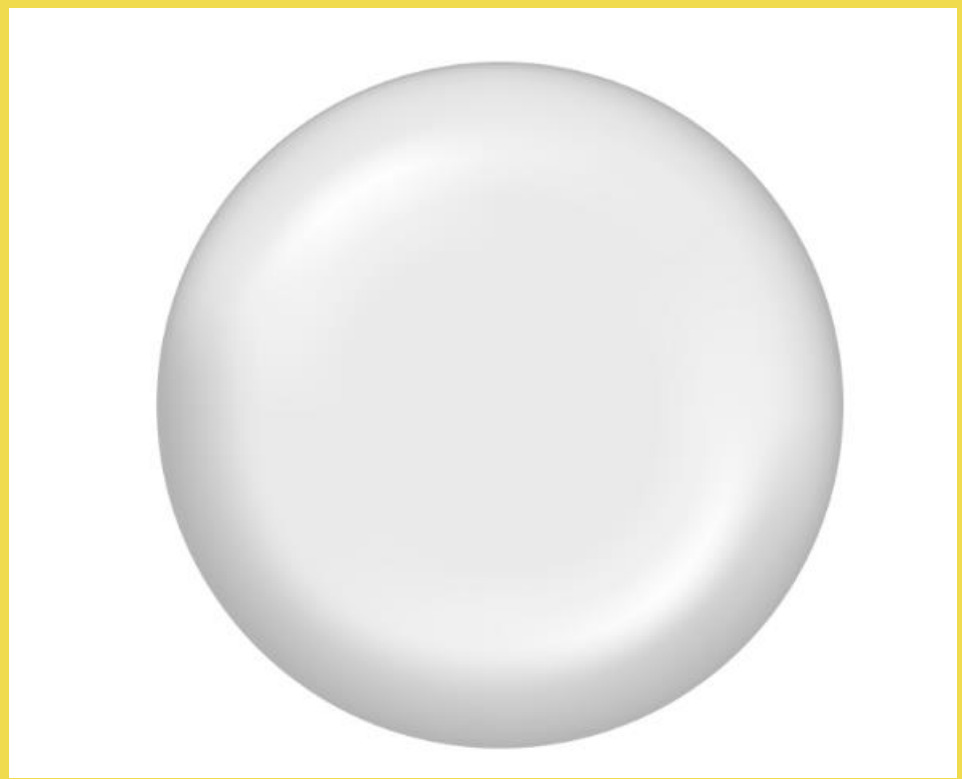


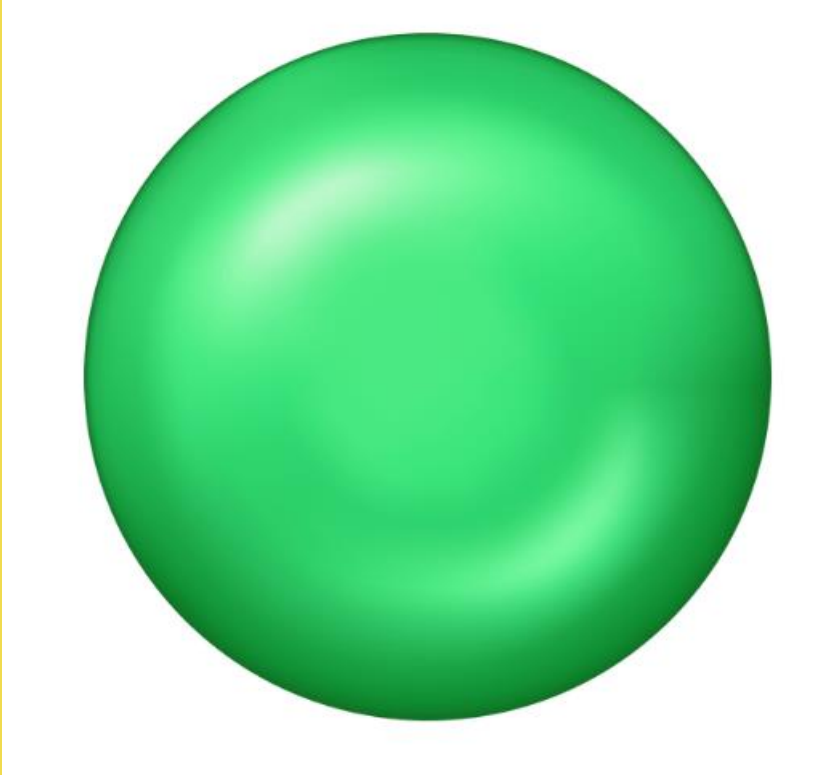
# JavaScript Fundamentals

## Why Study JavaScript?

JavaScript is one of the **3 languages** all web developers **must** learn:

1. **HTML** = Structure
2. **CSS** = Style
3. **JavaScript** = Script (Interactivity and Behavior)







const js = “free for everybody”;

# JavaScript Fundamentals

Numbers and Math

Strings

Variables (include Scope)

Booleans and If-Statements

Functions

Events

Objects

Document Object Model (DOM)

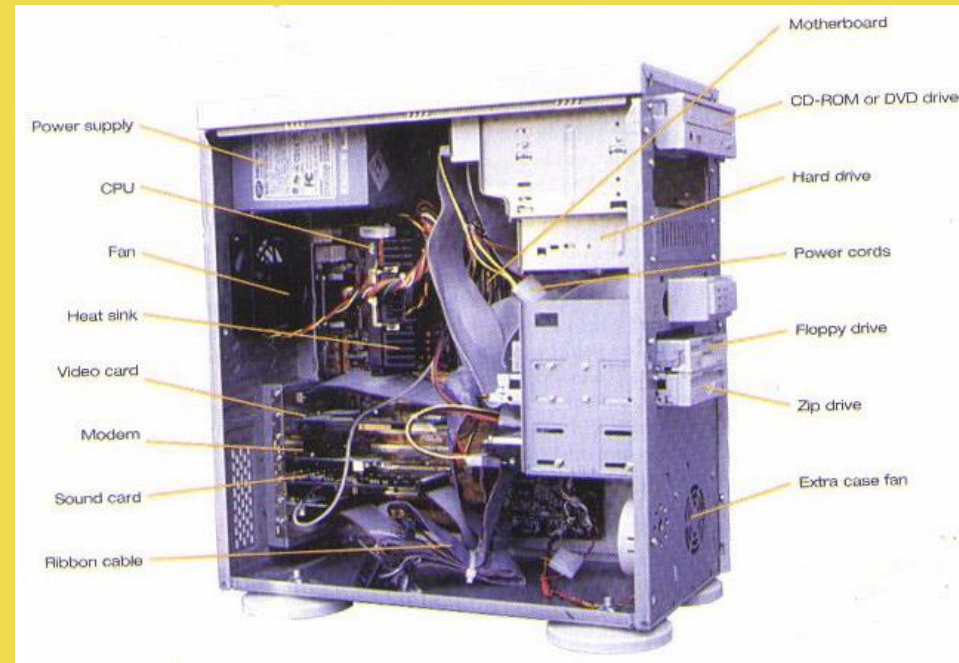
Arrays

Loops


# JavaScript Objects



# Real life objects



# Real life objects

Object	Properties	Methods
	<p>phone.brand= iPhone 15 Pro</p> <p>phone.displaySize = 6.1 inches</p> <p>phone.displayType = OLED</p> <p>phone.color = gold or yellow</p>	<p>phone.on()</p> <p>phone.off()</p>

# JavaScript Variables

```
let age = 30;  
let name = "John";
```

JavaScript variables  
are containers for  
data values.

VS

# JavaScript Objects

JavaScript objects can  
contain many values.

```
const person1 = {  
  name: "John",  
  age: 30  
};
```

Commonly, used to store  
and organize related  
data.

```
const students = [  
  { name: "Gracey" },  
  { name: "Gaeus" },  
];
```

Can be used to create  
more complex data  
structures. (eg. Arrays)

It is a common practice to declare objects  
with the `const` keyword.

ASI, or Automatic Semicolon Insertion.

# Object Properties

Property:Value pairs

Property	Value
firstName	Kevin
lastName	Durant
age	35
eyeColor	black

## Accessing Object Properties

You can access object properties in two ways:

*objectName.propertyName* called that *dot.notation*

or

*objectName["**property-name**"]* called the [*“bracket-natation”*]

# Object Properties

The **name:values (property-value)** pairs in JavaScript objects are called **properties**:

Property	Property Value
firstName	Kevin
lastName	Durant
age	35
eyeColor	black

## Manipulating Object Properties

You can add object property:

*objectName.propertyName; (does not exist in object)*

*You can delete object property:*

*delete objectName.propertyName;*

# Object Methods

They are **functions** that are associated with and defined within objects, and they allow you to perform specific operations or manipulations on the object's properties (data).

```
propertyName: function() {  
    return ("action you want to do with properties")  
}
```

## Accessing Object Methods

You access an object method with the following syntax:

*objectName.methodName()*

If you access a method **without** the () parentheses, it will return the **function definition...**

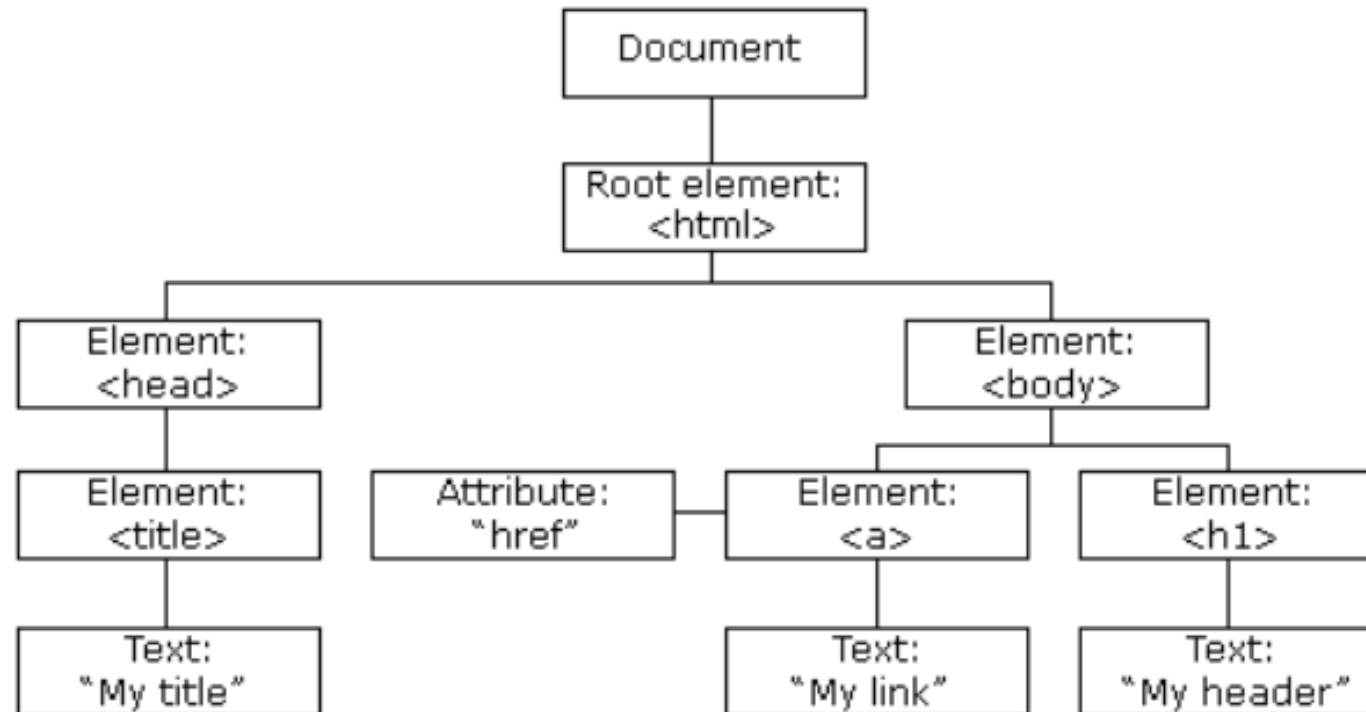
JavaScript HTML DOM

# The HTML DOM (Document Object Model)

When a web page is loaded, the browser creates a **Document Object Model** of the page.

DOM = another built-in object in JS (like console, Math, JSON, localStorage) → access using document

## The HTML DOM Tree of Objects





# The HTML DOM (Document Object Model)

With the object model, JavaScript gets all the power it needs to create dynamic HTML:

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

# The HTML DOM (Document Object Model)

With the object model, JavaScript gets all the power it needs to create dynamic HTML:

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

# JavaScript - HTML DOM Methods

The HTML DOM can be accessed with JavaScript (and with other programming languages). In the DOM, all HTML elements are defined as **objects**.

The programming interface is the properties and methods of each object:

```
<html>
<body>

<p id="demo"></p>

<script>
  document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>
</html>
```

A **method** is an action you can do (access or find an HTML element).

A **property** is a value that you can get or set (like changing the content of an HTML element).