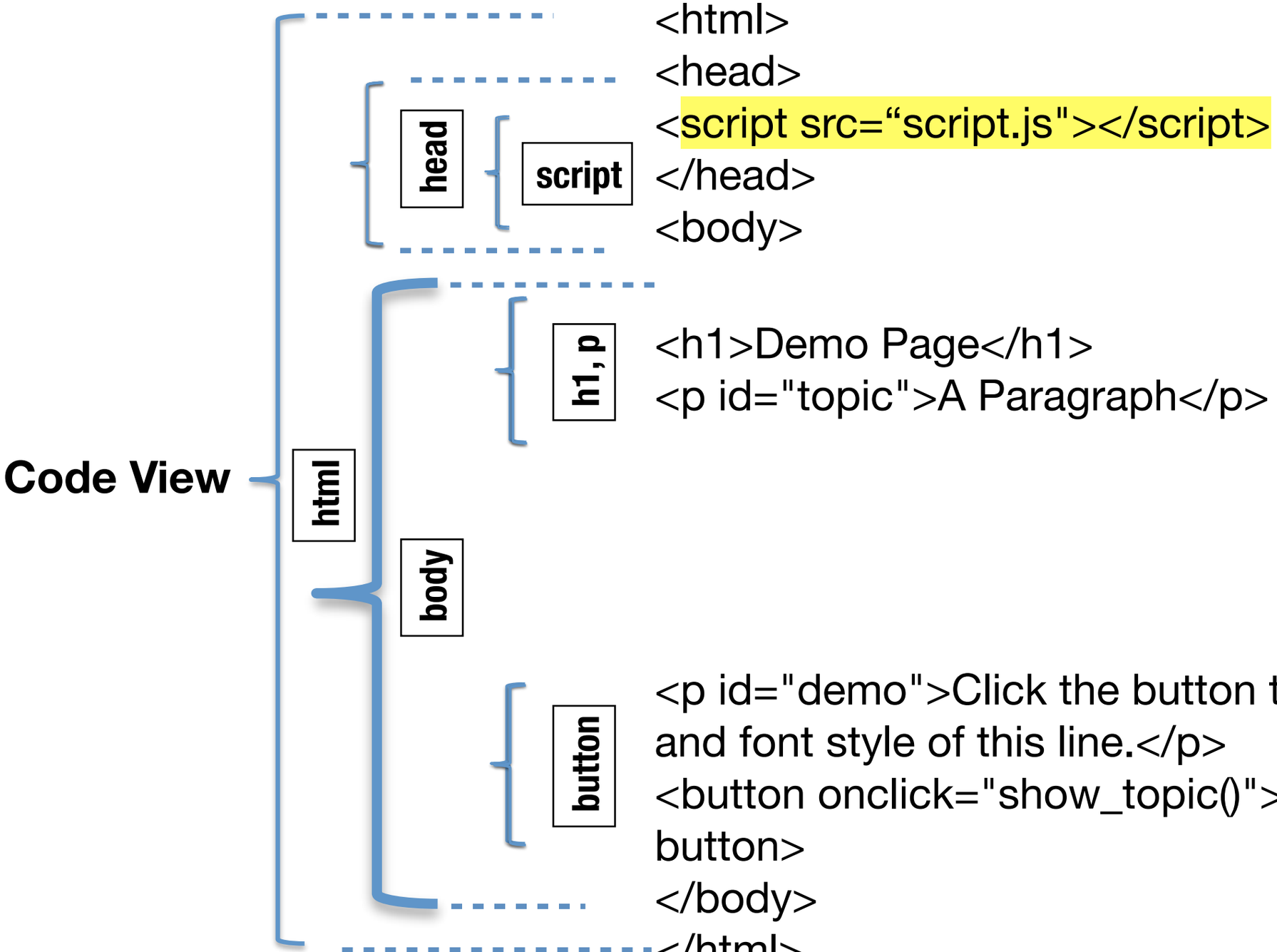
Code View




```
<!DOCTYPE html>
<html>
<head>
<script>
function show_topic() {
    var x = document.getElementById("demo");
    x.style.fontSize = "25px";
    x.style.color = "red";
}
</script>
</head>
<body>
<h1>Demo Page</h1>
<p id="demo">Click the button to change the color and
font style of this line.</p>
<button onclick="show_topic()">Click this button!</
button>

</body>
</html>
```

Similar to CSS, JS can be placed in an External File


Code View
html
head
script
body
h1, p
button
<!DOCTYPE html>
<html>
<head>
<script src="script.js"></script>
</head>
<body>
<h1>Demo Page</h1>
A Paragraph
Click the button to change the color and font style of this line.
<button onclick="show_topic()">Click this button!</button>
</body>
</html>

JavaScript Operations and Commands that Can Enrich the Transformation Process

Abstraction

Ignore irrelevant details to focus on essential features to come up with one solution or classification that works for multiple situations.

Basic Logical and Mathematical Operations

- == equal (comparing string and boolean)
- != not equal (comparing string and bollean)
- = equal (comparing numerical values)
- >= greater than or equal to (comparing numerical values)
- <= smaller or equal to (comparing numerical values)
- +,-,*,/,%, &&, ||,! (addition, subtraction, multiplication, division, modular, and, or, not)

Basic Structure of a JavaScript Function

```
<!DOCTYPE html>
<html>
<head><script>
function addition(a, b) {
    a = parseInt(a); b = parseInt(b);
    c = a + b;
    return c;
}
function get_values() {
    var a = prompt("Enter first number:");
    var b = prompt("Enter second number:");
    var z = addition(a,b); alert("The answer is:" + z);
}
</script></head>
<body>
    <button onclick="get_values();">Click here</button>
</body>
</html>
```

Basic Logical and Mathematical Operations

if (condition) {action} else {action}
Examples:

- if (boolean_var == true) {alert(“That is correct”);} else {alert(“That is incorrect”);}
- if (string_var != “David”) {alert(“Not Peter”);}
- if (num_var >= 8) {alert(“The number is greater than or equal to eight.”);} else {alert(“The number is smaller than eight.”);}

Input/Output Commands without Pop-up

- **Entering a variable values through HTML form - e.g.**

```
<script>
function guessInteger() {
        guess = document. forms[‘guessForm’][‘guessValue'].value;
        if (guess == ‘’) {
              document.getElementById(‘demo’).innerHTML = “Empty!”;
              return;
        } else {
              guess_int = parseInt(guess);
          if (guess_int) == 20)
              {document.getElementById(‘demo’).innerHTML = “Right!”;} else
              {document.getElementById(‘demo’).innerHTML = “Wrong!”;}
       return;
 }
</script>
<body>
<form name=‘guessForm’>
        <input name = “guessValue” class=“inputField”>
</form>
<button class=‘button’ onclick=‘guessInteger()'>Guess an Integer</button>
<div id=‘demo’></div>
</body>
```

More Advanced JS Data Structures: Array and Object

- **Array - a list of elements e.g.**
  var fruits = [“apple”,”grape”,”pear”];)

- **Object - a collection of properties represented in name:values pairs e.g.**

```
var student {
        student_id: 1155115511;
        student_fname: “Bernard”;
        student_lname:”Suen”;
        student_major: “EPIN”;
      }
```

Loop

Loop is an iterative programming construct suitable for handling JavaScript array and object.

```
for (initialization; condition; increment) {
    JavaScript statements
}
```

Try the following steps:

```
1) var fruits = [];
2) for (i=1; i< 10 ;i++) {
        fruits[i] =
   prompt("Enter
   fruit:");
   }
3) alert("fruits contain"
   + fruits);
```

$$Y = f(a,b)$$

“Ending Number” e.g.
b= 200

“Starting Number” e.g.
a=1

Compute next lower number from the top and next higher number from the bottom.
200->199->198->197….
1->2->3->4….
Sum the results

Sum of
a + b where
a = next lower number from the top and b = next higher from the bottom

“You may need a loop to complete this function.“

Functions in JavaScript Programming

- You can look at a function as a mini-system.
- A function is designed to transform input into output.
- You can execute a function within another function.
- A program can be viewed as a collection of functions decomposed into hierarchy of functions to get things done.
- Good programmer looks for patterns in job to be done and abstract common parameters, algorithms, and outcomes to be placed inside a function for code reuse.

The Grammar of JavaScript

- JavaScript is a programming language that can be used to write functions placed inside html or an external file.
- JavaScript can be placed between the <script> and </script> tags inside the <head> section or link to an external file through the script src link.
- JavaScript codes can be understood as a collection of functions that respond to events triggered by internal browser activities and external user interactions.
- JavaScript can be used to manipulate HTML elements and CSS styles.

Decomposition

Pattern

Abstraction

Algorithm

Automation & Testing

Computational thinking is about system and data.

JavaScript provides us with the capabilities to build system and transform data.

JavaScript Functions that Transform Input into Output

Basic Structure of a JavaScript Function

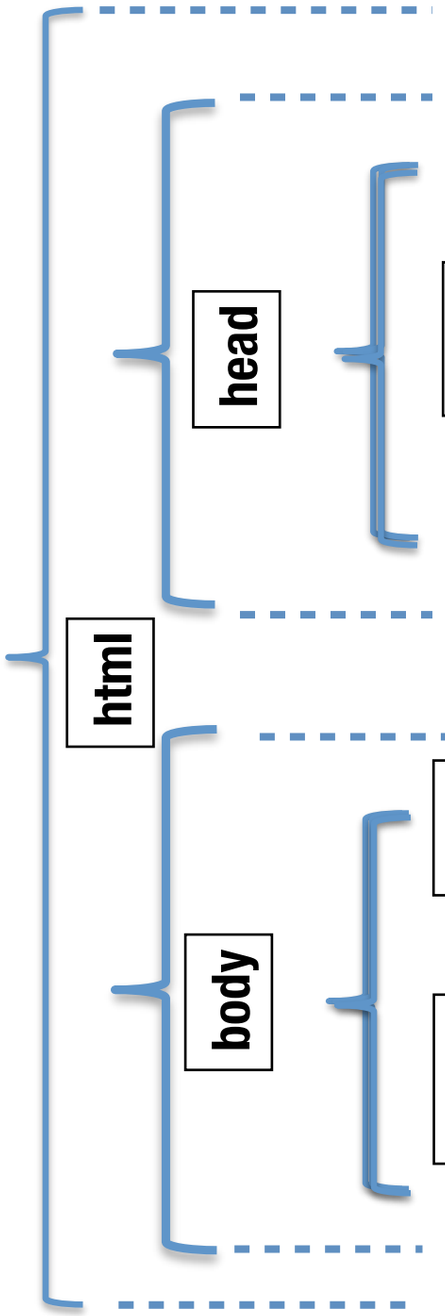
optional parameters

<head>
<script>
function function_name(parameter1, parameter 2…) {

Embed data type variables, input/output commands and logical and mathematical operators in the function to compute and return values.

}
</script>
</head>

Code View

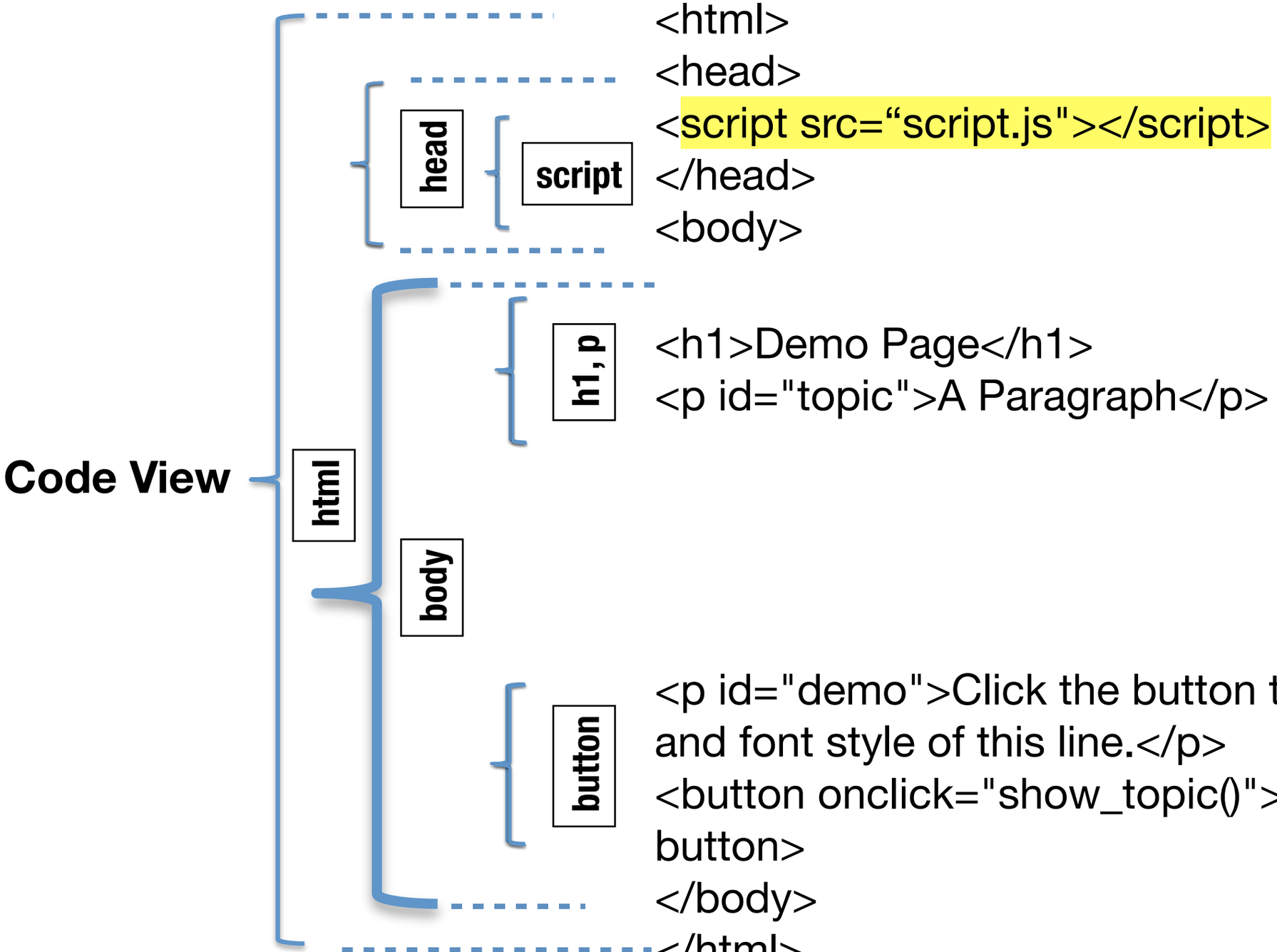



```
<!DOCTYPE html>
<html>
<head>
<script>
function show_topic() {
    var x = document.getElementById("demo");
    x.style.fontSize = "25px";
    x.style.color = "red";
}
</script>
</head>
<body>
<h1>Demo Page</h1>
<p id="demo">Click the button to change the color and
font style of this line.</p>
<button onclick="show_topic()">Click this button!</
button>

</body>
</html>
```

Similar to CSS, JS can be placed in an External File

See an Example


Code View
html
head
script
body
h1, p
button
<!DOCTYPE html>
<html>
<head>
<script src="script.js"></script>
</head>
<body>
<h1>Demo Page</h1>
A Paragraph
Click the button to change the color and font style of this line.
<button onclick="show_topic()">Click this button!</button>
</body>
</html>

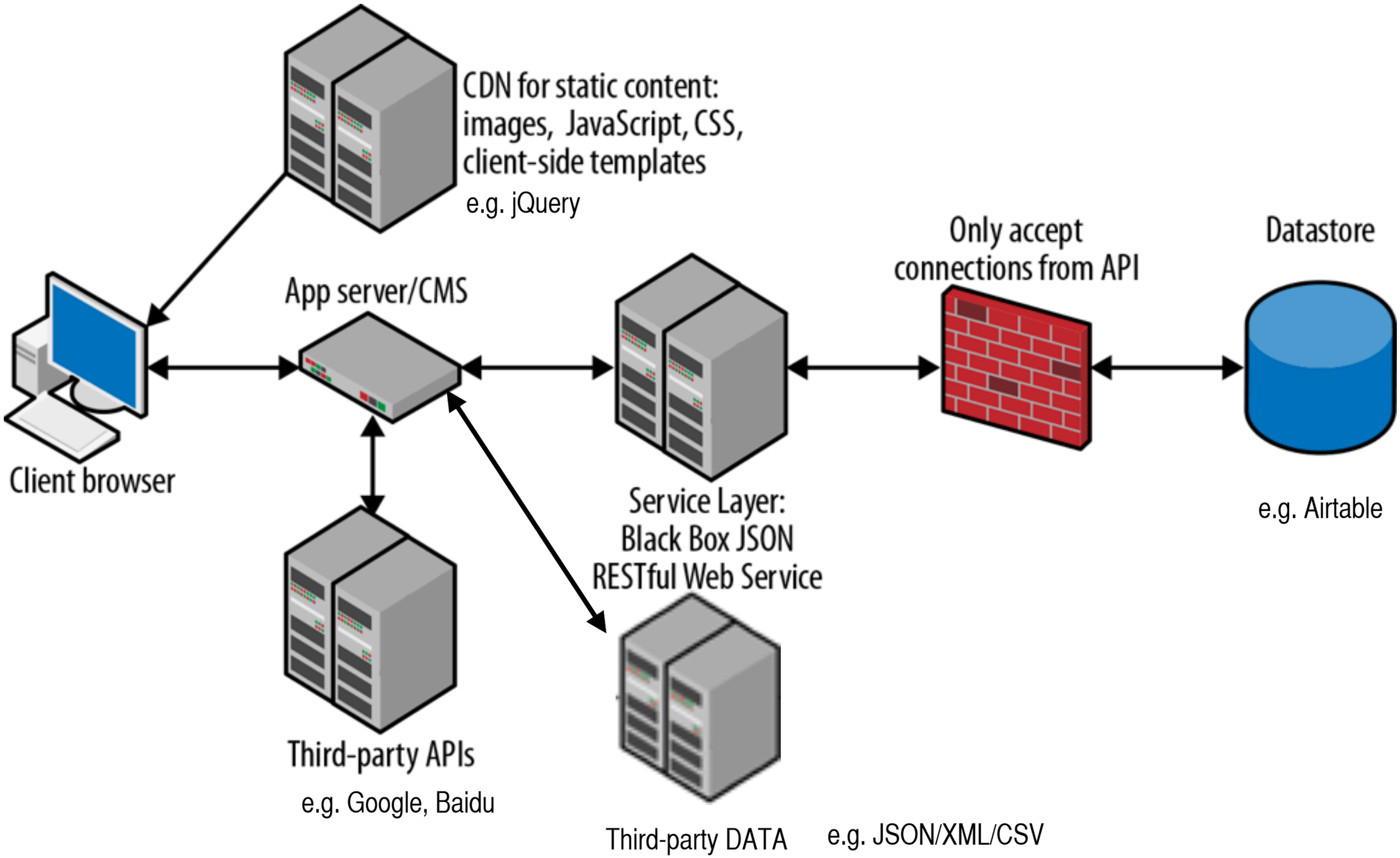
Functions in JavaScript Programming

- You can look at a function as a mini-system.
- A function is designed to transform input into output.
- You can execute a function within another function.
- A program can be viewed as a collections of functions decomposed into hierarchy of functions to get things done.
- Good programmer looks for patterns in job to be done and abstract common parameters, algorithms, and outcomes to be placed inside a function for code reuse (automation).

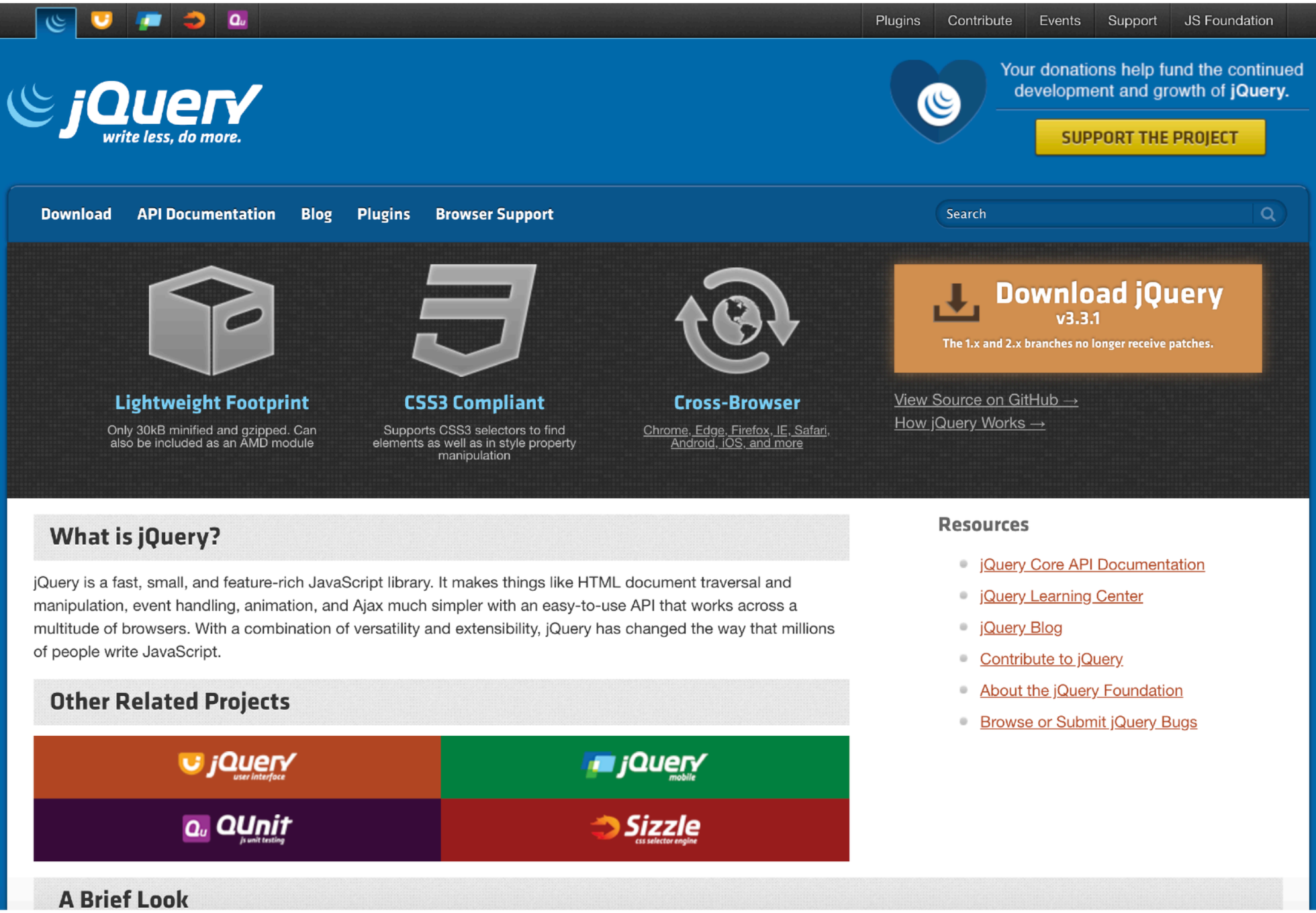
JavaScript Has Become the Most Popular Front-end Language with Pre-packaged Framework and Library Support

e.g. jQuery,
AngularJS, React,
Vue

e.g. jQuery
DataTable, D3,
C3D3, Leaflet



Source: http://chimera.labs.oreilly.com


Plugins
Contribute
Events
Support
JS Foundation
jQuery
write less, do more.
Your donations help fund the continued development and growth of jQuery.
SUPPORT THE PROJECT
Download
API Documentation
Blog
Plugins
Browser Support
Search
Lightweight Footprint
Only 30kB minified and gzipped. Can also be included as an AMD module
CSS3 Compliant
Supports CSS3 selectors to find elements as well as in style property manipulation
Cross-Browser
Chrome, Edge, Firefox, IE, Safari, Android, iOS, and more
Download jQuery
v3.3.1
The 1.x and 2.x branches no longer receive patches.
View Source on GitHub →
How jQuery Works →
What is jQuery?
jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.
Other Related Projects
jQuery user interface
jQuery mobile
QUnit js unit testing
Sizzle css selector engine
Resources
jQuery Core API Documentation
jQuery Learning Center
jQuery Blog
Contribute to jQuery
About the jQuery Foundation
Browse or Submit jQuery Bugs
A Brief Look

Basic jQuery Structure & Syntax

This means:

When the HTML document (i.e. DOM) is loaded, add the following jQuery functions (in orange) to your javascripts

```
$(document).ready(function() {
```

Your jQuery functions go inside here.

```
});
```

The following code demonstrates the basic structure of a jQuery function. The selector and method are two ingredients used to define a jQuery function:

```
$("<selector>").<method>('event',function(){
        The actual script goes here.
 });
```

Example:

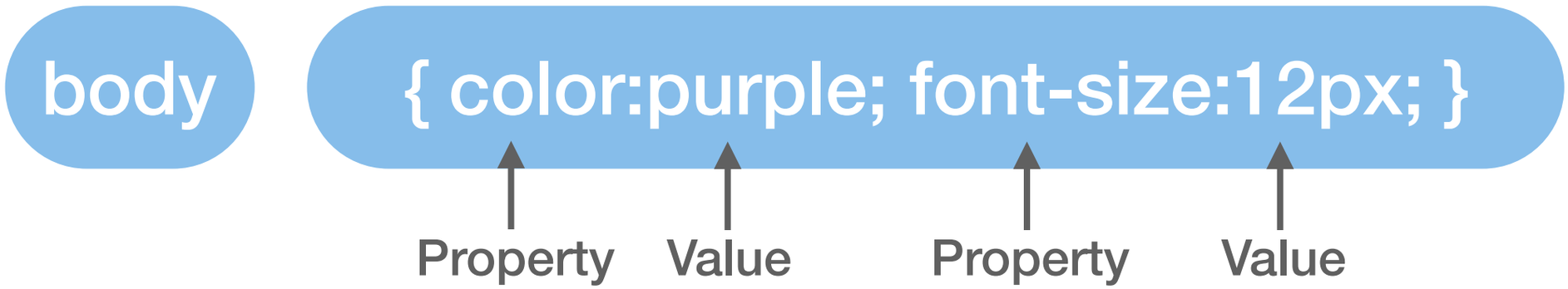
```
$("button#hide_h2").on('click',function(){
                $("h2").hide();
});
```

ADJECTIVE

CSS

Base

Selector Declaration Declaration

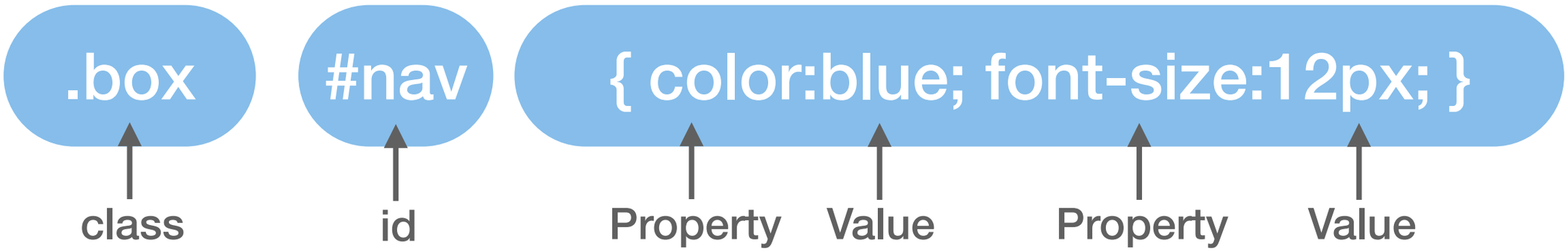


ADJECTIVE

CSS

Custom (e.g. class or id)
Selector

Declaration Declaration



Putting Everything Together

```
$(document).ready(function() {
        $(“button#hide_h2”).on(‘click’,function(){
                        $("h2").hide();
        });
        $(“button#show_h2”).on(‘click’,function(){
                        $(“h2").show();
                        $(“h2”).css(“color”,”blue”);
                        $(“h2”).html(“You clicked me.”);
});  });
```

The HTML Form and Input Objects(Tags)

Using Bootstrap Form and Button to Make Browser UI Look Better

The Grammar of JavaScript

- JavaScript is a programming language that can be used to write functions placed inside html or an external file.
- JavaScript can be placed between the <script> and </script> tags inside the <head> section or link to an external file through the script src link.
- JavaScript codes can be understood as a collection of functions that respond to events triggered by internal browser activities and external user interactions.
- JavaScript can be used to manipulate HTML elements and CSS styles.

Problem Set #4

Write a JavaScript program to create a puzzle (feel free to design a math or word puzzle as you wish). Ask the player to input values through HTML form and display the result inside a box (with an ID selector) positioned inside the browser.

Thank You!