JS-DOM

ANECO ACADEMY

What is DOM ?

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

- The **HTML DOM** model is constructed as a tree of **Objects**:

document is always at the top.
document is a special part of
the tree that you can use in
JavaScript to get access to the
entire DOM.

document is also like the root
of an upside down tree.

document

html

These are like the
branches of the tree.

head

body

meta

title

script

h1

div id="entry1"

My blog

My blog

h2

p

p

These are like the leaves
of the tree (because there
are no elements inside them,
just text).

Great
day bird
watching

Today
I saw
three...

I took a
couple of
photos...

The DOM includes the content of the page as well as the
elements. (We don't always show all the text content when
we draw the DOM, but it's there).

DOM continue..

- **HTML page as a puppet**, then the **DOM will be the strings** and the points at which they attach to the puppet
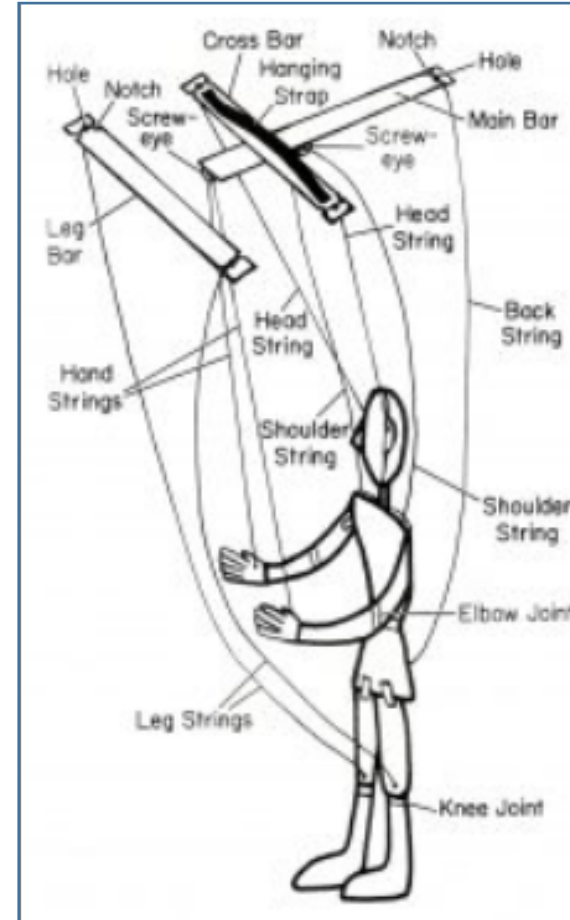
What we can do using DOM?

Using DOM, JavaScript gets all the power it needs to create/update 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

DOM Methods

- HTML **DOM methods are actions** you can perform (on HTML Elements)

- HTML **DOM properties are values** (of HTML Elements) that you can set or change

```
<html>
<body>

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

</body>
</html>
```

innerHTML is
a property

getElementById is
a **method**

DOM Methods

- **getElementById -** To change/modify the content of an HTML element

```
<script>
document.getElementById("p1").innerHTML = "New text!";
</script>
```

→ Changing HTML Content

```
<script>
document.getElementById("myImage").src = "landscape.jpg";
</script>
```

→ Changing the Value of an Attribute

```
<script>
 document.getElementById("myP").style.backgroundColor = "red";
</script>
```

→ Changing Style property

Style and class

```html
<html>
    <head>
        <style>
            .header{
                background-color: ☐indigo;
                font-size:28px;
                text-align: center;
            }
        </style>

    </head>
    <body>
        <h1 id="demo">Welcome to javascript</h1>
        <script>
            document.getElementById('demo').style.color = "red";
            document.getElementById('demo').setAttribute('class','header');
            document.getElementById('demo').setAttribute('name','header-details');
        </script>
    </body>
</html>
```
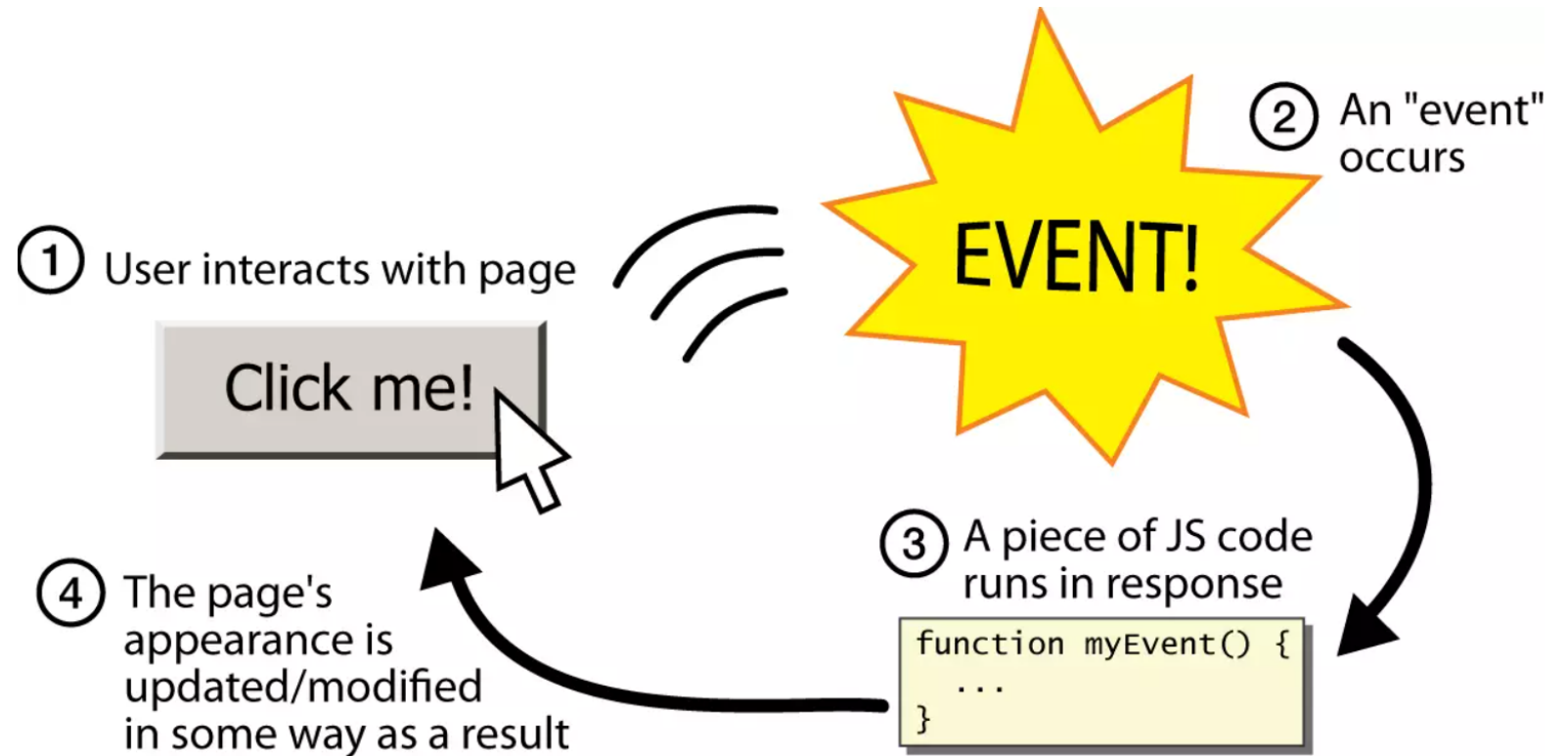
Events in JavaScript

An HTML event can be something the browser does, or something a user does

Here are some examples of HTML events:

➢ An HTML web page has finished loading

➢ An HTML input field was changed

➢ An HTML button was clicked

Events in JavaScript

Common HTML Events:

➢ **Onclick** - The user clicks an HTML element

➢ **Onchange** - An HTML element has been changed

Events in JavaScript

Common HTML Events:

➢ **Onclick** - The user clicks an HTML element

```
<div>
  <p class="name">Learning JS events</p>
  <button id="btn" onclick="changeColor()">Change to Blue</button>
</div>
```

```
function changeColor() {
    document.getElementById("btn").style.color = "blue";
}
```

➢ **Onchange** - An HTML element has been changed

```
<select id="cities" name="city" onchange="selectCity()">
        <option value="chn">Chennai</option>
        <option value="blr">Bengaluru</option>
        <option value="hyd" selected>Hyderabad</option>
    </select>
<p id="slcCty"></p>
```

```
function selectCity() {
            var cty = document.getElementById("cities").value;
            document.getElementById("slcCty").innerHTML
="Selected City: " + cty;
    }
```