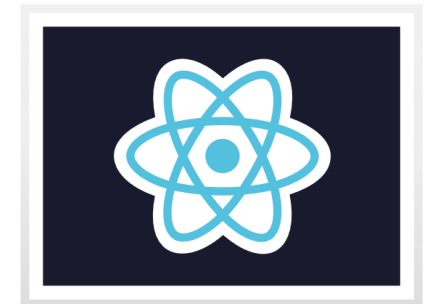
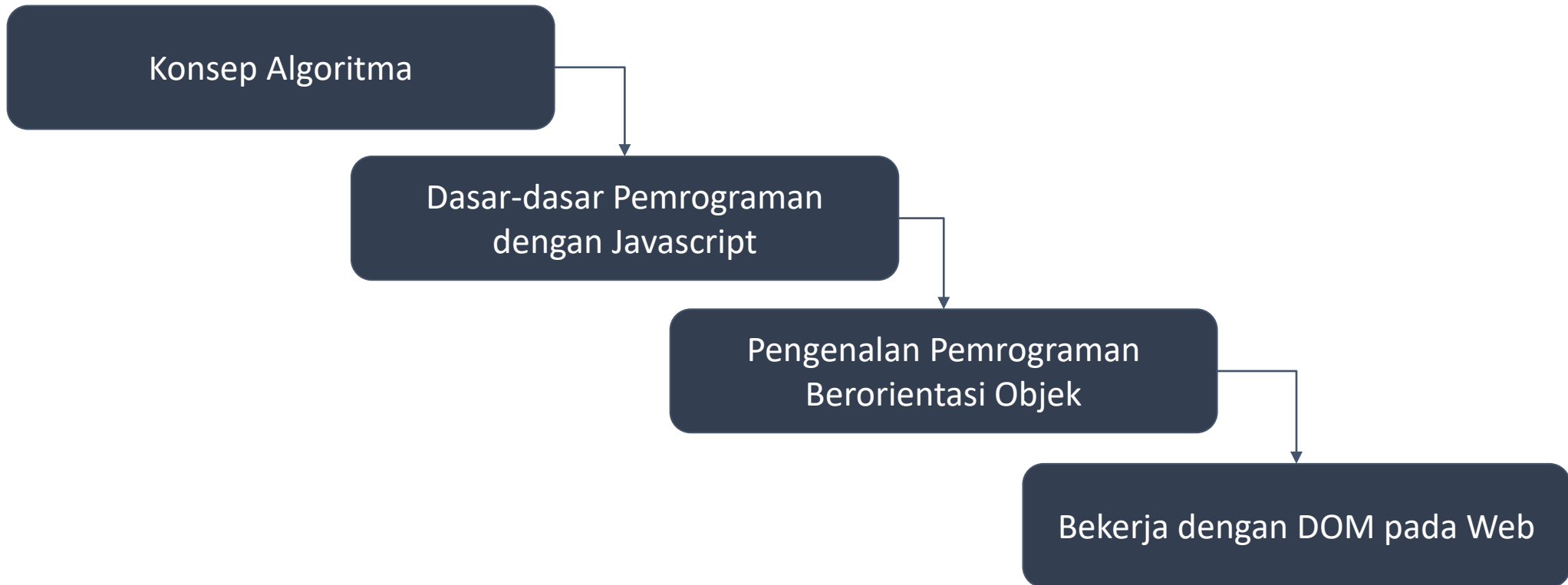


PEMROGRAMAN JAVASCRIPT



- **KODE** : MK03
- **MATAKULIAH** : PEMROGRAMAN JAVASCRIPT
- **SKS** : 4 SKS
- **SEMESTER** : 1 (SATU)



"Keberhasilan bukanlah milik orang yang pintar, keberhasilan adalah kepunyaan mereka yang senantiasa berusaha."

- BJ. Habibie -



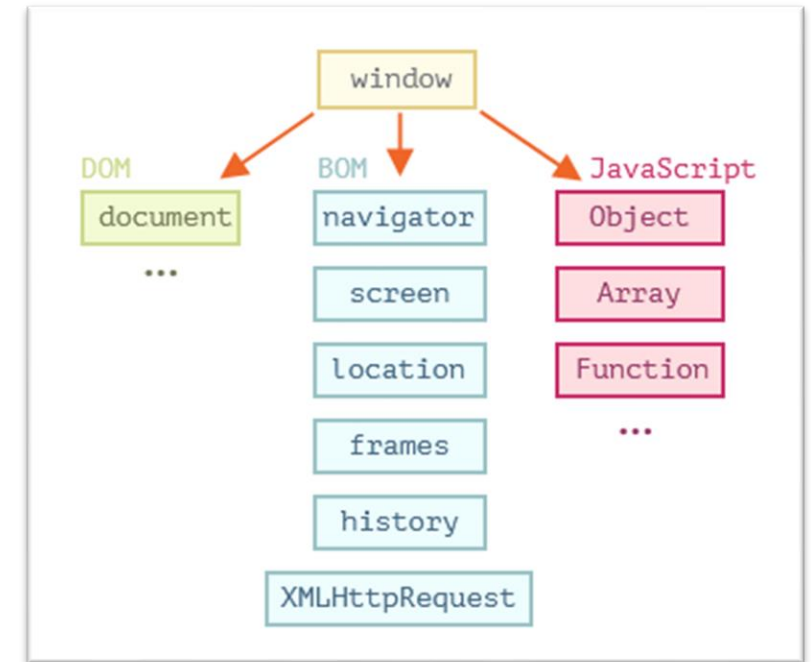
WORK WITH DOCUMENT

Ekosistem Browser

Pada browser terdapat objek root bernama window

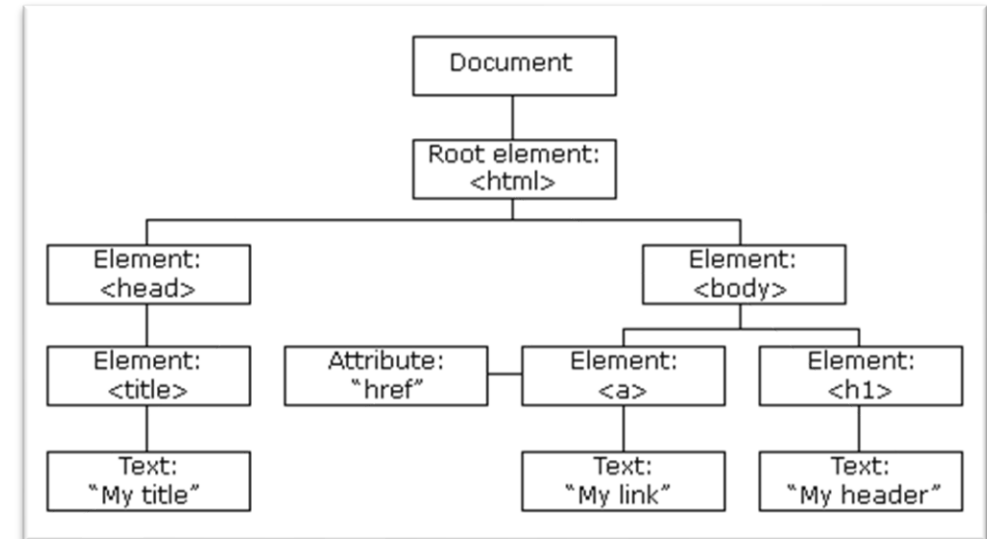
Fungsinya :

- Global objek untuk kode javascript
- Merepresentasikan browser window dan menyediakan metode untuk mengontrolnya



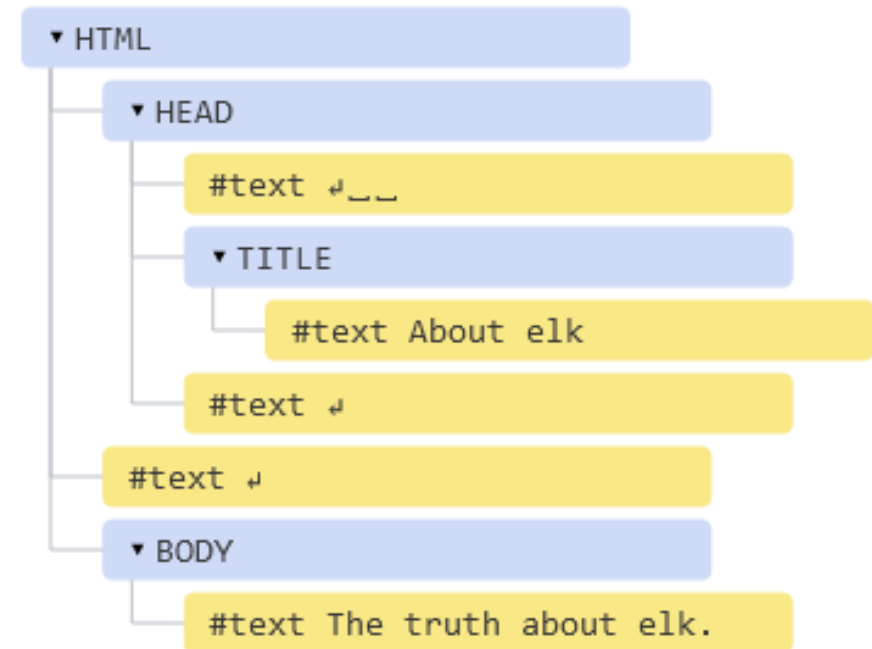
Document Object Model (DOM)

Saat halaman web dibuka, browser membuat HTML DOM dari halaman tersebut berupa objek. DOM memrepresentasikan semua konten halaman yang dapat dimodifikasi. Dengan DOM, Javascript mampu membuat HTML yang dinamis.



BROWSER DOM

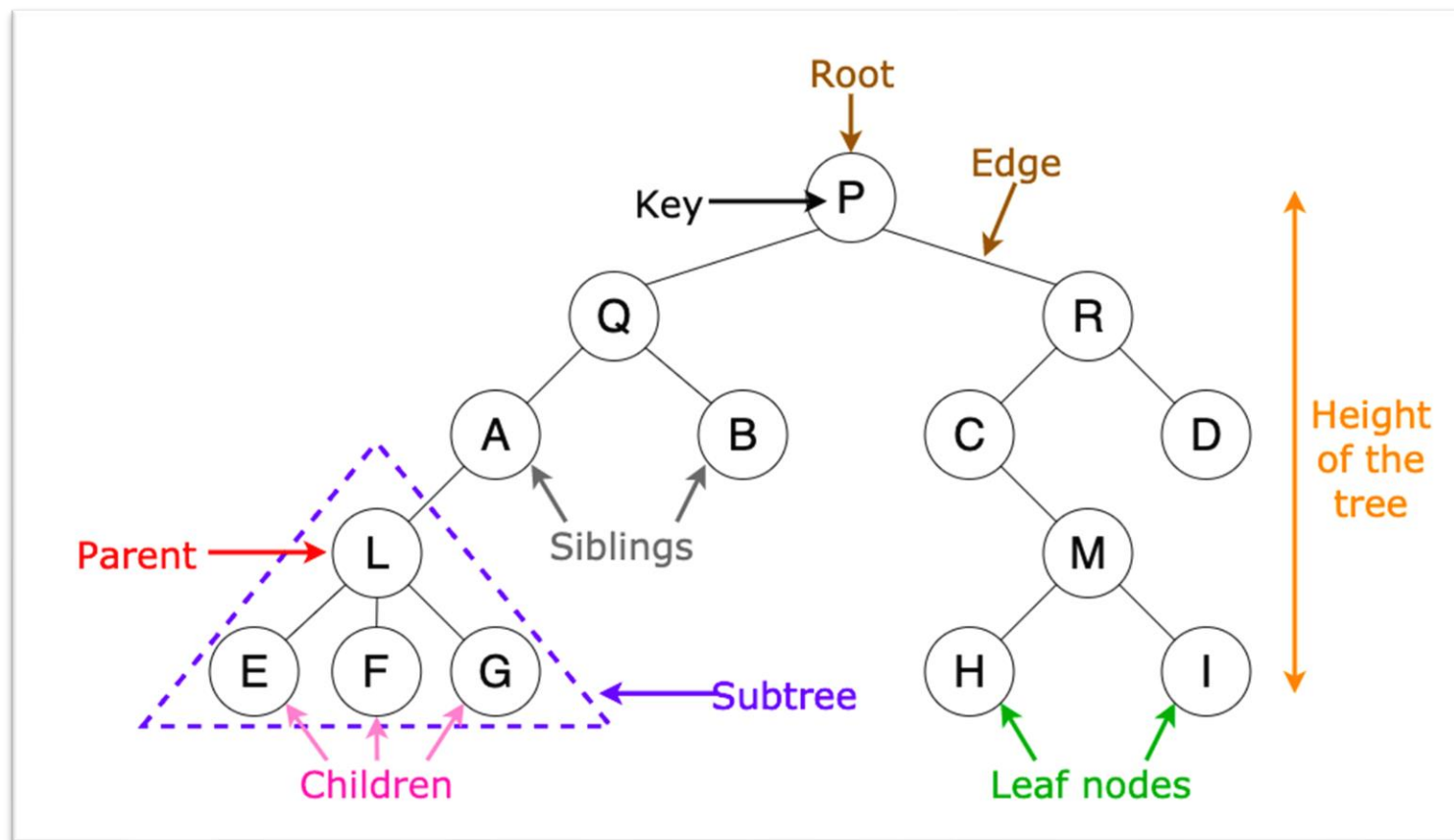
```
1 <!DOCTYPE HTML>
2 <html>
3 <head>
4   <title>About elk</title>
5 </head>
6 <body>
7   The truth about elk.
8 </body>
9 </html>
```



Navigasi HTML DOM

- Children
 - `childNodes`
 - `firstChild`
 - `lastChild`
- Ancestors
- Descendants
- Sibling
 - `nextSibling`
 - `previousSibling`
- Parent

BROWSER DOM



CLASS JAVASCRIPT

Kode HTML

```
<html>
<body>
  <div>Begin</div>
  <ul>
    <li>Information</li>
  </ul>
  <div>End</div>
</body>
</html>
```

Tampilan

Begin

- Information

End

Objek Body

```
> body = document.body
< ▼ <body>
  <div>Begin</div>
  ▼ <ul>
    ► <li>...</li>
  </ul>
  <div>End</div>
</body>

> typeof body
< "object"

> |
```

Document Object Model

DOM level 1 mendefinisikan 11 objek HTML, koleksi, dan properti (tahun 1998).

Masih valid sampai HTML5 yang digunakan sekarang.

Pada DOM level 3, ada beberapa objek, koleksi, dan properti yang ditambahkan

Properti	Keterangan	DOM
document.anchors	Returns all <a> elements that have a name attribute	1
document.applets	Deprecated	1
document.baseURI	Returns the absolute base URI of the document	3

DOM JAVASCRIPT

Properti	Keterangan	DOM
document.body	Returns the <body> element	1
document.cookie	Returns the document's cookie	1
document.doctype	Returns the document's doctype	3
document.documentElement	Returns the <html> element	3
document.documentMode	Returns the mode used by the browser	3
document.documentURI	Returns the URI of the document	3
document.domain	Returns the domain name of the document server	1
document.domConfig	Obsolete	3
document.embeds	Returns all <embed> elements	3

HTML DOM Method

- HTML DOM method adalah aksi yang dapat dilakukan ke elemen HTML
- HTML DOM properties adalah nilai dari elemen HTML yang dapat diubah/diatur
- Semua elemen HTML (tag) didefinisikan sebagai objek. Tag di dalamnya adalah objek turunan (children) dari objek tag yang mengurung.
- Contoh : `document.body` merepresentasikan `<body>` di dalam dokumen `<html>`

Mencari Element HTML

Beberapa method untuk mencari elemen HTML :

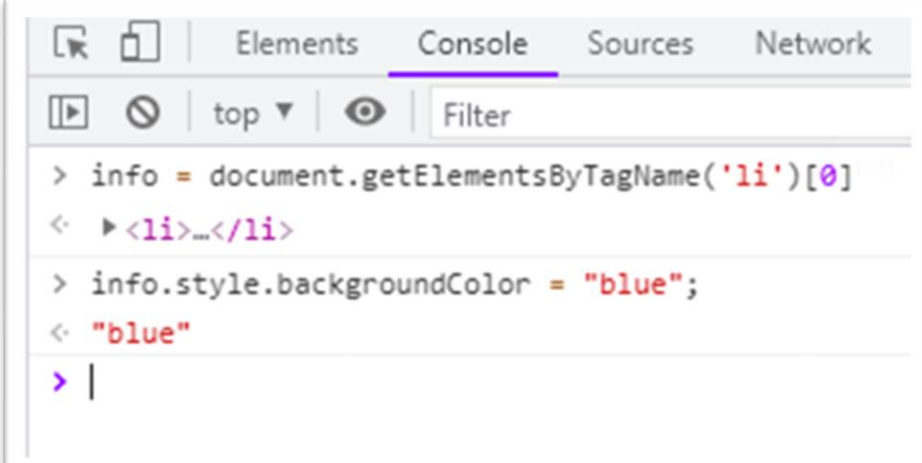
- `document.getElementById(id)`
- `document.getElementsByTagName(name)`
- `document.getElementsByClassName(name)`

Note, pencarian selain dengan Id menghasilkan collection of object HTML (lihat dari s pada `getElements`)
Bisa dipilih dengan kurung siku ke indexnya

```
> document.getElementsByTagName('li')
< HTMLCollection [li] 1
  ▶ 0: li
    length: 1
    __proto__: HTMLCollection
> document.getElementsByTagName('li')[0]
< <li>
  ::marker
  "Information"
  </li>
>
```

Mengubah Element HTML

- `element.innerHTML = <new html content>`
- `element.<attribute> = <new value>`
- `element.style.<property> = <new style>`
- `element.setAttribute(<attribute>, <value>)`



The screenshot shows a web browser's developer console with the 'Console' tab selected. The console displays the following JavaScript code and its output:

```
> info = document.getElementsByTagName('li')[0]
< ▶ <li>...</li>

> info.style.backgroundColor = "blue";
< "blue"

> |
```


Inner HTML

Properti innerHTML mengizinkan kita mengakses isi HTMLnya dalam bentuk string. Kita juga bisa memodifikasinya. Ini adalah cara paling powerful untuk mengubah konten halaman. Contoh :

```
document.body.innerHTML = 'The new BODY!';
```

Akan mengubah teks di dalam document.body ke teks baru tersebut.

textContent

Properti innerHTML menginput teks ke halaman sebagai objek HTML (masih memungkinkan ada tag di dalamnya). Lebih aman jika menggunakan properti textContent karena teksnya diperlakukan sebagai teks murni (pure).

Misalkan ada menu input seperti ini

```
let name = prompt("What's your name?", "");
```

```
elem.textContent = name;
```

textContent

Jika user memasukkan tag html, maka tag itu akan terakomodir di halaman kita (di objek elem). Tetapi jika menggunakan `elem.textContent = name;` tagnya akan dianggap sebagai bagian dari string.

OuterHTML

Properti Outer HTML tidak mengubah elemennya, tetapi mengganti elemen tersebut di DOM (elemen lamanya tetap ada)

OuterHTML

```
<div>Hello, world!</div>
```

```
<script>
```

```
let div = document.querySelector('div');
```

```
// ganti div.outerHTML dengan <p>...</p>
```

```
div.outerHTML = '<p>A new element</p>'; // (*)
```

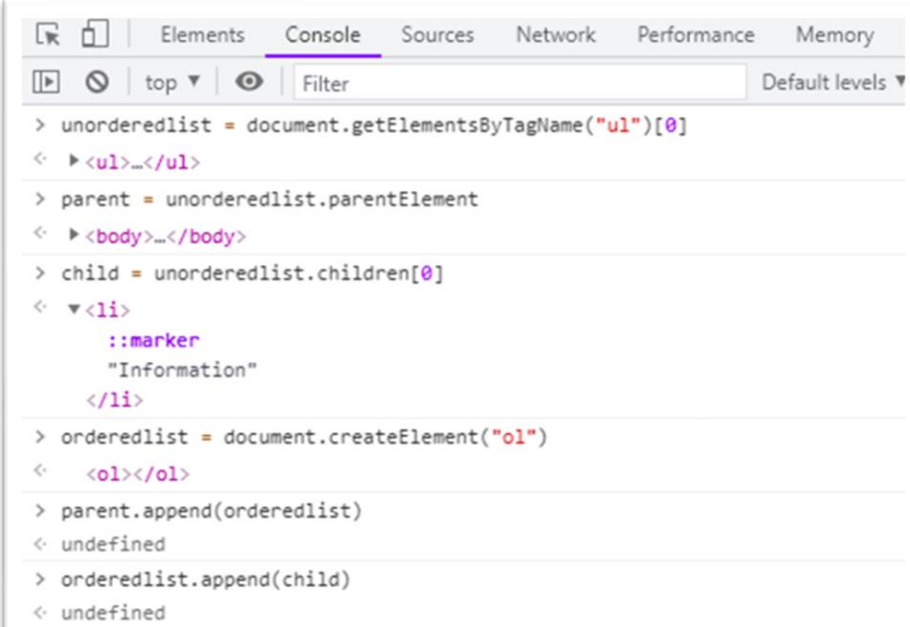
```
// ternyata div yang lama tetap bisa diakses
```

```
alert(div.outerHTML); // <div>Hello, world!</div> (**)
```

```
</script>
```

Mengganti dan Menghapus Elemen

- `document.createElement(element)`
- `document.removeChild(element)`
- `document.appendChild(element)`
- `document.replaceChild(new, old)`
- `document.write(text)`



```
> unorderedlist = document.getElementsByTagName("ul")[0]
< > <ul>...</ul>

> parent = unorderedlist.parentElement
< > <body>...</body>

> child = unorderedlist.children[0]
< > <li>
  ::marker
  "Information"
</li>

> orderedlist = document.createElement("ol")
< > <ol></ol>

> parent.append(orderedlist)
< > undefined

> orderedlist.append(child)
< > undefined
```

Example/Contoh :

```
const element = document.getElementById("artikel");
```

Jika elemennya ditemukan, maka metode ini akan mengembalikan elemennya sebagai objek di variabel elemen. Jika tidak ditemukan, maka akan mengembalikan null.

Query Selector

Untuk menemukan elemen HTML yang memenuhi CSS selector tertentu (id, class name, types, attributes, values of attributes, dll) bisa menggunakan method.

- `querySelector()` // untuk menemukan sebuah elemen
- `querySelectorAll()` // untuk menemukan semua elemen

Contoh, untuk mencari semua elemen `<p>` dengan kelas = 'artikel'

```
const x = document.querySelectorAll("p.artikel");
```


GetElementById* & Query Selector

Semua metode `getElementsBy` mengembalikan live collection, artinya sekumpulan object yang merepresentasikan keadaan dokumennya dan akan terupdate sesuai perubahan yang terjadi pada dokumen.

`querySelector` bersifat sebaliknya, dia mengembalikan static collection, artinya array dari elemen yang bersifat tetap.

GetElementById* & Query Selector

querySelector - Notepad

File Edit Format View Help

```
<div>First div</div>
```

```
<script>
  let divs = document.querySelectorAll('div');
  alert(divs.length); // 1
</script>
```

```
<div>Second div</div>
```

```
<script>
  alert(divs.length); // 1
</script>
```

getElements - Notepad

File Edit Format View Help

```
<div>First div</div>
```

```
<script>
  let divs = document.getElementsByTagName('div');
  alert(divs.length); // 1
</script>
```

```
<div>Second div</div>
```

```
<script>
  alert(divs.length); // 2
</script>
```

querySelector akan mengeluarkan alert 1 dan 1 (pemanggilan length yang kedua nilainya tidak berubah karena variabel divs static), sedangkan getElements menghasilkan 1 dan 2 (terupdate, live collections)

Dynamic HTML

Javascript juga dapat digunakan untuk membuat konten HTML yang dinamis, seperti jam yang menunjukkan waktu sesuai dengan lokasi user.

```
<!DOCTYPE html>
<html>
<body>

<script>
document.getElementById("demo").innerHTML = "Date : " + Date(); </script>

</body>
</html>
```

Dynamic Table

Objek table dibahas khusus

- Dapat dibuat dengan

```
let tbl = document.createElement("table");
```

- Untuk memunculkan elemen table dapat digunakan

```
tbl.rows          // untuk menampilkan semua elemen <tr> di table
```

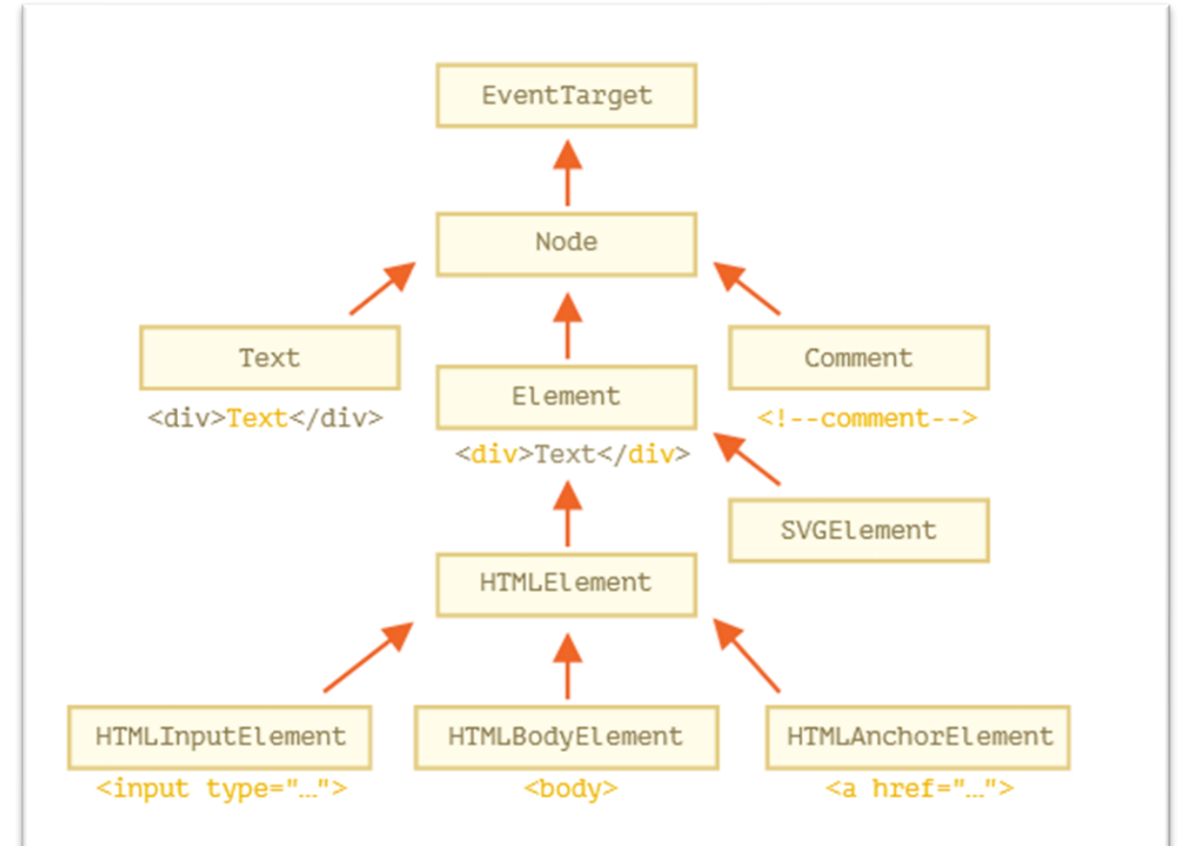
```
tbl.tBodies       // mengembalikan semua elemen <tbody> di table
```

Struktur Inheritance DOM

Semua Node DOM adalah keturunan dari root nodesnya, yaitu EventTarget

Note :

Ini bukan struktur di halaman webnya, tetapi struktur inheritancenya.



TUGAS JAVASCRIPT

1. Buatlah program konversi mata uang pada JavaScript
2. Gunakan Teknik DOM yang telah kalian pelajari
3. Simpan dalam file index.js
4. Upload pada akun github kalian dengan nama repository “Pertemuan-10-JS”



**Terima Kasih
Syukron Katsiron
Arigatou Gozaimasu**