

BOM

BROWSER OBJECT MODEL

(non-standart)

WINDOW

ГЛОБАЛЬНЫЙ ОБЪЕКТ

```
var someGlobVar = "hey";

console.log(someGlobVar);
console.log(window.someGlobVar);

window.anotherGlobVar = "blah";

console.log(window.anotherGlobVar);
console.log(anotherGlobVar);

console.log(window.window.window.anotherGlobVar);
```

```
console.log(window.setTimeout);  
console.log(window.Array);  
console.log(window.console.log);  
console.log(window.location);
```

LOCATION

```
console.log(location.href);  
console.log(location.toString());  
  
console.log(`Check out page href: ${location.href}`);  
console.log(`Check out page location: ${location}`);
```

```
location.origin  
location.protocol  
location.host           // with port number  
location.hostname      // w/o port number  
location.port  
location.pathname      // ?name=prop&name2=prop2  
location.hash
```

```
let ctrl = {
  somePerson: {
    name: "Bob",
    age: 20
  },
  anotherPerson: {
    name: "Sam",
    age: 30
  }
}

let hash = location.hash.slice(1);
ctrl.hasOwnProperty(hash) && renderPerson(ctrl[hash]);
```



```
// just for example

function renderPerson(data) {
  let $name = document.createElement("p"),
      $age = document.createElement("p");
  $body = document.getElementsByTagName("body")[0];

  $name.textContent = `Name is ${data.name}.`;
  $age.textContent = `Age is ${data.age}.`;
  $body.appendChild($name);
  $body.appendChild($age);
}
```

```
location.hash = "anotherPerson";

let hashChangeListener = () => {
  let hash = location.hash.slice(1);

  console.log(hash);
};

window.onhashchange = hashChangeListener;

// or

window.addEventListener("hashchange", hashChangeListener);
```


SCREEN

```
screen.width;  
screen.height;
```

```
screen.availWidth;  
screen.availHeight;
```

```
screen.pixelDepth
```


NAVIGATOR

```
console.log(navigator);
```

```
navigator.cookieEnabled;
```

```
navigator.doNotTrack;
```

```
navigator.language;
```

```
navigator.onLine;
```

```
navigator.platform;
```

HISTORY

DOM

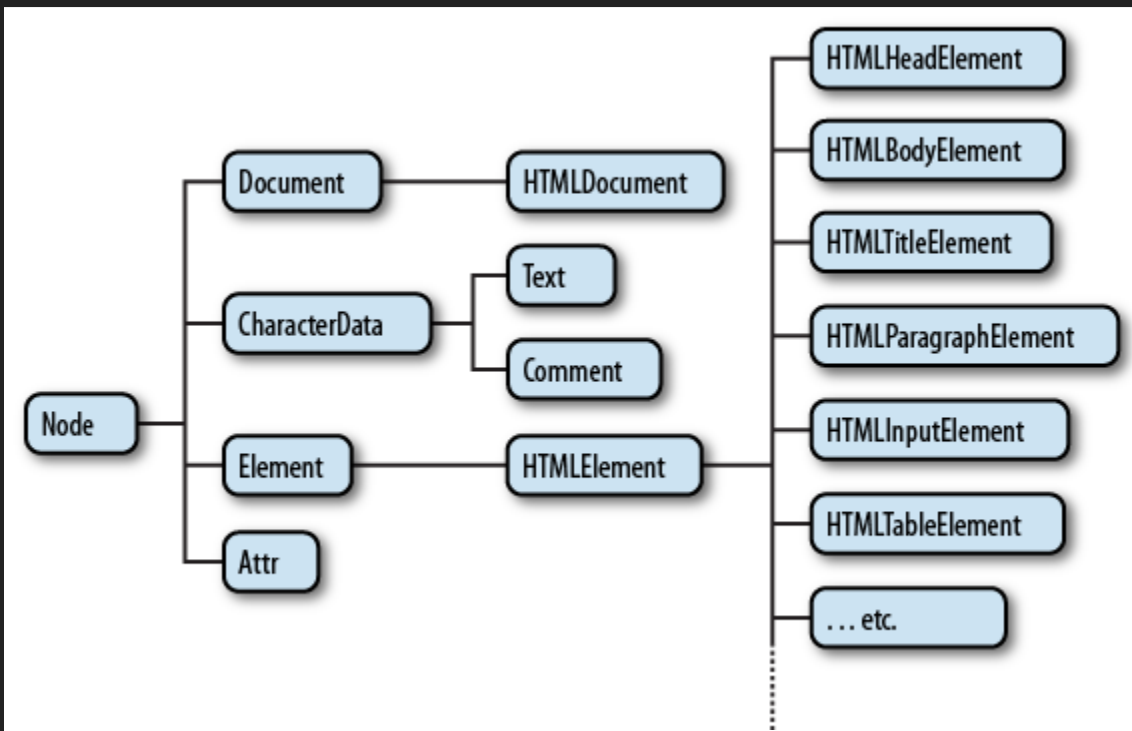
DOCUMENT OBJECT MODEL

The Document Object Model (DOM) is an application programming interface (API) for valid HTML and well-formed XML documents. It defines the logical structure of documents and the way a document is accessed and manipulated.

With the Document Object Model, programmers can build documents, navigate their structure, and add, modify, or delete elements and content. Anything found in an HTML or XML document can be accessed, changed, deleted, or added using the Document Object Model [...]

```
<ul>
  <li><a href="#">Some link</a></li>
  <li><a href="#">Some link</a></li>
</ul>
```

```
Node (ul)
|
|-Node (li)
| |
| |-Node (a)
|
|-Node (li)
| |
| |-Node (a)
```



Полезные ссылки

- [Modernizr](#)
- [Introduction to the DOM \(MDN\)](#)
- [Руководство по DOM \(MDN\)](#)
- [DevDocs/dom](#)