__/__/__

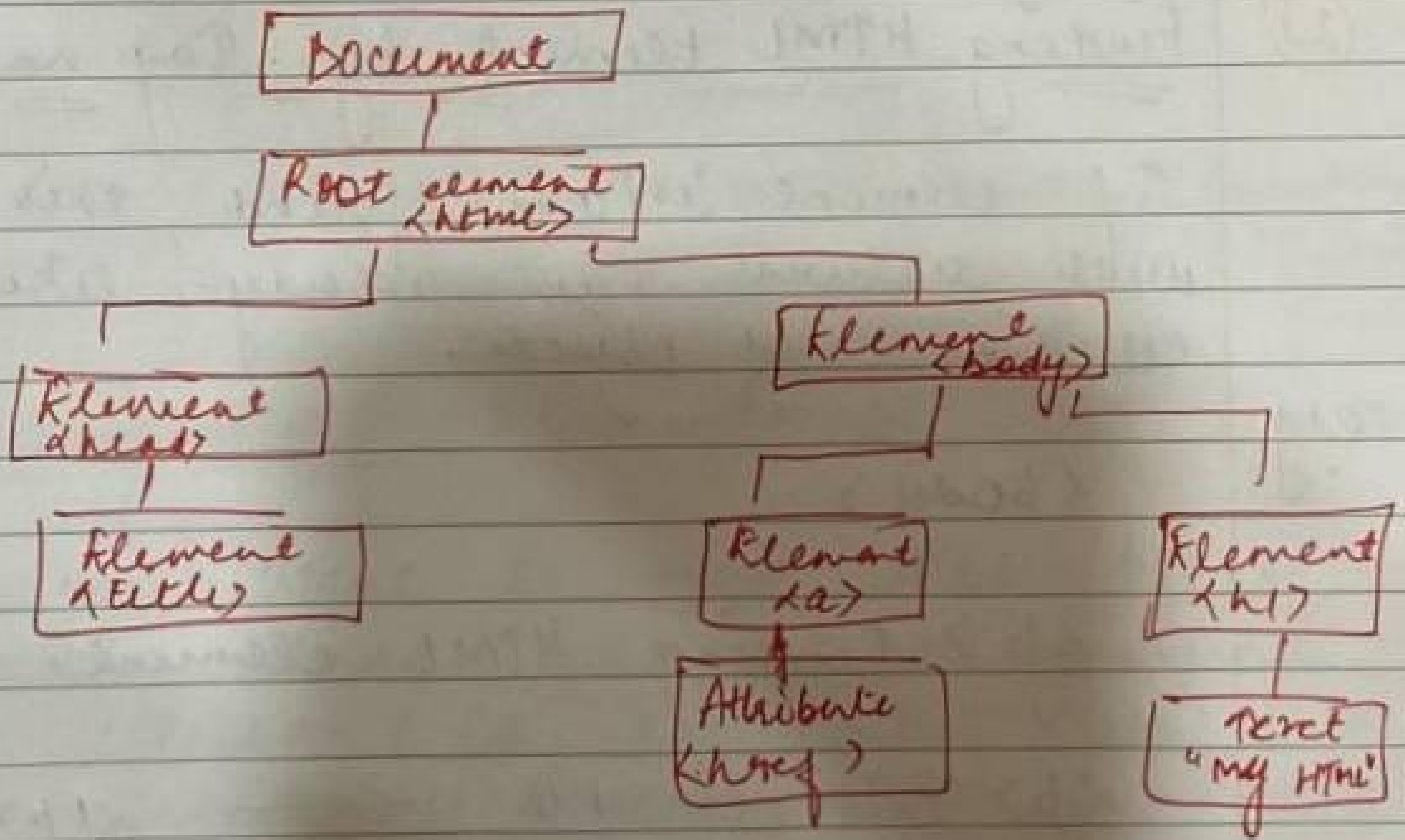
# DOM

With HTML DOM, javascript can access and change all the elements of an HTML document

⇒ HTML DOM is standard object model and programming interface for HTML. It defines

↳ HTML elements as Objects
↳ The properties of all HTML elements
↳ The methods to access all HTML elements
↳ The events for all HTML elements

HTML DOM Tree of Objects



example

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

  <script>
  document.getElementById("demo").innerHTML
                                    = "Hello Worl
```

__/__/__

</script>

Here innerHTML → property

~~document~~ getElementById() → method

DOM Finding Elements

① Finding HTML Elements By id

e.g.- document.getElementById("intro");

② Finding HTML Element by Tag name

If element is / are found, this method will return an array-like HTML collection of objects.

e.g
```
<body>

  <p> Finding HTML elements by Tag name</

  <p> This example ——— </p>

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

<script>

const element = document.getElementByTa
                                    name("p"
```

Scanned by TapScanner

__/__/__

document.getElementById("demo")
    .innerHTML = 'The text in
        first paragraph is :' +
            +element[0].innerHTML;

③ Finding elements by Class name

Same as previous one

eg.

```
<p class = " intro ">
<p class = " intro ">
  =
  =
  =
```

const x = document.getElementByClassName
    ("intro")

④ Finding HTML Elements by CSS Selectors

If you want to find all HTML elements that match a specified CSS selector ( id, class name, etc) use querySelectorAll() method.

eg.

```
<body>
<h2> HTML DOM </h2>
<p> Finding HTML Elements By Query
        Selector </p>

<p class="intro"> Hello World </p>
```

Scanned by TapScanner

__/__/__

```
<p class=" intro"> ______        </p>

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

<script>

const x = document. querySelectorAll ("p.intro");

document. getElementById("demo") . innerHTML
                     = 'The first paragraph
                       with index 0 with
                       class intro' +
                        x[0]. innerHTML ;

</script>
</body>
</html>
```

Note :→ If attribute is 'id then

e.g.
```
<p id = "intro"> ____     </p>
<p id =  "intro").         </p>

<script>

const x = document . querySelectorAll ("#intro");
```

⑤ Finding HTML Elements by HTML object collections.

_/_/_

```
<body>
<p> Finding HTML Elements Using document.form
                                    </p>

<form id = "frm1" >

FirstName: <input type = "text" name = "fnam
                      Value = "Rishi" ><br>

LastName: <Input type = "text" name = "lnam
                   Value = "Surat" > <br>

<input type = "submit" value = "submit">

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

<Script >

const x = document.forms ["frm1"];
                          (* It contains
                           length of
                           elements
                           in form
 let text = " ";

for (let i=0; i< x.length; i++)
{
   txt += x.elements[i].Value + "<br">"
}
```

__/__/__

```
document.getElementById("demo").innerHTML
                        = text;

</script>
```

Note

document.forms["form1"].length (gives number of elements of form)

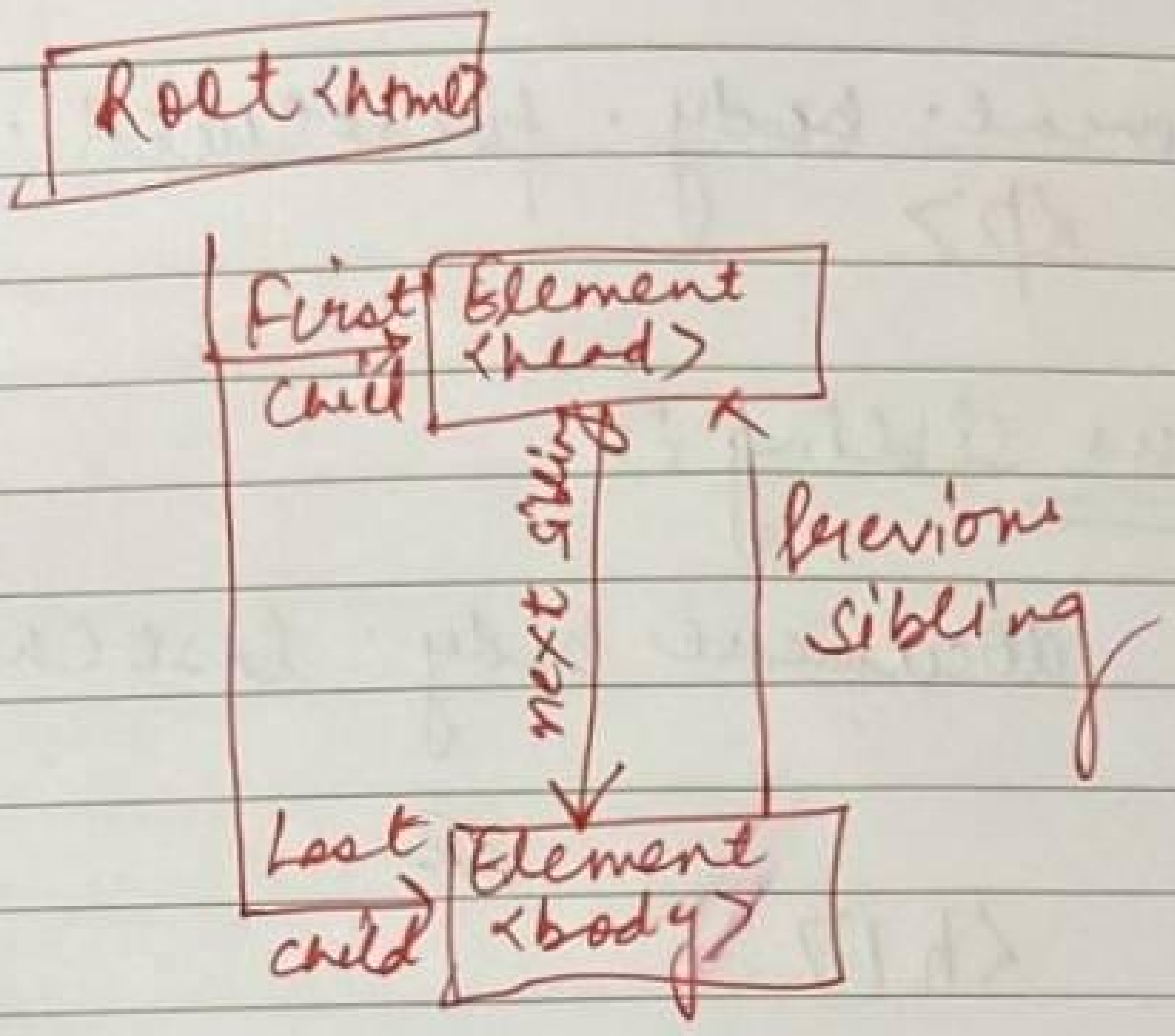
document.forms.length ( gives number of form tag)

## DOM Navigation

With HTML DOM, you can navigate the node tree using node relationships.
The nodes in the node tree have a hierarchical relationship to each other. The term parent, child and siblings are used to describe relationships.

```
<html>
  <head>
    <title> — </title>
  </head>
  <body>
      <h1> — </h1>
      <p> — </p>
    </body>
</html>
```

Scanned by TapScanner



→ ~~All~~ Explaination

We can use the following node properties to navigate b/w nodes with JS.

① parentNode :→ document . body . parent Node
Ans-<html>

② ChildNodes
document . head . childNodes (0);
Ans <title>

③ first child
document . body . first child
Ans h1

④ lastchild
document . body . lastchild
<p>

⑤ nextsibling :-

document. body. firstChild. nextSibling

Ans <p>

(6) Previous Sibling :->

document. body. lastChild. previous Sibling

Ans

<h1>

nodeName Property :->

```
<h1 id = "id01" > My first page </h1>
  <p id = "id02" > </p>

<script>
document. getElementById ("id02"). innerHTML
              = document. getElementById
                    ("id01"). nodeName
```

Ans H1

nodeValue

returns value of node.