Persistent

JavaScript:
Document Object Model

Persistent Interactive | Persistent University

Key learning points :

- DOM representation of HTML file
- Node
  - types
  - properties
  - methods
- Selecting Elements using – id, name & tag name
- DOM API
- Overview of Browser Object Model

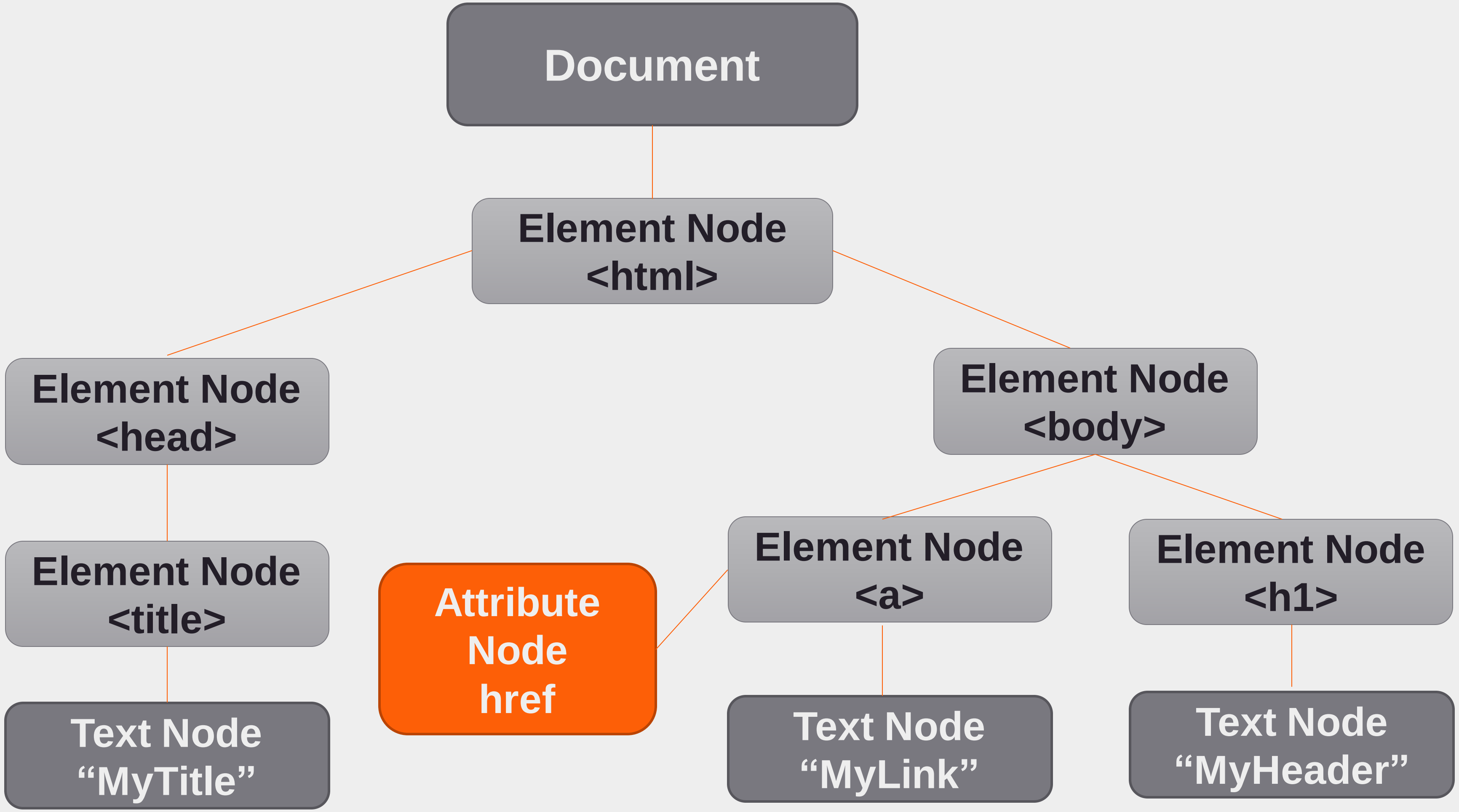
HTML DOM

- Defines a standard to access and manipulate HTML pages
- Modify element properties and data within the element
- Enables to dynamically create, manipulate and delete HTML elements
- Platform and language independent

Sample HTML file

```
<html>

        <head>

                <title>MyTitle</title>

        </head>

        <body>
                <h1>My Header </h1>

                <a href="Test.html" >My
                Link </a>

        </body>
</html>
```

DOM Structure



Document

Element Node <html>

Element Node <head>

Element Node <body>

Element Node <title>

Attribute Node href

Element Node <a>

Element Node <h1>

Text Node “MyTitle”

Text Node “MyLink”

Text Node “MyHeader”

Node Types

Element type	Description	NodeType
Element Node	Represents an element.	1
Attribute Node	Represents an attribute.	2
Text Node	Represents textual content in an element or attribute.	3
Comment Node	Represents a comment.	8
Document Node	Represents the entire document (the root-node of the DOM tree).	9

Using the Document Object

Writes data to the page

```
<html>
      <body>
             <script type="text/javascript">
             document.write("Hello World!")
             </script>
       </body>
<html>
```

Properties of a node

Property	Description
nodeName	Returns the name of a node Eg: IMG, H1,DIV
nodeType	Returns the type of node as numerical value. Eg: All element nodes : 1 All attribute nodes: 2
nodeValue	Sets or returns the value of a text node Eg: <p>Paragraph data</p>

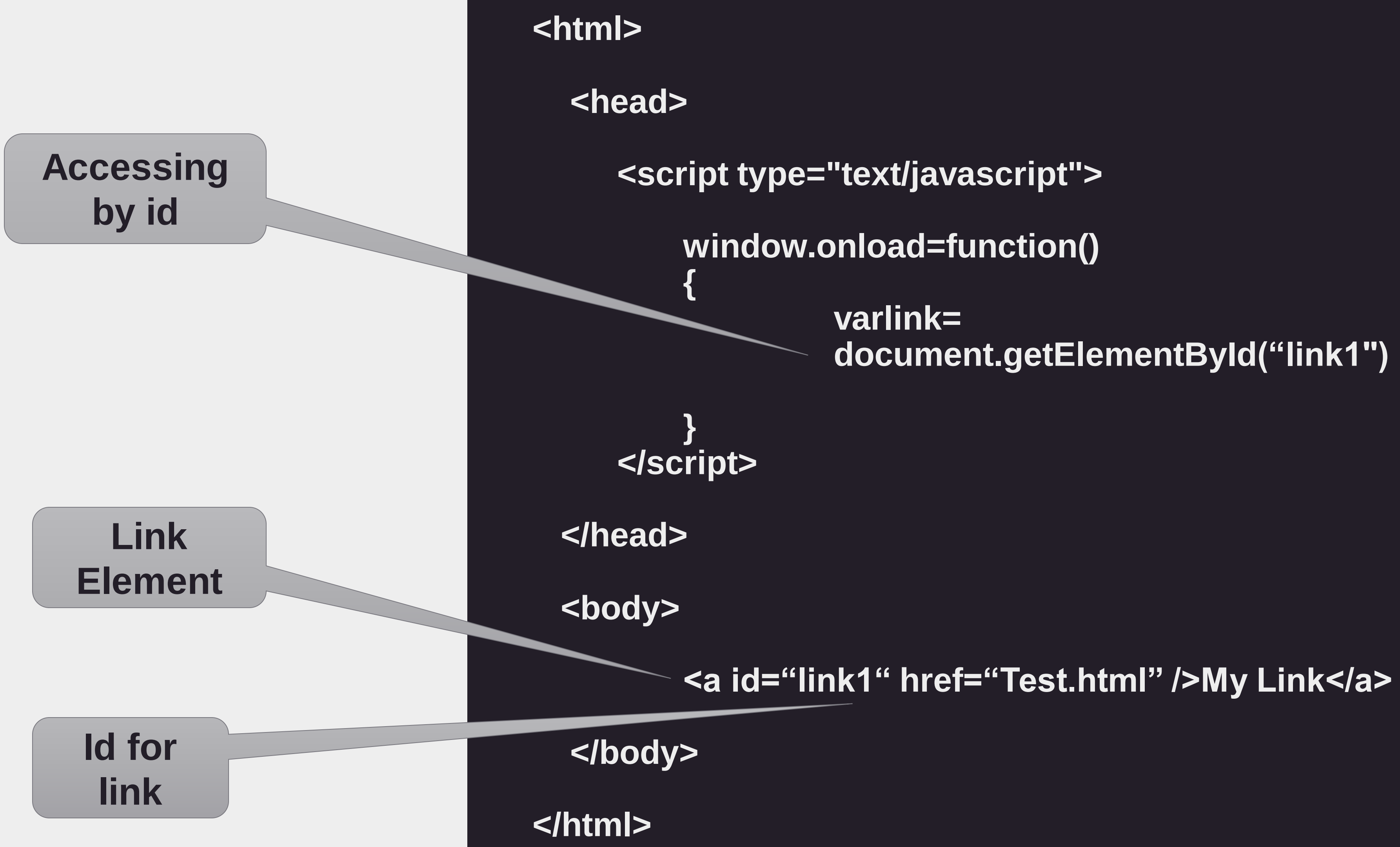
Different Ways of Selecting Elements

- document.getElementById(‘id’);

- document.getElementsByName(‘name’);

- document.getElementsByTagName(‘tagname’);

Use getElementById()



Accessing by id

Link Element

Id for link

```
<html>

  <head>

     <script type="text/javascript">

        window.onload=function()
        {
                 varlink=
                 document.getElementById(“link1")

        }
     </script>

  </head>

  <body>

        <a id=“link1“ href=“Test.html” />My Link</a>

  </body>

</html>
```

Use getElementsByName()

Array of input elements

Accessing by name

Name attribute

```
<head>
        <script type="text/javascript">

        window.onload=function() {

         var inputs
              =document.getElementsByName(“course)

              }

          </script>

</head>

<body>
              <input type = “text” name=“course” />
              <input type = “text” name=“course” />
 </body>
```

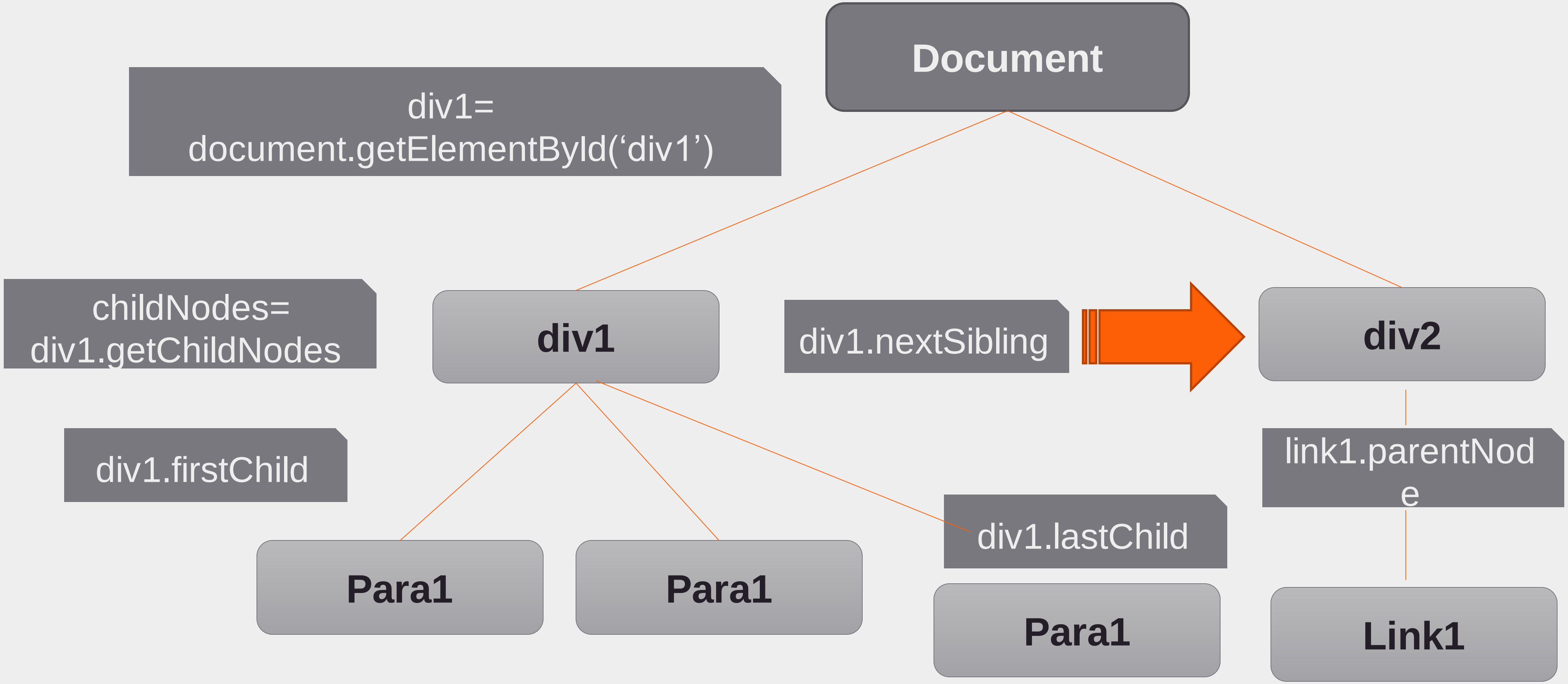
Use getElementsByTagName()

Array of para elements

Accessing by tag name

```
<html>

    <head>
         <script type="text/javascript">

                window.onload=function()
                {

                                var
allParas=document.getElementsByTagName(“p")


                }

         </script>

   </head>

   <body>

                <p> Paragraph 1</p>
                <p> Paragraph 2</p>
                <p> Paragraph 3</p>

    </body>

</html>
```

DOM Structure



div1= document.getElementById(‘div1’)

childNodes= div1.getChildNodes

div1.firstChild

div1.nextSibling

div1.lastChild

link1.parentNode

More Properties of a node

Property	Description
parentNode	Returns the parent node
childNodes	Returns an array of child nodes
firstChild	Returns the first child of a node
lastChild	Returns the last child of a node
previousSibling	Returns the previous node at the same level
nextSibling	Returns the next node at the same level

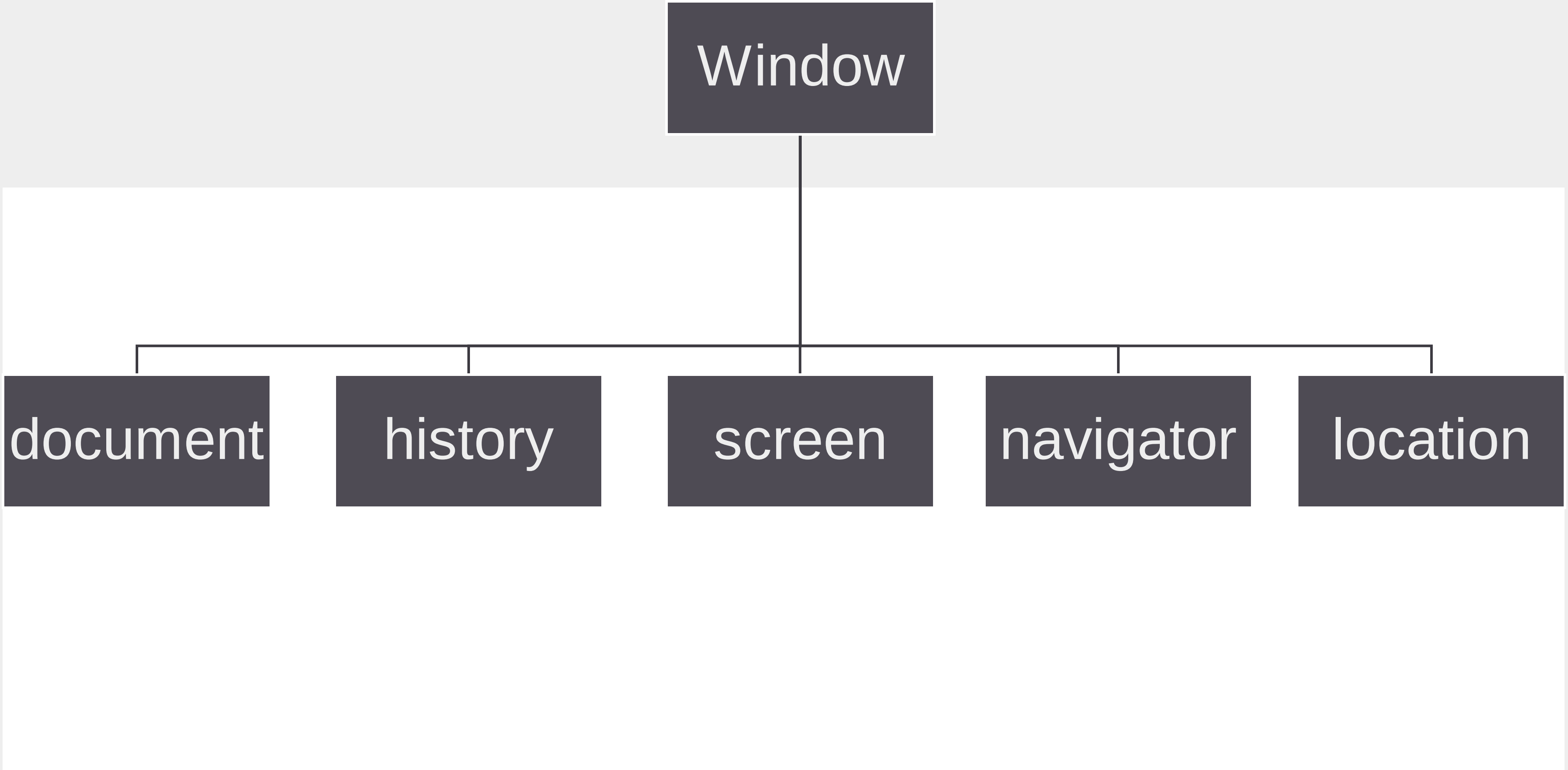
Methods to create nodes

Method	Description
createElement(‘tag’)	Creates new Element with the given tag
createTextNode(‘data’)	Creates a text node with some data
appendChild(newChild)	Appends a new child node to a parent node
insertBefore(newChild,referenceNode)	Inserts a new child node before a specified, existing, child node
replaceChild(newChild, oldChild)	Replaces a child node

Other Methods of a node

Method	Description
removeChild(Child)	Removes a child node
hasChildNodes()	Returns true/false if a node has any child nodes
hasAttributes()	Returns true/false if a node has any attributes
setAttribute(‘prop’,’value’)	Sets the attribute
getAttribute(‘prop’)	Gets the value for a attribute
removeAttribute(‘prop’)	Removes an attribute

Overview - Browser Object Model



Browser Objects

- Window object
  - Represents window in browser
  - Global Object of browser
  - Navigation & Opening windows
  - System dialogs
  - Intervals and Timeouts
  - Window sizing & positioning

- History object
  - Represents user navigation history
  - Each browser window, tab & frame has its own history object
  - Can navigate backward or forward without knowing exact url

Browser Objects continued..

- Screen object
  - Information about user screen like width, height etc
  - Its properties are different for different browsers
  - Less programmatic use

- Navigator object
  - Information about client’s browser like name, version, platform etc
  - Detecting plugins

- Location object
  - Information about current page
  - Redirect to new page

Summary : Session

With this we have come to an end of our session, where we discussed :

- DOM representation of HTML file
- Node Types and different ways of selecting elements
- Properties and methods of a node

At the end of this session, we expect you to :

- Understand concepts related to HTML DOM & its API
- Apply these concepts as per requirement

Appendix

- References
- Key Contacts

Reference Material : Books

- **Head First JavaScript Programming**
  - By: Eric T. Freeman; Elisabeth Robson
  - Publisher: O'Reilly Media, Inc.
- **Professional: JavaScript® for Web Developers**
  - By: Nicholas C. Zakas
  - Publisher: Wrox

Key Contacts

Persistent University

Tarun Kr. Joshi

tarun_joshi@persistent.co.in

Priya Singh

priya_singh@persistent.co.in

Shubhangi Kelkar

shubhangi_kelkar@persistent.co.in

Thank you!

Persistent Interactive | Persistent University