JS
B
jQuery

Daftar Isi

Pendahuluan BAB I

HTML (Hypertext Markup Language) adalah sistem pengkodean suatu dokumen untuk dapat ditampilkan pada halaman Web yang diatur dan distandarisasi oleh World Wide Web Consortium (W3C). Sistem pengkodean dokumen (Markup) ini terdiri dari berbagai simbol yang digunakan untuk menentukan struktur, format dan relasi antar bagian dari suatu dokumen. Yang disebut dengan dokumen disini – pada saat ini – tidak hanya berupa text tetapi dapat berupa berbagai macam sumber multi media.

Sistem pengkodean pada HTML merupakan kumpulan element yang digunakan untuk mengatur dan menampilkan web content. Suatu element ditandai oleh simbol berupa **<** *element_name* **>** yang disebut dengan Tag dan terdiri dari pasangan Opening Tag **<** *element_name* **>** dan Closing Tag **</** *element_name* **>**. Beberapa Element tidak memiliki Closing Tag dan disebut dengan Void Element.

Suatu element terdiri dari Opening maupun Closing Tag yang akan mengapit Content yaitu apa yang akan ditampilkan oleh Browser, dengan syntax **<** *element_name* **>** ….*content*…. **</** *element_name* **>**

Satu element dapat memiliki Attribute dan suatu element dapat berada dalam suatu element yang lain yang disebut dengan Nesting Element.

Sebagai contoh : <p class=”judul”> <strong>Hai</strong> apa kabar ? </p> → merupakan element yaitu terdiri dari opening tag, content dan closing tag dimana terdapat nesting element didalamnya.
Tag <p> dan </p> akan diintepretasikan oleh browser sebagai suatu paragraf karena p digunakan untuk menentukan bahwa format dari content adalah paragraf, dengan demikian content Hai apa kabar ? akan ditampilkan oleh browser sebagai suatu paragraf.
class=”judul” adalah attribute yaitu suatu kelengkapan, dalam contoh ini adalah attribute class dengan nama judul yang ditambahkan pada Tag <p>.
Tag <strong> digunakan untuk menebalkan suatu text sehingga hasil akhir yang akan ditampilkan browser menjadi **Hai** apa kabar ?

HTML sering disebut sebagai bagian dari bahasa pemrograman tetapi sebenarnya HTML tidak memiliki programming logic seperti bahasa pemrograman yang sebenarnya. Jadi yang harus selalu diingat bahwa HTML hanya digunakan untuk membentuk kerangka dari suatu halaman web dengan menggunakan pengkodean seperti dijelaskan di atas. Sedangkan konten atau isi dari halaman web tersebut jika ditempatkan langsung (embeded) pada file html disebut dengan Static Web dan apabila web content berasal dari luar file html tersebut disebut dengan Dynamic Web.

Dengan adanya Dynamic Web kemudian timbul istilah Client-side dan Server-side dimana HTML diposisikan berada pada Client-side karena sepenuhnya digunakan untuk membentuk tampilan yang akan dilihat oleh pengguna. Sedangkan web content nya sendiri berasal dari Server-side misalnya menggunakan PHP.

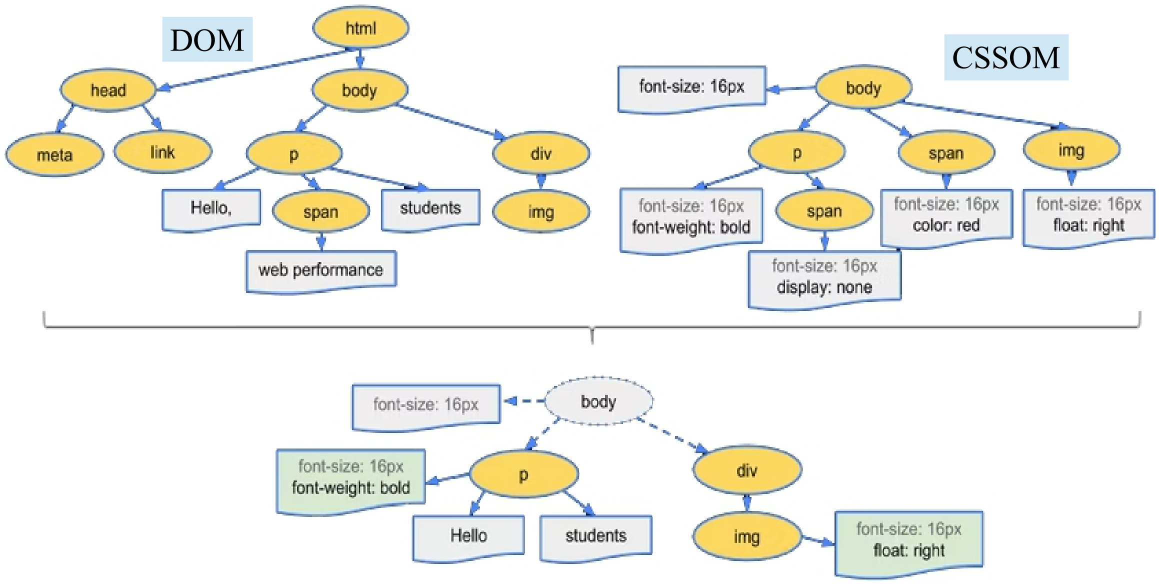
Contoh aplikasi dari Dynamic Web adalah untuk menampikan tabel yang berasal dari suatu database misalnya MySQL. PHP yang berada pada Server-side akan menarik data (Fetching) dari MySQL dan kemudian HTML yang berada pada Client-side akan mengurai (Parsing) data dari PHP tersebut sesuai dengan kerangka yang telah ditentukan dimana dalam contoh ini adalah kerangka dari suatu tabel.

Suatu file HTML yang bisa berada di remote server atau di local disk akan dibaca oleh suatu bagian dari Browser (Chrome, Firefox, Opera dan lain lain) yang disebut dengan Browser Engine. Pada Chrome, browser engine ini dinamakan Blink, Firefox menamakannya dengan Gecko dan Opera menamakannya Presto. Perbedaan antar browser engine dan perbedaan versi pada masing-masing engine dapat menyebabkan perbedaan pada rendering (proses menampilkan konten pada web) terutama pada Dynamic Web.

Browser engine akan melakukan parsing dari file html menjadi suatu unit yang disebut dengan token. Dengan menggunakan contoh di atas, <p class=”judul”> merupakan satu unit token berupa opening tag atau start tag dengan attribute, <strong> merupakan token berupa start tag tanpa attribute, Hai merupakan token berupa text, </strong> merupakan token berupa end tag tanpa attribute, apa kabar ? merupakan token berupa text, </p> merupakan token berupa end tag tanpa attribute.

Setiap token akan disusun dan dikonversi menjadi Java atau JavaScript object yang disebut dengan Node. Kemudian Node akan disusun berdasarkan hirarki yang membentuk Tree Structure dan disebut dengan Document Object Model (DOM).

Setiap element pada HTML seringkali harus diberikan “style” agar tampilannya menarik, untuk itu diperlukan CSS (Cascading Style Sheets). CSS adalah file terpisah dengan ekstensi .css CSS yang merupakan default dari Browser disebut dengan User Agent Stylesheet selanjutnya Browser Engine juga akan melakukan parsing terhadap file CSS dan akan menghasilkan CSS Object Model (CSSOM).



DOM dan CSSOM adalah struktur yang terpisah dimana CSSOM hanya akan mengimplementasikan style yang diperlukan sesuai dengan target Node yang ada pada DOM. Untuk menampilkannya pada layar monitor maka DOM dan CSSOM harus digabungkan dalam satu struktur baru yang disebut dengan Render Tree.

Struktur dari Render Tree hanya berisi node yang akan ditampilkan pada monitor komputer. Jadi jika suatu node memiliki attribute `display:none` maka node tersebut tidak akan ada pada Render Tree. Selanjutnya Browser Engine akan memulai proses menampilkan node yang harus ditampilkan pada layar monitor yang disebut dengan Rendering Sequence.

Rendering Sequence dimulai dengan proses Layout. Pada proses Layout, engine akan menentukan layout dari setiap node yang ada pada Render Tree seperti ukuran node dalam pixel dan posisi nya sesuai dengan Browser Viewport. Viewport adalah luas tampilan layar dan tidak selalu harus sama dengan ukuran layar monitor komputer atau smartphone atau media yang lain. Proses Layout ini dikenal juga dengan istilah Reflow atau Browser Reflow dan dapat juga terjadi jika pengguna melakukan kegiatan seperti scroll, resize window atau terjadi proses manipulasi pada element DOM melalui programming.

Proses berikutnya adalah proses Paint. Karena setiap element pada Render Tree bisa tumpang tindih dan juga bisa memiliki tampilan, posisi dan geometri (animasi) yang berbeda maka browser engine akan membuat layer untuk mengatasi hal tersebut. Selanjutnya adalah proses terakhir yaitu Compositing yaitu setiap layer (tidak selalu dalam keadaan utuh tapi bisa dibagi kedalam beberapa bagian) akan dikirimkan ke GPU (Graphics Processing Unit) untuk menggambarkannya pada layar monitor.

Selain HTML dan CSS, JavaScript juga digunakan untuk menambahkan fitur interaktif pada aplikasi web. Pada proses pembentukan DOM, setiap element pada file html akan diproses satu per satu mulai dari baris pertama hingga baris terakhir. Jika pada proses tersebut terdapat bagian yaitu tag `<script>` untuk menjalankan perintah JavaScript baik internal maupun eksternal maka proses pembentukan DOM akan ditunda sampai perintah JavaScript tersebut selesai diproses. Hal tersebut terjadi karena proses JavaScript tersebut bisa saja menyebabkan perubahan pada struktur DOM. Akses maupun manipulasi element DOM (JavaScript object) dapat dilakukan karena DOM ini merupakan Application Programming Interface (API)

Proses penundaan pembentukan DOM tersebut dapat ditunda sementara dengan menggunakan attribute `async` pada `<script>` tag. Attribute `async` ini hanya berfungsi untuk menunda proses eksekusi script yang berasal dari eksternal file yaitu sampai dengan proses download eksternal file tersebut selesai dilakukan. Begitu proses download selesai maka proses eksekusi script akan dijalankan dan pembentukan DOM akan ditunda.

Proses penundaan pembentukan DOM ini disebut Parser Blocking. Semua embeded script adalah Parser Blocking sementara eksternal script dapat ditunda sementara sampai file eksternal script tersebut selesai di download. Untuk memastikan tidak ada penundaan DOM, dapat menggunakan attribute `defer` pada `<script>` tag. Dengan demikian semua defered script akan dieksekusi sesuai dengan urutannya jika DOM telah selesai dibentuk.
Jadi pilihan menggunakan normal script, asynchronous script atau defered script harus

disesuaikan dengan kebutuhan.

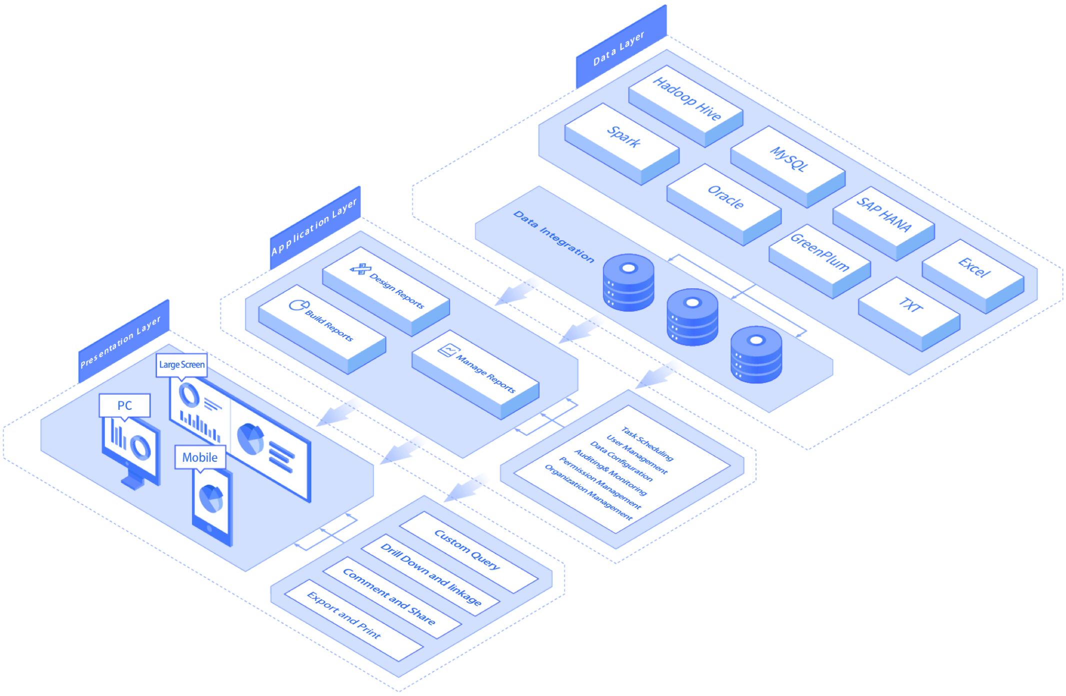
Sebenarnya proses pembentukan CSSOM juga menunda proses DOM secara tidak langsung. Jika browser mendapatkan baris dengan `<style>` tag baik embeded maupun eksternal maka CSSOM akan langsung diubah karena style dari dari suatu element akan mempengaruhi elemen yang hirarkinya berada dibawahnya, berbeda dengan DOM dimana pembentukan Node nya dilakukan secara berurutan satu persatu.

Proses rendering ini perlu dipahami dengan baik karena merupakan bagian dari proses loading suatu halaman web yang merupakan bagian krusial dari pengalaman pengguna ketika mengunjungi suatu situs web.

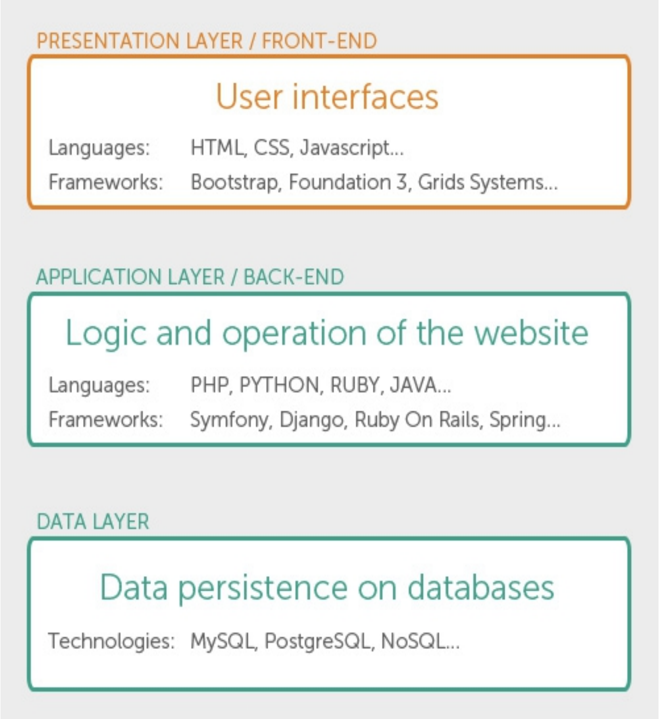
Dalam membangun Web Site dan atau Web Application terdapat berbagai macam metode yang digunakan dan kesemuanya dapat dirangkum dalam suatu bentuk desain dimana proses komputasi yang terjadi dipisahkan kedalam beberapa tingkatan.

Proses komputasi yang terjadi dalam Web Site dan Web Application memiliki perbedaan. Dimana pada masanya Web Site lebih banyak merupakan kumpulan Static Page sedangkan pada saat sekarang Web Site lebih banyak merupakan Web Application karena kebutuhan interaksi dengan pengguna baik dari internal maupun eksternal. Interaksi ini terjadi karena berbagai macam kebutuhan informasi yang diperlukan baik oleh pengguna maupun penyedia jasa.

Pemisahan proses komputasi ini secara umum disebut dengan Multi-tier Architecture atau N-tier Architecture dan yang paling umum digunakan Three Tier Architecture dimana fungsi Presentasi, Processing dan Management Data dibuat terpisah. Ketiga lapisan ini dikenal dengan Presentation Layer, Application Layer dan Data Layer. Pada Application Layer kadang ditambahkan Business Logic Layer.
Pemisahan proses komputasi ini melibatkan baik software maupun hardware.



Desain ini memastikan bahwa setiap layer harus mengikuti alur yang telah ditentukan, artinya presentation layer tidak dapat langsung melakukan akses ke database tanpa melalui application dan data layer. Dengan pemisahan ini developer bisa fokus dalam pengembangan aplikasi per layer karena berkurangnya ketergantungan antar layer selain kemudahan dalam maintenance, yang paling penting dalam aplikasi web adalah masalah keamanan data dimana dengan pemisahan ini kontrol terhadap keamanan data ini dilakukan secara bertingkat.



PRESENTATION LAYER / FRONT-END

User interfaces

Languages: HTML, CSS, Javascript...

Frameworks: Bootstrap, Foundation 3, Grids Systems...

APPLICATION LAYER / BACK-END

Logic and operation of the website

Languages: PHP, PYTHON, RUBY, JAVA...

Frameworks: Symfony, Django, Ruby On Rails, Spring...

DATA LAYER

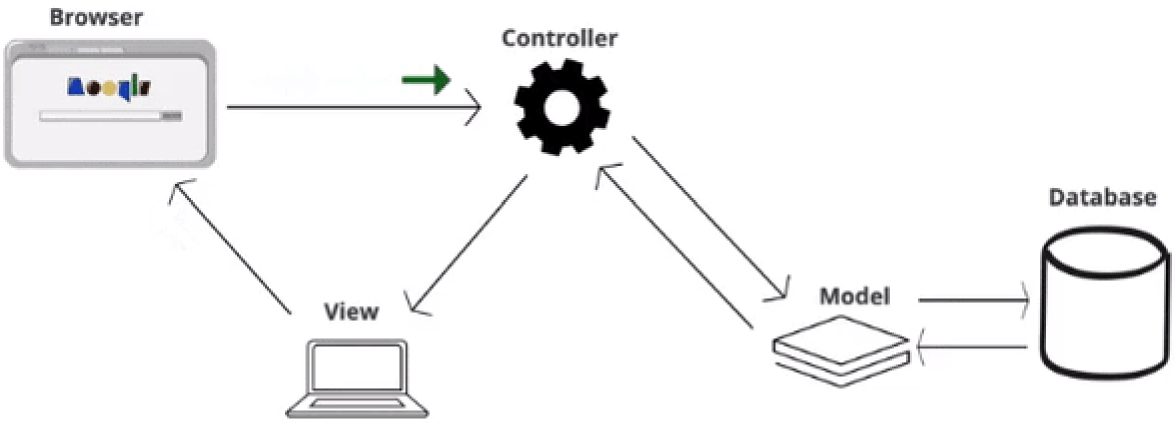
Data persistence on databases

Technologies: MySQL, PostgreSQL, NoSQL...

Dengan adanya pemisahan tersebut maka pada setiap layer juga terjadi pengelompokan pada penggunaan bahasa pemrograman yang digunakan walaupun ada bahasa pemrograman yang dapat digunakan pada setiap layer.

Pengelompokan ini terjadi karena jumlah pengguna dari bahasa pemrograman tersebut lebih banyak pada satu layer tertentu. Contohnya JavaScript sebenarnya juga bisa digunakan di application layer tetapi karena banyaknya penggunaan JavaScript di presentation layer maka saat ini bisa dikatakan digunakan sebagai standard di presentation layer selain HTML dan CSS.

Dalam membangun suatu aplikasi dari sisi software terdapat desain pembangunan aplikasi yang sudah menjadi standard yaitu MVC (Model – View – Controller) Framework. Seperti halnya pemisahan layer di atas, MVC juga menerapkan hal yang sama dimana pembangunan aplikasi dipisahkan menjadi 3 struktur yang saling berhubungan dan secara umum memastikan bahwa proses komputasi mengikuti alur yang telah ditentukan.

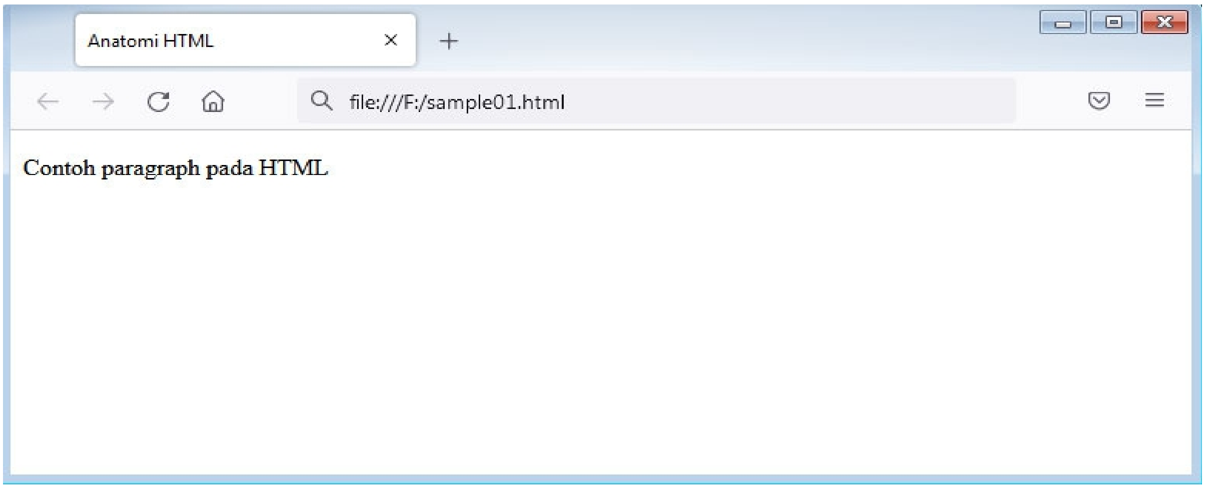


Keuntungan utama dengan menggunakan MVC Framework dalam pembangunan aplikasi adalah pemisahan ini menjadikan aplikasi menjadi bersifat modular dan dengan sifat modular ini maka proses pembangunan dan maintenance menjadi lebih spesifik sehingga penanganan setiap permasalahan menjadi lebih mudah dan dari sisi pengembangan aplikasi bisa menjadi lebih fokus.

A. Anatomi HTML

Secara garis besar anatomi dari HTML dapat diwakili oleh beberapa elemen dibawah ini

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Anatomi HTML</title>
  </head>
  <body>
    <p>Contoh paragraph     pada HTML </p>
  </body>
</html>
```



Contoh paragraph pada HTML

`<!DOCTYPE html>`	tag yang harus ada pada file html dan berada pada baris paling atas, digunakan untuk memastikan bahwa file html ini dapat berfungsi dengan benar terutama untuk cross browser.
`<html> </html>`	Merupakan elemen dimana seluruh elemen pada suatu halaman page berada. Disebut juga dengan Root Element.
`<head> </head>`	Element <head> dapat disebut sebagai container dari berbagai element yang tidak akan ditampilkan pada halaman web.
`<meta charset="utf-8">`	Element ini menentukan bahwa halaman web ini menggunakan character set UTF-8 yang merupakan kumpulan character generic yang dapat dibaca pengguna. Pada saat ini element ini diperlukan untuk menghindari banyak masalah karena perbedaan browser engine.
`<title> </title>`	Element menampikan judul pada browser tab dan juga digunakan sebagai browser bookmark.
`<body> </body>`	Merupakan elemen dimana semua element text, image, video, games dan lain lain yang akan ditampilkan pada halaman web berada.

`<p> </p>`	Menentukan bahwa content merupakan suatu paragraf.
white space	White space merupakan blank space yang terdapat diantara character. Pada contoh diatas terdapat pada : `<meta    charset="utf-8">` dan `<p>Contoh paragraph     pada HTML </p>` white space akan diabaikan oleh browser sehingga akan tersisa hanya satu blank space.

HTML Syntax :

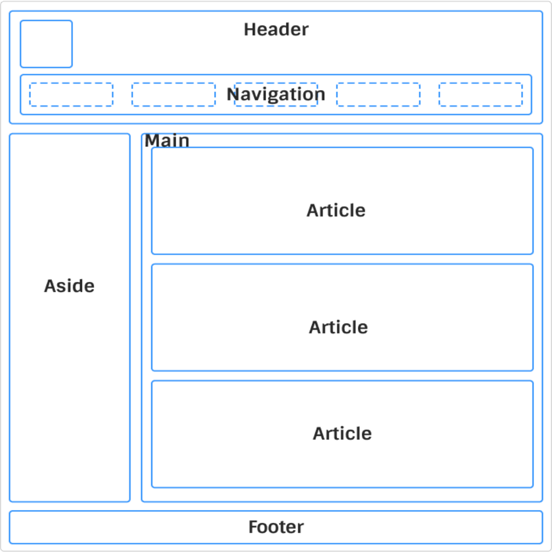
- Setiap element opening tag harus ditutup dengan closing tag kecuali void element.
- Setiap element yang berada didalam element lainnya, opening dan closing tagnya juga harus bersarang didalam element induknya.
- Attribute hanya dapat dituliskan di dalam opening tag.
- Parser Error hanya untuk kesalahan syntax.
- Selalu gunakan huruf kecil (lower case) dalam menuliskan elemen dan attribut.
- Attribute value gunakan tanda double quote "*value*" hindari penggunaan single quote '*value*'
- Untuk memastikan interpretasi dan indexing oleh search engine :`<meta charset="UTF-8">`
- Untuk responsive web : `<meta name="viewport" content="width=device-width, initial-scale=1.0">`
- Tanpa `<body>` dapat menyebabkan error pada browser lama.
- Tanpa `<html>` dan atau `<body>` dapat menyebabkan crash pada DOM dan XML.
- `<title>` digunakan oleh search engine untuk melakukan indexing

B. HTML Semantic

HTML Semantic adalah cara pengorganisasian halaman web dengan menggunakan semantic tag sehingga struktur dan layout halaman web mudah dipahami sedangkan pengaturan dan presentasi tampilan halaman web dilakukan dengan menggunakan CSS.

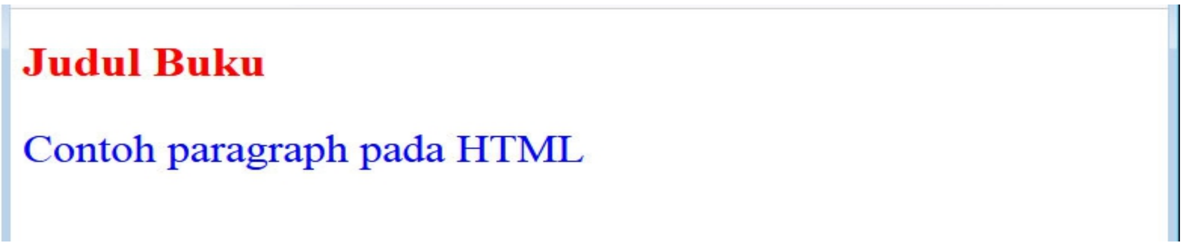
Beberapa contoh Semantic Tag :

```
<article>
<aside>
<details>
<figcaption>
<figure>
<footer>
<header>
<main>
<mark>
<nav>
<section>
<summary>
<time>
```



C. Anatomi CSS

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Anatomi HTML</title>
    <style>
       p,li {
          color: blue;
          font-size: 30px;
       }

        h1 { color: red;}
    </style>
  </head>
  <body>
    <h1>Judul Buku</h1>
    <p>Contoh paragraph pada HTML</p>
  </body>
</html>
```



Judul Buku

Contoh paragraph pada HTML

Pada contoh di atas, CSS berada pada element `<style>` disebut dengan embeded CSS. Jika menggunakan eksternal CSS maka menggunakan element `<link>` seperti contoh ini :

```
<link href="../bootstrap-5.1.3-dist/css/bootstrap.min.css" rel="stylesheet">
```

Eksternal CSS sebaiknya digunakan, sehingga seluruh halaman web dapat memiliki style yang seragam dan lebih efisien dibandingkan embeded CSS pada proses rendering.
Selain itu juga terdapat inline CSS yang sebaiknya dihindari seperti contoh ini :

```
<p style=”color: blue; font-size: 30px;”>Contoh paragraph pada HTML</p>
```

CSS yang digunakan pada contoh di atas yaitu :

```
p, li {
    color: blue;
    font-size: 30px;
}

h1 { color: red;}
```

CSS syntax dimulai dengan Selector yang merupakan HTML Tag yang menjadi target, pada contoh ini adalah `<p>` , `<li>` dan `<h1>` Apabila terdapat lebih dari satu selector dengan style yang sama maka setiap selector dipisahkan dengan delimiter.

Selanjutnya digunakan curly brace `{ }` untuk menentukan Property dan Value dari style yang diperlukan. Property selalu diakhir dengan tanda : dan Value selalu diakhiri dengan tanda ;

Setiap property memiliki value set sendiri. Misal `color` pada `<h1>` di atas, selain bisa menggunakan value `red` juga dapat menggunakan kode RGB dari red yaitu red(255, 0, 0) jadi selain `color: red;` dapat juga ditulis `color: rgb(255, 0,0);`

Pada contoh CSS di atas, seluruh tag `<p>` dan `<li>` akan memiliki `color: blue;` jika kemudian terdapat `<p>` yang dengan warna hijau maka terdapat 2 cara yang dilakukan yaitu
1. menggunakan inline CSS → `<p style="color: green;>` atau
2. dengan menggunakan attribute `class` sehingga dapat digunakan pada tag nya lain.

`.warnamerah {color: green;}` → perhatikan nama class didahului dengan tanda titik.

```
<p class="warnamerah">Paragraph berwarna hijau walaupun nama class nya
warnamerah</p>
```

Suatu CSS class juga dapat diterapkan pada suatu selector

```
p.warnamerah, li.warnabiru {
    color: blue;
    font-size: 30px;
}
```

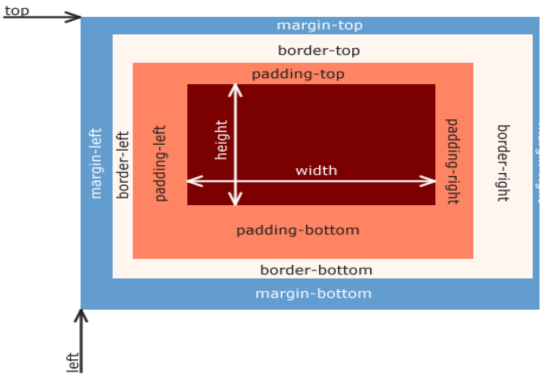
Pada CSS di atas style hanya berlaku pada element `<p class="warnamerah">` dan `<li class="warnabiru">`

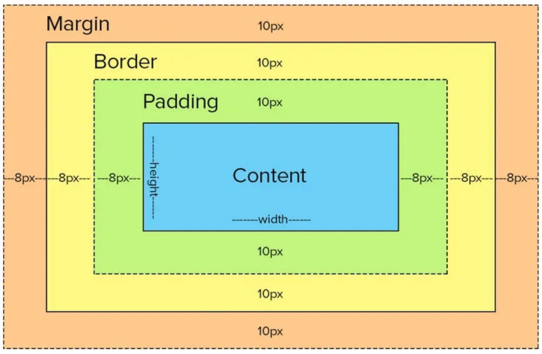
Untuk memberikan style pada suatu element berdasarkan keberadaannya pada suatu dokumen dapat dilakukan 2 cara yaitu :

article p {color: navyblue;} → berlaku untuk semua `<p>` yang berada didalam `<article>`

h1 + p {font-size: 200%;} → berlaku untuk semua `<p>` yang tepat berada dibawah `<h1>`

CSS akan selalu mendefinisikan suatu content berada dalam suatu kotak, jadi perhatikan dan pelajari layout pengaturan CSS di bawah ini.





C. Anatomi JavaScript.

Seperti halnya dengan CSS, JavaScript dapat berada didalam file html atau berada diluar file html dengan hanya menggunakan tag `<script>`.
`<script src="../opt/pilihan.js"></script>` → file html menggunakan script pilihan.js

Contoh script di bawah ini adalah embeded script dan tetap dapat ditempatkan sebagai file external dengan ekstensi js seperti pada contoh pilihan.js di atas.

```
<script type="text/javascript">
   var id_user = '<?php echo isset($_SESSION['user_id']) ? "VIEW" : "EDIT"; ?>';
   document.getElementById("regset").disabled = id_user==='VIEW' ? true : false;
</script>
```

jika menggunakan eksternal file, misal bernama `pilihan.js`

```
pilihan.js
var id_user = '<?php echo isset($_SESSION['user_id']) ? "VIEW" : "EDIT"; ?>';
document.getElementById("regset").disabled = id_user==='VIEW' ? true : false;
```

Selain itu terdapat metode penggunaan JavaScript yang disebut Inline JavaScript Handler seperti contoh di bawah ini :

```
function createParagraph() {
  const para = document.createElement('p');
  para.textContent = 'You clicked the button!';
  document.body.appendChild(para);
}

<button onclick="createParagraph()">Click me!</button>
```

Seperti telah dijelaskan sekilas pada bab sebelumnya, `<script>` memiliki attribut `async` dan `defer`.
Misal pada satu file html memerlukan eksternal script berurutan seperti dibawah ini :

```
<script async src="js/vendor/jquery.js"></script>
<script async src="js/script2.js"></script>
<script async src="js/script3.js"></script>
```

Dengan menggunakan attribute `async` maka selama ketiga file tersebut di download maka proses DOM tidak akan ditunda dan begitu selesai didownload maka proses eksekusi dimulai dengan tidak ada jaminan bahwa proses eksekusi akan berjalan sesuai dengan urutannya. Jadi bisa terjadi `script3.js` akan dieksekusi lebih dulu daripada `jquery.js` padahal pada `scrip3.js` terdapat proses yang memerlukan function yang terdapat pada `jquery.js`

Dengan demikian `async` sebaiknya digunakan jika script harus segera dieksekusi tidak memiliki ketergantungan kepada file script lain dan juga tidak ada proses yang harus dilakukan setelah DOM selesai terbentuk.

```
<script defer src="js/vendor/jquery.js"></script>
<script defer src="js/script2.js"></script>
```

```
<script defer src="js/script3.js"></script>
```

Pada contoh dengan menggunakan `defer` di atas, maka proses akan dipastikan dilakukan sesuai urutannya dan proses eksekusi hanya akan dilakukan setelah DOM selesai dibentuk.

D. HTML Reserved Character.

HTML menggunakan beberapa character dalam syntax code sehingga penggunaan character tersebut secara langsung sebaiknya dihindari.

Character	Entity	Keterangan
&	&amp;	Digunakan sebagai penanda entity
<	&lt;	Digunakan sebagai penanda tag
>	&gt;	Digunakan sebagai penanda tag
"	&quot;	Digunakan sebagai penanda value pada atrribute

Contoh penggunaan :

```
<p>&lt &amp &gt digunakan sebagai penanda Tag</p>
```

Output :

< & > digunakan sebagai penanda Tag

HTML Reference BAB IV

Basic HTML

Tag	Keterangan
`<!DOCTYPE>`	Menentukan tipe dokumen
`<html>`	HTML container `lang | xmlns`
`<head>`	Dokumen metadata container • `<title>` • `<style>` • `<base>` • `<link>` • `<meta>` • `<script>` • `<noscript>`
`<title>`	Document title
`<body>`	Document body
`<h1> ... <h6>`	HTML heading
`<p>`	Paragraf
`<br>`	Single line break
`<hr>`	Memberikan batas tematik pada satu content
`<!--...-->`	Comment

Format

Tag	Keterangan
`<abbr>`	Abbreviation atau Acronym
`<address>`	Contact information dari author/owner dari document/article
`<b>`	Bold text
`<bdi>`	Text yang berada didalam tag <bdi> akan diisolasi agar tidak terjadi perubahan arah penulisan text (dari kanan ke kiri atau dari kiri ke kanan)
`<bdo>`	Jika normal penulisan text dari kiri ke kanan maka akan dirubah menjadi dari kiri ke kanan
`<big>`	Big text. Non HTML5. Gunakan CSS.
`<blockquote>`	Mengambil quote (kutipan) dari sumber luar.
`<center>`	Centered text. Non HTML5. Gunakan CSS.
`<cite>`	Membuat kutipan, biasanya akan dirender sebagai *italic text*
`<code>`	Biasanya digunakan untuk menampilkan kode syntax dengan menggunakan `monospace` font.
`<del>`	Menampikan ~~text yang dicoret~~

`<dfn>`	Definition element. Menentukan suatu term (istilah) yang dijelaskan pada content
`<em>`	Emphasized (memberi penekanan) pada text
`<font>`	Menentukan font, color dan size dari text. Non HTML5. Gunakan CSS.
`<i>`	Memberikan pembedaan pada text, biasanya ditampilkan dalam bentuk *italic*
`<ins>`	Menampilkan text dengan garis bawah
`<kbd>`	Menampikan text sebagai keyboard input ditampikan dengan monospace font
`<mark>`	Memberi highlight pada text
`<meter>`	Memberikan tampilan meter gauge
`<pre>`	Preformatted text, mempertahankan content sesuai aslinya termasuk white-space
`<progress>`	Memberikan tampilan progress gauge
`<q>`	Membuat kutipan singkat, biasanya akan diberikan tanda “ ”

`<rp>`
`<rt>`
`<ruby>`
	Untuk menampikan text East Asian typography pada browser yang tidak mendukung Ruby Annotation. Digunakan secara bersamaan.
	<ruby>
	漢 <rt> ㄏㄢˋ </rt>
	</ruby>

`<s>`	Membuat ~~text tercoret (strikethrough)~~
`<samp>`	Menampilkan sample output komputer menggunakan monospace font
`<small>`	Smaller text
`<strike>`	Membuat text tercoret. Non HTML5. Gunakan `<del>` atau `<s>`
`<strong>`	Memberi penekanan pada text dengan tampilan bold
`<sub>`	Membuat $_{\text{subscripted text}}$
`<sup>`	Membuat $^{\text{superscripted text}}$
`<template>`	Membuat content yang disembunyikan / tidak ditampilkan pada saat page load
`<time>`	Menentukan specific time atau datetime
`<tt>`	Teletype text. Menampilkan text dengan monospace font. Non HTML5. Gunakan CSS.
`<u>`	Menampilkan text dengan underline
`<var>`	Menampilkan text seperti suatu variabel biasanya dalam bentuk *italic*
`<wbr>`	Word Break Opportunity, menetukan perkiraan posisi untuk membuat line break

Forms dan Input

Tag	Keterangan
`<form>`	Mendefinisikan Form

`<input>`	Membuat input (isian) field
`<textarea>`	Membuat input (isian) berupa text area
`<button>`	Membuat button
`<select>`	Membuat drop-down list
`<optgroup>`	Membuat satu group untuk beberapa option pada drop-down list
`<option>`	Membuat option pada drop-down list
`<label>`	Membuat label dari <input> element
`<fieldset>`	Menjadikan element didalam form menjadi satu group
`<legend>`	Membuat caption untuk <fieldset> element
`<datalist>`	Membuat pre-defined option untuk input controls
`<output>`	Menampilkan hasil kalkulasi

```
<form oninput="x.value=parseInt(a.value)+parseInt(b.value)">
  <input type="range" id="a" value="50">
  +<input type="number" id="b" value="25">
  =<output name="x" for="a b"></output>
</form>
```

Image

Tag	Keterangan
`<img>`	Menentukan image yang akan ditampikan

`<map>`
`<area>`
	Menentukan suatu image menjadi suatu area dengan koordinat tertentu

```
<img src="workplace.jpg" alt="Workplace" usemap="#workmap" width="400"
height="379">
<map name="workmap">
  <area shape="rect" coords="34,44,270,350" alt="Computer"
href="computer.htm">
  <area shape="rect" coords="290,172,333,250" alt="Phone" href="phone.htm">
  <area shape="circle" coords="337,300,44" alt="Cup of coffee"
href="coffee.htm">
</map>
```

`<canvas>`	Digunakan untuk membuat grafis secara interactive melalui script

`<figcaption>`
`<figure>`
	<figure> <img> dan <figcaption> contoh di bawah ini merupakan satu kesatuan element. Menghasilkan hal yang berbeda jika hanya menggunakan <img>

```
<figure>
  <img src="pic_trulli.jpg" alt="Trulli" style="width:100%">
  <figcaption>Fig.1 - Trulli, Puglia, Italy.</figcaption>
</figure>
```

`<picture>`	Membuat container dari kumpulan image, biasanya digunakan pada responsive design untuk menyesuaikan dengan viewport
`<svg>`	Membuat container untuk tampilan SVG(Scalable Vector Graphic)

Audio / Video

Tag	Keterangan
`<audio>`	Menentukan sumber audio
`<source>`	Menentukan sumber dari media elements (<video>, <audio> dan <picture>)

`<track>`

Menentukan sumber subtitle (<video> dan <audio>)

`default | kind | label | src | srclang`

`<video>`

Menentukan sumber video

`autoplay | controls | height | loop | muted | poster | preload | src | width`

Link

Tag

Keterangan

`<a>`

Membuat hyperlink

`download | href | hreflang | media | ping | referrerpolicy | rel | target | type`

`<link>`

Membuat link ke external resource

`crossorigin | href | hreflang | media | referrerpolicy | rel | sizes | type`

`<nav>`

Membuat group dari hyperlink

```
<nav>
  <a href="/html/">HTML</a> |
  <a href="/css/">CSS</a> |
  <a href="/js/">JavaScript</a> |
  <a href="/python/">Python</a>
</nav>
```

List

Tag

Keterangan

`<ul>`

Membuat unordered list yaitu bullet list

```
<ul>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>

</ul>
```

Output :

```
    •   Coffee
    •   Tea
    •   Milk
```

`<ol>`

Membuat ordered list yaitu list dengan nomer urut.

```
<ol>
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>

<ol start="50">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ol>
```

Output :

```
    1. Coffee
    2. Tea
    3. Milk

    50. Coffee
    51. Tea
    52. Milk
```

`<li>`	Membuat item dari list
`<dir>`	Menentukan directory list. Non HTML5. Gunakan `<ul>`

`<dl>`
`<dt>`
`<dd>`
	Membuat description list

```
<dl>
  <dt>Coffee</dt>
  <dd>Black hot drink</dd>
  <dt>Milk</dt>
  <dd>White cold drink</dd>
</dl>
```

Output :

```
Coffee
    Black hot drink
Milk
    White cold drink
```

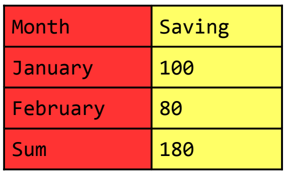
Tabel

Tag	Keterangan
`<table>`	Mendefinisikan table
`<caption>`	Membuat caption dari suatu table
`<th>`	Menentukan table header
`<tr>`	Menentukan table row
`<td>`	Menentukan table cell
`<thead>`	Membuat group header
`<tbody>`	Membuat group content
`<tfoot>`	Membuat group footer
`<col>`	Menentukan column properti dari setiap colomn yang berada pada `<colgroup>` element
`<colgroup>`	Membuat group column untuk keperluan formatting

```
<table>
  <colgroup>
    <col span="1" style="background-color:red">
    <col style="background-color:yellow">
  </colgroup>
  <thead>
    <tr>
      <th>Month</th>
      <th>Savings</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>January</td>
      <td>100</td>
    </tr>
    <tr>
      <td>February</td>
      <td>80</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td>Sum</td>
      <td>180</td>
    </tr>
  </tfoot>
```

```
</table>
```

Output :



Month	Saving
January	100
February	80
Sum	180

Style dan Semantic

Tag	Keterangan
`<style>`	Menentukan style
`<div>`	Menentukan division yaitu container dari element lain
`<span>`	Membentuk bagian dari suatu text atau dokumen sebagai inline container
`<header>`	Document header
`<footer>`	Document footer
`<main>`	Menentukan bagian utama dari dokumen
`<section>`	Membentuk section pada dokumen
`<article>`	Membentuk article
`<aside>`	Membentuk aside (informasi tambahan) dari dokumen yang ditampilkan
`<details>`	Membuat tanda ▶ yang jika di click akan menampilkan text yang disembunyikan
`<dialog>`	Membuat dialog box

`<summary>`	Membuat header text yang ditampilkan untuk <details> element
	`<details>`
	`  <summary>Epcot Center</summary>`
	`  <p>Epcot adalah theme park yang berada di Walt Disney World Resort.</p>`
	`</details>`
	
	Output :
	▶ `Epcot Center`

`<data>`	Menyediakan data yang dapat digunakan untuk data processing

Meta Info

Tag	Keterangan
`<head>`	Mendefinisikan informasi dari dokumen
`<meta>`	Mendefinisikan metadata dari dokumen
`<base>`	Mendefinisikan default URL dan link dokumen
`<basefont>`	Mendefinisikan default color, size, dan font dokumen. Non HTML5. Gunakan CSS.

Programming

Tag	Keterangan
`<script>`	Mendefinisikan script
`<noscript>`	Mendefinisikan content alternatif untuk user yang browsernya tidak mendukung client-side script
`<applet>`	Mendefinisikan applet (plug-in). Non HTML5. Gunakan <embed> atau <object>
`<embed>`	Mendefinisikan container untuk aplikasi eksternal (non-HTML)
`<object>`	Mendefinisikan embedded object
`<param>`	Mendefinisikan parameter object

HTML Global Attribute

Global attribute adalah attribute yang dapat digunakan diseluruh HTML element.

Attribute **Keterangan**

`accesskey`

Menentukan shortcut key untuk memgaktifkan/focus suatu element

```
<element accesskey="character">

<a href="https://www.example.com/html/" accesskey="h">HTML</a><br>
<a href="https://www.example.com/css/" accesskey="c">CSS</a>
```

`class`

Menentukan satu atau lebih class yang digunakan suatu element

```
<element class="classname">

<style>
  h1.intro {color: blue;}
</style>

<h1 class="intro">Header 1</h1>
```

`contenteditable`

Menentukan content dari suatu element bisa diedit atau tidak.

```
<element contenteditable="true|false">

<p contenteditable="true">Paragraf ini bisa diedit.</p>
```

`data-*`

Mendefinisikan data khusus

```
<element data-*="value">

<ul>
  <li data-animal-type="bird">Owl</li>
  <li data-animal-type="fish">Salmon</li>
  <li data-animal-type="spider">Tarantula</li>
</ul>
```

`dir`

Menentukan arah penulisan content

```
<element dir="ltr|rtl|auto">

<p dir="rtl">Menulis dari kanan ke kiri!</p>
```

`draggable`

Menentukan apakah element bisa ditarik geser

```
<element draggable="true|false|auto">

<p draggable="true">This is a draggable paragraph.</p>
```

`hidden`

Menentukan suatu element tidak ditampilkan ke layar

```
<element hidden> → true

<p hidden>Paragraf ini disembunyikan alias tidak ditampilkan ke layar.</p>
```

`id`

Menentukan ID suatu element

```
<element id="id">

<h1 id="myHeader">Hello World!</h1>
<button onclick="displayResult()">Change text</button>

<script>
function displayResult() {
  document.getElementById("myHeader").innerHTML = "Have a nice day!";
}
</script>
```

`lang` Informasi bahasa yang digunakan pada content

```
<element lang="language_code">

<p lang="fr">Ceci est un paragraphe.</p>
```

`spellcheck` Menentukan apakah suatu element perlu diperiksa spelling dan grammar

```
<element spellcheck="true|false">

<p contenteditable="true" spellcheck="true">This is a paragraph.</p>
```

`style` Mendefinisikan inline CSS style pada element

```
<element style="style_definition">

<h1 style="color:blue;text-align:center;">This is a header</h1>
<p style="color:green;">This is a paragraph.</p>
```

`tabindex` Menentukan tab order dari element, urutan ketika tombol keyboard Tab

```
<element tabindex="index_number">

<a href="https://www.oracle.com/" tabindex="2">Oracle</a>
<a href="http://www.google.com/" tabindex="1">Google</a>
<a href="http://www.microsoft.com/" tabindex="3">Microsoft</a>
```

`title` Memberikan ekstra informasi dari suatu element

`translate` Menentukan apakan suatu content perlu ditranslasi

Attribute	Tag	Keterangan

`accept`	`<input>`
	Khusus `<input type="file">`
		Menenentukan tipe file yang dapat diterima server
		`<form action="/action_page.php">`
		`  <input type="file" name="pic" accept="image/*">`
		`  <input type="submit">`
		`</form>`

`accept-charset`	`<form>`	Menentukan character set yang digunakan ketika form di submit
		`<form action="/action_page.php" accept-charset="ISO-8859-1">`
		`First name: <input type="text" name="fname"><br>`<br>`Last name: <input type="text" name="lname"><br>`<br>`<input type="submit" value="Submit">`
		`</form>`

`accesskey`	`Global Attribute`	
`action`	`<form>`	Menentukan kemana form data akan dikirim ketika form di submit
`align`	Non HTML5. Gunakan CSS.	

`alt`	`<area>`, `<img>`, `<input>`
	`<img>` wajib menggunakan `alt`
	Untuk membuat tooltip dari
		Mendefinisikan alternate text (text pengganti) ketika element gagal ditampikan
		`<img src="planets.gif" width="145" height="126"`

	`<img>` gunakan attribute `title`
	
	Untuk `<input>` hanya berlaku untuk `<input type="image">`
		`alt="Planets" usemap="#planetmap">`
		
		`<map name="planetmap">`
		`<area shape="rect" coords="0,0,82,126"`
		`href="sun.htm" alt="Sun">`
		`<area shape="circle" coords="90,58,3"`
		`href="mercur.htm" alt="Mercury">`
		`<area shape="circle" coords="124,58,8"`
		`href="venus.htm" alt="Venus">`
		`</map>`

`async`	`<script>`	Menentukan script di eksekusi secara asynchronous `<script src="demo_async.js" async></script>`

`autocomplete`	`<form>, <input>`	Menentukan apakah autocomplete enabled
		`<`*`element`*` autocomplete="on|off">`
		
		`<form action="/action_page.php"`
		`autocomplete="on">`
		`First name:<input type="text" name="fname"><br>`<br>`Last name: <input type="text" name="lname"><br>`<br>`E-mail: <input type="email" name="email"`
		`autocomplete="off">`
		`<br>`<br>`<input type="submit">`
		`</form>`

`autofocus`	`<button>, <input>, <select>, <textarea>`	Menentukan element yang secara otomatis mendapat focus ketika page load
		
		`<button type="button" autofocus>Click Me!`
		`</button>`

`autoplay`	`<audio>, <video>`	`<video controls autoplay>`
		`  <source src="movie.mp4" type="video/mp4">`
		`  <source src="movie.ogg" type="video/ogg">`
		`</video>`

`bgcolor`	Non HTML5. Gunakan CSS.	
`border`	Non HTML5. Gunakan CSS.	

`charset`	`<meta>, <script>`	Menentukan character encoding
		
		`<head>`
		`<meta charset="UTF-8">`
		`</head>`
		
		`<script src="myscripts.js" charset="UTF-`
		`8"></script>`

`checked`	`<input>`
	Khusus `<input type="checkbox">` dan `<input type="radio">`
		Menentukan <input> element yang harus dalam kondisi terpilih ketika page load
		
		`<form action="/action_page.php">`
		`<input type="checkbox" name="vehicle"`
		`value="Bike" checked> I have a bike<br>`<br>`<input type="checkbox" name="vehicle"`
		`value="Car"> I have a car<br>`<br>`input type="submit" value="Submit">`
		`</form>`

`cite`	`<blockquote>, <del>, <ins>, <q>`	Menentukan URL yang digunakan sebagai kutipan

```
<p>This is a text.
<ins cite="why_inserted.htm">This is an inserted
text.</ins></p>
```

class Global Attributes

color Non HTML5. Gunakan CSS.

cols <textarea>

Menentukan visible width dari text area

```
<textarea rows="2" cols="30">
Epcot adalah theme park yang berada di Walt
Disney World Resort.
</textarea>
```

colspan <td>, <th>

Menentukan berapa colom suatu cell dapat melebar

```
<table>
  <tr>
     <th colspan="2">Monthly Savings</th>
  </tr>
  <tr>
     <td>January</td>
     <td>$100</td>
  </tr>
  <tr>
     <td>February</td>
     <td>$80</td>
  </tr>
  <tr>
     <td colspan="2">Sum: $180</td>
  </tr>
</table>
```

content <meta>

Mendefinisikan value dari meta atribute atau http-equiv

```
<head>
<meta name="description" content="HTML
tutorials">
<meta name="keywords"
content="HTML,CSS,XML,JavaScript">
</head>
```

contenteditable Global Attributes

controls <audio>, <video>

Menentukan apakah control (play, pause, dll) audio/video ditampilkan

```
<audio controls>
  <source src="horse.ogg" type="audio/ogg">
  <source src="horse.mp3" type="audio/mpeg">
</audio>
```

coords <area>

Menentukan koordinat dari area

```
<area shape="rect" coords="0,0,82,126"
href="sun.htm" alt="Sun">
```

data <object>

Menentukan URL dari resource yang digunakan oleh object

```
<object width="400" height="400"
data="helloworld.swf"></object>
```

data-*	Global Attributes	
datetime	<del>, <ins>, <time>	Menentukan date dan time

default	<track>	Menentukan default track yang digunakan
		`<video width="320" height="240" controls>`
		`  <source src="forrest_gump.mp4"`
		`type="video/mp4">`
		`  <source src="forrest_gump.ogg"`
		`type="video/ogg">`
		`  <track src="subtitles_en.vtt" kind="subtitles"`
		`srclang="en" label="English" default>`
		`  <track src="subtitles_no.vtt" kind="subtitles"`
		`srclang="no" label="Norwegian">`
		`</video>`

defer	<script>	Menentukan script di eksekusi setelah proses DOM selesai `<script src="demo_defer.js" defer></script>`
dir	Global Attributes	

dirname	<input>, <textarea>	Menentukan file text direction (ekstensi .dir) yang akan dikirim ketika form di submit
		`<form action="/action_page.php">`
		`  First name: <input type="text" name="fname"`
		`dirname="fname.dir">`
		`  <input type="submit" value="Submit">`
		`</form>`

disabled	<button>, <fieldset>, <input>, <optgroup>, <option>, <select>, <textarea>	Menentukan element/group element yang di disable `<input type="text" name="lname" disabled><br>`
download	<a>, <area>	Menentukan file yang akan di download oleh user `<a href="/images/myimage.jpg" download>`
draggable	Global Attributes	

enctype	<form>
	Khusus <form method="post">
		Menentukan data encoder yang digunakan ketika form di submit
		`<form action="/action_page_binary.asp"`
		`method="post" enctype="multipart/form-data">`

for	<label>, <output>	Menentukan element yang terikat sesuai dengan ID
		`<input type="radio" id="html" name="fav_language"`
		`value="HTML">`
		`<label for="html">HTML</label><br>`

form	<button>, <fieldset>, <input>, <label>, <meter>, <object>, <output>, <select>, <textarea>	Menentukan pada form ID mana suatu element tergabung
		`<form action="/action_page.php" id="form1">`
		`First name: <input type="text" name="fname"><br>`

```
<input type="submit" value="Submit">
</form>

Last name: <input type="text" name="lname"
form="form1">
```

formaction

<button>, <input> Khusus type = "submit"

Atribute formaction akan membatalkan attribute action

```
<form action="/action_page.php" method="get">
First name: <input type="text" name="fname"><br>
Last name: <input type="text" name="lname"><br>
<button type="submit">Submit</button><br>
<button type="submit"
formaction="/action_page2.php">Submit to another
page</button>
</form>
```

headers

<td>, <th>

Mendefinisikan relasi header cell dengan tabel cell

```
<table>
  <tr>
    <th id="name" colspan="2">Name</th>
  </tr>
  <tr>
    <th headers="name">Firstname</th>
    <th headers="name">Lastname</th>
  </tr>
</table>
```

height

<canvas>, <embed>, <iframe>, <img>, <input>, <object>, <video>

Menentukan tinggi element

```
<iframe src="/default.asp" width="200"
height="200">
</iframe>
```

hidden

Global Attributes

high

<meter>

Menentukan nilai range tertinggi

```
<meter min="0" low="40" high="90" max="100"
value="95"></meter>
```

href

<a>, <area>, <base>, <link>

Menentukan URL tujuan dari hyperlink

```
<link rel="stylesheet" type="text/css"
href="theme.css">
```

hreflang

<a>, <area>, <link>

Menentukan bahasa yang digunakan pada dokumen tujuan

```
<a href="https://www.examples.com"
hreflang="en">HTML</a>
```

http-equiv

<meta>

Menyediakan HTTP header sebagai value atau informasi dari content attribut

http-equiv juga digunakan untuk mensimulasikan HTTP Response Header

```
<head>
<meta http-equiv="refresh" content="30">
</head>
```

`id`	`Global Attributes`	* `id` harus bersifat unik, artinya tidak boleh ada element dengan `id` yang sama * `id` digunakan sebagai reference pada CSS dengan tanda #

`ismap`	`<img>`	Menentukan suatu image merupakan server-side image map. Ketika image di click maka koordinatnya akan dikirimkan ke server sebagai URL string
		`<a href="/action_page.php">`
		`<img src="html.gif" alt="example.com" ismap>`
		`</a>`

`kind`	`<track>`	Mendefinisikan jenis dari text track `<track src="subtitles_en.vtt" kind="subtitles" srclang="en" label="English">`
`label`	`<track>, <option>, <optgroup>`	Mendefinisikan track title
`lang`	`Global Attributes`	

`list`	`<input>`	Merupakan penghubung <datalist> element
		`<input list="browsers">`
		
		`<datalist id="browsers">`
		`  <option value="Internet Explorer">`
		`  <option value="Firefox">`
		`  <option value="Google Chrome">`
		`  <option value="Opera">`
		`  <option value="Safari">`
		`</datalist>`

`loop`	`<audio>, <video>`	Menentukan bahwa audio/video akan secara otomatis diputar ulang jika telah selesai
		`<audio controls loop>`
		`  <source src="horse.ogg" type="audio/ogg">`
		`  <source src="horse.mp3" type="audio/mpeg">`
		`</audio>`

`low`	`<meter>`	Mendefinisikan value range terendah `<meter min="0" low="40" high="90" max="100" value="95"></meter>`
`max`	`<input>, <meter>, <progress>`	Mendefinisikan maximum value
`maxlength`	`<input>, <textarea>`	Mendefinisikan jumlah maximum character `Username: <input type="text" name="username" maxlength="10"><br>`

`media`	`<a>, <area>, <link>, <source>, <style>`	Mendefinisikan media yang digunakan
		`<element media="value">`
		
		`value` operator = `and | not | ,`
		`value` device = `all | print | screen | speech`
		`value` values = `aspect-ratio | color | color-index`

		| height | monochrome | orientation | resolution | scan | width <a href="att_a_media.asp?output=print" media="print and (resolution:300dpi)"> Open media attribute page for print.</a>

method	<form>	Mendefinisikan HTTP method yang digunakan untuk mengirim data
		<form method="get|post">
		<form action="/action_page.php" method="get">
		  First name: <input type="text"
		name="fname">
		
		  Last name: <input type="text" name="lname">
		
		  <input type="submit" value="Submit">
		</form>

min	<input>, <meter>	Mendefinisikan minimum value
multiple	<input>, <select> Khusus <input type="file"> <input type="email">	Pilihan dapat lebih dari satu <input type="file" id="files" name="files" multiple>
muted	<video>, <audio>	Menentukan suara audio/video diredam <video controls muted> <source src="movie.mp4" type="video/mp4"> <source src="movie.ogg" type="video/ogg"> </video>
name	<button>, <fieldset>, <form>, <iframe>, <input>, <map>, <meta>, <object>, <output>, <param>, <select>, <textarea>	Mendefinisikan nama element. * Hanya element dengan attribute name yang akan dikirimkan ke server * name tidak harus unik <form action="/action_page.php" method="get"> Choose your favorite subject: <button name="subject" type="submit" value="HTML">HTML</button> <button name="subject" type="submit" value="CSS">CSS</button> </form>
novalidate	<form>	Mendefinisikan form tidak perlu dilakukan validasi pada saat di submit <form action="/action_page.php" novalidate> E-mail: <input type="email" name="user_email"> <input type="submit"> </form>
onabort	<audio>, <embed>, <img>, <object>, <video>	<video id="myVideo" onabort="alert('Video load aborted')">
onafterprint	<body>	<body onafterprint="myFunction()">
onbeforeprint	<body>	<body onbeforeprint="myFunction()">
onbeforeunload	<body>	<body onbeforeunload="return myFunction()">
onblur	All visible elements loses focus.	<input type="text" name="fname" id="fname"

		onblur="myFunction()">
oncanplay	<audio>, <embed>, <object>, <video> Ready to start playing when it has enough buffer	<audio oncanplay="myFunction()">
oncanplaythrough	<audio>, <video> File can be played without pausing for buffering	<audio oncanplaythrough="myFunction()">
onchange	All visible elements.	<input type="text" name="txt" value="Hello" onchange="myFunction(this.value)">
onclick	All visible elements.	<button onclick="myFunction()">Click me</button>
oncontextmenu	All visible elements.	<div oncontextmenu="myFunction()" contextmenu="mymenu">
oncopy	All visible elements.	<p oncopy="myFunction()">Try to copy this text</p>
oncuechange	<track>	
oncut	All visible elements.	<p contenteditable="true" oncut="myFunction()">Try to cut this text</p>

ondblclick	All visible elements.	<p id="demo" ondblclick="myFunction()">Double-click me to change my text color.</p>
		
		<script>
		function myFunction() {
		 document.getElementById("demo").style.color = "red";
		}
		</script>

ondrag ondragend ondragenter ondragleave ondragover ondragstart ondrop	All visible elements.	<p draggable="true" ondrag="myFunction(event)">Drag me!</p>
ondurationchange	<audio>, <video>	Urutan Event yang terjadi pada saat audio/video loading 1. loadstart 2. durationchange 3. loadedmetadata 4. loadeddata 5. progress 6. canplay 7. canplaythrough <audio ondurationchange="myFunction()">
onemptied	<audio>, <video>	
onended	<audio>, <video>	<audio onended="thanks for listening">
onerror	<audio>, <body>, <embed>, <img>, <object>, <script>, <style>, <video>	<video onerror="myFunction()"> <video id="myVideo" > var vid = document.getElementById("myVideo");

		`vid.onerror = function() {`
		`  alert("Error! Something went wrong");`
		`}`

onfocus	All visible elements.	`<input type="text" id="fname" onfocus="myFunction(this.id)">`
onhashchange	`<body>`	
oninput	All visible elements.	`<input type="text" oninput="myFunction()">`
oninvalid	All visible elements.	`<input type="text" oninvalid="alert('You must fill out the form!');" required>`
onkeydown onkeypress onkeyup	All visible elements.	`<input type="text" onkeypress="displayResult()">`

onload	`<body>, <iframe>, <img>, <input>, <link>, <script>, <style>`	`<input type="image" onload="loadImage()" src="img_submit.gif" alt="Submit">`
		
		`<script>`
		`function loadImage() {`
		` alert("Image is loaded");`
		`}`
		`</script>`

onloadeddata	`<audio>, <video>`	`<audio onloadeddata="myFunction()">`
onloadedmetadata	`<audio>, <video>` meta data (dimensions, duration etc)	`<audio onloadedmetadata="myFunction()">`
onloadstart	`<audio>, <video>`	`<audio onloadstart="myFunction()">`
onmousedown onmousemove onmouseout onmouseover onmouseup onmousewheel	All visible elements.	`<img onmousemove="bigImg(this)" src="smiley.gif" alt="Smiley">`
onoffline ononline onpagehide onpageshow	`<body>`	`<body onpageshow="myFunction()">`
onpaste	All visible elements.	`<input type="text" onpaste="myFunction()" value="Paste something in here">`
onpause onplay onplaying	`<audio>, <video>`	`<video onplay="myFunction()">`
onpopstate	`<body>` window's history changes	
onprogress	`<audio>, <video>`	`<audio onprogress="myFunction()">`
onratechange	`<audio>, <video>` playback rate changes eg slowmotion, fastforward etc	`<video onratechange="myFunction()">`

onreset	`<form>`
	reset button
		`<form onreset="myFunction()">`
		`  Enter name: <input type="text">`
		`  <input type="reset">`
		`</form>`

onresize	`<body>`	`<body onresize="myFunction()">`

	browser window resized	
onscroll	All visible elements.	<div onscroll="myFunction()">
onsearch	<input> Khusus <input="search">	<input type="search" onsearch="myFunction()">
onseeked	<audio>, <video> seeking attribute is ended	<audio onseeked="myFunction()">
onseeking	<audio>, <video> seeking attribute is active	<video onseeking="myFunction()">
onselect	All visible elements.	<input type="text" onselect="myFunction()" value="Hello world!">
onshow	<menu>	<div contextmenu="mymenu"> <p>Right-click inside this box to see the context menu!</p> <menu type="context" id="mymenu" onshow="myFunction()"> <menuitem label="Refresh" onclick="window.location.reload();"></menuitem> </menu> </div>
onstalled	<audio>, <video> unable to fetch	
onstorage	<body> Web Storage area is updated	
onsubmit	<form>	<form onsubmit="myFunction()"> Enter name: <input type="text"> <input type="submit"> </form>
onsuspend	<audio>, <video> stopped before completely loaded	
ontimeupdate	<audio>, <video> playing position has changed	<audio ontimeupdate="myFunction()">
ontoggle	<details> open or close <details>	<details ontoggle="myFunction()">
onunload	<body> browser window has been closed	<body onunload="myFunction()">
onvolumechange	<audio>, <video>	<video onvolumechange="myFunction()">
onwaiting	<audio>, <video> pauses to buffer more data	<video onwaiting="myFunction()">
onwheel	All visible elements. mouse wheel rolls up or down	<div onwheel="myFunction()">Roll the mouse wheel over me</div>
open	<details>	<details open> <summary>Copyright 1999-2014.</summary> <p> - by MMP. All Rights Reserved.</p> <p>All content and graphics on this web site are the property of the company MMP.</p> </details>
optimum	<meter>	<meter value="0.3" high="0.9" low="0.1" optimum="0.5"></meter>
pattern	<input> regular expression	<form action="/action_page.php"> Country code: <input type="text" name="country_code" pattern="[A-Za-z]{3}" title="Three letter country code">

		<input type="submit"> </form> <form action="/action_page.php"> Password: <input type="password" name="pw" pattern="(?=.*\d)(?=.*[a-z])(?=.*[A-Z]).{8,}" title="Must contain at least one number and one uppercase and lowercase letter, and at least 8 or more characters"> <input type="submit"> </form>
placeholder	<input>, <textarea>	<form action="/action_page.php"> <label for="phone">Enter a phone number:</label> <input type="tel" id="phone" name="phone" placeholder="123-45-678" pattern="[0-9]{3}-[0-9]{2}-[0-9]{3}"> <small>Format: 123-45-678</small> <input type="submit"> </form>

poster	<video>
	image to be shown while the video is downloading
		<video controls poster="/images/html5.gif">
		  <source src="movie.mp4" type="video/mp4">
		  <source src="movie.ogg" type="video/ogg">
		</video>

preload	<audio>, <video>	<audio preload="auto|metadata|none"> <audio controls preload="none">

readonly	<input>, <textarea>	<form action="/action_page.php">
		  Country: <input type="text" name="country" value="Norway" readonly>
		
		  <input type="submit" value="Submit">
		</form>

rel	<a>, <area>, <form>, <link>	<*element* rel="*value*"> *value* = alternate | author | bookmark | external | help | license | next | nofollow | noopener | noreferrer | prev | search | tag <a rel="nofollow" href="http://www.functravel.com/">Cheap Flights</a>

required	<input>, <select>, <textarea>	<select required>
		  <option value="">None</option>
		  <option value="volvo">Volvo</option>
		  <option value="saab">Saab</option>
		  <option value="mercedes">Mercedes</option>
		  <option value="audi">Audi</option>
		</select>

reversed	<ol>	<ol reversed>
		  <li>Coffee</li>
		  <li>Tea</li>
		  <li>Milk</li>
		</ol>

rows	<textarea>	<textarea rows="4" cols="50"> At example.com you will learn how to make a website. We offer free tutorials in all web development technologies. </textarea>
rowspan	<td>, <th>	<tr> <td>January</td> <td>$100</td> <td rowspan="2">$50</td> </tr>

sandbox	<iframe> content extra set of restrictions	<iframe sandbox="*value*"> *value* = (*no value*)| allow-forms | allow-modals | allow-orientation-lock | allow-pointer-lock | allow-popups | allow-popups-to-escape-sandbox | allow-presentation | allow-same-origin | allow-scripts | allow-top-navigation | allow-top-navigation-by-user-activation <iframe src="demo_iframe_sandbox_form.htm" sandbox="allow-forms"></iframe>
scope	<th>	<th scope="col|row|colgroup|rowgroup"> <th scope="col">Month</th>

selected	<option>
	pre-selected when the page loads
		<select>
		  <option value="volvo">Volvo</option>
		  <option value="saab">Saab</option>
		  <option value="vw">VW</option>
		  <option value="audi" selected>Audi</option>
		</select>

shape	<area>	<area shape="default|rect|circle|poly"> <area shape="circle" coords="90,58,3" href="mercur.htm" alt="Mercury">

size	<input>, <select>	<form action="/action_page.php">
		  Email: <input type="text" name="email"
		size="35"><br><br> PIN: <input type="text" name="pin"
		maxlength="4" size="4"><br><br><input type="submit" value="Submit">
		</form>
		
		<select size="3">
		  <option value="volvo">Volvo</option>
		  <option value="saab">Saab</option>
		  <option value="opel">Opel</option>
		  <option value="audi">Audi</option>
		</select>

span	<col>, <colgroup>	<colgroup span="2" style="background:red"></colgroup>
spellcheck	Global Attributes	<*element* spellcheck="true|false"> <p contenteditable="true" spellcheck="true">This is a paragraph.</p>
src	<audio>, <embed>, <iframe>, <img>, <input>, <script>, <source>, <track>, <video>	<*element* src="*URL*"> <source src="horse.mp3" type="audio/mpeg">
srcdoc	<iframe>	<iframe srcdoc="*HTML_code*"> <iframe srcdoc="<p>Hello world!</p>" src="demo_iframe_srcdoc.htm"></iframe>

srclang	<track>	srclang="*language_code*"
		
		<track src="subtitles_en.vtt" kind="subtitles"
		srclang="en" label="English">

srcset	<img>, <source>	<*element* srcset="*URL*">

		<source media="(min-width:465px)" srcset="img_white_flower.jpg">

start	<ol>	<ol start="50">
		  <li>Coffee</li>
		  <li>Tea</li>
		  <li>Milk</li>
		</ol>

step	<input>	<input type="number" id="points" name="points" step="3">
style	Global Attributes	
tabindex	Global Attributes	
target	<a>, <area>, <base>, <form>	<*element* target="_blank|_self|_parent|_top| *framename*"> <area shape="rect" coords="0,0,82,126" href="sun.htm" alt="Sun" target="_blank">
title	Global Attributes	
translate	Global Attributes	
type	<a>, <button>, <embed>, <input>, <link>, <menu>, <object>, <script>, <source>, <style>	Specifies the type of element <a type="*media_type*"> <button type="button|submit|reset"> <embed type="*media_type*"> <input type="button|checkbox|color|date|datetime- local|email |file|hidden|image|month|number| password|radio|range|reset|search|submit|tel| text|time|url|week"> <link type="*media_type*"> <menu type="list|context|toolbar"> <object type="*media_type*"> <script type="*scripttype*"> <source type="*media_type*"> <style type="*media_type*">
usemap	<img>, <object>	<*element* usemap="*#mapname*"> <object data="planets.gif" width="145" height="126" usemap="#planetmap"></object>
value	<button>, <input>, <li>, <option>, <meter>, <progress>, <param>	<button name="subject" type="submit" value="fav_HTML">HTML</button>
width	<canvas>, <embed>, <iframe>, <img>, <input>, <object>, <video>	<canvas id="myCanvas" width="200" height="200" style="border:1px solid">
wrap	<textarea>	<textarea wrap="soft|hard"> <textarea rows="2" cols="20" wrap="hard"> At Example.com you will find free Web-building tutorials.

```
                                        </textarea>
```

HTML Window Event BAB VI

Window Event Attribute

Event pada window object (`<body>`) :
onafterprint | onbeforeprint | onbeforeunload | onerror | onhashchange | onload | onmessage
onoffline | ononline | onpagehide | onpageshow | onpopstate | onresize | onstorage | onunload

Form Event

Event yang pada umumnya digunakan HTML form :
onblur | onchange | oncontextmenu | onfocus | oninput | oninvalid | onreset
onsearch | onselect | onsubmit

Keyboard Event

onkeydown | onkeypress | onkeyup

Mouse Event

onclick | ondblclick | onmousedown | onmousemove | onmouseout | onmouseover
onmouseup | onmousewheel | onwheel

Drag Event

ondrag | ondragend | ondragenter | ondragleave | ondragover | ondragstart | ondrop | onscroll

Clipboard Event

oncopy | oncut | onpaste

Media Event

onabort | oncanplay | oncanplaythrough | oncuechange | ondurationchange | onemptied
onended | onerror | onloadeddata | onloadedmetadata | onloadstart | onpause | onplay
onplaying | onprogress | onratechange | onseeked | onseeking | onstalled
onsuspend | ontimeupdate | onvolumechange | onwaiting

Misc Event

ontoggle

Colors, Styles dan Shadow Property
fillStyle | strokeStyle | shadowColor | shadowBlur | shadowOffsetX | shadowOffsetY

Method
createLinearGradient() | createPattern() | createRadialGradient() |addColorStop()

Line Style Property
lineCap | lineJoin | lineWidth | miterLimit

Rectangle Method
rect() | fillRect() | strokeRect() | clearRect()

Path Method
fill() | stroke() | beginPath() | moveTo() | closePath() | lineTo() | clip() | quadraticCurveTo() | bezierCurveTo()
arc() | arcTo() | isPointInPath()

Transformation Method
scale() | rotate() | translate() | transform() | setTransform()

Text Property
font | textAlign | textBaseline

Text Method
fillText() | strokeText() | measureText()

Image Drawing Method
drawImage()

Pixel Manipulation Property
width | height | data

Pixel Manipulation Method
createImageData() | getImageData() | putImageData()

Compositing Property
globalAlpha | globalCompositeOperation

Other Method
save() | restore() | createEvent() | getContext() | toDataURL()

Audio/Video Method
addTextTrack() | canPlayType() | load() | play() | pause()

Audio/Video Property
audioTracks | autoplay | buffered | controller | controls | crossOrigin | currentSrc | currentTime
defaultMuted | defaultPlaybackRate | duration | ended | error | loop | mediaGroup | muted
networkState | paused | playbackRate | played | preload | readyState | seekable | seeking
src | startDate | textTracks | videoTracks | volume

Audio/Video Event
abort | canplay | canplaythrough | durationchange | emptied | ended | error | loadeddata |
loadedmetadata
loadstart | pause | play | playing | progress | ratechange | seeked | seeking | stalled
suspend | timeupdate | volumechange | waiting

HTTP Status Message

1xx: Information

Message:	Keterangan:
`100 Continue`	Server sudah menerima request header dan client kemudian mengirimkan request body
`101 Switching Protocols`	Meminta server untuk berganti protocol
`103 Early Hints`	Digunakan bersama Link header memberi kesempatan browser untuk memulai preloading resource sementara server menyiapkan response

2xx: Successful

Message:	Keterangan:
`200 OK`	Request OK (HTTP request = success)
`201 Created`	Request OK dan menyiapkan resource baru
`202 Accepted`	Request OK tetapi processing belum selesai
`203 Non-Authoritative Information`	Request OK tetapi returning information menyatakan berasal dari resource lain.
`204 No Content`	Request OK tetapi tanpa return content apapun
`205 Reset Content`	Request OK tetapi tanpa return content apapun dan document view harus di reset
`206 Partial Content`	Return content hanya sebagian karena range header

3xx: Redirection

Message:	Keterangan:
`300 Multiple Choices`	Link list. User dapat memilih link dengan maximum 5 alamat link
`301 Moved Permanently`	Page yang diminta berada di URL lain
`302 Found`	Page yang diminta berada di URL lain
`303 See Other`	Page yang diminta berada di URL lain
`304 Not Modified`	Page yang diminta berada belum pernah update sejak update terakhir
`307 Temporary Redirect`	Page yang diminta untuk sementara diarahkan ke URL baru
`308 Permanent Redirect`	Page yang diminta secara permanen diarahkan ke URL baru

4xx: Client Error

Message:	Keterangan:
`400 Bad Request`	Request tidak dapat diproses karena bad syntax
`401 Unauthorized`	Legal request tetapi server menolak untuk merespon.

`402 Payment Required`	*Reserved for future use*
`403 Forbidden`	Legal request tetapi server menolak untuk merespon.
`404 Not Found`	Page yang diminta tidak tersedia
`405 Method Not Allowed`	Request method tidak disupport page yang diminta
`406 Not Acceptable`	Server hanya dapat memberikan respon yang tidak dapat diterima client
`407 Proxy Authentication Required`	Client harus melakukan otentikasi pada proxy yang digunakan
`408 Request Timeout`	Server timed out saat menunggu request
`409 Conflict`	Terdapat conflict pada request
`410 Gone`	Page yang diminta tidak tersedia
`411 Length Required`	Content-Length" pada request tidak didefinisikan akan ditolak server
`412 Precondition Failed`	Request precondition setelah dievaluasi server menghasilkan false
`413 Request Too Large`	Request entity terlalu panjang
`414 Request-URI Too Long`	URI terlalu panjang. Terjadi karena perubahan dari POST request menjadi GET request
`415 Unsupported Media Type`	Tipe media uyang diminta tidak disupport server
`416 Range Not Satisfiable`	Client meminta potongan dari suatu file yang tidak dapat dilakukan server
`417 Expectation Failed`	Request-header field yang diminta tidak dapat disediakan server

5xx: Server Error

Message:	Keterangan:
`500 Internal Server Error`	Generic error message. Server tidak bisa memberikan error yang spesifik
`501 Not Implemented`	Server tidak mengenali request atau tidak dapat memprosesnya
`502 Bad Gateway`	Server yang berfungsi sebagai gateway atau proxy dan menerima invalid response
`503 Service Unavailable`	Server overloaded atau down
`504 Gateway Timeout`	Server yang berfungsi sebagai gateway atau proxy dan tidak menerima response pada waktunya
`505 HTTP Version Not Supported`	Server tidak mendukung versi HTTP protocol yang diminta
`511 Network Authentication Required`	Client harus melakukan otentikasi

HTTP Method BAB X

Hypertext Transfer Protocol (HTTP) berfungsi sebagai request-response protocol yang memungkinkan komunikasi terjadi antara client dan server.
Contoh : Client (browser) mengirimkan HTTP request ke server kemudian akan memberikan response kepada client. Server response berisi status dari client request dan bisa juga content yang diminta client.

Ada beberapa jenis HTTP Method tetapi yang akan disampaikan disini hanya yang valid digunakan pada HRML5 <form> dan AJAX yaitu GET, POST dan PUT

GET method (HTML5)

GET digunakan untuk melakukan request data dari sumber (URL) tertentu. Request tersebut akan dikirimkan ke URL dalam bentuk query string (name/value) seperti contoh ini:

```
/test/demo_form.php?name1=value1&name2=value2
```

GET request :

- Ditampilkan pada browser
- Bisa di cached
- Berada di browser history
- Bisa di bookmark
- Panjang maksimum URL adalah 2048 character
- Hanya dapat digunakan untuk ASCII character
- Yang harus diperhatikan : jangan digunakan untuk data yang sensitif
- Digunakan hanya untuk request data buka untuk modifikasi

POST Method (HTML5)

POST mengirimkan data ke server. Data yang dikirimkan disimpan pada HTTP request-body seperti contoh ini :

```
POST /test/demo_form.php HTTP/1.1
Host: example.com
name1=value1&name2=value2
```

POST request lebih banyak digunakan daripada GET.

PUT Method (AJAX)

PUT sama seperti halnya dengan POST. Perbedaan penggunaannya dapat dianalogikan dengan contoh ini :

PUT   : "UPDATE STUDENT SET address = "abc" where id="123";
POST : "INSERT INTO STUDENT(name, address) VALUES ("abc", "xyzzz");

A

align-content	stretch|center|flex-start|flex-end|space-between|space-around|initial|inherit
align-items	stretch|center|flex-start|flex-end|baseline|initial|inherit
align-self	auto|stretch|center|flex-start|flex-end|baseline|initial|inherit
all	initial|inherit|unset
animation	shorthand *animation-** property
animation-delay	*time*|initial|inherit
animation-direction	normal|reverse|alternate|alternate-reverse|initial|inherit
animation-duration	*time*|initial|inherit
animation-fill-mode	none|forwards|backwards|both|initial|inherit
animation-iteration-count	*number*|infinite|initial|inherit
animation-name	*keyframename*|none|initial|inherit
animation-play-state	paused|running|initial|inherit
animation-timing-function	linear|ease|ease-in|ease-out|ease-in-out|step-start|step-end|steps(int,start|end)|cubic-bezier(*n*,*n*,*n*,*n*)|initial|inherit

B

backface-visibility	visible|hidden|initial|inherit
background	shorthand *background-** property
background-attachment	scroll|fixed|local|initial|inherit
background-blend-mode	normal|multiply|screen|overlay|darken|lighten|color-dodge|saturation|color|luminosity
background-clip	border-box|padding-box|content-box|initial|inherit
background-color	*color*|transparent|initial|inherit
background-image	*url*|none|initial|inherit
background-origin	padding-box|border-box|content-box|initial|inherit
background-position	left top|left center|left bottom|right top|right center|right bottom|center top|center center|center bottom|*x% y%*|*xpos ypos*|initial|inherit
background-repeat	repeat|repeat-x|repeat-y|no-repeat|initial|inherit
background-size	auto|*length*|cover|contain|initial|inherit
border	shorthand property untuk *border-width, border-style* dan *border-color*
border-bottom	shorthand property untuk *border-bottom-width, border-bottom-style* dan *border-bottom-color*
border-bottom-color	*color*|transparent|initial|inherit
border-bottom-left-radius	*length*|*%* [*length*|*%*]|initial|inherit
border-bottom-right-radius	*length*|*%* [*length*|*%*]|initial|inherit
border-bottom-style	none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset|initial|inherit
border-bottom-width	medium|thin|thick|*length*|initial|inherit
border-collapse	separate|collapse|initial|inherit
border-color	*color*|transparent|initial|inherit
border-image	*source slice width outset repeat*|initial|inherit
border-image-outset	*length*|*number*|initial|inherit

border-image-repeat	stretch|repeat|round|space|initial|inherit
border-image-slice	_number_|_%_|fill|initial|inherit
border-image-source	none|_image_|initial|inherit
border-image-width	_number_|_%_|auto|initial|inherit
border-left	shorthand _border-left-*_ property
border-left-color	_color_|transparent|initial|inherit
border-left-style	none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset|initial|inherit
border-left-width	medium|thin|thick|_Length_|initial|inherit
border-radius	shorthand _border-*-radius_ property _1-4 length_|_%_ / _1-4 length_|_%_|initial|inherit
border-right	shorthand _border-right-*_ property
border-right-color	_color_|transparent|initial|inherit
border-right-style	none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset|initial|inherit
border-right-width	medium|thin|thick|_Length_|initial|inherit
border-spacing	_length_|initial|inherit
border-style	none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset|initial|inherit
border-top	shorthand property untuk _border-top-width, border-top-style_ dan _border-top-color_
border-top-color	_color_|transparent|initial|inherit
border-top-left-radius	_length_|_%_ [_length_|_%_]|initial|inherit
border-top-right-radius	_length_|_%_ [_length_|_%_]|initial|inherit
border-top-style	none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset|initial|inherit
border-top-width	medium|thin|thick|_length_|initial|inherit
border-width	medium|thin|thick|_length_|initial|inherit
bottom	auto|_length_|initial|inherit
box-decoration-break	slice|clone|initial|inherit|unset
box-shadow	none|_h-offset v-offset blur spread color_ |inset|initial|inherit
box-sizing	content-box|border-box|initial|inherit
break-after	auto|all|always|avoid|avoid-column|avoid-page|avoid-region|column|left|page|recto|region|right|verso|initial|inherit
break-before	auto|all|always|avoid|avoid-column|avoid-page|avoid-region|column|left|page|recto|region|right|verso|initial|inherit
break-inside	auto|all|always|avoid|avoid-column|avoid-page|avoid-region|column|left|page|recto|region|right|verso|initial|inherit

C

caption-side	top|bottom|initial|inherit
caret-color	auto|_color_|initial|inherit
@charset	_charset_
clear	none|left|right|both|initial|inherit
clip	auto|_shape_|initial|inherit
color	_color_|initial|inherit
column-count	_number_|auto|initial|inherit
column-fill	balance|auto|initial|inherit
column-gap	_length_|normal|initial|inherit
column-rule	shorthand _column-rule-*_ property
column-rule-color	_coLor_|initial|inherit
column-rule-style	none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset|

	initial|inherit
column-rule-width	medium|thin|thick|*length*|initial|inherit
column-span	none|all|initial|inherit
column-width	auto|*length*|initial|inherit
columns	shorthand property untuk *column-width* dan *column-count*
content	normal|none|counter|attr|*string*|open-quote|close-quote|no-open-quote|no-close-quote|*url*|initial|inherit
counter-increment	none|*id*|initial|inherit
counter-reset	none|*name number*|initial|inherit
cursor	alias|all-scroll|auto|cell|context-menu|col-resize|copy|crosshair|default|e-resize|ew-resize|grab|grabbing|help|move|n-resize|ne-resize|nesw-resize|ns-resize|nw-resize|nwse-resize|no-drop|none|not-allowed|pointer|progress|row-resize|s-resize|se-resize|sw-resize|text|*URL*|vertical-text|w-resize|wait|zoom-in zoom-out|initial|inherit

D

direction	ltr|rtl|initial|inherit
display	inline|block|contents|flex|grid|inline-block|inline-flex|inline-grid|inline-table|list-item|run-in|table|table-caption|table-column-group|table-header-group|table-footer-group|table-row-group|table-cell|table-column|table-row| none|initial|inherit

E

empty-cells	show|hide|initial|inherit

F

filter	none | blur() | brightness() | contrast() | drop-shadow() | grayscale() | hue-rotate() | invert() | opacity() | saturate() | sepia() | url()
flex	shorthand property untuk *flex-grow, flex-shrink dan flex-basis*
flex-basis	*number*|auto|initial|inherit
flex-direction	row|row-reverse|column|column-reverse|initial|inherit
flex-flow	shorthand property untuk *flex-direction* dan *flex-wrap*
flex-grow	*number*|initial|inherit
flex-shrink	*number*|initial|inherit
flex-wrap	*number*|initial|inherit
float	none|left|right|initial|inherit
font	shorthand property untuk *font-style, font-variant, font-weight, font-size/line-height* dan *font-family*

@font-face	font-family = “name”
	src = “font file”
	font-stretch = normal|condensed|ultra-condensed|extra-condensed|semi-condensed|expanded|semi-expanded|extra-expanded|ultra-expanded
	font-style = normal|italic|oblique
	font-weight = normal|bold|100|200|300|400|500|600|700|800|900
	
	@font-face {
	  font-family: myFirstFont;
	  src: url(sansation_bold.woff);
	  font-weight: bold;
	}

font-family	*family-name*|*generic-family*|initial|inherit
font-feature-settings	normal|*feature-value*
@font-feature-values	
font-kerning	auto|normal|none

font-language-override	
font-size	medium|xx-small|x-small|small|large|x-large|xx-large|smaller|larger|*Length*|initial|inherit
font-size-adjust	*number*|none|initial|inherit
font-stretch	ultra-condensed|extra-condensed|condensed|semi-condensed|normal|semi-expanded|expanded|extra-expanded|ultra-expanded|initial|inherit
font-style	normal|italic|oblique|initial|inherit
font-synthesis	
font-variant	normal|small-caps|initial|inherit
font-variant-alternates	
font-variant-caps	normal|small-caps|all-small-caps|petite-caps|all-petite-caps|unicase|titling-caps|initial|inherit|unset
font-variant-east-asian	
font-variant-ligatures	
font-variant-numeric	
font-variant-position	
font-weight	normal|bold|bolder|lighter|*number*|initial|inherit

G

gap	shorthand property untuk *row-gap* dan *column-gap*
grid	shorthand property untuk *grid-template-rows, grid-template-columns, grid-template-areas, grid-auto-rows, grid-auto-columns dan grid-auto-flow*
grid-area	shorthand property untuk *grid-row-start*, *grid-column-start*, *grid-row-end* dan *grid-column-end*
grid-auto-columns	auto|max-content|min-content|*length*
grid-auto-flow	row|column|dense|row dense|column dense
grid-auto-rows	auto|max-content|min-content|*length*
grid-column	shorthand property untuk *grid-column-start* dan *grid-column-end*
grid-column-end	auto|span *n*|*column-line*
grid-column-gap	*length*
grid-column-start	auto|span *n*|*column-line*
grid-gap	shorthand property untuk *grid-row-gap* dan *grid-column-gap*
grid-row	shorthand property untuk *grid-row-start* dan *grid-row-end*
grid-row-end	auto|*row-line*|span *n*
grid-row-gap	*length*
grid-row-start	auto|*row-line*
grid-template	shorthand property untuk *grid-template-rows*, *grid-template-columns* dan *grid-areas*
grid-template-areas	none|*itemnames*
grid-template-columns	none|auto|max-content|min-content|*length*|initial|inherit
grid-template-rows	none|auto|max-content|min-content|*length*|initial|inherit

H

hanging-punctuation	none|first|last|allow-end|force-end|initial|inherit
height	auto|*length*|initial|inherit
hyphens	none|manual|auto|initial|inherit

I

image-rendering	auto|smooth|high-quality|crisp-edges|pixelated|initial|inherit

@import	*url*|*string list-of-mediaqueries*
isolation	auto|isolate|initial|inherit

J

justify-content	flex-start|flex-end|center|space-between|space-around|space-evenly|initial|inherit

K

@keyframes	*animationname* {*keyframes-selector* {*css-styles;*}}

L

left	auto|*length*|initial|inherit
letter-spacing	normal|*length*|initial|inherit
line-break	
line-height	normal|*number*|*length*|initial|inherit
list-style	*list-style-type list-style-position list-style-image*|initial|inherit
list-style-image	none|*url*|initial|inherit
list-style-position	inside|outside|initial|inherit
list-style-type	disc|armenian|circle|cjk-ideographic|decimal|decimal-leading-zero|georgian|hebrew|hiragana|hiragana-iroha|katakana|katakana-iroha|lower-alpha|lower-greek|lower-latin|lower-roman|none|square|upper-alpha|upper-greek|upper-latin|upper-roman|initial|inherit

M

margin	*length*|auto|initial|inherit
margin-bottom	*length*|auto|initial|inherit
margin-left	*length*|auto|initial|inherit
margin-right	*length*|auto|initial|inherit
margin-top	*length*|auto|initial|inherit
mask	
mask-clip	
mask-composite	
mask-image	none|*image*|*url*|initial|inherit
mask-mode	match-source|luminance|alpha|initial|inherit
mask-origin	border-box|content-box|padding-box|margin-box|fill-box|stroke-box|view-box|initial|inherit
mask-position	left top|left center|left bottom|right top|right center|right bottom|center top|center center|center bottom|*x% y%* |*xpos ypos*|initial |inherit
mask-repeat	repeat|repeat-x|repeat-y|space|round|no-repeat|initial|inherit
mask-size	auto|*size*|contain|cover|initial|inherit
mask-type	
max-height	none|*length*|initial|inherit
max-width	none|*length*|initial|inherit

@media	@media not|only *mediatype* and *(mediafeature* and|or|not *mediafeature)*
	{
	CSS-Code;
	}
	Media type = all|print|screen|speech
	Media Feature = any-hover|any-pointer|aspect-ratio|color|color-gamut|color-index|grid|height|hover|inverted-colors|light-level|max-aspect-ratio|max-color|max-color-index|max-height|max-monochrome|max-resolution|max-width|min-aspect-ratio|min-color|min-color-index|min-height|min-monochrome|min-resolution|min-width|monochrome|

<table>
<tr><td></td><td>orientation|overflow-block|overflow-inline|pointer|resolution|scan|scripting|update|width</td></tr>
<tr><td>min-height</td><td>*length*|initial|inherit</td></tr>
<tr><td>min-width</td><td>*length*|initial|inherit</td></tr>
<tr><td>mix-blend-mode</td><td>normal|multiply|screen|overlay|darken|lighten|color-dodge|color-burn|difference|exclusion|hue|saturation|color|luminosi</td></tr>
</table>

O

<table>
<tr><td>object-fit</td><td>fill|contain|cover|scale-down|none|initial|inherit</td></tr>
<tr><td>object-position</td><td>*position*|initial|inherit</td></tr>
<tr><td>opacity</td><td>*number*|initial|inherit</td></tr>
<tr><td>order</td><td>*number*|initial|inherit</td></tr>
<tr><td>orphans</td><td>*integer*|initial|inherit</td></tr>
<tr><td>outline</td><td>*outline-width outline-style outline-color*|initial|inherit</td></tr>
<tr><td>outline-color</td><td>invert|*color*|initial|inherit</td></tr>
<tr><td>outline-offset</td><td>*length*|initial|inherit</td></tr>
<tr><td>outline-style</td><td>none|hidden|dotted|dashed|solid|double|groove|ridge|inset|outset|initial|inherit</td></tr>
<tr><td>outline-width</td><td>medium|thin|thick|*length*|initial|inherit</td></tr>
<tr><td>overflow</td><td>visible|hidden|scroll|auto|initial|inherit</td></tr>
<tr><td>overflow-wrap</td><td>normal|anywhere|break-word|initial|inherit</td></tr>
<tr><td>overflow-x</td><td>visible|hidden|scroll|auto|initial|inherit</td></tr>
<tr><td>overflow-y</td><td>visible|hidden|scroll|auto|initial|inherit</td></tr>
</table>

P

<table>
<tr><td>padding</td><td>Shorthand property untuk *padding-**</td></tr>
<tr><td>padding-bottom</td><td>*length*|initial|inherit</td></tr>
<tr><td>padding-left</td><td>*length*|initial|inherit</td></tr>
<tr><td>padding-right</td><td>*length*|initial|inherit</td></tr>
<tr><td>padding-top</td><td>*length*|initial|inherit</td></tr>
<tr><td>page-break-after</td><td>auto|always|avoid|left|right|initial|inherit</td></tr>
<tr><td>page-break-before</td><td>auto|always|avoid|left|right|initial|inherit</td></tr>
<tr><td>page-break-inside</td><td>auto|avoid|initial|inherit</td></tr>
<tr><td>perspective</td><td>*length*|none</td></tr>
<tr><td>perspective-origin</td><td>*x-axis y-axis*|initial|inherit</td></tr>
<tr><td>pointer-events</td><td>auto|none</td></tr>
<tr><td>position</td><td>static|absolute|fixed|relative|sticky|initial|inherit</td></tr>
</table>

Q

<table>
<tr><td>quotes</td><td>none|*string*|initial|inherit</td></tr>
</table>

R

<table>
<tr><td>resize</td><td>none|both|horizontal|vertical|initial|inherit</td></tr>
<tr><td>right</td><td>auto|*length*|initial|inherit</td></tr>
<tr><td>row-gap</td><td>*length*|normal|initial|inherit</td></tr>
</table>

S

<table>
<tr><td>scroll-behavior</td><td>auto|smooth|initial|inherit</td></tr>
</table>

T

tab-size	*number*|*length*|initial|inherit
table-layout	auto|fixed|initial|inherit
text-align	left|right|center|justify|initial|inherit
text-align-last	auto|left|right|center|justify|start|end|initial|inherit
text-combine-upright	
text-decoration	
text-decoration-color	*color*|initial|inherit
text-decoration-line	none|underline|overline|line-through|initial|inherit
text-decoration-style	solid|double|dotted|dashed|wavy|initial|inherit
text-decoration-thickness	auto|from-font|*length/percentage*|initial|inherit
text-indent	*length*|initial|inherit
text-justify	auto|inter-word|inter-character|none|initial|inherit
text-orientation	
text-overflow	clip|ellipsis|*string*|initial|inherit
text-shadow	*h-shadow v-shadow blur-radius color*|none|initial|inherit
text-transform	none|capitalize|uppercase|lowercase|initial|inherit
text-underline-position	
top	auto|*length*|initial|inherit
transform	none|*transform-functions*|initial|inherit
transform-origin	*x-axis y-axis z-axis*|initial|inherit
transform-style	flat|preserve-3d|initial|inherit
transition	shorthand property untuk *transition-**
transition-delay	*time*|initial|inherit
transition-duration	*time*|initial|inherit
transition-property	none|all|*property*|initial|inherit
transition-timing-function	linear|ease|ease-in|ease-out|ease-in-out|step-start|step-end|steps(int,start|end)|cubic-bezier(*n*,*n*,*n*,*n*)|initial|inherit

U

unicode-bidi	normal|embed|bidi-override|initial|inherit
user-select	auto|none|text|all

V

vertical-align	baseline|*length*|sub|super|top|text-top|middle|bottom|text-bottom|initial|inherit
visibility	visible|hidden|collapse|initial|inherit

W

white-space	normal|nowrap|pre|pre-line|pre-wrap|initial|inherit
widows	*integer*|initial|inherit
width	auto|*value*|initial|inherit
word-break	normal|break-all|keep-all|break-word|initial|inherit
word-spacing	normal|*length*|initial|inherit
word-wrap	normal|break-word|initial|inherit
writing-mode	horizontal-tb|vertical-rl|vertical-lr

Z

z-index	auto|*number*|initial|inherit

CSS Selector BAB XII

Selector	Contoh	Keterangan
`.class`	.intro	Setiap element dengan class="intro"
`.class1.class2`	.name1.name2	Setiap element yang menggunakan class *name1* dan *name2* secara bersamaan → class=”name1 name2”
`.class1 .class2`	.name1 .name2	Setiap element dengan class *name2* dimana *name2* merupakan turunan dari *name1*
`#id`	#firstname	Setiap element dengan id="firstname"
`*`	*	Seluruh element
`element`	p	Setiap <p> element
`element.class`	p.intro	Setiap <p> element dengan class="intro"
`element,element`	div, p	Setiap <div> element dan <p> element
`element element`	div p	Setiap <p> element yang berada didalam <div> element
`element>element`	div > p	Setiap <p> element yang parent elementnya adalah <div>
`element+element`	div + p	Setiap <p> element pertama yang langsung berada dibawah <div> element
`element1~element2`	p ~ ul	Setiap <ul> element yang didahului oleh <p> element
`[attribute]`	[target]	Setiap element dengan attribute target
`[attribute=value]`	[target=_blank]	Setiap elements dengan target="_blank"
`[attribute~=value]`	[title~=flower]	Setiap element dengan title attribute yang berisi huruf "flower"
`[attribute|=value]`	[lang|=en]	Setiap element dengan lang attribute sama dengan "en" atau dimulai dengan "en-"
`[attribute^=value]`	a[href^="https"]	Setiap <a> element dengan href attribute dimulai dengan "https"
`[attribute$=value]`	a[href$=".pdf"]	Setiap <a> element dengan href attribute diakhiri dengan ".pdf"
`[attribute*=value]`	a[href*="school"]	Setiap <a> element dengan href attribute yang berisi "w3schools"
`:active`	a:active	Setiap active link
`::after`	p::after	Menyisipkan setelah content pada setiap <p> element
`::before`	p::before	Menyisipkan sebelum content pada setiap <p> element

:checked	input:checked	Setiap <input> element dengan status terpilih
:default	input:default	Setiap <input> element dengan status default
:disabled	input:disabled	Setiap <input> element dengan status disable
:empty	p:empty	Setiap <p> element yang tidak memiliki hirarki element dibawahnya
:enabled	input:enabled	Setiap <input> element dengan status enabled
:first-child	p:first-child	Setiap <p> element yang menjadi urutan pertama dari hirarki diatasnya.
::first-letter	p::first-letter	Setiap huruf pertama dari setiap <p> element
::first-line	p::first-line	Setiap baris pertama dari setiap <p> element
:first-of-type	p:first-of-type	Setiap <p> element yang menjadi <p> element pertama dari hirarki diatasnya
:focus	input:focus	Setiap input element yang sedang menjadi focus
:fullscreen	:fullscreen	Setiap element yang berada pada full-screen mode
:hover	a:hover	Setiap <a> ketika mouse over
:in-range	input:in-range	Setiap input element dimana value berada dalam range
:indeterminate	input:indeterminate	Setiap input element dalam status indeterminate
:invalid	input:invalid	Setiap input element dengan invalid value
:lang(language)	p:lang(it)	Setiap <p> element dengan lang attribute sama dengan "it" (Italian)Setiap <p> element yang menjadi <p> element pertama dari hirarki diatasnya
:last-child	p:last-child	Setiap <p> element yang menjadi urutan terakhir dari hirarki diatasnya.
:last-of-type	p:last-of-type	Setiap <p> element yang menjadi <p> element terakhir dari hirarki diatasnya
:link	a:link	Setiap unvisited link
::marker	::marker	Setiap item dengan status marker
:not(selector)	:not(p)	Setiap element selain <p> element
:nth-child(n)	p:nth-child(2)	Setiap <p> element yang menjadi urutan nomer 2 dari hirarki diatasnya.
:nth-last-child(n)	p:nth-last-child(2)	Setiap <p> element yang menjadi urutan nomer 2 dari urutan terakhir dari hirarki diatasnya.
:nth-last-of-type(n)	p:nth-last-of-type(2)	Setiap <p> element yang menjadi <p> element kedua dari urutan terkahir dari hirarki diatasnya
:nth-of-type(n)	p:nth-of-type(2)	Setiap <p> element yang menjadi <p> element kedua dari hirarki diatasnya
:only-of-type	p:only-of-type	Setiap <p> element yang hanya menjadi <p> element dari hirarki diatasnya
:only-child	p:only-child	Setiap <p> element yang hanya menjadi satu-

		satunya element dari hirarki diatasnya
`:optional`	input:optional	Setiap input element tanpa atrtibute "required"
`:out-of-range`	input:out-of-range	Setiap input element dengan value berada diluar range
`::placeholder`	input::placeholder	Setiap input element yang menggunakan attribute "placeholder"
`:read-only`	input:read-only	Setiap input element dengan attribute"readonly"
`:read-write`	input:read-write	Setiap input element dimana attribute "readonly" tidak ditentukan.
`:required`	input:required	Setiap input element dengan attribute "required"
`:root`	:root	Document root element
`::selection`	::selection	Setiap bagian element yang dipilih oleh user
`:target`	#news:target	Target dengan ID #news element
`:valid`	input:valid	Setiap input element dengan valid value
`:visited`	a:visited	Setiap visited link

CSS Function

Function	Keterangan
`attr()`	value dari attribute dari suatu element
`calc()`	Kalkulasi utuk menentukan value
`conic-gradient()`	Create conic gradient
`counter()`	value dari suatu counter
`cubic-bezier()`	Cubic Bezier curve
`hsl()`	Warna berdasarkan Hue-Saturation-Lightness model (HSL)
`hsla()`	Warna berdasarkan Hue-Saturation-Lightness-Alpha model (HSLA)
`linear-gradient()`	Create linear gradient
`max()`	Menggunakan value terbesar dari value list (value dipisahkan dengan tanda comma)
`min()`	Menggunakan value terkecil dari value list (value dipisahkan dengan tanda comma)
`radial-gradient()`	Create radial gradient
`repeating-conic-gradient()`	Pengulangan conic gradient
`repeating-linear-gradient()`	Pengulangan linear gradient
`repeating-radial-gradient()`	Pengulangan radial gradient
`rgb()`	Warna berdasarkan Red-Green-Blue model (RGB)
`rgba()`	Warna berdasarkan Red-Green-Blue-Alpha model (RGBA)
`var()`	Value dari custom property

A. Absolute Length

Absolute length unit bersifat tetap artinya panjang yang tercantum akan ditampilkan sesuai dengan unit yang digunakan. Absolute Lenght tidak disarankan digunakan pada layar monitor kecuali sebagai output pada printer.

Unit	Keterangan
cm	centimeter
mm	millimeter
in	inch (1inch = 96px = 2.54cm)
px *	pixel (1px = 1/96 inch)
pt	point (1pt = 1/72 inch)
pc	pica (1pc = 12 pt)

**Pixels (px) bersifat relatif tergantung pada peralatan. Untuk layar low-dpi, 1px berarti 1 pixel (dot) pada layar tetapi pada printer dan layar high resolution 1px dapat berarti multi pixel.

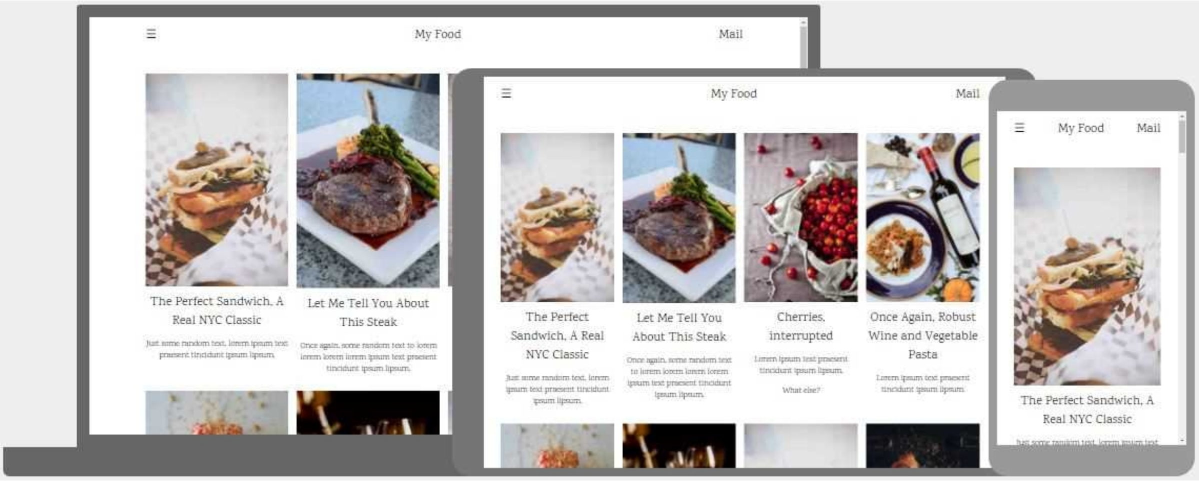
B. Relative Length

Relative length units mendasarkan ukurannya berdasarkan unit property element lain. Relative length unit sebaiknya digunakan karena penyesuaian ukuran bekerja dengan baik untuk berbagai macam ukuran layar.

Unit	Keterangan
em	font-size (contoh : 2em berarti ukuran 2 kali lebih besar dari current font)
ex	font height
ch	Relative to the width of the "0" (zero)
rem	font-size berdasarkan root element
vw	1% lebar viewport*
vh	1% tinggi viewport*
vmin	1% ukuran terkecil dari viewport
vmax	1% ukuran terbesar dari viewport
%	parent element

BOOTSTRAP 5 BAB XV

Bootstrap adalah front-end framework yang banyak digunakan pada web development terutama untuk Responsive Web Design. Responsive Web adalah desain web yang secara otomatis menyesuaikan dengan layar yang digunakan mulai dari smartphone hingga desktop


My Food
Mail
The Perfect Sandwich, A Real NYC Classic
Let Me Tell You About This Steak
Cherries, interrupted
Once Again, Robust Wine and Vegetable Pasta

Untuk menggunakan Bootstrap diperlukan satu file CSS dan satu file JavaScript (jika menggunakan komponen seperti `model`, `tooltip`, `popover` dan lain lain yang menggunakan JavaScript) dengan cara menggunakan link dari CDN (Content Delivery Network) seperti contoh ini :

```
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/css/bootstrap.min.css"
rel="stylesheet">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.1.3/dist/js/bootstrap.bundle.min.js"></script>
```

Atau download dari https://getbootstrap.com/ dan tempatkan pada server. Dengan menggunakan CDN kedua file tersebut tidak perlu didownload berulang-ulang karena akan menggunakan cache yang tersedia sehingga page load menjadi lebih cepat.

Bootstrap menggunakan komponen HTML dan CSS yang sesuai dengan tipe HTML5 document, karena itu seluruh element di bawah ini harus tersedia.

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
  </head>
</html>
```

* `lang` disesuaikan

Bootstrap memerlukan seluruh content selalu berada dalam container agar dapat berfungsi sebagai responsive web. Karena itu Bootstrap memiliki 3 class container yaitu `.container` yaitu fixed width container dan `.container-fluid` adalah full width container yang akan melebar mengikuti lebar viewport dan `.container-{breakpoint}` yang akan melebar 100% hingga batas Breakpoint.

A. Breakpoint

Breakpoint adalah bagian dari layout Bootstrap yang akan mengendalikan layout agar sesuai dengan viewport atau ukuran layar device.

class	Extra small <576px *none*	Small ≥576px sm	Medium ≥768px md	Large ≥992px lg	Extra Large ≥1200px xl	XXL ≥1400px xxl
max-width	100%	540px	720px	960px	1140px	1320px

`.container`, `max-width` pada setiap responsive breakpoint
`.container-fluid`, width: 100% pada setiap breakpoint
`.container-{breakpoint}`, `width: 100%` sampai dengan breakpoint

Contoh :

```
<div class="container-lg">.container-lg</div>
```

→ container akan dibuat dengan lebar 100% selama belum lebih dari 960px

B. Grid

Bootstrap Grid dibuat dengan menggunakan flexbox dan dapat terbagi maximum 12 kolom. Setiap kolom akan di posisikan sesuai dengan viewport. Setiap antar kolom terdapat gutter (gap) dengan default 0.75rem di kiri dan 0.75rem di bagian kanan kolom

<table>
<tr><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td><td>span 1</td></tr>
<tr><td colspan="4">span 4</td><td colspan="4">span 4</td><td colspan="4">span 4</td></tr>
<tr><td colspan="4">span 4</td><td colspan="8">span 8</td></tr>
<tr><td colspan="6">span 6</td><td colspan="6">span 6</td></tr>
<tr><td colspan="12">span 12</td></tr>
</table>

Class yang digunakan adalah :

```
.col-# (extra small device)
.col-sm-# (small device)
.col-md-# (medium device)
.col-lg-# (large device)
.col-xl-# (xlarge device)
.col-xxl-# (xxlarge device)
```

# adalah jumlah kolom

Contoh :

```
<div class="row">
   <div class="col-sm-3">.col-sm-3</div>
   <div class="col-sm-3">.col-sm-3</div>
   <div class="col-sm-3">.col-sm-3</div>
   <div class="col-sm-3">.col-sm-3</div>
</div>
```

Alert

Tag name	Description
alert-primary	<div class="alert alert-primary" role="alert"> Text message to be displayed </div>
alert-secondary	<div class="alert alert-secondary" role="alert"> Text message to be displayed </div>
alert-success	<div class="alert alert-success" role="alert"> Success text to be displayed </div>
alert-info	<div class="alert alert-info" role="alert"> Info message to be displayed </div>
alert-warning	<div class="alert alert-warning" role="alert"> Warning message to be displayed </div>
alert-danger	<div class="alert alert-danger" role="alert"> Error message to be displayed, something really went wrong </div>
alert-light	<div class="alert alert-light" role="alert"> Light font and style to display alert </div>
alert-dark	<div class="alert alert-dark" role="alert"> Dark font and style to display alert message </div>
alert-link	<div class="alert alert-dark" role="alert"> <a href="#!" class="alert-link">I am an alert link</a> </div>
alert-dismissible	<div class="alert alert-info alert-dismissible"> <a href="#" class="close" data-dismiss="alert" aria-label="close">&times;</a> Alert message with dismissible (closing) link </div>
alert-heading	<h4 class="alert-heading">Alert success with heading</h4>

Badge

badge	<h1>Heading text <span class="badge badge-secondary">Badge same size as heading text</span></h1>
badge-pill	<span class="badge badge-pill badge-primary">Primary</span>
badge-primary	<span class="badge badge-primary">Primary badge text variation</span>
badge-secondary	<span class="badge badge-primary">Secondary badge text variation</span>
badge-success	Success badge
badge-info	Info badge
badge-warning	Warning badge
badge-danger	Danger badge
badge-light	Light badge
badge-dark	Dark badge

Breadcrumb

breadcrumb

```
<nav aria-label="breadcrumb">
   <ol class="breadcrumb">
      <li class="breadcrumb-item"><a href="#">Root</a></li>
      <li class="breadcrumb-item active" aria-
current="page">packages</li>
   </ol>
</nav>
```

Button

btn

```
<button type="button" class="btn">Basic</button>
```

btn-primary

```
<button type="button" class="btn-primary">Primary</button>
```

btn-secondary

```
<button type="button" class="btn-secondary">secondary button</button>
```

btn-success

```
<button type="button" class="btn-success">Success</button>
```

btn-info

```
<button type="button" class="btn-info">Info button</button>
```

btn-warning

```
<button type="button" class="btn-warning">Warning button</button>
```

btn-danger

```
<button type="button" class="btn-danger">Danger</button>
```

btn-light

```
<button type="button" class="btn-light">light button</button>
```

btn-dark

```
<button type="button" class="btn-dark">Dark button </button>
```

btn-link

```
<button type="button" class="btn-link">link in button</button>
```

btn-outline-primary

```
<button type="button" class="btn-outline-primary">Primary
outline</button>
```

btn-outline-secondary

```
<button type="button" class="btn-outline-secondary">secondary
outline</button>
```

btn-outline-success

```
<button type="button" class="btn-outline- success "> success
outline</button>
```

btn-outline-info

```
<button type="button" class="btn-outline- info "> info
outline</button>
```

btn-outline-warning

```
<button type="button" class="btn-outline- warning "> warning
outline</button>
```

btn-outline-danger

```
<button type="button" class="btn-outline- danger "> danger
outline</button>
```

btn-outline-light

```
<button type="button" class="btn-outline-light">light outline and
text</button>
```

btn-outline-dark

```
<button type="button" class="btn-outline-dark">Dark outline</button>
```

Button Group

btn-group

```
<div class="btn-group">
--individual button items
</div>
```

btn-group-lg

```
<div class="btn-group btn-group-lg">--individual button items</div>
```

btn-group-sm

```
<div class="btn-group btn-group-sm">--individual button items</div>
```

btn-group-vertical

```
<div class="btn-group- vertical ">--individual button items</div>
```

btn-group (nested)

```
<div class="btn-group">
    <button type="button" class="btn btn-primary dropdown-toggle"
data-toggle="dropdown">
       ButtonGroupItem
     </button>
     <div class="dropdown-menu">
        --add dropdown menu items
     </div>
</div>
```

btn-toolbar

```
<div class="btn-toolbar" role="toolbar">
   <div class="btn-group" role="group" aria-label="grp1">
      --group one button types
   </div>
   <div class="btn-group" role="group" aria-label="grp2">
      --group two button types
   </div>
</div>
```

Button Modifier

btn-lg	<button type="button" class="btn-primary btn-lg">Large</button>
btn-sm	<button type="button" class="btn-primary btn-sm">Small</button>
btn-block	<button type="button" class="btn btn-primary btn-block">Full-Width Button</button>
active button	<button type="button" class="btn btn-primary active">Active button with style of primary</button>
disabled button	<button type="button" class="btn btn-warning" disabled>Disabled button</button> for a link (<a>), disabled has to be used in the class - class="btn btn-warning disabled”

checkbox as button	<div class="btn-group btn-group-toggle" data-toggle="buttons">
	   <label class="btn btn-secondary active">
	      <input type="checkbox" checked autocomplete="off"> checkbox1
	   </label>
	   <label class="btn btn-secondary">
	      <input type="checkbox" autocomplete="off"> checkbox2
	   </label>
	</div>

spinner buttons	<button class="btn btn-primary">
	   <span class="spinner-border spinner-border-sm"></span>
	   Loading..
	</button>

Card

card	<div class="card">
card-body	<div class="card-body">Contents of the card</div>
card-title	<h2 class="card-title">Title of card in h2</h2>
card-subtitle	<h6 class="card-subtitle">Card subtitle</h6>
card-text	<p class="card-text">text.text.text</p>
card-link	<a href="#!" class="card-link">link in the card</a>
card-img-top	<img class="card-img-top" src="myimpimage.png">
middle image	<img src="myimpimage.png">
card-img-bottom	<img class="card-img-bottom" src="myimpimage.png">
card-img-overlay	<div class="card-img-overlay"> <p class="card-text">See the image at background</p> </div>

list-group	<ul class="list-group">
	   <li class="list-group-item">Item 1</li>
	   <li class="list-group-item">Item 2</li>
	</ul>

card-header	<div class="card-header"> This is text for card header </div>
h*.card-header	<h3 class="card-header">Card header in h3</h3>
card-footer	<div class="card-footer"> footer content </div>
card-group	<div class="card-group"> …card 1 div tags …card 2 div tags </div>
card-deck	<div class="card-deck"> …card 1 div tags …card 2 div tags </div>
card-columns	<div class="card-columns"> …card 1 div tags …card 2 div tags </div>
card bg-... text-...	<div class = "card text-<color> bg-<color>"></div>

Carousel

Adds slide show for going through elements in a cyclic fashion

carousel slide	`<div class="carousel slide" data-ride="carousel">`
carousel-fade	`<div class="carousel slide carousel-fade" data-ride="carousel">`
carousel-inner	`<div class="carousel-inner">`

carousel-item	`<div class="carousel-item">`
	`<div class="carousel-item active">`

carousel-indicators	`<ul class="carousel-indicators">`
	`   <li data-target="#demo" data-slide-to="0" class="active"></li>`
	`   <li data-target="#demo" data-slide-to="1"></li>`
	`</ul>`

carousel-caption	`<div class="carousel-caption">`
carousel-control-prev	`<a class="carousel-control-prev" href="#demo" data-slide="prev"></a>`
carousel-control-next	`<a class="carousel-control-next" href="#demo" data-slide="prev"></a>`

Collapse

collapse

```
<p>
   <a class="btn btn-primary" data-toggle="collapse"
href="#expandcollapse" aria-expanded="false" aria-
controls="expandcollapse">
Click to expand/collapse
   </a>
</p>
<div class="collapse" id="expandcollapse">
   <div class="card card-body">
      lot of text
   </div>
</div>
```

accordion

```
<div id="accordion">
   <div class="card">
      <div class="card-header">
         <a class="card-link" data-toggle="collapse" href="#item1">
            Item #1
         </a>
      </div>
      <div id="item1" class="collapse show" data-parent="#accordion">
         <div class="card-body">
            text for expand/collapse
         </div>
      </div>
   </div>
   <div class="card">
      <div class="card-header">
         <a class="collapsed card-link" data-toggle="collapse"
href="#item2">
            Item #2
         </a>
      </div>
      <div id="item2" class="collapse" data-parent="#accordion">
         <div class="card-body">
            text for expand/collapse
         </div>
      </div>
   </div>
</div>
```

Dropdown

dropdown

```
<div class="dropdown">
and then in the button or link definition, add
data-toggle="dropdown" attribute</div>
```

dropdown (split)

dropup

dropup (split)	
dropright	
dropleft	
dropdown-header	
dropdown-item-text	<span class="dropdown-item-text">Plain Text</span>
dropdown-divider	
dropdown-item disabled	
dropdown-item-active	
dropdown-menu-right	<div class="dropdown-menu dropdown-menu-right">

Form

form-group	<div class="form-group">
form-inline	<form class="form-inline" action="someactionpage.jsp">

form using the grid	<form>
	   <div class="row">
	      <div class="col">
	         <input type="text" class="form-control" id="login id" placeholder="Enter login id" name="login">
	      </div>
	      <div class="col">
	         <input type="password" class="form-control" placeholder="Enter password" name="pwd">
	      </div>
	   </div>
	</form>

form-control	<input type="password" class="form-control" ….
form-control-lg	<input type="password" class="form-control-lg" ….
form-control-sm	<input type="password" class="form-control-sm" ….
form-control-file	<input type="file" class="form-control-file" id="formcontrolfileupload">
form-control-plaintext	<input type="text" readonly class="form-control-plaintext" id="defaulttext" value="someexample@gmail.com">
form-control-range	<input type="range" class="form-control-range">
form-check	<div class="form-group form-check"> <input class="form-check-input" type="checkbox" name = "Agree">
form-check-inline	

disabled items	<fieldset disabled>
	   <div class="form-group">
	      <label for="disabledTextInput">Disabled text box</label>
	     <input type="text" id="disabledTextInput" class="form-control" placeholder="Disabled input box">
	   </div>
	</fieldset>

readonly	<input class="form-control" type="text" placeholder="I am a read only text" readonly>

Form input group

input-group	<div class="input-group">
input-group-prepend	<div class="input-group-prepend"> <span class="input-group-text">+91</span> </div>
input-group-append	<div class="input-group-append"> <span class="input-group-text">@gmail.com</span> </div>
input-group-sm	<div class="input-group input-group-sm">
input-group-lg	<div class="input-group input-group-lg">
checkbox	<div class="input-group">

```
   <div class="input-group-*">
      <div class="input-group-text">
         <input type="checkbox">
      </div>
   </div>
   <input type="text" class="form-control">
</div>
```

radio

```
<div class="input-group">
   <div class="input-group-*">
      <div class="input-group-text">
         <input type="radio">
      </div>
   </div>
   <input type="text" class="form-control">
</div>
```

dropdown

```
<div class="input-group">
   <div class="input-group-*">
      ...code to add dropdown (check dropdown section)
   </div>
   <input type="text" class="form-control" label="Text input with
dropdown button">
</div>
```

Custom Form

custom-checkbox

```
<form>
   <div class="custom-control custom-checkbox">
      <input type="checkbox" class="custom-control-input"
id="customCheckBox" name="chkboxexample">
      <label class="custom-control-label" for="customCheckBox">Custom
checkbox</label>
   </div>
</form>
```

custom-radio

```
<form>
   <div class="custom-control custom-radio">
     <input type="radio" class="custom-control-input" id="customRadio"
name="radioexample">
     <label class="custom-control-label" for="customRadio">Custom
Radio button</label>
   </div>
</form>
```

custom-switch

```
<div class="custom-control custom-switch">
   <input type="checkbox" class="custom-control-input"
id="customSwitch">
   <label class="custom-control-label" for="customSwitch">Toggle
switch element</label>
</div>
```

custom-select

```
<select class="custom-select">
   <option selected>Custom select menu</option>
   <option value="1">Val1</option>
   <option value="2">Val2</option>
   <option value="3">Val3</option>
</select>
```

custom-file

```
<div class="custom-file">
   <input type="file" class="custom-file-input" id="customFile">
   <label class="custom-file-label" for="customFile">Choose your
file</label>
</div>
```

custom-range

```
<form>
   <label for="customRange">Custom range</label>
   <input type="range" class="custom-range" id="customRange">
</form>
```

Grid

container

```
<div class="container">content</div>
```

container-fluid

```
<div class="container-fluid">full-width container</div>
```

row

```
<div class="row">
   ... column elements for the row
```

```
</div>
```

col-# (<576px)

```
<div class="col-2">
<div class="col-4">
<div class="col-8">
...
```

col-sm-# (≥576px)

```
<div class="col-sm-2">
<div class="col-sm-4">
<div class="col-sm-8">
```

col-md-# (≥768px)

```
<div class="col-md-2">
<div class="col-md-4">
<div class="col-md-8">
```

col-lg-# (≥992px)

```
<div class="col-lg-2">
<div class="col-lg-4">
<div class="col-lg-8">
```

col-xl-# (≥1200px)

```
<div class="col-xl-2">
<div class="col-xl-4">
<div class="col-xl-8">
```

col

```
<div class="col">
```

col-*

```
<div class="col-xl">
```

no-gutters

```
<div class="row no-gutters">… column definitions</div>
```

offset-*-#

```
<div class="col-3 offset-6">
<div class="col-md-3 offset-md-6">
```

order-#

```
<div class="col order-md-2">Ordered column</div>
<div class="col order-1">
```

nested columns

```
<div class="row">
   <div class="col-sm-10">
      Level 1: small column with 10
      <div class="row">
         <div class="col-8 col-sm-3">
            Nested column 1
         </div>
         <div class="col-4 col-sm-3">
            Nested column 2
         </div>
      </div>
   </div>
</div>
```

Images

img-fluid

```
<img src="/images/drawing.png" class="img-fluid" alt="Your dynamic
image">
```

img-thumbnail

```
<img class="img-thumbnail" src="/images/drawing.png" alt="Thumbnail
image">
```

Jumbotron

jumbotron

```
<div class="jumbotron">
   <h1>heading</h1>
   <p>Everything will be displayed inside a big grey box</p>
</div>
```

jumbotron-fluid

```
<div class="jumbotron jumbotron-fluid">
   <div class="container">
      <h1>I am fluid jumbotron</h1>
      <p>I occupy the whole horizontal grey area</p>
   </div>
</div>
```

List Group

list-group

```
<ul class="list-group">
   <li class="list-group-item">one</li>
   <li class="list-group-item">three</li>
   <li class="list-group-item">two</li>
</ul>
```

list-group-item active

```
<li class="list-group-item active">Active item</li>
```

list-group-item disabled

```
<li class="list-group-item disabled">disabled list item</li>
```

list-group-item-action

```
<div class="list-group">
   <a href="#" class="list-group-item list-group-item-action">One</a>
   <a href="#" class="list-group-item list-group-item-action">Two</a>
</div>
```

list-group-item-primary

```
<li class="list-group-item list-group-item-primary">Primary color
</li>
```

list-group-item-secondary

```
<li class="list-group-item list-group-item-secondary">secondary color
</li>
```

list-group-item-success

```
<li class="list-group-item list-group-item-success">success color
</li>
```

list-group-item-info

```
<li class="list-group-item list-group-item-info">Info color </li>
```

list-group-item-warning

```
<li class="list-group-item list-group-item-warning">Warning color
</li>
```

list-group-item-danger

```
<li class="list-group-item list-group-item-danger">Danger color </li>
```

list-group-item-light

```
<li class="list-group-item list-group-item-light">Light color </li>
```

list-group-item-dark

```
<li class="list-group-item list-group-item-dark">Dark color </li>
```

list-group with badges

```
<li class="list-group-item">Starred
<span class="badge badge-success badge-pill">30</span>
```

list-group with d-flex

```
<li class="list-group-item d-flex justify-content-between align-items-
center">
```

list-group-horizontal

```
<ul class="list-group list-group-horizontal">
```

list-group-flush

```
<ul class="list-group list-group-flush">
```

Media object

media

```
<div class="media">
   <img src="img_1.png" alt="Action hero" class="mr-3 square"
style="width:60px;">
   <div class="media-body">
      -- some content adjacent to the media object
   </div>
</div>
```

nested media

```
<div class="media">
   <img src="img_1.png" alt="Action hero" class="mr-3 square"
style="width:60px;">
   <div class="media-body">
      -- some content adjacent to the media object
      <div class="media">
         <img src="img_2.png" alt="Characteristics" class="mr-3
square" style="width:60px;">
         <div class="media-body">
             -- some content adjacent to the nested media object
         </div>
      </div>
   </div>
</div>
```

d-flex align-self-start

```
<img src="img_1.png" class="align-self-start mr-3" style="width:60px">
```

d-flex align-self-center

```
<img src="img_1.png" class="align-self-center mr-3"
style="width:60px">
```

d-flex align-self-end

```
<img src="img_1.png" class="align-self-end mr-3" style="width:60px">
```

right aligned media

```
<div class="media">
   <img src="img_1.png" alt="Action hero" class="mr-3 square"
style="width:60px;">
   <div class="media-body">
      -- some content adjacent to the media object
   </div>
   <img src="img_1.png" alt="Action hero" class="mr-3 square"
style="width:60px;">
</div>
```

Modal

modal	<button type="button" class="btn btn-primary" data-toggle="modal" data-target="#modalname"> Open modal dialog </button> <div class="modal" id="modalname"> -- inside content </div>
modal-dialog-centered	<div class="modal-dialog modal-dialog-centered">
modal-dialog-scrollable	<div class="modal-dialog modal-dialog-scrollable">
modal fade	<div class="modal fade"></div>
modal-lg	<div class="modal-dialog modal-lg">
modal-sm	<div class="modal-dialog modal-sm">
modal-xl	<div class="modal-dialog modal-xl">

Nav

ul.nav	<ul class="nav"> <li class="nav-item"> <a class="nav-link" href="#">navigation link</a> … other nav links
nav.nav	<nav class="nav"> <a class="nav-link" href="#">navigation link</a>
nav justify-content-*	<ul class="nav justify-content-center"> <ul class="nav justify-content-end">
nav flex-column	<ul class="nav flex-column">
nav-tabs	<ul class="nav nav-tabs"> <li class="nav-item"> <a class="nav-link active" href="#">Active/Current</a> </li>
nav-pills	<ul class="nav nav-pills">
nav-fill	<ul class="nav nav-pills nav-fill">
nav-justified	<ul class="nav nav-pills nav-justified"> <ul class="nav nav-tabs nav-justified">
nav-tabs with dropdown	<ul class="nav nav-tabs"> <li class="nav-item"> <a class="nav-link active" href="#">Current nav</a> </li> <li class="nav-item dropdown"> ... code for dropdown
nav-pills with dropdown	<ul class="nav nav-pills"> <li class="nav-item"> <a class="nav-link active" href="#">Current nav</a> </li> <li class="nav-item dropdown"> ... code for dropdown

Navbar

navbar	<ul class="navbar-nav">
navbar-brand	<a class="navbar-brand" href="#">Brand</a>
navbar with form	<nav class="navbar navbar-expand-md bg-primary navbar-dark"> <form class="form-inline">
navbar-text	<span class="navbar-text">
navbar-dark bg-dark	<nav class="navbar navbar-dark bg-dark">
navbar-light	<nav class="navbar navbar-light"></nav>
navbar fixed-top	<nav class="navbar fixed-top navbar-dark">
navbar fixed-bottom	<nav class="navbar fixed-bottom navbar-light">

navbar sticky-top	<nav class="navbar sticky-top navbar-light">
collapse navbar-collapse	<button class="navbar-toggler" type="button" data-toggle="collapse" data-target="#collapsibleNavbar"> <div class="collapse navbar-collapse" id="collapsibleNavbar">
navbar-toggler	<button class="navbar-toggler" type="button"
navbar-expand-*	<nav class="navbar navbar-expand-sm bg-dark">

Pagination

pagination	<ul class="pagination">
	   <li class="page-item"><a class="page-link" href="#">Prev</a></li>
	   <li class="page-item"><a class="page-link" href="#">1</a></li>
	   <li class="page-item"><a class="page-link" href="#">Next</a></li>
	</ul>

page-item disabled	<li class="page-item disabled"> <a class="page-link" href="#!" tabindex="-1">Previous</a> </li>
page-item active	<li class="page-item active"><a class="page-link" href="#">3</a></li>
pagination-lg	<ul class="pagination pagination-lg">
pagination-sm	<ul class="pagination pagination-sm">

Popover

popovers	<a href="#" title="mypopoverdata" data-toggle="popover" data-placement="top" data-content="Content">click me to see details</a>
dismissible popover	<a href="#" title="Dismissible popover" data-toggle="popover" data-trigger="focus" data-content="Click anywhere to close the popover">Click</a>
hover popover	<a href="#" title="Header" data-toggle="popover" data-trigger="hover" data-content="content">Hover over me to see popover content</a>

Progress

progress	<div class="progress">
progress-bar	<div class="progress-bar" role="progressbar" style="width: 41%"</div>
progress-bar with label	<div class="progress-bar" style="width:70%">progress details</div>
progress-bar with height	<div class="progress-bar" style="width:60%;height:12px"></div>
progress-bar bg-*	<div class="progress"> <div class="progress-bar bg-secondary" role="progressbar" style="width: 30%" aria-valuenow="25"</div> </div>

multiple progress-bar	<div class="progress">
	   <div class="progress-bar bg-danger" style="width:20%">no charge</div>
	   <div class="progress-bar bg-warning" style="width:20%">charge 40%</div>
	   <div class="progress-bar bg-success" style="width:30%">charge 70%</div>
	</div>

progress-bar-striped	<div class="progress-bar progress-bar-striped" style="width:30%"></div>
progress-bar-striped bg-*	<div class="progress-bar progress-bar-striped bg-success" role="progressbar" style="width: 40%"</div>
progress-bar-animated	<div class="progress-bar progress-bar-striped progress-bar-animated" role="progressbar" style="width: 20%"></div>

Scrollspy

data-spy

Spinner

spinner-border	`<div class="spinner-border" role="status">`

spinner-border text-*	`<div class="spinner-border text-primary" role="status">`
	`   <span class="sr-only"></span>`
	`</div>`

spinner-border-sm	`<div class="spinner-border spinner-border-sm" role="status">`
spinner-grow	`<div class="spinner-grow" role="status">`
spinner-grow text-*	`<div class="spinner-grow text-primary" role="status">`
spinner-grow-sm	`<div class="spinner-grow spinner-grow-sm" role="status">`

Tables

table	`<table class="table">…table contents</table>`
thead-light	`<thead class="thead-light">`
thead-dark	`<thead class="thead-dark">`
table-striped	`<table class="table table-striped">`
table-bordered	`<table class="table table-bordered">`
table-borderless	`<table class="table table-borderless">`
table-hover	`<table class="table table-hover">`
table-sm	`<table class="table table-sm">`
table-*-responsive	`<table class="table table-sm-responsive"><!-- ... table content --></table>`
table-reflow	
table-active	`<tr class="table-active">`
table-primary	`<tr class="table-primary">`
table-secondary	`<tr class="table-secondary">`
table-success	`<tr class="table-success">`
table-info	`<tr class="table-info">`
table-warning	`<tr class="table-warning">`
table-danger	`<tr class="table-danger">`
table-light	`<tr class="table-light">`
table-dark	`<tr class="table-dark">`

Toasts

toast	`<div class="toast">`
	`   <div class="toast-header">`
	`      The Header`
	`   </div>`
	`   <div class="toast-body">`
	`      The text for toast body`
	`   </div>`
	`</div>`

Tooltips

tooltip	`<a href="#" data-toggle="tooltip" data-placement="auto" title="Good day!">Thanks for hovering</a>`

Typography

display-# (1-4)	`<h1 class="display-1">Display 1</h1>`
	`<h1 class="display-2">Display 2</h1>`
	`<h1 class="display-3">Display 3</h1>`
	`<h1 class="display-4">Display 4</h1>`

lead	<p class="lead">This paragraph stands out amongst others</p>
blockquote	<blockquote class="blockquote"> <p>Lot of content</p> </blockquote>
blockquote-footer	<footer class="blockquote-footer">Footer text</footer>
list-unstyled	<ul class="list-unstyled">
list-inline	<ul class="list-inline">

dl-horizontal	<dl class = "dl-horizontal">
	   <dt>Description</dt>
	   <dd>Item</dd>
	</dl>

mark	<mark>highlight text</mark>
abbr	<abbr>ABC</abbr>
code	<code>span</code>
kbd	<kbd>ctrl + c</kbd>

Utility: Borders

border	<span class="border border-dark">Hi there</span>
border-*-0	<span class="p-1 border border-primary border-0">I have no border</span> <span class="p-1 border border-primary border-right-0">I have border on all sides except right</span>
border-*	<span class="border border-primary">I have blue border</span> <span class="border border-success">I have green border</span>
rounded	<img class="rounded" src="/images/myimg.png">
rounded-sm	<img class="rounded-sm" src="/images/myimg.png">
rounded-lg	<img class="rounded-lg" src="/images/myimg.png">
rounded-circle	<img class="rounded-circle" src="/images/myimg.png">
rounded-0	<img class="rounded-0" src="/images/myimg.png">
rounded-*	<img class="rounded-top" src="/images/myimg.png">

Utility: Colors

bg-primary	<div class="bg-primary text-white">text is white and background is blue</div>
bg-secondar	<div class="bg-secondary"> background is grey</div>
bg-success	<div class="bg-success"> background is green</div>
bg-info	<div class="bg-info"> background is light blue</div>
bg-warning	<div class="bg-warning"> background is yellow</div>
bg-danger	<div class="bg-danger"> background is red</div>
bg-light	<div class="bg-light"> background is white/light grey</div>
bg-dark	<div class="bg-dark"> background is black</div>
bg-white	<div class="bg-white"> background is white</div>
text-*	<p class="text-primary">Text color is blue</p> <p class="text-success">Text color is greeen</p> ...and so on

Utility: Display

d-*-block	<span class="d-block bg-primary">display block</span> <span class="d-sm-block bg-success">display small block</span>
d-*-flex	<div class="d-flex bg-secondary">...</div> <span class="d-md-flex bg-info">...</span>
d-*-inline	<div class="d-inline bg-secondary">...</div> <div class="d-md-inline bg-info">...</div>

d-*-inline-block	<div class="d-inline-block bg-secondary">...</div> <div class="d-lg-inline-block bg-info">...</div>
d-*-inline-flex	<div class="d-inline-flex bg-secondary">...</div> <div class="d-lg-inline-flex bg-info">...</div>
d-*-none	<div class="d-none">...</div> <div class="d-lg-none">...</div>
d-*-table	<div class="d-table">...</div> <div class="d-lg-table">...</div>
d-*-table-cell	<div class="d-table-cell">...</div> <div class="d-lg-table-cell">...</div>
d-print-...	<div class="d-print-block">Display content as block on print</div>

Utility: Flexbox

flex-*-column	<div class="d-flex flex-column"> <!--default size--> <div class="d-flex flex-sm-column">...</div>
flex-*-column-reverse	<div class="d-flex flex-column-reverse">
flex-*-row	<div class="d-flex flex-row"> <div class="d-flex flex-md-row">...</div>
flex-*-row-reverse	<div class="d-flex flex-row-reverse"> <div class="d-flex flex-lg-row-reverse">...</div>
flex-*-nowrap	<div class="d-flex flex-nowrap">..</div>
flex-*-wrap	<div class="d-flex flex-wrap"> ... </div> <div class="d-flex flex-xl-wrap"> ... </div>
flex-*-wrap-reverse	<div class="d-flex flex-wrap-reverse"> ... </div> <div class="d-flex flex-xl-wrap-reverse"> ... </div>
flex-fill	<div class="p-2 flex-fill bg-*">Flex item</div>
flex-*-grow-1	<div class="p-2 flex-grow-1 bg-primary">Flex grow</div>
flex-*-grow-0	<div class="p-2 flex-grow-0 bg-info">dont let me grow</div>
flex-*-shrink-1	<div class="p-2 flex-shrink-1 bg-danger">Flex shrink</div>
flex-*-shrink-0	<div class="p-2 flex-shrink-0 bg-primary">Flex no shrink</div>
justify-content-*-start	<div class="d-flex justify-content-start"> <div class="d-flex justify-content-sm-start">...</div>
justify-content-*-end	<div class="d-flex justify-content-end"> <div class="d-flex justify-content-sm-end">...</div>
justify-content-*-center	<div class="d-flex justify-content-center"> <div class="d-flex justify-content-sm-center">...</div>
justify-content-*-between	<div class="d-flex justify-content-between"> <div class="d-flex justify-content-sm-between">...</div>
justify-content-*-around	<div class="d-flex justify-content-around"> <div class="d-flex justify-content-sm-around">...</div>
align-content-*-start	<div class="d-flex flex-wrap align-content-start"> <div class="d-flex align-content-sm-start">...</div>
align-content-*-end	<div class="d-flex flex-wrap align-content-end"> <div class="d-flex align-content-sm-end">...</div>
align-content-*-center	<div class="d-flex flex-wrap align-content-center"> <div class="d-flex align-content-sm-center">...</div>
align-content-*-around	<div class="d-flex flex-wrap align-content-around"> <div class="d-flex align-content-sm-around">...</div>
align-content-*-stretch	<div class="d-flex flex-wrap align-content-stretch"> <div class="d-flex align-content-sm-stretch">...</div>
align-items-*-baseline	<div class="d-flex flex-wrap align-items-baseline"> <div class="d-flex align-items-sm-baseline">...</div>
align-items-*-stretch	<div class="d-flex align-items-stretch"> <div class="d-flex align-items-sm-stretch">...</div>
align-self-*-start	<div class="align-self-start">flex item align</div> <div class="align-self-md-start">...</div>
align-self-*-end	<div class="align-self-end">flex item align</div> <div class="align-self-md-end">...</div>
align-self-*-center	<div class="align-self-center">flex item align</div> <div class="align-self-md-center">...</div>

align-self-*-baseline	<div class="align-self-baseline">flex item align</div> <div class="align-self-md-baseline">...</div>
align-self-*-stretch	<div class="align-self-stretch">flex item align</div> <div class="align-self-md-stretch">...</div>
order-*-#	<div class="order-12">first item</div> <!--will be displayed 12th--> <div class="order-sm-3">ordered flex item</div>

Utility: Misc

close	<button type="button" class="close">
embed-responsive	<div class="embed-responsive embed-responsive-24by16"> <iframe class="embed-responsive-item" src="<video link>" allowfullscreen=""></iframe> </div>
shadow-none	<div class="shadow-none p-2 mb-5 bg-dark rounded">No shadow</div>
shadow-sm	<div class="shadow-sm p-2 mb-5 bg-light rounded">Small Shadow</div>
shadow	<div class="shadow p-2 mb-5 bg-light rounded">Default shadow setting</div>
shadow-lg	<div class="shadow-lg p-2 mb-5 bg-white">Larger shadow</div>
invisible	<div class="invisible">invisible element</div>
visible	<div class="visible">visible element</div>
sr-only	<span class="sr-only">Hide me everywhere except screen readers</span>
stretched-link	<a href="#" class="stretched-link">stretch the link to container</a>

Utility: Positioning

align-*	<span class="align-middle">middle</span>

clearfix	<div class="clearfix">
	   <span class="float-left">leftFloat</span>
	   <span class="float-right">rightFloat</span>
	</div>

fixed-top	<nav class="fixed-top">Fixed on page top</nav>
fixed-bottom	<nav class="fixed-bottom">Fixed on page bottom</nav>
sticky-top	<div class="sticky-top">Sticks on top</div>
float-*-left	<div class="float-sm-left">Float left on views small or wider</div> <div class="float-left">Float left on all view sizes</div>
float-*-right	<div class="float-md-right">Float right on views small or wider</div> <div class="float-right">Float right on all view sizes</div>
float-*-none	<div class="float-lg-none">no float on views small or wider</div> <div class="float-none"> no float on any view sizes</div>

Utility: Sizing

w-*	<div class="w-25 p-3" style="background-color: blue">Width 25% and size 3</div>
h-*	<div class="h-50 d-inline-block" style="background-color: red">Height is 50%</div>
mw-*	<div class="mw-100 bg-success">Max Width 100%</div>
mh-*	<div class="mw-75 bg-success">Max height 75%</div>

Utility: Spacing

m-1 / m-*-#	<div class="m-4 bg-success d-inline-block">div with 4 spaces</div>
mt-1 / mt-*-#	<div class="mt-3 bg-warning d-inline-block">Spaced top margin</div>
mr-1 / mr-*-#	<div class="mr-3 bg-warning d-inline-block">Spaced right margin</div>
mb-1 / mb-*-#	<div class="mb-3 bg-warning d-inline-block">Spaced bottom margin</div>
ml-1 / ml-*-#	<div class="ml-3 bg-warning d-inline-block">Spaced left margin</div>

`mx-1 / mx-*-#`	`<div class="mx-3 bg-primary d-inline-block">Spaced x axis margin</div>`
`my-1 / my-*-#`	`<div class="my-3 bg-primary d-inline-block">Spaced y axis margin</div>`
`m-n1 / m-*-n#`	`<div class="m-n5 bg-primary d-inline-block">Negative spacing</div>`
`p-1 / p-*-#`	`<div class="p-3 bg-danger d-inline-block">Padding on all sides with box size 3</div>`
`pt-1 / pt-*-#`	`<div class="pt-3 bg-danger d-inline-block">Padding on top with box size 3</div>`
`pr-1 / pr-*-#`	`<div class="pr-3 bg-danger d-inline-block">Padding on right with box size 3</div>`
`pb-1 / pb-*-#`	`<div class="pb-3 bg-danger d-inline-block">Padding on bottom with box size 3</div>`
`pl-1 / pl-*-#`	`<div class="pl-3 bg-danger d-inline-block">Padding on left with box size 3</div>`
`px-1 / px-*-#`	`<div class="px-3 bg-danger d-inline-block">Padding on left and right with box size 3</div>`
`py-1 / py-*-#`	`<div class="py-3 bg-danger d-inline-block">Padding on top and bottom with box size 3</div>`

Utility: Text

`font-weight-bold`	`<p class="font-weight-bold">Text in bold</p>`
`font-weight-bolder`	`<p class="font-weight-bolder">Bolder text</p>`
`font-weight-light`	`<p class="font-weight-light">Light text</p>`
`font-weight-lighter`	`<p class="font-weight-lighter">Lighter than light</p>`
`font-weight-normal`	`<p class="font-weight-normal">Normal weight </p>`
`text-decoration-none`	`<p class="text-decoration-none">No decorations on text</p>`
`font-italic`	`<p class="font-italic">Italic text (Ctrl+i) in word</p>`
`text-monospace`	`<p class="text-monospace">monospace text</p>`
`text-justify`	`<p class="text-justify">This is a justified text</p>`
`text-nowrap`	`<p class="text-nowrap">No wrapping</p>`
`text-*-left`	`<p class="text-sm-left">Left align text on viewports that are SM (small) or wider.</p>`
`text-*-right`	`<p class="text-md-right">right align text on viewports that are md (medium) or wider.</p>`
`text-*-center`	`<p class="text-lg-center">center align text on viewports that are lg (large) or wider.</p>`
`text-lowercase`	`<p class="text-lowercase">text in lowercase</p>`
`text-uppercase`	`<p class="text- uppercase">text in uppercase </p>`
`text-capitalize`	`<p class="text-capitalize">capitalized each word</p>`
`text-truncate`	`<p class="text-truncate">truncate this text while displaying because its long</p>`
`text-body`	`<p class="text-body">Default way of displaying text</p>`
`text-black-50`	`<p class="text-black-50">Text that's 50% opaque</p>`
`text-white-50`	`<p class="text-white-50">Text that's 50% opaque white</p>`
`text-muted`	Muted text. Muted link.
`text-hide`	`<h1 class="text-hide">Hide this text and show background image</h1>`
`text-break`	`<p class="text-break">very very long text, that needs to be wrapped to maintain alignment</p>`

A. Syntax

Pada file HTML, Javascript diaplikasikan dengan menggunakan `<script>` tag. Terdapat 2 macam penggunaan yaitu :

1. Embedded Script

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

   Pada browser dengan versi yang lebih baru `type="text/javascript"` tidak perlu dituliskan.

2. External Script

```
<script src="URL"></script>
```

   URL adalah Absolute atau relative path dan nama file javascript dengan ekstensi .js yang digunakan.

Pemograman pada JavaScript seperti halnya pemrograman pada bahasa yang lain menggunakan ketentuan dalam cara penulisan kode-kode programnya.

1. Tanda `//` digunakan untuk satu baris comment artinya baris dengan tanda `//` (seperti contoh di atas) tidak akan diproses oleh interpreter JavaScript.

2. Untuk beberapa baris comment digunakan tanda `/* ... */`

3. Curly brace → `{ ... }` merupakan tanda bahwa semua kode program yang ada diantaranya merupakan satu block kode.

4. Setiap baris kode diakhiri dengan tanda semicolon **;** walaupun bukan suatu keharusan terutama pada browser terbaru, sebaiknya setiap baris kode tetapi diakhiri dengan ; untuk kemudahan dalam membaca program dan debugging.

B. Pemrosesan Script

```
<!DOCTYPE html>
<html>
   <head>
      <script type="text/javascript">
         alert( 'First script Block ');
         alert( 'First script Block - Second Line ');
      </script>
   </head>
   <body>
      <h1>Test Page</h1>
      <script type="text/javascript">
         alert( 'Second script Block' );
      </script>
      <p>Some more HTML</p>
      <script type="text/JavaScript">
         alert( 'Third script Block' );
         function doSomething() {
            alert( 'Function in Third script Block' );
         }
      </script>
   </body>
</html>
```

Pada contoh kode di atas, browser akan memproses satu persatu setiap baris kode (statement) mulai dari baris paling awal hingga akhir, jika merupakan bagian dari HTML maka akan digunakan untuk membentuk DOM Tree dan jika bertemu dengan baris `<script>` maka (pada contoh di atas) proses pembentukan DOM Tree akan ditunda untuk menjalankan apa yang ada didalam `<script>`

Pada contoh di atas terdapat 3 `<script>` untuk menampilkan pop up message → `alert` Terdapat 5 `alert` yang dituliskan pada contoh di atas namun hanya akan ditampilkan sebanyak 4 `alert` karena `alert` yang terakhir berada didalam `function` dan dari seluruh baris kode diatas tidak ada baris kode yang ditulis untuk memanggil `function doSomething` tersebut.

`function` adalah suatu block code yang dibuat untuk melakukan proses tertentu yang akan sering dilakukan secara berulang. Karena akan sering dipanggil maka tentunya `function` harus diberi nama, pemberian nama `function` sebaiknya menggunakan camelCase notation seperti contoh di atas yaitu `doSomething` dan nama yang diberikan sebaiknya mencerminkan proses yang dilakukan. Setelah nama kemudian ditambahkan tanda `()`. Fungsi tanda `()` adalah untuk mengapit parameter yang digunakan didalam function. Misalnya `doSomething(`*`param1, param2, … ,paramN`*`)`

C. Variabel

Variabel adalah suatu lokasi penyimpanan di dalam memory komputer yang bersifat sementara digunakan untuk menyimpan suatu nilai. Variable juga diberi nama dan bersifat case sensitive, yang dapat dimulai dengan huruf atau underscore dan dilanjutkan dengan huruf atau angka. Variabel `aVar` dan `Avar` adalah 2 variabel yang berbeda.

Terdapat 3 jenis variable pada JavaScript yaitu :

1. `var` : keyword `var` digunakan untuk mendeklarasikan suatu variabel yang dapat memiliki scope global (variabel dideklarasikan diluar funtion) dan scope function/local (variabel dideklarasikan didalam suatu function). Variabel dengan scope global dapat digunakan didalam function tetapi variable dengan scope function/local tidak dapat digunakan diluar function dimana variabel tersebut dideklarasikan.
`var` variabel dapat dideklarasikan ulang dan dapat diubah-ubah nilainya.

Global Scope	Local Scope	Hoisting

`<script>`
`  var a = 10`
`  function f(){`
`    console.log(a)`
`  }`
`  f();`
	`<script>`
	`  function f() {`
	`    var a = 10;`
	`    console.log(a)`
	`  }`
	`  f();`
		`console.log (greet);`
		`var greet = "say hello"`
		Interpreter akan merubahnya menjadi :

`console.log(a);`
`</script>`
 	
Output:
`10`
`10`
	`console.log(a); //error undefined`
	`</script>`
	
	Output :
	`10`
	`ReferenceError: a is not defined`
		`var greet;`
		`console.log(greet); // error undefined`
		`greet = "say hello"`

Hoisting adalah mekanisme mengangkat scope dari variable dan function ke scope yang lebih tinggi sebelum kode dieksekusi.

2. `let` : keyword `let` digunakan untuk memperbaiki kelemahan pada `var` yaitu `let` hanya memiliki block scope jadi hanya berlaku dimana `let` variable dideklarasikan didalam block `{ … }`

```
    let greeting = "Hi";
    if (true) {
        let greeting = "Apa Kabar";
        console.log(greeting); // menampilkan Apa Kabar
    }
    console.log(greeting); // menampilkan Hi
```

Contoh di atas, `greeting` merupakan 2 variabel yang berbeda karena didefinisikan di block scope yang berbeda.
Variabel `let` dapat diubah-ubah nilainya tetapi tidak dapat dideklarasikan ulang.

3. `const` : variable `const` sama seperti variabel `let` kecuali nilai variable `const` tidak dapat dirubah.

** deklarasi variable tanpa menggunakan keyword `var`, `let` atau `const` akan selalu dianggap sebagai global variabel walaupun dideklarasikan didalam suatu `function`

D. Tipe Data

Terdapat dua jenis tipe data yang digunakan oleh variabel pada JavaScript yaitu :

1. Tipe Primitif

- String : tipe data yang dapat menampung huruf dan angka. Misal : "Test 123 percobaan"
- Number : menampung bilangan bulat maupun desimal. Misal : 123 123.45 JavaScript selalu menggunakan 64 bit floating number.
- Boolean : hanya memiliki dua nilai yaitu true dan false
- `null` : `null` sebenarnya bukan tipe data tapi digunakan untuk menentukan suatu variabel tidak memiliki nilai artinya nilai variabel tersebut adalah `null` atau nothing.
- `Undefined` : nilai variabel belum didefinisikan walaupun variabel tersebut sudah dideklarasikan.

2. Tipe Komposit : adalah tipe data yang nilai nya tidak hanya satu tetapi bersifat majemuk.

- Array : tipe data yang terdiri dari pasangan key dan value. Key merupakan nomor index dari array dan value adalah tipe data yang digunakan. Nomor index selalu dimulai dari angka `0`.

  Terdapat 3 jenis array dalam bahasa pemrograman secara umum yaitu :

  - Indexed Array
  - Multidimension Array
  - Associative Array

  Yang ada pada JavaScript adalah Indexed Array seperti contoh dibawah ini, Multidimension Array pada JavaScript adalah array didalam array sedangkan Associative Array pada JavaScript dikenal sebagai Object.

  Array dapat dideklarasikan dengan beberapa cara :

```
<script>
    var myArray = [1, 2, 3];
    var yourArray = ["red", "blue", "green"];
    var initArray = new Array( "First item", "Second item", "Third Item" );
    var preArray = new Array( 5 );
    var anArray = new Array();

    document.write(myArray[1] + "<br>");
    document.write(yourArray[3] + "<br>");
    document.write(initArray[2] + "<br>");

    preArray[5] = "Nomor urut 5";
    anArray[0] = "Dimulai dari 0";
    document.write(preArray[5] + "<br>");
    document.write("Jumlah set preArray = " + preArray.length +"<br>");
    document.write(anArray[0] + "<br>");
 </script>
```

  Output :

```
2
undefined
Third Item
Nomor urut 5
Jumlah set preArray = 6
Dimulai dari 0
```

  Selain key berupa index number, juga dapat digunakan keyword disebut dengan associative array. Tapi harus diingat bahwa hal ini seharusnya tidak dilakukan karena bisa mengacaukan index array.

```
        yourArray["sky"] = "blue";
        document.write(yourArray["sky"] + "<br>");  // blue

        var fruit = "dragon"
        myArray[fruit] = "purple"
        document.write(myArray["dragon"] + "<br>");  // purple
```

  Jika ingin menggunakan keyword maka gunakan Object.

- Object : Object adalah container dari berbagai macam tipe data. Pada OOP, object diasosiasikan sebagai manifestasi dari suatu benda didunia nyata, misalnya kendaraan.

Gambar kendaraan disebelah ini yaitu Fiat Multipla adalah suatu Object. Sebagai suatu object, kendaraan memiliki berbagai macam data misalnya :

Brand	: Fiat
Type	: Multipla
Color	: Gold
Wheel	: 4
Transmission	: Automatic

Transmission sendiri merupakan suatu object juga karena memiliki berbagai macam data didalamnya. Selain itu untuk mengoperasikan kendaraan diperlukan beragam fungsi seperti fungsi untuk menyalakan mesin, menyalakan lampu dan sebagainya.

Dari kumpulan data tersebut maka suatu object dideklarasikan. Pada JavaScript, salah satu cara pendeklarasian suatu object menggunakan syntax :

```
const car= {
  brand : "Fiat",
  type : "Multipla",
  color : "Gold",
  wheel : 4,
  gearbox : 1,
  transmission : function() {
    return this.gearBox===0 ? "Manual" : "Automatic" ;
  }
}
```

`brand`, `type`, `color`, `wheel` dan `gearbox` disebut dengan property dimana setiap property memiliki value. Sedangkan `transmission` disebut dengan method yaitu suatu `function`. this adalah keyword bukan variabel, pada contoh ini `this` digunakan untuk merujuk suatu property pada object yaitu `car`. Jadi `this.gearbox` adalah sama dengan `car.gearbox`

Perhatikan dengan cermat contoh penggunaan object dibawah ini :

```
<!DOCTYPE html>
<html>
   <body>
      <p id="demo"></p>
      <script>
      let light = false; //Light Off
      const car = {
         brand : "Fiat",
         type : "Multipla",
         color : "Gold",
         wheel : 4,
         gearbox : 1,
          gearshift : ["N", 1, 2, 3, "R"],
          fuse : true,
          headLight : function(status) {
             return status && this.fuse ? "Light ON" : "Light OFF";
          },
          transmission : function() {
              return this.gearbox===0 ? "Manual" : "Automatic" ;
           }
      }

      car.gearbox = 0;
      light = true;
      document.write(car.transmission() + "<br>");
      document.write(car.headLight(light) + "<br>");
      document.write(car.gearshift[4] + "<br>");
      document.write(car.headLight);

      let txt = "";
      for (let x in car) {
        txt += car[x] + "<br>";
      }

      document.getElementById("demo").innerHTML = txt;
      </script>
   </body>
</html>
```

Output :

```
Fiat
Multipla
Gold
4
0
N,1,2,3,R
true
function(status) {return status && this.fuse ? "Light ON" : "Light
OFF";}
function() {return this.gearbox===0 ? "Manual" : "Automatic" ;}

Manual
Light ON
R
function(status) {return status && this.fuse ? "Light ON" : "Light
OFF";}
```

** Penulisan `car.gearbox` atau `this.fuse` misalnya dapat digantikan dengan `car["gearbox"]` atau `this["fuse"]` tetapi untuk konsistensi dengan bahasa pemrograman yang lain biasakan untuk tetap menggunakan tanda **.** Sedangkan tanda [...] digunakan untuk array.

E. Event

Event adalah peristiwa atau tindakan yang terjadi dari suatu tahapan proses komputasi yang dapat dideteksi atau dikenali oleh software baik karena tindakan yang dilakukan oleh user seperti menekan tombol keyboard, menggerakan mouse, input dari sensor atau dari systemnya sendiri misal proses loading halaman web yang memiliki beberapa tahapan.

Suatu Event ini jika dikehendaki akan dibuatkan Event Listener atau kadang disebut Event Handler dan akan menjadi trigger (pemicu) agar Event Handler tersebut diproses. Event Handler sendiri berisi kode program yang digunakan untuk menangani terjadinya suatu Event. Contoh di bawah ini menggunakan event `onclick` pada `<button>`

```
<!DOCTYPE html>
<html>
   <body>
      <script>
      function clickButton(){
         return document.getElementById('demo').innerHTML="Button #3 click";
      }
      </script>

      <button onclick="document.getElementById('demo').innerHTML='Apa Kabar'">1.
Click me</button>
      <button onclick="this.innerHTML='Click me again'">2. Click me</button>
      <button onclick="clickButton()">3. Click me</button>

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

Khusus untuk menangani error yang terjadi pada sistem ketika sedang dieksekusi, menggunakan event handler `try … catch … finally … throw`

`try … catch` digunakan untuk menentukan block statement yang akan ditangani event error nya

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

<script>
   try {
      adddlert("Welcome guest!");
   }
   catch(err) {
      document.getElementById("demo").innerHTML = err.message;
   }
</script>
```

Output :

```
adddlert is not defined
```

`throw` digunakan untuk membuat error message sendiri

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

<script>
   let x = 9;
   try {
      if (x > 5) throw "Too High";
   }
   catch(err) {
      document.getElementById("demo").innerHTML =  "Error : " + err;
   }
</script>
```

Output :

```
Error : Too High
```

`finally` digunakan sebagai pengalih jika terjadi error

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

<script>
   try {
      adddlert("Welcome guest!");
   }
   catch(err) {
      document.getElementById("demo").innerHTML = err.message;
   }
   finally {
      document.getElementById("demo").innerHTML = "Yang benar adalah 
alert(\"Welcome guest!\")";
   }
</script>
```

Output :

```
Yang benar adalah alert("Welcome guest!")
```

** \" pada `Welcome guest!` disebut dengan escape character dari " karena harus ditampilkan sebagai " sedang syntax JavaScript tidak memungkinkannya.

F. Operator

JavaScript memiliki beberapa jenis operator seperti +, - , ==, && dan lain lain tetapi yang perlu diperhatikan adalah operator +

Perhatikan contoh di bawah ini :

```
let x = 5 + 5;          → 10
let y = "5" + 5;        → 55
let z = "Hello" + 5;    → Hello5
```

1. Arithmetic Operator

Operator	Keterangan	Contoh :		
+	Penambahan	x = y + 2	y = 5	x = 7
-	Pengurangan	x = y - 2	y = 5	x = 3
*	Perkalian	x = y * 2	y = 5	x = 10
/	Pembagian	x = y / 2	y = 5	x = 2.5
%	Modulus	x = y % 2	y = 5	x = 1
++	Increment	x = ++y Nilai y bertambah 1 x akan mendapat nilai sebelum y bertambah	y = 6	x = 6

		x = y++	y = 6	x = 5
		Nilai y bertambah 1 x akan mendapat nilai setelah y bertambah		
--	Decrement	x = --y	y = 4	x = 4
		x = y--	y = 4	x = 5

Supaya tidak membingungkan dan tidak melakukan kesalahan

```
let y = 7;
y++;   // y=8
--y;  // y=7
```

Jadi jangan pindahkan nilai y ke variable lain

** Perkalian, Pembagian dan Modulus akan dikerjakan lebih dahulu jika tidak ada tanda ( ) untuk menentukan prioritas perhitungan.

2. String Operator

Operator	Contoh	text1	text2	text3
+	text3 = text1 + text2	"Good "	"Morning"	"Good Morning"
+=	text1 += text2	"Good "	"Morning"	"Good Morning"

3. Comparison Operator

Operator	Keterangan		
==	sama	`x == 8`	false

===	sama baik value dan tipe	`let x == 5`
		`x === "5"`
			false

		`x === 5`	true
!=	tidak sama	`x != 8`	true
!==	tidak sama baik value maupun tipe	`x !== "5"`	true
		`x !== 5`	false
>	Lebih besar	`x > 8`	false
<	Lebih kecil	`x < 8`	true
>=	Lebih besar dan sama dengan	`x >= 8`	false
<=	Lebih kecil dan sama dengn	`x <= 8`	true

4. Ternary Operator

Syntax	Contoh
`variablename = (condition) ? Value1 : value2`	voteable = (age < 18) ? "Nope" : "Okay";

5. Logical Operator

Operator	Keterangan	Contoh : x = 6 dan y = 3	
`&&`	and	`(x < 10 && y > 1)`	→ true
`||`	or	`(x === 5 || y === 5)`	→ false
`!`	not	`!(x === y)`	→ true

6. Type Casting (Merubah tipe data)

`datatype` → string

`String(10)`	→ `"10"`
`(10).toString()`	→ `"10"`
`String(true)`	→ `"true"`
`true.toString()`	→ `"true"`
`String(false)`	→ `"false"`
`false.toString()`	→ `"false"`
`String(new Date('2019-01-22'))`	
`(new Date('2019-01-22')).toString()`	
`String(null)`	→ `"null"`
`String(undefined)`	→ `"undefined"`
`String(NaN)`	→ `"NaN"`

`datatype` → number

`Number("1")`	→ `1`
`Number("0")`	→ `0`
`Number(" 1 ")`	→ `1`
`Number("")`	→ `0`
`Number("12.2")`	→ `12.2`
`Number("10,000")`	→ `NaN`
`Number(true)`	→ `1`
`Number(false)`	→ `0`
`Number(null)`	→ `0`
`Number(undefined)`	→ `NaN`
`Number(NaN)`	→ `NaN`

G. Condition & Loop

1. Condition

a. `if`

```
if (condition) {
    // condition = true
}

if (condition) {
    // condition = true
} else {
    // condition = false
}
```

```
if (condition1) {
    // condition1 = true
} else if (condition2) {
    // condition1 = false and condition2 = true
} else {
    // condition1 = false and condition2 = false
}

if (time < 10) {
    greeting = "Good morning";
} else if (time < 20) {
    greeting = "Good day";
} else {
    greeting = "Good evening";
}
```

b. switch

```
switch(expression) {
case x:
    // code block
    break;
case y:
    // code block
    break;
default:
    // code block
}

switch (new Date().getDay()) {
case 0:
    day = "Sunday";
    break;
case 1:
    day = "Monday";
    break;
case 2:
    day = "Tuesday";
    break;
case 3:
    day = "Wednesday";
    break;
 case 4:
    day = "Thursday";
    break;
case 5:
    day = "Friday";
    break;
case 6:
   day = "Saturday";
}
```

2. Loop

a. for

```
for (statement1; statement2; statement3) {
    // code block
}

for (let i = 0; i < 5; i++) {
    text += "The number is " + i + "<br>";
}
```

- nilai awal `i` ditentukan sama dengan `0` kemudian loop akan berakhir ketika nilai `i` sama dengan 5 → proses loop akan terus terjadi selama nilai `i` < 5. Selama proses loop berlangsung, nilai `i` akan bertambah 1 dari nilai `i` terakhir.
- `statement1` bersifat optional dan juga sekaligus bisa memiliki beberapa value dipisahkan dengan comma → semicolon `;` sebagai delimiter harus tetap digunakan

  ** tanpa `statement1`

  ```
  let i = 2;
  let len = cars.length;
  let text = "";
  for (; i < len; i++) {
      text += cars[i] + "<br>";
  }
  ```

  ** multiple value

  ```
  for (let i = 0, len = cars.length, text = ""; i < len; i++) {
      text += cars[i] + "<br>";
  }
  ```

- `statement2` juga bersifat optional, jika tidak digunakan maka didalam blok kode harus ditambahkan keyword `break` untuk keluar dari proses loop. Karena `statement2` digunakan sebagai kodisi untuk menentukan keberlangsungan proses loop. Jika nilai `statement2 = true` maka proses loop akan terus berlangsung. Jika `statement2` tidak digunakan maka nilainya akan selalu true.
- *statement3* digunakan sebagai increment atau decrement dan juga bersifat optional. Jika tidak digunakan maka value increment atau decrement dilakukan didalam kode blok.
- Perhatikan jika terjadi Loop Scope seperti contoh ini :

  ```
  let i = 5;
  for (let i = 0; i < 10; i++) {
      // code block
  }
  // nilai i disini adalah 5 karena sebenarnya terdapat 2 variabel i
  yang berbeda scope

  var i = 5;
  for (var i = 0; i < 10; i++) {
      // code block
  }
  //nilai i disini adalah 10 karena sifat dari keyword var
  ```

b. for of → digunakan untuk melakukan loop pada object yang dapat dilakukan pengulangan seperti array, map, nodelist dan lain lain

```
const cars = ["BMW", "Volvo", "Mini"];
let text = "";
for (let x of cars) {
    text += x;
}
```

```
let language = "JavaScript";
let text = "";
for (let x of language) {
    text += x;
}
```

c. `for in` → digunakan untuk melakukan loop dari property suatu Object

```
for (key in object) {
    // code block
}

const car = {brand:"Fiat", type:"Multipla", color:"Gold"}
let text = "";
for (let x in car) {
    text += car[x];
}
```

- `for in` akan melakukan pengulangan dengan menggunakan key (x) . Key ini kemudian digunakan untuk mengakses property value dari object car
- walaupun dapat digunakan pada array, for in sebaiknya tidak digunakan karena ouputnya tidak dapat diprediksi.

d. `while` → digunakan untuk melakukan looping selama kondisi nya true.

```
while (condition) {
    // code block
}

while (i < 10) {
    text += "The number is " + i;
    i++;
}
```

- sebelum melakukan eksekusi code block, `condition` akan diperiksa terlebih dahulu.

e. `do while` → sama seperti `while` tetapi code block akan dieksekusi terlebih dahulu sebelum `condition` diperiksa.

```
do {
    // code block
}
while (condition);

do {
    text += "The number is " + i;
    i++;
}
while (i < 10);
```

H. Function

Terdapat beberapa cara function dideklarasikan

1. Declared function

```
function functionName(param1, … ,paramN) {
   // code to be executed
}
```

2. Expression

```
const x = function (a, b) {return a * b};
let z = x(4, 3);
```

3. Function Constructor

```
const myFunction = new Function("a", "b", "return a * b");
let x = myFunction(4, 3);
```

- Function tidak harus memiliki return statement, jika keyword `return` tidak digunakan maka return value dari function tersebut adalah `undefined`
- `parameter` dari function dapat memiliki default value

  ```
  function myFunction(x, y = 2) {
     // function code
  }
  ```
- jika `argument` lebih sedikit dari `parameter` yang disediakan maka `parameter` yang tidak memiliki `argument` akan bernilai `undefined`
- `parameter` tidak memiliki tipe data
- Jika nilai dari suatu `argument` diubah didalam suatu function maka nilai asli dari `parameter` tidak akan ikut berubah. Tetapi jika suatu object dijadikan argument maka apabila nilai property dari object diubah didalam function maka nilai asli dari property tersebut akan ikut berubah.

I. Best Practice

- nama variable menggunakan lowercase dan selalu dimulai dengan huruf
- `const` global variabel menggunakan UPPERCASE dan dimulai dengan huruf
- varaible String selalu diapit dengan "..."
- biasakan untuk selalu mendeklarasikan terutama global variabel diawal kode.
- gunakan `const` untuk mendeklarasikan object dan array untuk menghidari terjadi perubahan tipe data secara tidak sengaja.
- nama function menggunakan camelCase dan selalu dimulai dengan huruf
- nama function yang berhubungan dengan proses CRUD gunakan awalan `get`, `set` dan `del`
- gunakan spasi pada operator dan setelah tanda comma
- statement selalu diakhiri dengan ;
- selalu gunakan indentasi pada penulisan blok statement
- gunakan === untuk melakukan pembandingan.
- letakan <script> dibagian akhir file HTML untuk mempercepat proses loading atau menggunakan `defer`.

JavaScript Object BAB XVIII

Pada JavaScript segala sesuatunya adalah Object kecuali sedikit yaitu tipe data primitif tetapi String, Number dan Boolean tetap bisa diubah menjadi Object dengan menggunakan keyword `new` dan Object adalah variabel.

Pada contoh sebelumnya, Fiat Multipla dijadikan suatu object literal yaitu object yang dideklarasikan secara langsung seperti contoh tersebut. Cara lain dalam mendeklarasikan object literal adalah dengan cara membuat empty object dan menambahkan setiap property yang diperlukan seperti contoh ini :

```
const car= {};
car.brand = "Fiat";
car.type = "Multipla";
car.color = "Gold";
```

Selain secara literal, object dapat dideklarasikan dengan menggunakan keyword `new` seperti contoh ini :

```
const car= new Object();
car.brand = "Fiat";
car.type = "Multipla";
car.color = "Gold";
```

Object pada JavaScript bersifat mutable artinya value yang terkandung pada suatu property dapat diubah dan Object selalu direferensikan kepada property nya, pada contoh di atas adalah `brand`, `type` dan `color`.

`const xyz = car;` → statement ini tidak akan membuat object baru yaitu object `xyz` karena `xyz` tetap menggunakan property dari object `car`. Jadi jika `xyz.brand = "BMW"` maka `car.brand` juga menjadi `"BMW"`

Array dapat memiliki object didalamnya :

```
const cars = [
    {name:"Ford", models:["Fiesta", "Focus", "Mustang"]},
    {name:"BMW", models:["320", "X3", "X5"]},
    {name:"Fiat", models:["Panda", "Multipla"]}
]
document.write(cars[1].models[2]);     → X5
```

Suatu object juga dapat memiliki object dan array didalamnya :

```
const cars = {
    brand : {
        ford : ["Fiesta", "Focus", "Mustang"],
        bmw  : ["320", "X3", "X5"],
        fiat : ["500", "Panda"]},
    color : "Black"
}
```

```
document.write(cars.brand.ford[0]);    → Fiesta
```

Untuk menambahkan dan menghapus method atau property dari suatu object :

```
const cars = {
    brand : "Ford",
    color : "Black",
    gearbox : 0
}
// menambah property dan method
cars.wheel = 4;
cars.transmission = function() {return this.gearbox===0 ? "Manual" : "Automatic";} ;
// menghapus property dan method
delete cars.wheel;
delete cars.transmission;
```

Object Constructor adalah function yang digunakan untuk menciptakan object. Contructor digunakan dengan menggunakan keyword `new`. Setiap kali digunakan maka akan menciptakan object baru.

```
function car (brand, model, color, gearbox) {
    this.brand = brand;   // this digunakan untuk menciptakan property dan method
    this.model = model;
    this.color = color;
    this.gearbox = gearbox;
    this.transmission = function() {
        return this.gearbox===0 ? "Manual" : "Automatic";
    }
}

const ford = new car("Ford", "Fiesta", "Black", 0); //object baru
const bmw = new car("BMW", "X5", "White", 1);       //object baru
ford.wheel = 4; //menambah property wheel hanya pada object ford

document.write(ford.brand + " " + ford.transmission() + "<br>");
document.write(bmw.brand + " " + bmw.transmission() + "<br>");
document.write(ford.wheel + "<br>");
document.write(bmw.wheel);
```

Output :

```
Ford Manual
BMW Automatic
4
undefined
```

Untuk menambahkan property dan method pada object constructor menggunakan keyword property `prototype`

```
car.prototype.wheel = 4;
car.prototype.fuse = true;
car.prototype.headLight = function(status) {
            return status && this.fuse ? "Light ON" : "Light OFF";
         }
```

JavaScript Reference BAB XIX

Array Method

Method Keterangan

`concat()`

```
array1.concat(array2, array3, ..., arrayX)

const arr1 = ["Banana", "Orange"];
const arr2 = ["Apple", "Mango"];
const arr3 = ["Kiwi"];
const fruits = arr1.concat(arr2,arr3);
```

`copyWithin()`

```
array.copyWithin(target, start, end)

const fruits = ["Banana", "Orange", "Apple", "Mango", "Kiwi"];
fruits.copyWithin(2, 0, 2);
```

`entries()`

```
array.entries()

const fruits = ["Banana", "Orange", "Apple", "Mango", "Kiwi"];
const f = fruits.entries();
for (let x of f) {
   document.getElementById("demo").innerHTML += x;
} ;
```

`every()`

```
array.every(function(currentValue, index, arr), thisValue)

const ages = [32, 33, 16, 40];
ages.every(checkAge);
function checkAge(age) {
    return age > 18;
} ;
```

`fill()`

```
array.fill(value, start, end)

const fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.fill("Kiwi");

const fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.fill("Kiwi", 2, 4);
```

`filter()`

```
array.filter(function(currentValue, index, arr), thisValue)

const ages = [32, 33, 16, 40];
const result = ages.filter(checkAdult);
function checkAdult(age) {
    return age >= 18;
} ;
```

`find()`

```
array.find(function(currentValue, index, arr),thisValue)

const ages = [3, 10, 18, 20];
function checkAge(age) {
     return age > 18;
} ;
function myFunction() {
    document.getElementById("demo").innerHTML = ages.find(checkAge);
} ;
```

findIndex()

```
array.findIndex(function(currentValue, index, arr), thisValue)

const ages = [3, 10, 18, 20];
ages.findIndex(checkAge);
function checkAge(age) {
    return age > 18;
} ;
```

forEach()

```
array.forEach(function(currentValue, index, arr), thisValue)

const numbers = [65, 44, 12, 4];
numbers.forEach(myFunction)
function myFunction(item, index, arr) {
    arr[index] = item * 10;
} ;
```

from()

```
Array.from(object, mapFunction, thisValue)

Array.from("ABCDEFG")
```

includes()

```
array.includes(element, start)

const fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.includes("Banana", 3);
```

indexOf()

```
array.indexOf(item, start)

const fruits = ["Banana", "Orange", "Apple", "Mango", "Apple"];
let index = fruits.indexOf("Apple", 3);
```

isArray()

```
Array.isArray(obj)

const fruits = ["Banana", "Orange", "Apple", "Mango"];
let result = Array.isArray(fruits);
```

join()

```
array.join(separator)

const fruits = ["Banana", "Orange", "Apple", "Mango"];
let text = fruits.join(" and ");
```

keys()

```
array.keys()

const fruits = ["Banana", "Orange", "Apple", "Mango"];
const keys = fruits.keys();
let text = "";
for (let x of keys) {
    text += x + "<br>";
} ;
```

lastIndexOf()

```
array.lastIndexOf(item, start)

const fruits = ["Orange", "Apple", "Mango", "Apple", "Banana", "Apple"];
let index = fruits.lastIndexOf("Apple");
```

map()

```
array.map(function(currentValue, index, arr), thisValue)

const numbers = [65, 44, 12, 4];
const newArr = numbers.map(myFunction)

function myFunction(num) {
    return num * 10;
}
```

pop()

```
array.pop()

const fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.pop(); // remove last element
```

push()

```
array.push(item1, item2, ..., itemX)

const fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.push("Kiwi", "Lemon"); // add new elements
```

reduce()

```
array.reduce(function(total, currentValue, currentIndex, arr),
initialValue)
```

** reduce from left to right

```
const numbers = [15.5, 2.3, 1.1, 4.7];
document.getElementById("demo").innerHTML = numbers.reduce(getSum, 0);

function getSum(total, num) {
return total + Math.round(num);
} ;
```

reduceRight()

```
array.reduceRight(function(total, currentValue, currentIndex,
arr), initialValue)
```

** reduce from right to left

```
const numbers = [2, 45, 30, 100];
document.getElementById("demo").innerHTML = numbers.reduceRight(getSum);

function getSum(total, num) {
    return total - num;
} ;
```

reverse()

```
array.reverse()

const fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.reverse();
```

shift()

```
array.shift()
```

** remove first element

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.shift();
```

slice()

```
array.slice(start, end)

const fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];
const myBest = fruits.slice(-3, -1);
```

some()

```
array.some(function(value, index, arr), this)

const ages = [3, 10, 18, 20];
ages.some(checkAdult);
function checkAdult(age) {
    return age > 18;
}
```

sort()

```
array.sort(compareFunction)

const fruits = ["Banana", "Orange", "Apple", "Mango"];
```

```
fruits.sort();
fruits.reverse();
```

`splice()`

```
array.splice(index, howmany, item1, ....., itemX)
```

** at position 2, add 2 element

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.splice(2, 0, "Lemon", "Kiwi");
```

** at position 2, remove 2 element

```
const fruits = ["Banana", "Orange", "Apple", "Mango", "Kiwi"];
fruits.splice(2, 2);
```

`toString()`

```
array.toString()
```

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
let text = fruits.toString();
```

`unshift()`

```
array.unshift(item1, item2, ..., itemX)
```

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.unshift("Lemon","Pineapple");
```

`valueOf()`

```
array.valueOf()
```

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
const myArray = fruits.valueOf();
```

Array Property

Property — Keterangan

`constructor`

```
array.constructor
```

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
let text = fruits.constructor;
```

`length`

```
array.length
```

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
let length = fruits.length;
```

** set the length

```
const fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.length = 2;
```

`prototype`

```
Array.prototype.name = value
```

```
Array.prototype.myUcase = function() {
    for (let i = 0; i < this.length; i++) {
        this[i] = this[i].toUpperCase();
    }
}
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.myUcase();
```

Boolean Method

Method Keterangan

`toString()` *boolean*.toString()

```
let bool = true;
let text = bool.toString();
```

`valueOf()` *boolean*.valueOf()

```
let bool = false;
let value = bool.valueOf();
```

Boolean Property

Property Keterangan

`constructor` *boolean*.constructor

```
let bool = false;
let text = bool.constructor;
```

`prototype` Boolean.prototype.*name* = *value*

```
Boolean.prototype.myColor = function() {
   if (this.valueOf() == true) {
       return "green";
   } else {
       return = "red";
   }
}
let a = true;
let color = a.myColor();
```

Class Method

Method Keterangan

`constructor()` constructor(*parameters*)

```
class Car {
    constructor(brand) { // Constructor
        this.carname = brand;
    }
}
myCar = new Car("Ford");
```

Class Keyword

Keyword Keterangan

`extends` ** class inheritance

class *childClass* extends *parentClass*

```
class Car {
   constructor(brand) {
      this.carname = brand;
   }
   present() {
      return 'I have a ' + this.carname;
   }
} ;

class Model extends Car {
   constructor(brand, mod) {
      super(brand);
      this.model = mod;
   }
   show() {
      return this.present() + ', it is a ' + this.model;
   }
}

myCar = new Model("Ford", "Mustang");
document.getElementById("demo").innerHTML = myCar.show();
```

static

```
static methodName()

class Car {
   constructor(brand) {
      this.carname = brand;
   }
   static hello() {   // static method
      return "Hello!!";
   }
}

myCar = new Car("Ford");

// Call 'hello()' on the class Car
document.getElementById("demo").innerHTML = Car.hello();

// error call hello() from myCar
document.getElementById("demo").innerHTML = myCar.hello();



class Car {
   constructor(brand) {
      this.carname = brand;
   }
   static hello(x) {
      return "Hello " + x.carname;
   }
} ;

myCar = new Car("Ford");
// call hello() from myCar, by using parameter
document.getElementById("demo").innerHTML = Car.hello(myCar);
```

super

```
super(arguments); // calls the parent constructor (only
inside the constructor)
super.parentMethod(arguments); // calls a parent method

class Car {
   constructor(brand) {
      this.carname = brand;
   }
   present() {
      return 'I have a ' + this.carname;
   }
} ;

class Model extends Car {
   constructor(brand, mod) {
      super(brand);
      this.model = mod;
   }
   show() {
      return this.present() + ', it is a ' + this.model;
   }
} ;

mycar = new Model("Ford", "Mustang");
document.getElementById("demo").innerHTML = mycar.show();
```

Date Method

```
new Date();
new Date(milliseconds);
new Date(dateString);
new Date(year, month, day, hours, minutes, seconds, milliseconds);
```

UTC = Coordinated Universal Time

Method Keterangan

`getDate()`

** day of the month (from 1-31)

```
Date.getDate()

const d = new Date("July 21, 1983 01:15:00");
let day = d.getDate();
```

`getDay()`

** day of the week (from 0-6) Sunday = 0, Monday = 1 ...

```
Date.getDay()

const d = new Date();
let day = d.getDay();

const d = new Date("July 21, 1983 01:15:00");
let day = d.getDay();
```

`getFullYear()`

```
Date.getFullYear()

const d = new Date("July 21, 1983 01:15:00");
let year = d.getFullYear();
```

`getHours()`

** hour (from 0-23)

```
Date.getHours()

const d = new Date("July 21, 1983 01:15:00");
let hour = d.getHours();
```

`getMilliseconds()`

** milliseconds (from 0-999)

```
Date.getMilliseconds()

const d = new Date("July 21, 1983 01:15:00:526");
let ms = d.getMilliseconds();
```

`getMinutes()`

** minutes (from 0-59)

```
Date.getMinutes()
```

```
const d = new Date("July 21, 1983 01:15:00");
let minutes = d.getMinutes();
```

`getMonth()`

** month (from 0-11)

```
Date.getMonth()
```

```
const month =
["January","February","March","April","May","June","July","August",
"September","October","November","December"];

const d = new Date();
let name = month[d.getMonth()];
```

`getSeconds()`

** seconds (from 0-59)

```
Date.getSeconds()
```

```
const d = new Date();
let seconds = d.getSeconds();
```

`getTime()`

** milliseconds from Jan 1 1970 00:00 UTC

```
Date.getTime()
```

```
const d = new Date();
let time = d.getTime();
```

`getTimezoneOffset()`

** time difference between UTC time and local time, in minutes

```
Date.getTimezoneOffset()
```

```
const d = new Date();
let diff = d.getTimezoneOffset();
```

`getUTCDate()`

** day of the month, according to universal time (from 1-31)

```
Date.getUTCDate()
```

`getUTCDay()`

** day of the week, according to universal time (from 0-6)

```
Date.getUTCDay()
```

`getUTCFullYear()`

** year, according to universal time

```
Date.getUTCFullYear()
```

`getUTCHours()`

** hour, according to universal time (from 0-23)

```
Date.getUTCHours()
```

`getUTCMilliseconds()`

** milliseconds, according to universal time (from 0-999)

```
Date.getUTCMilliseconds()
```

`getUTCMinutes()`

** minutes, according to universal time (from 0-59)

```
Date.getUTCMinutes()
```

`getUTCMonth()`

** month, according to universal time (from 0-11)

```
Date.getUTCMonth()
```

`getUTCSeconds()`

** seconds, according to universal time (from 0-59)

```
Date.getUTCSeconds()
```

`getYear()`

Gunakan `getFullYear()`

`now()`

** milliseconds since Jan 1, 1970 00:00:00

```
Date.now()

let ms = Date.now();
```

`parse()`

** milliseconds since January 1, 1970

```
Date.parse(datestring)

const minute = 1000 * 60;
const hour = minute * 60;
const day = hour * 24;
const year = day * 365;
const d = Date.parse("March 21, 2012");
let years = Math.round(d / year);
```

`setDate()`

```
Date.setDate(day)

const d = new Date("July 21, 1983 01:15:00");
d.setDate(15);
```

`setFullYear()`

```
Date.setFullYear(year, month, day)

const d = new Date();
d.setFullYear(2020, 10, 3);

const d = new Date();
d.setFullYear(d.getFullYear(), d.getMonth() - 6);
```

`setHours()`

```
Date.setHours(hour, min, sec, millisec)

const d = new Date();
d.setHours(15, 35, 1);

const d = new Date();
d.setHours(d.getHours() - 48);
```

`setMilliseconds()`

```
Date.setMilliseconds(millisec)

const d = new Date();
d.setMilliseconds(192);
```

`setMinutes()`

```
Date.setMinutes(min, sec, millisec)

const d = new Date();
d.setMinutes(17);

const d = new Date();
d.setMinutes(d.getMinutes() - 90);
```

`setMonth()`

```
Date.setMonth(month, day)

const d = new Date();
d.setMonth(4);
```

`setSeconds()`

```
Date.setSeconds(sec, millisec)

const d = new Date();
d.setSeconds(35);
```

`setTime()`

** milliseconds after/before January 1, 1970

```
Date.setTime(millisec)

const d = new Date();
d.setTime(1332403882588);
```

`setUTCDate()` — `Date.setUTCDate(day)`

`setUTCFullYear()` — `Date.setUTCFullYear(year, month, day)`

`setUTCHours()` — `Date.setUTCHours(hour, min, sec, millisec)`

`setUTCMilliseconds()` — `Date.setUTCMilliseconds(millisec)`

`setUTCMinutes()` — `Date.setUTCMinutes(min, sec, millisec)`

`setUTCMonth()` — `Date.setUTCMonth(month, day)`

`setUTCSeconds()` — `Date.setUTCSeconds(sec, millisec)`

`setYear()` — Gunakan `setFullYear()`

`toDateString()` — `Date.toDateString()`

```
const d = new Date();
let text = d.toDateString();
```

`toGMTString()` — Gunakan `toUTCString()`

`toISOString()` — ** ISO standard date string

`Date.toISOString()`

```
const d = new Date();
let text = d.toISOString();
```

`toJSON()` — ** JSON string

`Date.toJSON()`

```
const d = new Date();
let text = d.toJSON();
```

`toLocaleDateString()` — `Date.toLocaleDateString()`

```
const d = new Date();
let text = d.toLocaleDateString();
```

`toLocaleTimeString()` — `Date.toLocaleTimeString()`

```
const d = new Date();
let text = d.toLocaleTimeString();
```

`toLocaleString()` — `Date.toLocaleString(locales, options)`

locales :

id-ID Indonesian (Indonesia)

options :

	Legal values
dateStyle	"full" "long" "medium" "short"
timeStyle	"full" "long" "medium" "short"
localeMatcher	"best-fit" (default) "lookup"

timeZone	hour12 false true
hourCycle	"h11" "h12" "h23" "h24"
formatMatcher	"basic" "best-fit" (default)
weekday	"long" "short" "narrow"
year	"2-digit" "numeric"
month	"2-digit" "long" "narrow" "numeric" "short"
day	"2-digit" "numeric"
hour	"2-digit" "numeric"
minute	"2-digit" "numeric"
second	"2-digit" "numeric"
timeZoneName	"long" "short"

```
const d = new Date();
let text = d.toLocaleString();
```

toString()

Date.toString()

```
const d = new Date();
let text = d.toString();
```

toTimeString()

Date.toTimeString()

```
const d = new Date();
let text = d.toTimeString();
```

toUTCString()

Date.toUTCString()

```
const d = new Date();
let text = d.toUTCString();
```

UTC()

Date.UTC(*year*, *month*, *day*, *hours*, *minutes*, *seconds*, *millisec*)

```
let ms = Date.UTC(2020, 02, 30);
```

valueOf()

Date.valueOf()

```
const d = new Date();
```

```
let ms = d.valueOf();
```

Date Property

Property Keterangan

constructor

```
Date.constructor

const d = new Date();
let text = d.constructor;
```

prototype

```
Date.prototype.name = value

Date.prototype.myMonth = function()
{
if (this.getMonth()==0) {return "January"};
if (this.getMonth()==1) {return "February"};
if (this.getMonth()==2) {return "March"};
if (this.getMonth()==3) {return "April"};
if (this.getMonth()==4) {return "May"};
if (this.getMonth()==5) {return "June"};
if (this.getMonth()==6) {return "July"};
if (this.getMonth()==7) {return "August"};
if (this.getMonth()==8) {return "September"};
if (this.getMonth()==9) {return "October"};
if (this.getMonth()==10) {return "November"};
if (this.getMonth()==11) {return "December"};
}
const d = new Date();
let month = d.myMonth();
```

Error Property

Property Keterangan

name

```
errorObj.name

name :
RangeError
ReferenceError
SyntaxError
TypeError
URIError

try {
   adddlert("Welcome guest!");
}
catch(err) {
   document.getElementById("demo").innerHTML = err.name;
}
```

message

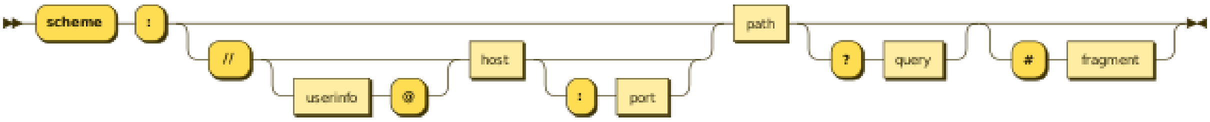
```
errorObj.message

try {
adddlert("Welcome guest!");
}
catch(err) {
```

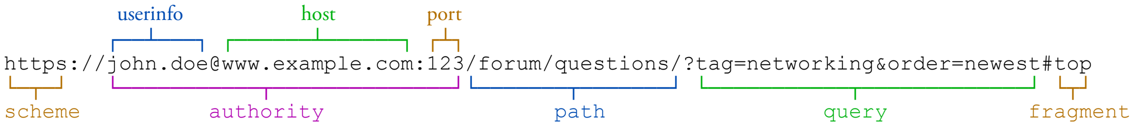
```
document.getElementById("demo").innerHTML = err.message;
}
```

JavaScript Global Method

URI = Uniform Resource Identifier

URI = scheme + URL (Uniform Resource Locator)




userinfo
host
port
https://john.doe@www.example.com:123/forum/questions/?tag=networking&order=newest#top
scheme
authority
path
query
fragment

Function Keterangan

decodeURI()

decodeURI(*uri*)

```
let uri = "my test.asp?name=ståle&car=saab";
let encoded = encodeURI(uri);
let decoded = decodeURI(encoded);
```

decodeURIComponent()

decodeURIComponent(*uri*)

```
let uri = "https://example.com/my test.asp?name=ståle&car=saab";
let encoded = encodeURIComponent(uri);
let decoded = decodeURIComponent(encoded);
```

encodeURI()

encodeURI(*uri*)

```
let uri = "my test.asp?name=ståle&car=saab";
let encoded = encodeURI(uri);
```

encodeURIComponent()

encodeURIComponent(*uri*)

```
let uri = "https://w3schools.com/my test.asp?name=ståle&car=saab";
let encoded = encodeURIComponent(uri);
```

escape()

Gunakan `encodeURI()` atau `encodeURIComponent()`

eval()

eval(*string*)

Jangan gunakan `eval()` karena menyebabkan resiko pada security, gunakan kode langsung atau function

```
let x = 10;
let y = 20;
let text = "x * y";
let result = eval(text); // security risk

let x = 10;
let y = 20;
let text = "x * y";
```

isFinite()

```
isFinite(value)

let result = isFinite(123);
```

isNaN()

```
NaN = Not-a-Number

isNaN('123');
isNaN('Hello');
isNaN('2005/12/12');
```

Number()

```
Number(value)

Number(true); // 1
Number(false); // 0
Number(new Date()); //milliseconds since January 1, 1970 00:00:00
```

parseFloat()

```
parseFloat(value)

parseFloat(10);             // 10
parseFloat("10");           // 10
parseFloat("10.33");        // 10.33
parseFloat("34 45 66");     // 34
parseFloat("He was 40");    // NaN
```

parseInt()

```
parseInt(string, radix)
radix :
```

2 = binary, 8 = octal, 10 = decimal (default), 16 = hexadecimal.

```
ParseInt("10");             // 10
parseInt("10.00");          // 10
parseInt("10.33");          // 10
parseInt("34 45 66");       // 34
parseInt(" 60 ");           // 60
parseInt("40 years");       // 40
parseInt("He was 40");      // NaN
```

String()

String(*value*)

```
String(new Date());
String("12345");
String(12345);
```

unescape()

Gunakan `decodeURI()` atau `decodeURIComponent()`

JavaScript Global Property

Property	Keterangan

`Infinity`	`Infinity adalah bilangan yang melebihi 1.797693134862315E+308.`
	`-Infinity adalah bilangan yang melebihi -1.797693134862315E+308.`

`NaN`	"Not-a-Number" value
`undefined`	Variable belum didefinisikan

```
let x;
if (x === "undefined") {
    text = "x is undefined";
} else {
    text = "x is defined";  // x is defined
}

let x;
if (typeof x === "undefined") {
    text = "x is undefined";  // x is undefined
} else {
    text = "x is defined";
}
```

JSON Method

Method	Keterangan

`parse()`	`JSON.parse(string, function)`
	function is used to transform the result
	
	`var obj = JSON.parse('{"firstName":"John", "lastName":"Doe"}');`
	`document.getElementById("demo").innerHTML = obj.firstName; // John`

`stringify()`	`JSON.stringify(obj, replacer, space)`
	
	`var obj = {"name":"John", "age":30, "city":"New York"};`
	`var myJSON = JSON.stringify(obj);`
	`document.getElementById("demo").innerHTML = myJSON;`
	
	`Output :`
	`{"name":"John","age":30,"city":"New York"}`

Math Object Method

Method	Keterangan
`abs(x)`	`let x = Math.abs(-7.25);`
`acos(x)`	`let x = Math.acos(0.5);`
`acosh(x)`	** hyperbolic arccosine of x `let x = Math.acosh(2);`
`asin(x)`	`let x = Math.asin(0.5);`
`asinh(x)`	** hyperbolic arcsine of x `let x = Math.asinh(1);`
`atan(x)`	** arctangent of x as a numeric value between -PI/2 and PI/2 radians
`atan2(y, x)`	** arctangent of the quotient of its arguments
`atanh(x)`	** hyperbolic arctangent of x
`cbrt(x)`	** cubic root of x
`ceil(x)`	** rounded upwards to the nearest integer
`clz32(x)`	** number of leading zeros in a 32-bit binary representation of x
`cos(x)`	** cosine of x (x is in radians)
`cosh(x)`	** hyperbolic cosine of x

exp(x)	** E^x, where E is Euler's number (approximately 2.7183)
expm1(x)	** value of Ex minus 1
floor(x)	** rounded downwards to the nearest integer
fround(x)	** nearest (32-bit single precision) float representation of a number
log(x)	** natural logarithmof x
log10(x)	** base-10 logarithm of x
log1p(x)	** natural logarithm of 1 + x
log2(x)	** base-2 logarithm of x
max(x, y, z, ..., n)	** number with the highest value
min(x, y, z, ..., n)	** number with the lowest value
pow(x, y)	** value of x to the power of y
random()	** random number between 0 and 1

```
Math.floor((Math.random() * 10) + 1);  // between 0 and 10
Math.floor((Math.random() * 100) + 1);  // between 0 and 100
```

round(x)	** roundsx to the nearest integer
sign(x)	** sign of a number (checks whether it is positive, negative or zero)
sin(x)	** sine of x (x is in radians)
sinh(x)	** hyperbolic sine of x
sqrt(x)	** square root of x
tan(x)	** tangent of an angle
tanh(x)	** hyperbolic tangent of a number
trunc(x)	** integer part of a number (x)

Math Object Property

Property	Keterangan
E	Euler's number (approx. 2.718)
LN2	natural logarithm of 2 (approx. 0.693)
LN10	natural logarithm of 10 (approx. 2.302)
LOG2E	base-2 logarithm of E (approx. 1.442)
LOG10E	base-10 logarithm of E (approx. 0.434)
PI	PI (approx. 3.14)
SQRT1_2	square root of 1/2 (approx. 0.707)
SQRT2	square root of 2 (approx. 1.414)

Number Method

Method	Keterangan

`isFinite()`	Checks whether a value is a finite number
`isInteger()`	Checks whether a value is an integer
`isNaN()`	Checks whether a value is Number.NaN
`isSafeInteger()`	Checks whether a value is a safe integer
`toExponential(x)`	Converts a number into an exponential notation
`toFixed(x)`	Formats a number with x numbers of digits after the decimal point
`toLocaleString()`	Converts a number into a string, based on the locale settings
`toPrecision(x)`	Formats a number to x length
`toString()`	Converts a number to a string
`valueOf()`	Returns the primitive value of a number

Number Property

Property	Keterangan
`constructor`	Returns the function that created JavaScript's Number prototype
`MAX_VALUE`	Returns the largest number possible in JavaScript
`MIN_VALUE`	Returns the smallest number possible in JavaScript
`NEGATIVE_INFINITY`	Represents negative infinity (returned on overflow)
`NaN`	Represents a "Not-a-Number" value
`POSITIVE_INFINITY`	Represents infinity (returned on overflow)
`prototype`	Allows you to add properties and methods to an object

Bitwise Operator

Bit operators hanya berfungsi pada 32 bits numbers. Pada saat operasi semua akan dikonversi ke 32 bit number. Kemudian hasilnya akan dikembalikan menjadi JavaScript number.

Operator	Description	Example	Same as	Result	Decimal
&	AND	x = 5 & 1	0101 & 0001	0001	1
|	OR	x = 5 | 1	0101 | 0001	0101	5
~	NOT	x = ~ 5	~0101	1010	10
^	XOR	x = 5 ^ 1	0101 ^ 0001	0100	4
<<	Left shift	x = 5 << 1	0101 << 1	1010	10
>>	Right shift	x = 5 >> 1	0101 >> 1	0010	2

JavaScript Statement Identifier

Statement Keterangan

break	Exit dari switch atau loop
class	Deklarasi class
const	Deklarasi variabel dengan nilai konstan
continue	Breaks one iteration (in the loop) if a specified condition occurs, and continues with the next iteration in the loop
debugger	Stops the execution of JavaScript, and calls (if available) the debugging function
do ... while	Executes a block of statements and repeats the block while a condition is true
for	Loops through a block of code a number of times
for ... in	Loops through the properties of an object
for ... of	Loops through the values of an iterable object
function	Declares a function
if ... else ... else if	Marks a block of statements to be executed depending on a condition
let	Deklarasi variabel
return	Stops the execution of a function and returns a value from that function
switch	Marks a block of statements to be executed depending on different cases
throw	Throws (generates) an error
try ... catch ... finally	Marks the block of statements to be executed when an error occurs in a try block, and implements error handling
var	Deklarasi variabel
while	Marks a block of statements to be executed while a condition is true

String Method

Method	Keterangan

charAt()	`string.charAt(index)`
	
	`let text = "HELLO WORLD";`
	`let letter = text.charAt(text.length-1);`

charCodeAt()	`string.charCodeAt(index) // Unicode`
	
	`let text = "HELLO WORLD";`
	`let code = text.charCodeAt(1);`

concat()	`string.concat(string1, string2, ..., stringN)`
	
	`let text1 = "Hello";`
	`let text2 = "world!";`
	`let text3 = "Have a nice day!";`
	`let result = text1.concat(" ", text2, " ", text3);`

endsWith()	`string.endsWith(searchvalue, length)`
	
	`let text = "Hello world, welcome to the universe.";`
	`text.endsWith("world", 11);`

fromCharCode()	`String.fromCharCode(n1, n2, ..., nX)`

```
let text = String.fromCharCode(72, 69, 76, 76, 79); //HELLO
```

includes() *string*.includes(*searchvalue, start*)

```
let text = "Hello World, welcome to the universe.";
let result = text.includes("world", 12);
```

indexOf() *string*.indexOf(*searchvalue, start*)

```
let text = "Hello world, welcome to the universe.";
text.indexOf("e", 5);
```

lastIndexOf() *string*.lastIndexOf(*searchvalue, start*)

```
let text = "Hello planet earth, you are a great planet.";
let result = text.lastIndexOf("Planet"); // -1
```

localeCompare() *string*.localeCompare(*compareString*)

```
let text1 = "ab";
let text2 = "cd";
let result = text1.localeCompare(text2);
```

match() *string*.match(*match*)

```
let text = "The rain in SPAIN stays mainly in the plain";
text.match("ain");
```

repeat() *string*.repeat(*count*)

```
let text = "Hello world!";
let result = text.repeat(4);
```

replace() *string*.replace(*searchValue, newValue*)

```
let text = "Visit Microsoft!";
let result = text.replace("Microsoft", "Oracle");
```

search() *string*.search(*searchValue*)

```
let text = "Mr. Blue has a blue house";
let position = text.search("blue");
```

slice() *string*.slice(*start, end*)

```
let text = "Hello world!";
let result = text.slice(0, 5);
```

split() *string*.split(*separator, limit*)

```
let text = "How are you doing today?";
const myArray = text.split(" ");
let word = myArray[1];
```

startsWith() *string*.startsWith(*searchValue, start*)

```
let text = "Hello world, welcome to the universe.";
text.startsWith("world", 7);
```

substr() *string*.substr(*start, length*)

```
let text = "Hello world!";
let result = text.substr(1, 4);"
```

substring() *string*.substring(*start, end*)

```
let text = "Hello world!";
let result = text.substring(1, 4);
```

toLocaleLowerCase()

string.toLocaleLowerCase()

```
let text = "Hello World!";
let result = text.toLocaleLowerCase();
```

toLocaleUpperCase()

string.toLocaleUpperCase()

```
let text = "Hello World!";
let result = text.toLocaleUpperCase();
```

toLowerCase()

string.toLowerCase()

```
let text = "Hello World!";
let result = text.toLowerCase();
```

toString()

string.toString()

```
let text = "Hello World!";
let result = text.toString();
```

toUpperCase()

string.toUpperCase()

```
let text = "Hello World!";
let result = text.toUpperCase();
```

trim()

string.trim()

```
let text = "   Hello World!   ";
let result = text.trim();
```

valueOf()

string.valueOf()

```
let text = "Hello World!";
let result = text.valueOf();
```

String Property

Property   Keterangan

constructor

string.constructor

```
let message = "Hello World!";
let text = message.constructor;
```

length

string.length

```
let text = "Hello World!";
let length = text.length;
```

prototype

object.prototype.*name* = *value*

```
function employee(name, jobtitle, born) {
   this.name = name;
   this.jobtitle = jobtitle;
   this.born = born;
}
employee.prototype.salary = 2000;

const fred = new employee("Fred Flintstone", "Caveman", 1970);
```

HTML DOCUMENT OBJECT BAB XX

HTML DOM

Property / Method — Keterangan

`activeElement`

`document.activeElement`

```
const element = document.activeElement.tagName;
```

`addEventListener()`

`document.addEventListener(`*`event, function, Capture`*`)`
Event name. Jangan gunakan "on" prefix : "onclick" → "click"
`capture` Optional (default = false).
true – dieksekusi pada capturing phase.
false – dieksekusi pada bubbling phase.

```
document.addEventListener("click", myFunction);

function myFunction() {
   document.getElementById("demo").innerHTML = "Hello World";
}

document.addEventListener("click", function(){
   document.getElementById("demo").innerHTML = "Hello World";
});
```

`adoptNode()`

`document.adoptNode(`*`node`*`)`
`node` Required.
Node yang berasal dari document lain.

```
const frame = document.getElementById("myFrame");
const h1 = frame.contentWindow.document.getElementsByTagName("H1")[0];
const node = document.adoptNode(h1);
```

`anchors` Deprecated

`applets` Deprecated

`baseURI`

`document``.baseURI`

```
let base = document.baseURI;
```

`body`

`document.body`

`document.body = `*`newContent`*

```
const myBody = document.body.innerHTML;

document.body.style.backgroundColor = "yellow";
```

`charset` Deprecated

`characterSet`

`document.characterSet`

```
encoding = document.characterSet;
```

`close()`

`document.close()`

```
const myWindow = window.open();
myWindow.document.open();
myWindow.document.write("<h1>Hello World!</h1>");
myWindow.document.close();
```

`cookie`

`document.cookie`

document.cookie = *newCookie*

```
let allCookies = document.cookie;

document.cookie="username=John Doe; expires=Thu, 18 Dec 2013 12:00:00
UTC; path=/";
```

createAttribute()

document.createAttribute(*name*)

```
// Create a style attribute:
const att = document.createAttribute("style");

// Set the value of the style attribute:
att.value = "color:red";

// Add the style attribute to the first h1:
const h1 = document.getElementsByTagName("h1")[0];
h1.setAttributeNode(att);
```

createComment()

document.createComment(*text*)

```
const comment = document.createComment("My comments");
document.body.appendChild(comment);
```

createDocumentFragment()

document.createDocumentFragment()

```
const fruits = ["Banana", "Orange", "Mango"];

// Create a document fragment:
const dFrag = document.createDocumentFragment();

// Add li elements to the fragment:
for (let x in fruits) {
  const li = document.createElement('li');
  li.textContent = fruits[x];
  dFrag.appendChild(li);
}

// Add fragment to a list:
document.getElementById('myList').appendChild(dFrag);
```

createElement()

document.createElement(*type*)

```
const para = document.createElement("p");
para.innerHTML = "This is a paragraph.";
document.getElementById("myDIV").appendChild(para);
```

createEvent()

document.createEvent(*type*)

```
type :  AnimationEvent
        ClipboardEvent
        DragEvent
        FocusEvent
        HashChangeEvent
        InputEvent
        KeyboardEvent
        MouseEvent
        PageTransitionEvent
        PopStateEvent
        ProgressEvent
        StorageEvent
        TouchEvent
        TransitionEvent
        UiEvent
        WheelEvent
```

```
const ev = document.createEvent("MouseEvent");
ev.initMouseEvent("mouseover", true, true, window, 0, 0, 0, 0, 0,
false, false, false, false, 0, null);

document.getElementById("myDiv").dispatchEvent(ev);
```

`createTextNode()`

`document.createTextNode(`*`text`*`)`

```
const para = document.createElement("p");
const textNode = document.createTextNode("Hello World");
para.appendChild(textNode);
```

`defaultView`

`document.defaultView`

```
const view = document.defaultView;
let width = view.innerWidth;
let height = view.innerHeight;
```

`designMode`

Get:
`document.designMode`

Set:
`document.designMode = "on|off"`

```
document.designMode;
document.designMode = "on";
```

`doctype`

`document.doctype`

```
const doctypeObj = document.doctype.name;
```

`documentElement`

`document.documentElement`

```
document.documentElement.nodeName;
```

`documentMode`

Deprecated

`documentURI`

Get :
`document.documentURI`

Set :
`document.documentURI = locationURI`

```
let uri = document.documentURI;
```

`domain`

`document.domain`

```
let myDomain = document.domain;
```

`domConfig`

Deprecated

`embeds`

`document.embeds`

```
let num = document.embeds.length;
```

`execCommand()`

Deprecated

`forms`

`document.forms`

Property	
`length`	The number of elements in the collection.

Method	Description
`[`*`index`*`]`	Returns the element with the specified index (starts at 0). Returns null if the index is out of range.
`item(`*`index`*`)`	Returns the element with the specified index (starts at 0). Returns `null` if the index is out of range.
`namedItem(`*`id`*`)`	Returns the element with the specified id.

Returns `null` if the id does not exist.

```
let num = document.forms.length;
let id = document.forms[0].id;
let id = document.forms.item(0).id;
let html = document.forms.namedItem("myCarForm").innerHTML;
```

`getElementById()`

```
document.getElementById(elementID)
```

```
const myElement = document.getElementById("demo");
myElement.style.color = "red";

document.getElementById("demo").style.color = "red";
```

`getElementsByClassName()`

```
document.getElementsByClassName(classname)
```

get class `example` and `color`

```
const collection = document.getElementsByClassName("example color");
```

`getElementsByName()`

Deprecated

`getElementsByTagName()`

```
document.getElementsByTagName(tagname)
```

```
document.getElementsByTagName("p")[0].innerHTML = "Hello World!";
```

`hasFocus()`

```
document.hasFocus()
```

```
if (document.hasFocus()) {
   text = "The document has focus.";
} else {
   text = "The document does NOT have focus.";
}
```

`head`

```
document.head
```

```
document.head.firstElementChildtagName;
```

`images`

```
document.images
```

```
const myImages = document.images;
let text = "";
for (let i = 0; i < myImages.length; i++) {
   text += myImages[i].src + "<br>";
}
```

`implementation`

document.implementation

```
document.implementation.hasFeature("DOM", "1.0");
```

`importNode()`

```
document.importNode(node, deep)
deep : true | false
```

```
const frame = document.getElementsById("myFrame");
const h1 = frame.contentWindow.document.getElementsByTagName("H1")[0];
const node = document.importNode(h1, true);
```

`inputEncoding`

Deprecated

`lastModified`

```
document.lastModified
```

```
let text = document.lastModified;
const date = new Date(document.lastModified);
```

`links`

```
document.links
```

```
let numb = document.links.length;
let url = document.links[0].href;
let url = document.links.item(0).href;
```

`normalize()`

Removes empty Text nodes, and joins adjacent nodes

```
document.normalize()
```

```
document.normalize();
```

`normalizeDocument()`

Deprecated

`open()`

```
document.open()
```

```
const myWindow = window.open();
const myWindow.document.open();
const myWindow.document.write("<h1>Hello World!</h1>");
const myWindow.document.close();
```

`querySelector()`

```
document.querySelector(CSS selectors)
```

```
document.querySelector("p");
document.querySelector(".classexample");
```

`querySelectorAll()`

```
document.querySelectorAll(CSS selectors)
```

```
document.querySelectorAll(".example");
```

`readyState`

```
document.readyState
```

```
document.readyState;
```

`referrer`

Returns the URL of the document that loaded the current document

```
document.referrer
```

```
document.referrer;
```

`removeEventListener()`

```
document.removeEventListener(event, function,
capture)
```

```
document.removeEventListener("mousemove", myFunction);
```

`renameNode()`

Deprecated

`scripts`

```
document.scripts
```

```
document.scripts.length;
document.scripts[0].text;
document.scripts.item(0).text;
```

`strictErrorChecking`

Deprecated

`title`

```
document.title
document.title = newTitle
```

```
document.title;
document.title = "A new title";
```

`URL`

```
document.URL
```

```
document.URL;
```

`write()`

```
document.write(exp1, exp2, exp3, ...)
```

```
document.write("Hello World!");
document.write("<h2>Hello World!</h2><p>Have a nice day!</p>");
```

`writeln()`

```
document.writeln(exp1, exp2, exp3, ...)
```

```
<body>

<p>Note that write() does NOT add a new line after each statement:</p>
<pre>
<script>
document.write("color:mediumblue">"Hello World!");
document.write("color:mediumblue">"Have a nice day!");
</script>
</pre>

<p>Note that writeln() add a new line after each statement:</p>
<pre>
<script>
document.writeln("color:mediumblue">"Hello World!");
document.writeln("color:mediumblue">"Have a nice day!");
</script>
</pre>

</body>
```

Element Object

Property / Method — Keterangan

`accessKey`

element.accessKey
element.accessKey = *character*

```
document.getElementById("myAnchor").accessKey = "w";
document.getElementById("myAnchor").accessKey;
```

`addEventListener()`

element.addEventListener(*event*, *function*, *useCapture*)
useCapture : true | false

```
element.addEventListener("click", function() {
   document.getElementById("demo").innerHTML = "Hello World";
});
```

`appendChild()`

element.appendChild(*node*)

```
const node = document.createElement("li");
const textnode = document.createTextNode("Water");
node.appendChild(textnode);
document.getElementById("myList").appendChild(node);
```

`attributes`

node.attributes

```
const nodeMap = document.getElementById("myImg").attributes;
let text = "";
for (let i = 0; i < nodeMap.length; i++) {
  text += nodeMap[i].name + " = " + nodeMap[i].value + "<br>";
}
document.getElementById("demo").innerHTML = text;
```

`blur()`

removes focus from an element
HTMLElementObject.blur()

```
document.getElementById("myAnchor").blur();
```

`childElementCount`

element.childElementCount

```
let numb = document.getElementById("myDIV").childElementCount;
```

`childNodes`

element.childNodes

```
const nodeList = document.body.childNodes;
let numb = document.getElementById("myDIV").childNodes.length;
```

children

```
element.children

const collection = document.body.children;
```

classList

```
element.classList

const list = element.classList;
list.add("myStyle");

const list = element.classList;
list.remove("myStyle");

const list = element.classList;
list.toggle("myStyle");
```

className

```
HTMLElementObject.className

HTMLElementObject.className = class

element.className = "myStyle";

let value = document.getElementById("myDIV").className;

if (element.className == "myStyle") {
   element.className = "newStyle";
} else {
  element.className = "myStyle";
}
```

click()

```
element.click()

<input type="checkbox" id="myCheck" onmouseover="myFunction()";

<script>
function myFunction() {
  document.getElementById("myCheck").click();
}
</script>
```

clientHeight

```
element.clientHeight

const element = document.getElementById("myDIV");
let text = "clientHeight: " + element.clientHeight + "px<br>";
text += "clientWidth: " + element.clientWidth + "px";
```

clientLeft

```
element.clientLeft

const elmnt = document.getElementById("myDIV");
let text = "Border top width: " + elmnt.clientTop + "px<br>";
text += "Border left width: " + elmnt.clientLeft + "px";
```

clientTop

```
element.clientTop

const element = document.getElementById("myDIV");
let text = "clientTop: " + element.clientTop + "px<br>";
text += "clientLeft: " + element.clientLeft + "px";
```

clientWidth

```
element.clientWidth

const element = document.getElementById("myDIV");
let text = "clientHeight: " + element.clientHeight + "px<br>";
text += "clientWidth: " + element.clientWidth + "px";
```

cloneNode()

```
node.cloneNode(deep)
deep : true | false
```

```
const node = document.getElementById("myList2").lastChild;
const clone = node.cloneNode(true);

document.getElementById("myList1").appendChild(clone);
```

closest()

```
element.closest(selectors)

const element = document.getElementById("myElement");
const closest = element.closest(".container");

const element = document.getElementById(".container, .wrapper");
const closest = element.closest(".container");
```

compareDocumentPosition()

```
node1.compareDocumentPosition(node2)
ReturnValue
1       Kedua node tidak berada pada dokumen yang sama.
2       posisi node1 berada setelah node2
4       posisi node1 berada sebelum node2
8       posisi node1 berada didalam node2
16      posisi node2 berada didalam node1
32      kedua node merupakan attribute dari element yang sama

const p1 = document.getElementById("p1");
const p2 = document.getElementById("p2");
let position = p1.compareDocumentPosition(p2);
```

contains()

```
node.contains(node)

const span = document.getElementById("mySPAN");
let answer = document.getElementById("myDIV").contains(span);
```

contentEditable

```
element.contentEditable

ement.contentEditable = value

value
"true" - The content is editable
"false" - The content is not editable
"inherit"- Default. Is editable if parent element is editable

<p id="myP" contenteditable="true">I am editable.</p>

document.getElementById("myP").contentEditable;

document.getElementById("myP").contentEditable = "true";
```

dir

```
element.dir

element.dir = "ltr|rtl|auto"

document.getElementById("myP").dir = "rtl";

document.body.dir;

document.body.dir = "rtl";
```

exitFullscreen()

firstChild

```
element.firstChild
node.firstChild

document.getElementById("myList").firstChild.innerHTML;

let text = document.getElementById("mySelect").firstChild.text;
```

firstElementChild

```
element.firstElementChild

let text = element.firstElementChild.innerHTML;
let text = document.getElementById("myDIV").firstElementChild.tagName;
let text = document.getElementById("mySelect").firstElementChild.text;
```

focus()

```
element.focus()
```

```
document.getElementById("myAnchor").focus();
document.getElementById("myText").focus();
window.onload = function() {
   document.getElementById("myText").focus();
}
```

getAttribute()

```
element.getAttribute(name)
```

```
let text = element.getAttribute("class");
let text = myAnchor.getAttribute("target");
```

getAttributeNode()

```
element.getAttributeNode(name)
```

```
const element = document.getElementsByTagName("H1")[0];
let text = element.getAttributeNode("class").value;
```

getBoundingClientRect()

```
element.getBoundingClientRect()
```

```
const rect = element.getBoundingClientRect();
```

getElementsByClassName()

```
element.getElementsByClassName(classname)
```

```
const list = document.getElementsByClassName("example")[0];
list.getElementsByClassName("child")[0].innerHTML = "Milk";

const element = document.getElementById("myDIV");
const nodes = element.getElementsByClassName("child");
let number = nodes.length;

const element = document.getElementById("myDIV");
element.getElementsByClassName("child")[1].style.fontSize = 24px";
```

getElementsByTagName()

```
element.getElementsByTagName(tagname)
```

```
const list = document.getElementsByTagName("UL")[0];
list.getElementsByTagName("li")[0].innerHTML = "Milk";

const element = document.getElementById("myDIV");
const nodes = element.getElementsByTagName("p");
let numb = nodes.length;

const element = document.getElementById("myDIV");
element.getElementsByTagName("p")[1].style.fontSize = "24px";
```

hasAttribute()

```
element.hasAttribute(name)
```

```
let answer = myButton.hasAttribute("onclick");
if (element.hasAttribute("target")) {
  element.setAttribute("target", "_self");
}
```

hasAttributes()

```
element.hasAttributes()
node.hasAttributes()
```

```
let answer = document.body.hasAttributes();
```

hasChildNodes()

```
element.hasChildNodes()
```

```
let answer = element.hasChildNodes();
if (element.hasChildNodes()) {
   element.removeChild(element.childNodes[0]);
}
```

id

```
element.id
element.id = id

let id = document.getElementsByTagName("a")[0].id;
document.getElementById("demo").id = "newid";
const element = document.getElementById("myP");
element.style.fontSize = "30px";
```

innerHTML

```
element.innerHTML

element.innerHTML = text

let html = document.getElementById("myP").innerHTML;
document.getElementById("demo").innerHTML = "I have changed!";
let html = document.getElementById("myList").innerHTML;
element.innerHTML = "";
```

innerText

```
element.innerText
node.innerText

element.innerText = text
node.innerText = text

let text = element.innerText;
```

insertAdjacentElement()

```
element.insertAdjacentElement(position, element)
node.insertAdjacentElement(position, element)
position :  afterbegin
            afterend
            beforebegin
            beforeend

const span = document.getElementById("mySpan");
const h2 = document.getElementById("myH2");
h2.insertAdjacentElement("afterend", span);
```

insertAdjacentHTML()

```
element.insertAdjacentHTML(position, html)
node.insertAdjacentHTML(position, html)
position :  afterbegin
            afterend
            beforebegin
            beforeend

const h2 = document.getElementById("myH2");
let html = "<p>My new paragraph.</p>";
h2.insertAdjacentHTML("afterend", html);
```

insertAdjacentText()

```
const h2 = document.getElementById("myH2");
let text = "My inserted text";
h2.insertAdjacentText("afterend", text);
```

insertBefore()

```
element.insertBefore(new, existing)
node.insertBefore(new, existing)

const newNode = document.createElement("li");
const textNode = document.createTextNode("Water");
newNode.appendChild(textNode);

const list = document.getElementById("myList");
list.insertBefore(newNode, list.children[0]);

const node = document.getElementById("myList2").lastElementChild;
const list = document.getElementById("myList1");
list.insertBefore(node, list.children[0]);
```

```
const node = document.getElementById("myList2").lastElementChild;
const list = document.getElementById("myList1");
list.insertBefore(node, null);
```

isContentEditable

```
element.isContentEditable

let answer = document.getElementById("myP").isContentEditable;
```

isDefaultNamespace()

```
element.isDefaultNamespace(namespaceURI)
node.isDefaultNamespace(namespaceURI)

let answer =
element.isDefaultNamespace("http://www.example.org/1999/xhtml");
```

isEqualNode()

```
element.isEqualNode(node)
node.isEqualNode(node)

var item1 = document.getElementById("myList1").firstChild;
var item2 = document.getElementById("myList2").firstChild;
var x = item1.isEqualNode(item2);
```

isSameNode()

```
node.isSameNode(node)

var item1 = document.getElementById("myList1"); // An <ul> element
with id="myList"
var item2 = document.getElementsByTagName("UL")[0]; // The first <ul>
element in the document
var x = item1.isSameNode(item2);
```

isSupported()

Deprecated

lang

```
element.lang
element.lang = lang_code

let text = element.lang;
```

lastChild

```
element.lastChild
node.lastChild

document.getElementById("myList").lastChild.innerHTML;
let text = document.getElementById("mySelect").lastChild.text;
```

lastElementChild

```
element.lastElementChild

const element = document.getElementById("myList")
let html = element.lastElementChild.innerHTML;

const element =document.getElementById("myDIV")
let tag = element.lastElementChild.tagName;

const element = document.getElementById("mySelect")
let text = element.lastElementChild.text;
```

matches()

```
element.matches(selectors)

const element = document.getElementById("demo");
let answer = element.matches(".container");

const element = document.getElementById("demo");
let answer = element.matches(".container, .wrapper");
```

namespaceURI

```
element.namespaceURI

let uri = element.namespaceURI;
```

nextSibling

```
element.nextSibling
node.nextSibling
```

```
document.getElementById("item1").nextSibling.innerHTML;
```

nextElementSibling

```
element.nextElementSibling

let text = element.nextElementSibling.innerHTML;
```

nodeName

```
element.nodeName
node.nodeName

document.getElementById("myP").nodeName;
document.body.nodeName;
const nodes = document.body.childNodes;

let text = "";
for (let i = 0; i < nodes.length; i++) {
  text += nodes[i].nodeName + "<br>";
}
```

nodeType

```
node.nodeType

var x = document.getElementById("myP").nodeType;
```

nodeValue

```
node.nodeValue
node.nodeValue = value

document.getElementsByTagName("BUTTON")[0].childNodes[0].nodeValue;
```

normalize()

```
node.normalize()

document.getElementById("demo").normalize();
```

offsetHeight

```
element.offsetHeight

const elmnt = document.getElementById("myDIV");
let text = "Height with padding and border: " + elmnt.offsetHeight +
"px<br>";
text += "Width with padding and border: " + elmnt.offsetWidth + "px";
```

offsetWidth

```
element.offsetWidth

const elmnt = document.getElementById("myDIV");
let text = "Height with padding and border: " + elmnt.offsetHeight +
"px<br>";
text += "Width with padding and border: " + elmnt.offsetWidth + "px";
```

offsetLeft

```
element.offsetLeft

const element = document.getElementById("myDIV");
let pos = element.offsetLeft;

const element = document.getElementById("myDIV");
let pos1= element.offsetTop;
let pos2= element.offsetLeft;
```

offsetParent

```
element.offsetParent

let node = document.getElementById("myDIV").offsetParent;
```

offsetTop

```
element.offsetTop

const element = document.getElementById("myDIV");
let pos = element.offsetTop;
const element = document.getElementById("test");
let pos1 = element.offsetTop;
let pos2 = element.offsetLeft;
```

outerHTML

```
element.outerHTML
element.outerHTML = text
```

```
document.getElementsByTagName("h2")[0] = "<h3>Changed!</h3>";
element.outerHTML = "<h2>This is a h2 element</h2>";
let html = document.getElementsByTagName("h1")[0].outerHTML;
alert(html);
```

outerText

```
node.outerText
node.outerText = text
```

```
document.getElementById("myH1").outerText = "Changed content!";
```

ownerDocument

```
node.ownerDocument
```

```
var x = document.getElementById("myP").ownerDocument.nodeType;
```

parentNode

```
element.parentNode
node.parentNode
```

```
let name = document.getElementById("myLI").parentNode.nodeName;
<div>
   <span onclick="this.parentNode.style.display = 'none';">x</span>
</div>
```

parentElement

```
node.parentElement
```

```
var x = document.getElementById("myLI").parentElement.nodeName;
<div>
  <span onclick="this.parentElement.style.display = 'none';">x</span>
</div>
```

previousSibling

```
element.previousSibling
node.previousSibling
```

```
document.getElementById("item2").previousSibling.innerHTML;
```

previousElementSibling

```
element.previousElementSibling
```

```
let text = element.previousElementSibling.innerHTML;
```

querySelector()

```
element.querySelector(CSS selectors)
```

```
var x = document.getElementById("myDIV");
x.querySelector(".example").innerHTML = "Hello World!";
```

querySelectorAll()

```
element.querySelectorAll(CSS selectors)
```

```
var x = document.getElementById("myDIV").querySelectorAll(".example");

x[0].style.backgroundColor = "red";
```

remove()

```
element.remove()
node.remove()
```

```
const element = document.getElementById("demo");
element.remove();
```

removeAttribute()

```
element.removeAttribute(name)
```

```
document.getElementsByTagName("H1")[0].removeAttribute("class");
document.getElementById("myAnchor").removeAttribute("href");
```

removeAttributeNode()

```
element.removeAttributeNode(node)
```

```
const element = document.getElementsByTagName("H1")[0];

const attr = element.getAttributeNode("class");
```

```
element.removeAttributeNode(attr);
```

removeChild()

```
element.removeChild(node)
node.removeChild(node)

const list = document.getElementById("myList");
list.removeChild(list.firstElementChild);

const list = document.getElementById("myList");

if (list.hasChildNodes()) {
  list.removeChild(list.children[0]);
}

const list = document.getElementById("myList");

while (list.hasChildNodes()) {
  list.removeChild(list.firstChild);
}
```

removeEventListener()

```
element.removeEventListener(event, function,
capture)
capture : true | false

myDIV.removeEventListener("mousemove", myFunction);
```

replaceChild()

```
node.replaceChild(newnode, oldnode)

const newNode = document.createTextNode("Water");
const element = document.getElementById("myList").children[0];

element.replaceChild(newNode, element.childNodes[0]);
```

requestFullscreen()

Shows an element in fullscreen mode

scrollHeight

```
element.scrollHeight

const element = document.getElementById("content");
let x = element.scrollHeight;
let y = element.scrollWidth;

const element = document.getElementById("content");
let x = element.scrollHeight;
let y = element.scrollWidth;
```

scrollIntoView()

```
element.scrollIntoView(align)
align : true | false

const element = document.getElementById("content");
element.scrollIntoView();
```

scrollLeft

```
element.scrollLeft
element.scrollLeft = pixels

const element = document.getElementById("myDIV");
let x = elmnt.scrollLeft;
let y = elmnt.scrollTop;

const element = document.getElementById("myDIV");
element.scrollLeft = 50;
element.scrollTop = 10;

const element = document.getElementById("myDIV");
element.scrollLeft += 50;
element.scrollTop += 10;
```

scrollTop

```
element.scrollTop
element.scrollTop = pixels
```

```
const element = document.getElementById("myDIV");
let x = elmnt.scrollLeft;
let y = elmnt.scrollTop;
```

`scrollWidth`

```
element.scrollWidth

const element = document.getElementById("content");
let x = element.scrollHeight;
let y = element.scrollWidth;

element.style.height = element.scrollHeight + "px";
element.style.width = element.scrollWidth + "px";
```

`setAttribute()`

```
element.setAttribute(name, value)

element.setAttribute("class", "democlass");
```

`setAttributeNode()`

```
element.setAttributeNode(node)

const attr = document.createAttribute("class");
attr.value = "democlass";

const h1 = document.getElementsByTagName("H1")[0];
h1.setAttributeNode(attr);
```

`style`

```
element.style.property
element.style.property = value

document.getElementById("myH1").style.color = "red";
let value = document.getElementById("myP").style.borderTop;
```

`tabIndex`

```
element.tabIndex
element.tabIndex = number

document.getElementById("myAnchor1").tabIndex = "3";
document.getElementById("myAnchor2").tabIndex = "2";
document.getElementById("myAnchor3").tabIndex = "1";

let order = document.getElementsByTagName("A")[0].tabIndex;
```

`tagName`

```
element.tagName

let name = document.getElementById("demo").tagName;

const element = event.target;
let name = element.tagName;
```

`textContent`

```
element.textContent
node.textContent

let text = element.textContent;
element.textContent = "I have changed!";
let text = document.getElementById("myList").textContent;
```

`title`

```
element.title
element.title = text

let title = element.title;
element.title = "The World's Largest Web Development Site";
```

`toString()`

Converts an element to a string

Attribute Object

HTML attribute merupakan bagian dari HTML element.
Attribute Property

Property	Keterangan
`isId`	Deprecated

`name`	*`attribute`*`.name`
	`let aName = element.attributes[0].name;`

`value`	*`attribute`*`.value`
	`attribute``.value = `*`value`*
	`let value = element.attributes[0].value;`
	`let value = element.getAttributeNode("id").value;`

`specified`	*`attribute`*`.specified`
	`document.getElementById("myDiv").getAttributeNode("style").specified`

NamedNodeMap Property & Method

Method	Keterangan

`getNamedItem()`	*`namednodemap`*`.getNamedItem(`*`nodename`*`)`
	`const nodeMap = document.getElementById("light").attributes;`
	`let value = nodeMap.getNamedItem("src").value;`
	`const nodeMap = document.getElementById("myButton");`
	`let value = nodeMap.getNamedItem("onclick").value;`

`item()`	*`namednodemap`*`[`*`index`*`]`
	`const nodeMap = document.getElementById("myDiv").attributes;`
	`let name1 = nodeMap.item(0).name;`
	`let name2 = nodeMap.item(1).name;`
	`const nodeMap = document.getElementById("myDiv").attributes;`
	`let name1 = nodeMap[0].name;`
	`let name2 = nodeMap[1].name;`

`length`	*`namednodemap`*`.length`
	`let num x = document.getElementById("myButton").attributes.length;`

`removeNamedItem()`	*`namednodemap`*`.removeNamedItem(`*`nodename`*`)`
	`const nodeMap = document.getElementById("myInput").attributes;`
	`nodeMap.removeNamedItem("type");`

`setNamedItem()`	*`namednodemap`*`.setNamedItem(`*`node`*`)`
	`const nodeMap = document.getElementsByTagName("H1")[0].attributes;`
	`const node = document.createAttribute("class");`
	`node.value = "democlass";`
	`nodeMap.setNamedItem(node);`

HTML DOM Event

Event	Keterangan	Berasal dari

abort	<element onabort="myScript">
	object.onabort = function(){myScript};
	
	<video onabort="myFunction()">
	
	object.addEventListener("abort", myScript);
		UiEvent, Event

afterprint	<element onafterprint="myScript">
	object.onafterprint = function(){myScript};
	object.addEventListener("afterprint", myScript);
	
	<body onafterprint="myFunction()">
		Event

animationend	object.addEventListener("webkitAnimationEnd", myScript); // Code for Chrome, Safari and Opera
	object.addEventListener("animationend", myScript); // Standard syntax
	
	var x = document.getElementById("myDIV");
	
	// Code for Chrome, Safari and Opera
	x.addEventListener("webkitAnimationEnd", myEndFunction);
	
	// Standard syntax
	x.addEventListener("animationend", myEndFunction);
		AnimationEvent

animationiteration	object.addEventListener("webkitAnimationIteration", myScript); // Code for Chrome, Safari and Opera
	object.addEventListener("animationiteration", myScript); // Standard syntax
	
	var x = document.getElementById("myDIV");
	
	// Code for Chrome, Safari and Opera
	x.addEventListener("webkitAnimationIteration", myRepeatFunction);
	
	// Standard syntax
	x.addEventListener("animationiteration", myRepeatFunction);
		AnimationEvent

animationstart	object.addEventListener("webkitAnimationStart", myScript); // Code for Chrome, Safari and Opera
	object.addEventListener("animationstart", myScript); // Standard syntax
	
	var x = document.getElementById("myDIV");
	
	// Code for Chrome, Safari and Opera
	x.addEventListener("webkitAnimationStart", myStartFunction);
	
	// Standard syntax
	x.addEventListener("animationstart", myStartFunction);
		AnimationEvent

beforeprint	<element onbeforeprint="myScript">
	object.onbeforeprint = function(){myScript};
	object.addEventListener("beforeprint", myScript);
	
	<body onbeforeprint="myFunction()">
		Event

beforeunload	<element onbeforeunload="myScript">
	object.onbeforeunload = function(){myScript};
	object.addEventListener("beforeunload", myScript);
	
	<body onbeforeunload="return myFunction()">
		UiEvent, Event

blur	<element onblur="myScript">
	object.onblur = function(){myScript};
	object.addEventListener("blur", myScript);
	
	<input type="text" onblur="myFunction()">
		FocusEvent

canplay	<element oncanplay="myScript">
	object.oncanplay = function(){myScript};
	object.addEventListener("canplay", myScript);
	
	<video oncanplay="myFunction()">
		Event

canplaythrough	<element oncanplaythrough="myScript">
	object.oncanplaythrough = function(){myScript};
	object.addEventListener("canplaythrough", myScript);
	
	<video oncanplaythrough="myFunction()">
		Event

change	<element onchange="myScript">
	object.onchange = function(){myScript};
	object.addEventListener("change", myScript);
	
	<select onchange="myFunction()">
		Event

click	<element onclick="myScript">
	object.onclick = function(){myScript};
	object.addEventListener("click", myScript);
	
	<button onclick="myFunction()">Click me</button>
		MouseEvent

contextmenu	<element oncontextmenu="myScript">
	object.oncontextmenu = function(){myScript};
	object.addEventListener("contextmenu", myScript);
	
	<div oncontextmenu="myFunction()" contextmenu="mymenu">
		MouseEvent

copy	<element oncopy="myScript">
	object.oncopy = function(){myScript};
	object.addEventListener("copy", myScript);
	
	<input type="text" oncopy="myFunction()" value="Try to copy this text">
		ClipboardEvent

cut	<element oncut="myScript">
	object.oncut = function(){myScript};
	object.addEventListener("cut", myScript);
	
	<input type="text" oncut="myFunction()" value="Try to cut this text">
		ClipboardEvent

dblclick	<element ondblclick="myScript">
	object.ondblclick = function(){myScript};
	object.addEventListener("dblclick", myScript);
	
	<p ondblclick="myFunction()">Double-click me</p>
		MouseEvent

drag	<element ondrag="myScript">
	object.ondrag = function(){myScript};
	object.addEventListener("drag", myScript);
	
	<p draggable="true" ondrag="myFunction(event)">Drag me!</p>
		DragEvent

dragend	<element ondragend="myScript">
	object.ondragend = function(){myScript};
	object.addEventListener("dragend", myScript);
	
	<p draggable="true" ondragend="myFunction(event)">Drag me!</p>
		DragEvent

dragenter	<element ondragenter="myScript"> object.ondragenter = function(){myScript};	DragEvent

	object.addEventListener("dragenter", myScript); <div ondragenter="myFunction(event)"></div>	

dragleave	<element ondragleave="myScript">
	object.ondragleave = function(){myScript};
	object.addEventListener("dragleave", myScript);
	
	<div ondragleave="myFunction(event)"></div>
		DragEvent

dragover	<element ondragover="myScript">
	object.ondragover = function(){myScript};
	object.addEventListener("dragover", myScript);
	
	<div ondragover="myFunction(event)"></div>
		DragEvent

dragstart	<element ondragstart="myScript">
	object.ondragstart = function(){myScript};
	object.addEventListener("dragstart", myScript);
	
	<p draggable="true" ondragstart="myFunction(event)">Drag me!</p>
		DragEvent

drop	<element ondrop="myScript">
	object.ondrop = function(){myScript};
	object.addEventListener("drop", myScript);
	
	<div ondrop="myFunction(event)"></div>
		DragEvent

durationchange	<element ondurationchange="myScript">
	object.ondurationchange = function(){myScript};
	object.addEventListener("durationchange", myScript);
	
	<video ondurationchange="myFunction()">
		Event

ended	<element onended="myScript">
	object.onended = function(){myScript};
	object.addEventListener("ended", myScript);
	
	<audio onended="myFunction()">
		Event

error	<element onerror="myScript">
	object.onerror = function(){myScript};
	object.addEventListener("error", myScript);
	
	<img src="image.gif" onerror="myFunction()">
		ProgressEvent, UiEvent, Event

focus	<element onfocus="myScript">
	object.onfocus = function(){myScript};
	object.addEventListener("focus", myScript);
	
	<input type="text" onfocus="myFunction()">
		FocusEvent

focusin	<element onfocusin="myScript">
	object.onfocusin = function(){myScript};
	object.addEventListener("focusin", myScript);
	
	<input type="text" onfocusin="myFunction()">
		FocusEvent

focusout	<element onfocusout="myScript">
	object.onfocusout = function(){myScript};
	object.addEventListener("focusout", myScript);
	
	<input type="text" onfocusout="myFunction()">
		FocusEvent

fullscreenchange	<element onfullscreenchange="myScript">
	object.onfullscreenchange = function(){myScript};
	object.addEventListener("fullscreenchange", myScript);
	
	document.addEventListener("fullscreenchange", function() {
	  output.innerHTML = "fullscreenchange event fired!";
	});
		Event

fullscreenerror	<element onfullscreenerror="myScript">
	object.onfullscreenerror = function(){myScript};
	object.addEventListener("fullscreenerror", myScript);
	
	document.addEventListener("fullscreenerror", function() {
	  alert("Fullscreen denied")
	});
		Event

hashchange	<element onhashchange="myScript">
	object.onhashchange = function(){myScript};
	object.addEventListener("hashchange", myScript);
	
	<body onhashchange="myFunction()">
		HashChangeEvent

input	<element oninput="myScript">
	object.oninput = function(){myScript};
	object.addEventListener("input", myScript);
	
	<input type="text" oninput="myFunction()">
		InputEvent, Event

invalid	<element oninvalid="myScript">
	object.oninvalid = function(){myScript};
	object.addEventListener("invalid", myScript);
	
	<input type="text" oninvalid="alert('You must fill out the form!');" required>
		Event

keydown	<element onkeydown="myScript">
	object.onkeydown = function(){myScript};
	object.addEventListener("keydown", myScript);
	
	<input type="text" onkeydown="myFunction()">
		KeyboardEvent

keypress	<element onkeypress="myScript">
	object.onkeypress = function(){myScript};
	object.addEventListener("keypress", myScript);
	
	<input type="text" onkeypress="myFunction()">
		KeyboardEvent

keyup	<element onkeyup="myScript">
	object.onkeyup = function(){myScript};
	object.addEventListener("keyup", myScript);
	
	<input type="text" onkeyup="myFunction()">
		KeyboardEvent

load	<element onload="myScript">
	object.onload = function(){myScript};
	object.addEventListener("load", myScript);
	
	<body onload="myFunction()">
		UiEvent, Event

loadeddata	<element onloadeddata="myScript">
	object.onloadeddata = function(){myScript};
	object.addEventListener("loadeddata", myScript);
	
	<video onloadeddata="myFunction()">
		Event

loadedmetadata	<element onloadedmetadata="myScript">
	object.onloadedmetadata = function(){myScript};
	object.addEventListener("loadedmetadata", myScript);
	
	<video onloadedmetadata="myFunction()">
		Event

loadstart	<element onloadstart="myScript">
	object.onloadstart = function(){myScript};
	object.addEventListener("loadstart", myScript);
	
	<video onloadstart="myFunction()">
		ProgressEvent

message	object.onmessage = function(){myScript};
	object.addEventListener("message", myScript);
		Event

	var source = new EventSource("demo_sse.php");
	source.onmessage = function(event) {
	  document.getElementById("myDIV").innerHTML += event.data + "
	";
	};	

mousedown	<element onmousedown="myScript">
	object.onmousedown = function(){myScript};
	object.addEventListener("mousedown", myScript);
	<p onmousedown="myFunction()">Click the text!</p>
		MouseEvent

mouseenter	<element onmouseenter="myScript">
	object.onmouseenter = function(){myScript};
	object.addEventListener("mouseenter", myScript);
	<img onmouseenter="bigImg(this)" src="smiley.gif" alt="Smiley">
		MouseEvent

mouseleave	<element onmouseleave="myScript">
	object.onmouseleave = function(){myScript};
	object.addEventListener("mouseleave", myScript);
	<img onmouseleave="normalImg(this)" src="smiley.gif" alt="Smiley">
		MouseEvent

mousemove	<element onmousemove="myScript">
	object.onmousemove = function(){myScript};
	object.addEventListener("mousemove", myScript);
	<div onmousemove="myFunction()">Move the cursor over me</div>
		MouseEvent

mouseover	<element onmouseover="myScript">
	object.onmouseover = function(){myScript};
	object.addEventListener("mouseover", myScript);
	<img onmouseover="bigImg(this)" src="smiley.gif" alt="Smiley">
		MouseEvent

mouseout	<element onmouseout="myScript">
	object.onmouseout = function(){myScript};
	object.addEventListener("mouseout", myScript);
	<img onmouseout="normalImg(this)" src="smiley.gif" alt="Smiley">
		MouseEvent

mouseup	<element onmouseup="myScript">
	object.onmouseup = function(){myScript};
	object.addEventListener("mouseup", myScript);
	<p onmouseup="mouseUp()">Click the text!</p>
		MouseEvent

mousewheel	Deprecated.	WheelEvent

offline	<element onoffline="myScript">
	object.onoffline = function(){myScript};
	object.addEventListener("offline", myScript);
	<body onoffline="myFunction()">
		Event

online	<element ononline="myScript">
	object.ononline = function(){myScript};
	object.addEventListener("online", myScript);
	<body ononline="myFunction()">
		Event

open	object.onopen = function(){myScript};
	object.addEventListener("open", myScript);
		Event

	var source = new EventSource("demo_sse.php");
	source.onopen = function() {
	  document.getElementById("myH1").innerHTML = "Getting server updates";
	};	

pagehide	<element onpagehide="myScript">
	object.onpagehide = function(){myScript};
	object.addEventListener("pagehide", myScript);
	
	<body onpagehide="myFunction()">
		PageTransitionEvent

pageshow	<element onpageshow="myScript">
	object.onpageshow = function(){myScript};
	object.addEventListener("pageshow", myScript);
	
	<body onpageshow="myFunction()">
		PageTransitionEvent

paste	<element onpaste="myScript">
	object.onpaste = function(){myScript};
	object.addEventListener("paste", myScript);
	
	<input type="text" onpaste="myFunction()" value="Paste something in here">
		ClipboardEvent

pause	<element onpause="myScript">
	object.onpause = function(){myScript};
	object.addEventListener("pause", myScript);
	
	<video onpause="myFunction()">
		Event

play	<element onplay="myScript">
	object.onplay = function(){myScript};
	object.addEventListener("play", myScript);
	
	<video onplay="myFunction()">
		Event

playing	<element onplaying="myScript">
	object.onplaying = function(){myScript};
	object.addEventListener("playing", myScript);
	
	<video onplaying="myFunction()">
		Event

popstate		PopStateEvent

progress	<element onprogress="myScript">
	object.onprogress = function(){myScript};
	object.addEventListener("progress", myScript);
	
	<video onprogress="myFunction()">
		Event

ratechange	<element onratechange="myScript">
	object.onratechange = function(){myScript};
	object.addEventListener("ratechange", myScript);
	
	<video onratechange="myFunction()">
		Event

resize	<element onresize="myScript">
	object.onresize = function(){myScript};
	object.addEventListener("resize", myScript);
	
	<body onresize="myFunction()">
		UiEvent, Event

reset	<element onreset="myScript">
	object.onreset = function(){myScript};
	object.addEventListener("reset", myScript);
	
	<form onreset="myFunction()">
	  Enter name: <input type="text">
	  <input type="reset">
	</form>
		Event

scroll	<element onscroll="myScript">
	object.onscroll = function(){myScript};
	object.addEventListener("scroll", myScript);
	
	<div onscroll="myFunction()">
		UiEvent, Event

search	<element onsearch="myScript">
	object.onsearch = function(){myScript};
	object.addEventListener("search", myScript);
	
	<input type="search" onsearch="myFunction()">
		Event

seeked	<element onseeked="myScript">
	object.onseeked = function(){myScript};
	object.addEventListener("seeked", myScript);
	
	<video onseeked="myFunction()">
		Event

seeking	<element onseeking="myScript">
	object.onseeking = function(){myScript};
	object.addEventListener("seeking", myScript);
	
	<video onseeking="myFunction()">
		Event

select	<element onselect="myScript">
	object.onselect = function(){myScript};
	object.addEventListener("select", myScript);
	
	<input type="text" onselect="myFunction()">
		UiEvent, Event

show	<element onshow="myScript">
	object.onshow = function(){myScript};
	object.addEventListener("show", myScript);
	
	<div contextmenu="mymenu">
	  <p>Right-click inside this box to see the context menu!
	  <menu type="context" id="mymenu" onshow="myFunction()">
	    <menuitem label="Refresh"
	onclick="window.location.reload();"></menuitem>
	  </menu>
	</div>
		Event

stalled	<element onstalled="myScript">
	object.onstalled = function(){myScript};
	object.addEventListener("stalled", myScript);
	
	<video onstalled="myFunction()">
		Event

storage		StorageEvent

submit	<element onsubmit="myScript">
	object.onsubmit = function(){myScript};
	object.addEventListener("submit", myScript);
	
	<form onsubmit="myFunction()">
	  Enter name: <input type="text">
	  <input type="submit">
	</form>
		Event

suspend	<element onsuspend="myScript">
	object.onsuspend = function(){myScript};
	object.addEventListener("suspend", myScript);
	
	<video onsuspend="myFunction()">
		Event

timeupdate	<element ontimeupdate="myScript">
	object.ontimeupdate = function(){myScript};
	object.addEventListener("timeupdate", myScript);
	
	<video ontimeupdate="myFunction()">
		Event

toggle	<element ontoggle="myScript"> object.ontoggle = function(){myScript};	Event

	object.addEventListener("toggle", myScript); <details ontoggle="myFunction()">	

touchcancel	<element ontouchcancel="myScript">
	object.ontouchcancel = myScript;
	object.addEventListener("touchcancel", myScript);
	<p ontouchcancel="myFunction(event)">Touch me!</p>
		TouchEvent

touchend	<element ontouchend="myScript">
	object.ontouchend = myScript;
	object.addEventListener("touchend", myScript);
	<p ontouchend="myFunction(event)">Touch me!</p>
		TouchEvent

touchmove	<element ontouchmove="myScript">
	object.ontouchmove = myScript;
	object.addEventListener("touchmove", myScript);
	<p ontouchmove="myFunction(event)">Touch me!</p>
		TouchEvent

touchstart	<element ontouchstart="myScript">
	object.ontouchstart = myScript;
	object.addEventListener("touchstart", myScript);
	<p ontouchstart="myFunction(event)">Touch me!</p>
		TouchEvent

transitionend	object.addEventListener("transitionend",
	myScript); // Standard syntax
	document.getElementById("myDIV").addEventListener("transitionend", myFunction);
		TransitionEvent

unload	<element onunload="myScript">
	object.onunload = function(){myScript};
	object.addEventListener("unload", myScript);
	<body onunload="myFunction()">
		UiEvent, Event

volumechange	<element onvolumechange="myScript">
	object.onvolumechange = function(){myScript};
	object.addEventListener("volumechange", myScript);
	<video onvolumechange="myFunction()">
		Event

waiting	<element onwaiting="myScript">
	object.onwaiting = function(){myScript};
	object.addEventListener("waiting", myScript);
	<video onwaiting="myFunction()">
		Event

wheel	<element onwheel="myScript">
	object.onwheel = function(){myScript};
	object.addEventListener("wheel", myScript);
	document.getElementById("myDIV").addEventListener("wheel", myFunction);
	function myFunction() {
	  this.style.fontSize = "35px";
	}
		WheelEvent

HTML DOM Event Property & Method

Property/Method	Keterangan	Berasal dari

altKey

```
event.altKey

if (event.altKey) {
  alert("The ALT key was pressed!");
} else {
  alert("The ALT key was NOT pressed!");
}
```

MouseEvent

altKey

```
event.altKey

var x = document.getElementById("demo");
if (event.altKey) {
  x.innerHTML = "The ALT key was pressed!";
} else {
  x.innerHTML = "The ALT key was NOT pressed!";
}
```

KeyboardEvent, TouchEvent

animationName

```
event.animationName

var x = document.getElementById("myDIV");
x.addEventListener("animationstart", myStartFunction);

function myStartFunction(event) {
  this.innerHTML = "Animation-name is: " +
event.animationName;
}
```

AnimationEvent

bubbles

```
event.bubbles

var x = event.bubbles;
```

Event

button

```
event.button
Return Value :
0 : Left mouse button
1 : Wheel button or middle button (if present)
2 : Right mouse button

alert("You pressed button: " + event.button)
```

MouseEvent

buttons

```
event.buttons
Return Value :
1 : Left mouse button
2 : Right mouse button
4 : Wheel button or middle button
8 : Fourth mouse button (typically the "Browser Back"
button)
16 : Fifth mouse button (typically the "Browser
Forward" button)

var x = event.buttons;
```

MouseEvent

cancelable

```
event.cancelable

var x = event.cancelable;
```

Event

charCode

```
event.charCode

var x = event.charCode;
```

KeyboardEvent

changeTouches

TouchEvent

clientX

```
event.clientX

var x = event.clientX;     // Get the horizontal
coordinate
var y = event.clientY;     // Get the vertical coordinate
var coor = "X coords: " + x + ", Y coords: " + y;
```

MouseEvent, TouchEvent

clientY

```
event.clientY
```

MouseEvent,

```
var x = event.clientX;     // Get the horizontal
coordinate
var y = event.clientY;     // Get the vertical coordinate
var coor = "X coords: " + x + ", Y coords: " + y;
```

TouchEvent

clipboardData

ClipboardData

code

event.code

```
var x = event.code;
```

KeyboardEvent

composed

Event

ctrlKey

event.ctrlKey

```
if (event.ctrlKey) {
  alert("The CTRL key was pressed!");
} else {
  alert("The CTRL key was NOT pressed!");
}
```

MouseEvent

ctrlKey

event.ctrlKey

```
var x = document.getElementById("demo");
if (event.ctrlKey) {
  x.innerHTML = "The CTRL key was pressed!";
} else {
  x.innerHTML = "The CTRL key was NOT pressed!";
}
```

KeyboardEvent, TouchEvent

currentTarget

event.currentTarget

```
alert(event.currentTarget);
```

Event

data

event.data

```
function myFunction(event) {
  var x = event.data;
}
```

InputEvent

dataTransfer

DragEvent,InputEvent

defaultPrevented

event.defaultPrevented

```
document.getElementById("myAnchor").addEventListener("clic
k", function(event){
  event.preventDefault()
  alert("Was preventDefault() called: " +
event.defaultPrevented);
});
```

Event

deltaX

event.deltaX

```
function myFunction(event) {
  var x = event.deltaX;
}
```

WheelEvent

deltaY

event.deltaY

```
function myFunction(event) {
  var y = event.deltaY;
}
```

WheelEvent

deltaZ

event.deltaZ

```
function myFunction(event) {
  var z = event.deltaZ;
}
```

WheelEvent

deltaMode

```
event.deltaMode

function myFunction(event) {
  var x = event.deltaMode;
  // 0 = pixels
  // 1 = lines
  // 2 = pages
}
```

WheelEvent

detail

```
event.detail

var x = event.detail;
```

UiEvent

elapsedTime

```
event.elapsedTime

var x = document.getElementById("myDIV");
x.addEventListener("animationiteration",
myRepeatFunction);

function myRepeatFunction(event) {
  this.innerHTML = "Elapsed time: " + event.elapsedTime;
}
```

AnimationEvent

elapsedTime

```
event.elapsedTime

document.getElementById("myDIV").addEventListener("transitionend", myFunction);

function myFunction(event) {
  this.innerHTML = "Transition lasted: " +
event.elapsedTime + " seconds";
}
```

eventPhase

```
event.eventPhase
0. NONE
1. CAPTURING_PHASE - The event flow is in capturing phase
2. AT_TARGET - The event flow is in target phase
3. BUBBLING_PHASE - The event flow is in bubbling phase
```

Event

getTargetRanges()

InputEvent

getModifierState()

```
event.getModifierState(modifierKey)
modifierKey :    "Alt"
                 "AltGraph"
                 "CapsLock"
                 "Control"
                 "Meta"
                 "NumLocK"
                 "ScrollLock"
                 "Shift"

var x = event.getModifierState("CapsLock");
```

MouseEvent

inputType

```
event.inputType

function myFunction(event) {
  var x = event.inputType;
}
```

InputEvent

isComposing

InputEvent,
KeyboardEvent

isTrusted

```
event.isTrusted

function myFunction(event) {
  if ("isTrusted" in event) {
    if (event.isTrusted) {
```

Event

	alert ("The " + event.type + " event is trusted.");
	} else {
	alert ("The " + event.type + " event is not trusted.");
	}
	} else {
	alert ("The isTrusted property is not supported by your browser");
	}
	}	

key	*event*.key var x = event.key;	KeyboardEvent
key		StorageEvent
keyCode	event.keyCode let unicode = event.keyCode;	KeyboardEvent
location	*event*.location var x = event.location;	KeyboardEvent
lengthComputable		ProgressEvent
loaded		ProgressEvent

metaKey	*event*.metaKey
	
	if (event.metaKey) {
	alert("The META key was pressed!");
	} else {
	alert("The META key was NOT pressed!");
	}
		MouseEvent

metaKey		KeyboardEvent, TouchEvent
MovementX		MouseEvent
MovementY		MouseEvent
newValue		StorageEvent
newURL	*event*.newURL event.newURL;	HasChangeEvent
offsetX		MouseEvent
offsetY		MouseEvent
oldValue		StorageEvent
oldURL	*event*.oldURL event.oldURL;	HasChangeEvent
onemptied		

pageX	*event*.pageX
	
	var x = event.pageX; // Get the horizontal coordinate
	var y = event.pageY; // Get the vertical coordinate
	var coor = "X coords: " + x + ", Y coords: " + y;
		MouseEvent

pageY	*event*.pageY	MouseEvent

	var x = event.pageX; // Get the horizontal coordinate
	var y = event.pageY; // Get the vertical coordinate
	var coor = "X coords: " + x + ", Y coords: " + y;	

persisted	*event*.persisted
	
	function myFunction(event) {
	  alert(event.persisted);
	}
		PageTransitionEvent

preventDefault()	*event*.preventDefault()
	
	document.getElementById("myAnchor").addEventListener("click", function(event){
	  event.preventDefault()
	});
		Event

propertyName	*event*.propertyName
	
	document.getElementById("myDIV").addEventListener("transitionend", myFunction);
	
	function myFunction(event) {
	  this.innerHTML = "Property name is: " +
	event.propertyName;
	}
		AnimationEvent,
		TransitionEvent

pseudoElement		AnimationEvent, TransitionEvent
region		MouseEvent

relatedTarget	*event*.relatedTarget
	
	<p onmouseover="getRelatedElement(event)">Mouse over this
	paragraph.</p>
	
	<script>
	function getRelatedElement(event) {
	  alert("The cursor just exited the " +
	event.relatedTarget.tagName + " element.");
	}
	</script>
		MouseEvent

relatedTarget	*event*.relatedTarget
	
	function getRelatedElement(event) {
	  alert(event.relatedTarget.tagName);
	}
		FocusEvent

repeat		KeyboardEvent

screenX	*event*.screenX
	
	var x = event.screenX; // Get the horizontal
	coordinate
	var y = event.screenY; // Get the vertical coordinate
	var coor = "X coords: " + x + ", Y coords: " + y;
		MouseEvent

screenY	*event*.screenY
	
	var x = event.screenX; // Get the horizontal
	coordinate
	var y = event.screenY; // Get the vertical coordinate
	var coor = "X coords: " + x + ", Y coords: " + y;
		MouseEvent

shiftKey	*event*.shiftKey if (event.shiftKey) {	MouseEvent

```
  alert("The SHIFT key was pressed!");
} else {
  alert("The SHIFT key was NOT pressed!");
}
```

`shiftKey` *event*.shiftKey KeyboardEvent, TouchEvent

```
var x = document.getElementById("demo");
if (event.shiftKey) {
  x.innerHTML = "The SHIFT key was pressed!";
} else {
  x.innerHTML = "The SHIFT key was NOT pressed!";
}
```

`state` PopStateEvent

`stopImmediatePropagation()` *event*.stopImmediatePropagation() Event

```
var x = document.getElementById("myBtn");
x.addEventListener("click", myFunction);
x.addEventListener("click", someOtherFunction);

function myFunction(event) {
  alert ("Hello World!");
  event.stopImmediatePropagation();
}

// This function will not be executed
function someOtherFunction() {
  alert ("I will not get to say Hello World");
}
```

`stopPropagation()` *event*.stopPropagation() Event

```
function func1(event) {
  alert("DIV 1");
  event.stopPropagation();
}
```

`storageArea` StorageEvent

`target` *event*.target Event

```
alert(event.target);
```

`targetTouches` *event*.targetTouches TouchEvent

```
function countTouches(event) {
  var x = event.targetTouches.length;
}
```

`timeStamp` *event*.timeStamp Event

```
var n = event.timeStamp;
```

`total` ProgressEvent

`touches` *event*.touches TouchEvent

```
function countTouches(event) {
  var x = event.touches.length;
}
```

`transitionend` TransitionEvent

`type` *event*.type Event

```
var x = event.type;
```

`url` StorageEvent

`which`	*event*.which
	`0 : No button`
	`1 : Left mouse button`
	`2 : Wheel button or middle button (if present)`
	`3 : Right mouse button`
	`alert("You pressed button: " + event.which)`
		MouseEvent

`which`	`event.which`
	`var x = event.which;`
		KeyboardEvent

`view`	`event.view`
	`var x = event.view;`
		UiEvent

Event Object

Event Object	Keterangan Penggunaan
`Event`	Indu dari seluruh event object
`AnimationEvent`	CSS animasi
`ClipboardEvent`	Modifikasi clipboard
`DragEvent`	Interaksi drag & drop
`FocusEvent`	Focus event
`HashChangeEvent`	Perubahan bagian dari URL pada anchor
`InputEvent`	User input
`KeyboardEvent`	Interaksi keyboard
`MouseEvent`	Interaksi mouse
`PageTransitionEvent`	Navigasi menuju dan keluar dari web page
`PopStateEvent`	Perubahan pada history entry
`ProgressEvent`	Progress loading external resources
`StorageEvent`	Perubahan pada window's storage area.
`TouchEvent`	Interaksi touch screen
`TransitionEvent`	CSS transition
`UiEvent`	Interaksi pada user interface
`WheelEvent`	Interaksi mousewheel

HTMLCollection

HTMLCollection adalah kumpulan dari HTML Node yang serupa dengan array dan diakses berdasarkan index (dimulai dari 0) :

Property / Method	Description

`item()`	`HTMLCollection.item(index)`
	`HTMLCollection[index]`

```
const collection = document.getElementsByTagName("p").item(0);
let text = collection.innerHTML;

const collection = document.getElementsByTagName("p")[0];
let text = collection.innerHTML;

document.getElementsByTagName("p")[0].innerHTML = "Paragraph changed";
```

`length`

```
HTMLCollection.length

let number = document.getElementsByTagName("p").length;
const collection = document.getElementsByClassName("myclass");

for (let i = 0; i < collection.length; i++) {
  collection[i].style.fontSize = "24px";
}
```

`namedItem()`

```
HTMLCollection.namedItem(name)
HTMLCollection[name]

const collection = document.getElementsByTagName("p");
const element = collection.namedItem("myElement");
let text = element.innerHTML;

const collection = document.getElementsByTagName("p");
const element = collection["myElement"];
let text = element.innerHTML;
```

Style object

Style Object Property

Property Keterangan

`alignContent`

```
object.style.alignContent
object.style.alignContent = "stretch|center|flex-start|flex-end|
space-between|space-around|initial|inherit"

document.getElementById("main").style.alignContent = "space-between";
```

`alignItems`

```
object.style.alignItems
object.style.alignItems = "stretch|center|flex-start|flex-end|
baseline|initial|inherit"

document.getElementById("main").style.alignItems = "flex-start";
```

`alignSelf`

```
object.style.alignSelf
object.style.alignSelf = "auto|stretch|center|flex-start|flex-end|
baseline|initial|inherit"

document.getElementById("myBlueDiv").style.alignSelf = "stretch";
```

`animation`

```
object.style.animation
object.style.animation = "name duration timingFunction delay
iterationCount direction fillMode playState"

document.getElementById("myDIV").style.animation = "mynewmove 4s 2";
```

`animationDelay`

```
object.style.animationDelay
object.style.animationDelay = "time|initial|inherit"

document.getElementById("myDIV").style.animationDelay = "1s";
```

`animationDirection`

```
object.style.animationDirection
object.style.animationDirection = "normal|reverse|alternate|
alternate-reverse|initial|inherit"

document.getElementById("myDIV").style.animationDirection = "reverse";
```

animationDuration

```
object.style.animationDuration
object.style.animationDuration = "time|initial|inherit"

document.getElementById("myDIV").style.animationDuration = "3s";
```

animationFillMode

```
object.style.animationFillMode
object.style.animationFillMode = "none|forwards|backwards|both|
initial|inherit"

document.getElementById("myDIV").style.animationFillMode = "forwards";
```

animationIterationCount

```
object.style.animationIterationCount
object.style.animationIterationCount = "number|infinite|initial|
inherit"

document.getElementById("myDIV").style.animationIterationCount =
"infinite";
```

animationName

```
object.style.animationName
object.style.animationName = "none|keyframename|initial|inherit"

document.getElementById("myDIV").style.animationName = "myNEWmove";
```

animationTimingFunction

```
object.style.animationTimingFunction
object.style.animationTimingFunction = "linear|ease|ease-in|ease-
out|cubic-bezier(n, n, n, n)|initial|inherit"

document.getElementById("myDIV").style.animationTimingFunction =
"linear";
```

animationPlayState

```
object.style.animationPlayState
object.style.animationPlayState = "running|paused|initial|inherit"

document.getElementById("myDIV").style.animationPlayState = "paused";
```

background

```
object.style.background
object.style.background = "color|image|repeat|attachment|position|
size| origin|clip|initial|inherit"

document.body.style.background = "#f3f3f3 url('img_tree.png') no-repeat
right top";
```

backgroundAttachment

```
object.style.backgroundAttachment
object.style.backgroundAttachment = "scroll|fixed|local|initial|
inherit"

document.body.style.backgroundAttachment = "fixed";
```

backgroundColor

```
object.style.backgroundColor
object.style.backgroundColor = "color|transparent|initial|inherit"

document.body.style.backgroundColor = "red";
```

backgroundImage

```
object.style.backgroundImage
object.style.backgroundImage = "url('URL')|none|initial|inherit"

document.body.style.backgroundImage = "url('img_tree.png')";
```

backgroundPosition

```
object.style.backgroundPosition
object.style.backgroundPosition = value
value : top left|top center|top right|center left|center center|
center right|bottom left|bottom center|bottom right|x% y%|xpos
ypos|initial|inherit

document.body.style.backgroundPosition = "top right";
```

backgroundRepeat

```
object.style.backgroundRepeat
object.style.backgroundRepeat = "repeat|repeat-x|repeat-y|no-
repeat|initial|inherit"

document.body.style.backgroundRepeat = "repeat-y";
```

backgroundClip

```
object.style.backgroundClip
object.style.backgroundClip = "border-box|padding-box|content-box|
initial|inherit"

document.getElementById("myDIV").style.backgroundClip = "content-box";
```

backgroundOrigin

```
object.style.backgroundOrigin
object.style.backgroundOrigin = "padding-box|border-box|content-
box|initial|inherit"

document.getElementById("myDIV").style.backgroundOrigin = "content-box";
```

backgroundSize

```
object.style.backgroundSize
object.style.backgroundSize = "auto|length|cover|contain|intial|
inherit"

document.getElementById("myDIV").style.backgroundSize = "60px 120px";
```

backfaceVisibility

```
object.style.backfaceVisibility
object.style.backfaceVisibility = "visible|hidden|initial|inherit"

document.getElementById("myDIV").style.backfaceVisibility = "hidden";
```

border

```
object.style.border
object.style.border = "width style color|initial|inherit"

document.getElementById("myDiv").style.border = "thick solid #0000FF";
```

borderBottom

```
object.style.borderBottom
object.style.borderBottom = "width style color|initial|inherit"

document.getElementById("myDiv").style.borderBottom = "thick solid
#0000FF";
```

borderBottomColor

```
object.style.borderBottomColor
object.style.borderBottomColor = "color|transparent|initial|
inherit"

document.getElementById("myDiv").style.borderBottomColor = "red";
```

borderBottomLeftRadius

```
object.style.borderBottomLeftRadius
object.style.borderBottomLeftRadius = "length|% [length|%]|
initial|inherit"

document.getElementById("myDIV").style.borderBottomLeftRadius = "25px";
```

borderBottomRightRadius

```
object.style.borderBottomRightRadius
object.style.borderBottomRightRadius = "length|% [length|%]|
initial|inherit"

document.getElementById("myDIV").style.borderBottomRightRadius = "25px";
```

borderBottomStyle

```
object.style.borderBottomStyle
object.style.borderBottomStyle = value

document.getElementById("myDiv").style.borderBottomStyle = "solid";
```

borderBottomWidth

```
object.style.borderBottomWidth
object.style.borderBottomWidth = "thin|medium|thick|length|
initial|inherit"

document.getElementById("myDiv").style.borderBottomWidth = "10px";
```

borderCollapse

```
object.style.borderCollapse
object.style.borderCollapse = "separate|collapse|initial|inherit"

document.getElementById("myTable").style.borderCollapse = "collapse";
```

borderColor

```
object.style.borderColor
object.style.borderColor = "color|transparent|initial|inherit"

document.getElementById("myDiv").style.borderColor = "red";
```

borderImage

```
object.style.borderImage
object.style.borderImage = "source|slice|width|outset|repeat|
initial|inherit"

document.getElementById("myDIV").style.borderImage = "url(border.png) 30
30 round";
```

borderImageOutset

```
object.style.borderImageOutset
object.style.borderImageOutset = "length|number|initial|inherit"

document.getElementById("myDIV").style.borderImageOutset = "5px 10px
20px 15px";
```

borderImageRepeat

```
object.style.borderImageRepeat
object.style.borderImageRepeat = "stretch|repeat|round|initial|
inherit"

document.getElementById("myDIV").style.borderImageRepeat = "round";
```

borderImageSlice

```
object.style.borderImageSlice
object.style.borderImageSlice = "number|%|fill|initial|inherit"

document.getElementById("myDIV").style.borderImageSlice = "50% 10%";
```

borderImageSource

```
object.style.borderImageSource
object.style.borderImageSource = "none|image|initial|inherit"

document.getElementById("myDIV").style.borderImageSource =
"url(border.png)";
```

borderImageWidth

```
object.style.borderImageWidth
object.style.borderImageWidth = "number|%|auto|initial|inherit"

document.getElementById("myDIV").style.borderImageWidth = "20px 30px";
```

borderLeft

```
object.style.borderLeft
object.style.borderLeft = "width style color|initial|inherit"

document.getElementById("myDiv").style.borderLeft = "thick solid
#0000FF";
```

borderLeftColor

```
object.style.borderLeftColor
object.style.borderLeftColor = "color|transparent|initial|inherit"

document.getElementById("myDiv").style.borderLeftColor = "red";
```

borderLeftStyle

```
object.style.borderLeftStyle
object.style.borderLeftStyle = value
value : dotted|dashed|solid|double|groove|ridge|inset|outset|
initial|inherit

document.getElementById("myDiv").style.borderLeftStyle = "solid";
```

borderLeftWidth

```
object.style.borderLeftWidth
object.style.borderLeftWidth = "thin|medium|thick|length|initial|
inherit"

document.getElementById("myDiv").style.borderLeftWidth = "10px";
```

borderRadius

```
object.style.borderRadius
object.style.borderRadius = "1-4 length|% / 1-4 length|%|initial|
inherit"

document.getElementById("myDIV").style.borderRadius = "25px";
```

borderRight

```
object.style.borderRight
object.style.borderRight = "width style color|initial|inherit"

document.getElementById("myDiv").style.borderRight = "thick solid
#0000FF";
```

borderRightColor

```
object.style.borderRightColor
object.style.borderRightColor = "color|transparent|initial|
inherit"

document.getElementById("myDiv").style.borderRightColor = "red";
```

borderRightStyle

```
object.style.borderRightStyle
object.style.borderRightStyle = value
value : dotted|dashed|solid|double|groove|ridge|inset|outset|
initial|inherit

document.getElementById("myDiv").style.borderRightStyle = "solid";
```

borderRightWidth

```
object.style.borderRightWidth
object.style.borderRightWidth = "thin|medium|thick|length|initial|
inherit"

document.getElementById("myDiv").style.borderRightWidth = "10px";
```

borderSpacing

```
object.style.borderSpacing
object.style.borderSpacing = "length length|initial|inherit"

document.getElementById("myTable").style.borderSpacing = "20px";
```

borderStyle

```
object.style.borderStyle
object.style.borderStyle = value
value : dotted|dashed|solid|double|groove|ridge|inset|outset|
initial|inherit

document.getElementById("myDiv").style.borderStyle = "solid";
```

borderTop

```
object.style.borderTop
object.style.borderTop = "width|style|color|initial|inherit"

document.getElementById("myDiv").style.borderTop = "thick solid
#0000FF";
```

borderTopColor

```
object.style.borderTopColor
object.style.borderTopColor = "color|transparent|initial|inherit"

document.getElementById("myDiv").style.borderTopColor = "red";
```

borderTopLeftRadius

```
object.style.borderTopLeftRadius
object.style.borderTopLeftRadius = "length|% [length|%]|initial|
inherit"

document.getElementById("myDIV").style.borderTopLeftRadius = "25px";
```

borderTopRightRadius

```
object.style.borderTopRightRadius
object.style.borderTopRightRadius = "length|% [length|%]|initial|
inherit"

document.getElementById("myDIV").style.borderTopRightRadius = "25px";
```

borderTopStyle

```
object.style.borderTopStyle
object.style.borderTopStyle = value |initial|inherit
value : dotted|dashed|solid|double|groove|ridge|inset|outset|
initial|inherit

document.getElementById("myDiv").style.borderTopStyle = "solid";
```

borderTopWidth

```
object.style.borderTopWidth
object.style.borderTopWidth = "thin|medium|thick|length|initial|
inherit"

document.getElementById("myDiv").style.borderTopWidth = "10px";
```

borderWidth

```
object.style.borderWidth
object.style.borderWidth = "thin|medium|thick|length|initial|
inherit"

document.getElementById("myDiv").style.borderWidth = "thick";
```

bottom

```
object.style.bottom
object.style.bottom = "auto|length|%|initial|inherit"

document.getElementById("myBtn").style.bottom = "100px";
```

boxDecorationBreak

boxShadow

```
object.style.boxShadow
object.style.boxShadow = "none|h-shadow v-shadow blur spread color
|inset|initial|inherit"

document.getElementById("myDIV").style.boxShadow = "10px 20px 30px
blue";
```

boxSizing

```
object.style.boxSizing
object.style.boxSizing = "content-box|border-box|initial|inherit"

document.getElementById("myDIV").style.boxSizing = "border-box";
```

captionSide

```
object.style.captionSide
object.style.captionSide = "top|bottom|initial|inherit"

document.getElementById("myCap").style.captionSide = "bottom";
```

caretColor

```
object.style.caretColor
object.style.caretColor = "auto|color|initial|inherit"

document.getElementById("input1").style.caretColor = "red";
```

clear

```
object.style.clear
object.style.clear = "none|left|right|both|initial|inherit"

document.getElementById("demo").style.clear = "left";
```

clip

```
object.style.clip
object.style.clip = "auto|rect(top right bottom left)|initial|
inherit"

document.getElementById("myImg").style.clip = "rect(0px 75px 75px 0px)";
```

color

```
object.style.color
object.style.color = "color|initial|inherit"

document.getElementById("myH2").style.color = "#ff0000";
document.getElementById("myP").style.color = "magenta";
document.getElementById("myP2").style.color = "blue";
document.getElementById("myDiv").style.color = "lightblue";
```

columnCount

```
object.style.columnCount
object.style.columnCount = "number|auto|initial|inherit"

document.getElementById("myDIV").style.columnCount = 3;
```

columnFill

```
object.style.columnFill
object.style.columnFill = "balance|auto|initial|inherit"

document.getElementById("myDIV").style.columnFill = "balance";
```

columnGap

```
object.style.columnGap
object.style.columnGap = "length|normal|initial|inherit"

document.getElementById("myDIV").style.columnGap = "50px";
```

columnRule

```
object.style.columnRule
object.style.columnRule = "column-rule-width column-rule-style
column-rule-color|initial|inherit"

document.getElementById("myDIV").style.columnRule = "3px outset blue";
```

columnRuleColor

```
object.style.columnRuleColor
object.style.columnRuleColor = "color|initial|inherit"
```

```
document.getElementById("myDIV").style.columnRuleColor = "blue";
```

columnRuleStyle

```
object.style.columnRuleStyle
object.style.columnRuleStyle = "none|hidden|dotted|dashed|solid|
double|groove|ridge|inset|outset|
initial|inherit"

document.getElementById("myDIV").style.columnRuleStyle = "dotted";
```

columnRuleWidth

```
object.style.columnRuleWidth
object.style.columnRuleWidth = "medium|thin|thick|length|initial|
inherit"

document.getElementById("myDIV").style.columnRuleWidth = "10px";
```

columns

```
object.style.columns
object.style.columns = "auto|column-width column-count|initial|
inherit"

document.getElementById("myDIV").style.columns = "100px 3";
```

columnSpan

```
object.style.columnSpan
object.style.columnSpan = "1|all|initial|inherit"

document.getElementById("myDIV").style.columnSpan = "all";
```

columnWidth

```
object.style.columnWidth
object.style.columnWidth = "auto|length|initial|inherit"

document.getElementById("myDIV").style.columnWidth = "100px";
```

content

counterIncrement

```
object.style.counterIncrement
object.style.counterIncrement = "none|id|initial|inherit"

document.getElementById("myH1").style.counterIncrement = "subsection";
```

counterReset

```
object.style.counterReset
object.style.counterReset = "none|name number|initial|inherit"

document.body.style.counterReset = "section";
```

cssFloat

```
object.style.cssFloat
object.style.cssFloat = "left|right|none|initial|inherit"

function floatRight() {
  document.getElementById("myImg").style.cssFloat = "right";
}

function floatLeft() {
  document.getElementById("myImg").style.cssFloat = "left";
}
```

cursor

```
object.style.cursor
object.style.cursor = value
value :
alias|all-scroll|auto|cell|context-menu|col-resize|copy|crosshair|
default|e-resize|ew-resize|help|move|n-resize|ne-resize|nesw-
resize|ns-resize|nw-resize|nwse-resize|no-drop|none|not-allowed|
pointer|progress|row-resize|s-resize|se-resize|sw-resize|text|
URL|vertical-text|w-resize|wait|zoom-in|zoom-out|initial|inherit

document.getElementById("demo").style.cursor = "pointer";
```

direction

```
object.style.direction
object.style.direction = "ltr|rtl|initial|inherit"

document.getElementById("demo").style.direction = "rtl";
```

display

```
object.style.display
object.style.display = value
```

```
value : block|compact|flex|inline|inline|inline-block|inline-flex|
inline-table|list-item|marker|none|run-in|table|table-caption|
table-cell|table-column|table-column-group|table-footer-group|
table-header-group|table-row|table-row-group|initial|inherit
```

```
document.getElementById("myDIV").style.display = "none";
```

emptyCells

```
object.style.emptyCells
object.style.emptyCells = "show|hide|initial|inherit"
```

```
function show() {
  document.getElementById("myTable").style.emptyCells = "show";
}

function hide() {
  document.getElementById("myTable").style.emptyCells = "hide";
}
```

filter

```
object.style.filter
object.style.filter = "none | blur() | brightness() | contrast() |
drop-shadow() | grayscale() | hue-rotate() | invert() | opacity()
| saturate() | sepia()"
```

```
document.getElementById("myImg").style.filter = "grayscale(100%)";
```

flex

```
object.style.flex
object.style.flex = "flex-grow flex-shrink flex-basis|auto|
initial|inherit"
```

```
for (i = 0; i < y.length; i++) {
  y[i].style.flex = "1";
}
```

flexBasis

```
object.style.flexBasis
object.style.flexBasis = "number|auto|initial|inherit"
```

```
document.getElementById("myBlueDiv").style.flexBasis = "200px";
```

flexDirection

```
object.style.flexDirection
object.style.flexDirection = "row|row-reverse|column|column-
reverse|initial|inherit"
```

```
document.getElementById("main").style.flexDirection = "column-reverse";
```

flexFlow

```
object.style.flexFlow
object.style.flexFlow = "flex-direction flex-wrap|initial|inherit"
```

```
document.getElementById("main").style.flexFlow = "column nowrap";
```

flexGrow

```
object.style.flexGrow
object.style.flexGrow = "number|initial|inherit"
```

```
document.getElementById("myBlueDiv").style.flexGrow = "5";
```

flexShrink

```
object.style.flexShrink
object.style.flexShrink = "number|initial|inherit"
```

```
document.getElementById("myBlueDiv".style.flexShrink = "5";
```

flexWrap

```
object.style.flexWrap
object.style.flexWrap = "nowrap|wrap|wrap-reverse|initial|inherit"
```

```
document.getElementById("main".style.flexWrap = "wrap";
```

font

```
object.style.font
object.style.font = "font-style font-variant font-weight font-
size/line-height|caption|icon|menu|
message-box|small-caption|status-bar|initial|inherit;"
```

```
document.getElementById("demo").style.font = "italic bold 20px
arial,serif";
```

`fontFamily`

```
object.style.fontFamily
object.style.fontFamily = "font1, font2, etc.|initial|inherit"

document.getElementById("demo").style.fontFamily =
"Impact,Charcoal,sans-serif";
```

`fontSize`

```
object.style.fontSize
object.style.fontSize = "value|initial|inherit"
value : xx-small|x-small|small|medium|large|x-large|xx-large|
smaller|larger|length|%|initial|inherit

document.getElementById("demo").style.fontSize = "x-large";
```

`fontStyle`

```
object.style.fontStyle
object.style.fontStyle = "normal|italic|oblique|initial|inherit"

document.getElementById("demo").style.fontStyle = "italic";
```

`fontVariant`

```
object.style.fontVariant
object.style.fontVariant = "normal|small-caps|initial|inherit"

document.getElementById("demo").style.fontVariant = "small-caps";
```

`fontWeight`

```
object.style.fontWeight
object.style.fontWeight = "normal|lighter|bold|bolder|value|
initial|inherit"

document.getElementById("demo").style.fontWeight = "900";
```

`fontSizeAdjust`

```
object.style.fontSizeAdjust
object.style.fontSizeAdjust = "none|number|initial|inherit"

document.getElementById("demo").style.fontSizeAdjust = "0.58";
```

`fontStretch`

Selects a normal, condensed, or expanded face from a font family

`hangingPunctuation`

Specifies whether a punctuation character may be placed outside the line box

`height`

```
object.style.height
object.style.height = "auto|length|%|initial|inherit"

document.getElementById("myBtn").style.height = "50px";
```

`hyphens`

Sets how to split words to improve the layout of paragraphs

`icon`

Provides the author the ability to style an element with an iconic equivalent

`imageOrientation`

Specifies a rotation in the right or clockwise direction that a user agent applies to an image

`isolation`

```
object.style.isolation
object.style.isolation = "auto|isolate|initial|inherit"

document.getElementById("d").style.isolation = "isolate";
```

`justifyContent`

```
object.style.justifyContent
object.style.justifyContent = "flex-start|flex-end|center|space-
between|space-around|initial|inherit"

document.getElementById("main").style.justifyContent = "space-between";
```

`left`

```
object.style.left
object.style.left = "auto|length|%|initial|inherit"

document.getElementById("myBtn").style.left = "100px";
```

`letterSpacing`

```
object.style.letterSpacing
object.style.letterSpacing = "normal|length|initial|inherit"

document.getElementById("demo").style.letterSpacing = "15px";
```

`lineHeight`

```
object.style.lineHeight
object.style.lineHeight = "normal|number|length|%|initial|inherit"
```

```
document.getElementById("myDiv").style.lineHeight = "3";
```

listStyle

```
object.style.listStyle
object.style.listStyle = "type position image|initial|inherit"

document.getElementById("myList").style.listStyle = "decimal inside";
```

listStyleImage

```
object.style.listStyleImage
object.style.listStyleImage = "none|url|initial|inherit"

document.getElementById("myList").style.listStyleImage =
"url('sqorange.gif')";
```

listStylePosition

```
object.style.listStylePosition
object.style.listStylePosition = "outside|inside|initial|inherit"

document.getElementById("myUL").style.listStylePosition = "inside";
```

listStyleType

```
object.style.listStyleType
object.style.listStyleType = value
value : armenian|circle|cjk-ideographic|decimal|decimal-leading-
zero|disc|georgian|hebrew|
hiragana|hiragana-iroha|katakana|katakana-iroha|lower-alpha|lower-
greek|lower-latin|lower-roman|
none|square|upper-alpha|upper-latin|upper-roman|initial|inherit

document.getElementById("myList").style.listStyleType = "upper-roman";
```

margin

```
object.style.margin
object.style.margin = "%|length|auto|initial|inherit"

document.getElementById("myDiv").style.margin = "50px 10px 20px 30px";
```

marginBottom

```
object.style.marginBottom
object.style.marginBottom = "%|length|auto|initial|inherit"

document.getElementById("myDiv").style.marginBottom = "50px";
```

marginLeft

```
object.style.marginLeft
object.style.marginLeft = "%|length|auto|inherit"

document.getElementById("myDiv").style.marginLeft = "50px";
```

marginRight

```
object.style.marginRight
object.style.marginRight = "%|length|auto|initial|inherit"

document.getElementById("myDiv").style.marginRight = "50px";
```

marginTop

```
object.style.marginTop
object.style.marginTop = "%|length|auto|initial|inherit"

document.getElementById("myDiv").style.marginTop = "50px";
```

maxHeight

```
object.style.maxHeight
object.style.maxHeight = "none|length|%|initial|inherit"

document.getElementById("myDIV").style.maxHeight = "15px";
```

maxWidth

```
object.style.maxWidth
object.style.maxWidth = "none|length|%|initial|inherit"

document.getElementById("myDiv").style.maxWidth = "100px";
```

minHeight

```
object.style.minHeight
object.style.minHeight = "length|%|initial|inherit"

document.getElementById("myDIV").style.minHeight = "100px";
```

minWidth

```
object.style.minWidth
object.style.minWidth = "length|%|initial|inherit"

document.getElementById("myDIV").style.minWidth = "400px";
```

navDown

navIndex

navLeft

navRight

navUp

objectFit

```
object.style.objectFit
object.style.objectFit = "fill|contain|cover|scale-down|none|
initial|inherit"

document.getElementById("myImg").style.objectFit = "cover";
```

objectPosition

```
object.style.objectPosition
object.style.objectPosition = "position|initial|inherit"

document.getElementById("myImg").style.objectPosition = "0 10%";
```

opacity

```
object.style.opacity
object.style.opacity = "number|initial|inherit"

document.getElementById("myDIV").style.opacity = "0.5";
```

order

```
object.style.order
object.style.order = "number|initial|inherit"

document.getElementById("myRedDIV").style.order = "4";
document.getElementById("myBlueDIV").style.order = "3";
document.getElementById("myGreenDIV").style.order = "1";
document.getElementById("myPinkDIV").style.order = "2";
```

orphans

```
object.style.orphans
object.style.orphans = "number|initial|inherit"
```

outline

```
object.style.outline
object.style.outline = "width|style|color|initial|inherit"

document.getElementById("myDiv").style.outline = "thick solid #0000FF";
```

outlineColor

```
object.style.outlineColor
object.style.outlineColor = "color|invert|initial|inherit"

document.getElementById("myDiv").style.outlineColor = "#00ff00";
```

outlineOffset

```
object.style.outlineOffset
object.style.outlineOffset = "length|initial|inherit"

document.getElementById("myDIV").style.outlineOffset = "15px";
```

outlineStyle

```
object.style.outlineStyle
object.style.outlineStyle = value

document.getElementById("myDiv").style.outlineStyle = "solid";
```

outlineWidth

```
object.style.outlineWidth
object.style.outlineWidth = "thin|medium|thick|length|initial|
inherit"

document.getElementById("myDiv").style.outlineWidth = "10px";
```

overflow

```
object.style.overflow
object.style.overflow = "visible|hidden|scroll|auto|initial|
inherit"

document.getElementById("myDIV").style.overflow = "scroll";
```

overflowX

```
object.style.overflowX
object.style.overflowX = "visible|hidden|scroll|auto|initial|
inherit"

document.getElementById("myDIV").style.overflowX = "scroll";
```

overflowY

```
object.style.overflowY
object.style.overflowY = "visible|hidden|scroll|auto|initial|
inherit"

document.getElementById("myDIV").style.overflowY = "scroll";
```

padding

```
object.style.padding
object.style.padding = "%|length|initial|inherit"

document.getElementById("myDiv").style.padding = "50px 10px 20px 30px";
```

paddingBottom

```
object.style.paddingBottom
object.style.paddingBottom = "%|length|initial|inherit"

document.getElementById("myDiv").style.paddingBottom = "50px";
```

paddingLeft

```
object.style.paddingLeft
object.style.paddingLeft = "%|length|initial|inherit"

document.getElementById("myDiv").style.paddingLeft = "50px";
```

paddingRight

```
object.style.paddingRight
object.style.paddingRight = "%|length|initial|inherit"

document.getElementById("myDIV").style.paddingRight = "50px";
```

paddingTop

```
object.style.paddingTop
object.style.paddingTop = "%|length|initial|inherit"

document.getElementById("myDiv").style.paddingTop = "50px";
```

pageBreakAfter

```
object.style.pageBreakAfter
object.style.pageBreakAfter = "auto|always|avoid|emptystring|left|
right|initial|inherit"

document.getElementById("footer").style.pageBreakAfter = "always";
```

pageBreakBefore

```
object.style.pageBreakBefore
object.style.pageBreakBefore = "auto|always|avoid|emptystring|
left|right|initial|inherit"

document.getElementById("footer").style.pageBreakBefore = "always";
```

pageBreakInside

```
object.style.pageBreakInside
object.style.pageBreakInside = "auto|avoid|initial|inherit"

document.getElementById("footer").style.pageBreakInside = "avoid";
```

perspective

```
object.style.perspective
object.style.perspective = "length|none"

document.getElementById("myDIV").style.perspective = "50px";
```

perspectiveOrigin

```
object.style.perspectiveOrigin
object.style.perspectiveOrigin = "x-axis y-axis|initial|inherit"

document.getElementById("myDIV").style.perspectiveOrigin = "10px 50%";
```

position

```
object.style.position
object.style.position = "static|absolute|fixed|relative|sticky|
initial|inherit"

document.getElementById("myDIV").style.position = "absolute";
```

quotes

```
object.style.quotes
object.style.quotes = "none|string string string string|initial|
inherit"

document.getElementById("myQ").style.quotes = "'\253' '\273'";
```

resize

```
object.style.resize
object.style.resize = "none|both|horizontal|vertical|initial|
inherit"

document.getElementById("myDIV").style.resize = "both";
```

right

```
object.style.right
object.style.right = "auto|length|%|initial|inherit"

document.getElementById("myDIV").style.right = "100px";
```

scrollBehavior

```
object.style.scrollBehavior = "auto|smooth|initial|inherit"

document.documentElement.style.scrollBehavior = "smooth";
```

tableLayout

```
object.style.tableLayout
object.style.tableLayout = "auto|fixed|initial|inherit"

document.getElementById("myTable").style.tableLayout = "fixed";
```

tabSize

```
object.style.tabSize
object.style.tabSize = "number|length|initial|inherit"

document.getElementById("myPRE").style.tabSize = "16";
```

textAlign

```
object.style.textAlign
object.style.textAlign = "left|right|center|justify|initial|
inherit"

document.getElementById("demo").style.textAlign = "center";
```

textAlignLast

```
object.style.textAlignLast
object.style.textAlignLast = "auto|left|right|center|justify|
start|end|initial|inherit"

document.getElementById("myDIV").style.textAlignLast = "right";
```

textDecoration

```
object.style.textDecoration
object.style.textDecoration = "none|underline|overline|line-
through|blink|initial|inherit"

document.getElementById("demo").style.textDecoration = "underline
overline";
```

textDecorationColor

```
object.style.textDecorationColor
object.style.textDecorationColor = "color|initial|inherit"

document.getElementById("demo").style.textDecorationColor = "red";
```

textDecorationLine

```
object.style.textDecorationLine
object.style.textDecorationLine = "none|underline|overline|line-
through|initial|inherit"

document.getElementById("demo").style.textDecorationLine = "line-
through";
```

textDecorationStyle

```
object.style.textDecorationStyle
object.style.textDecorationStyle = "solid|double|dotted|dashed|
wavy|initial|inherit"

document.getElementById("demo").style.textDecorationStyle = "wavy";
```

textIndent

```
object.style.textIndent
object.style.textIndent = "length|%|initial|inherit"

document.getElementById("myDiv").style.textIndent = "50px";
```

textJustify

textOverflow

```
object.style.textOverflow
object.style.textOverflow = "clip|ellipsis|string|initial|inherit"
```

```
document.getElementById("myDIV").style.textOverflow = "ellipsis";
```

textShadow

```
object.style.textShadow
object.style.textShadow = "none|h-shadow v-shadow blur color|
initial|inherit"

document.getElementById("demo").style.textShadow = "5px 5px 1px #ff0000,
10px 10px 1px #0000ff";
```

textTransform

```
object.style.textTransform
object.style.textTransform = "none|capitalize|uppercase|lowercase|
initial|inherit"

document.getElementById("demo").style.textTransform = "capitalize";
```

top

```
object.style.top
object.style.top = "auto|length|%|initial|inherit"

document.getElementById("myBtn").style.top = "100px";
```

transform

```
object.style.transform
object.style.transform = "none|transform-functions|initial|
inherit"
none|matrix(n, n, n, n, n, n)|matrix3d(n, n, n, n, etc....)|
translate(x, y)|translate3d(x, y, z)|translateX(x)|
translateY(y)|translateZ(z)|scale(x, y)|scale3d(x, y, z)|
scaleX(x)|scaleY(y)|scaleZ(z)|rotate(angle)|rotate3d(x, y, z,
angle)|rotateX(angle)|rotateY(angle)|rotateZ(angle)|
skew(x-angle, y-angle)|skewX(angle)|skewY(angle)|perspective(n)|
initial|inherit

document.getElementById("myDIV").style.transform = "rotate(7deg)";
```

transformOrigin

```
object.style.transformOrigin
object.style.transformOrigin = "x-axis|y-axis|z-axis|initial|
inherit"

document.getElementById("myDIV").style.transformOrigin = "0 0";
```

transformStyle

```
object.style.transformStyle
object.style.transformStyle = "flat|preserve-3d|initial|inherit"

document.getElementById("myDIV").style.transformStyle = "preserve-3d";
```

transition

```
object.style.transition
object.style.transition = "property duration timing-function
delay|initial|inherit"

document.getElementById("myDIV").style.transition = "all 2s";
```

transitionProperty

```
object.style.transitionProperty
object.style.transitionProperty = "none|all|property|initial|
inherit"

document.getElementById("myDIV").style.transitionProperty =
"width,height";
```

transitionDuration

```
object.style.transitionDuration
object.style.transitionDuration = "time|initial|inherit"

document.getElementById("myDIV").style.transitionDuration = "1s";
```

transitionTimingFunction

```
object.style.transitionTimingFunction
object.style.transitionTimingFunction = "ease|linear|ease-in|ease-
out|ease-in-out|cubic-bezier()|
initial|inherit"

document.getElementById("myDIV").style.transitionTimingFunction =
"linear";
```

transitionDelay

```
object.style.transitionDelay
object.style.transitionDelay = "time|initial|inherit"
```

```
document.getElementById("myDIV").style.transitionDelay = "2s";
```

unicodeBidi

```
object.style.unicodeBidi
object.style.unicodeBidi = "normal|embed|bidi-override|initial|
inherit"

document.getElementById("myP").style.unicodeBidi = "bidi-override";
```

userSelect

```
object.style.userSelect
object.style.userSelect = "auto|none|text|all"

document.getElementById("myDiv").style.userSelect = "none";
```

verticalAlign

```
object.style.verticalAlign
object.style.verticalAlign = value
value : length|%|baseline|sub|super|top|text-top|middle|bottom|
text-bottom|initial|inherit

document.getElementById("myTd").style.verticalAlign = "bottom";
```

visibility

```
object.style.visibility
object.style.visibility = "visible|hidden|collapse|initial|
inherit"

document.getElementById("myP").style.visibility = "hidden";
```

whiteSpace

width

```
object.style.width
object.style.width = "auto|length|%|initial|inherit"

document.getElementById("myBtn").style.width = "300px";
```

wordBreak

```
object.style.wordBreak
object.style.wordBreak = "normal|break-all|keep-all|initial|
inherit"

document.getElementById("myDIV").style.wordBreak = "break-all";
```

wordSpacing

```
object.style.wordSpacing
object.style.wordSpacing = "normal|length|initial|inherit"

document.getElementById("myP").style.wordSpacing = "50px";
```

wordWrap

```
object.style.wordWrap
object.style.wordWrap = "normal|break-word|initial|inherit"

document.getElementById("myDIV").style.wordWrap = "break-word";
```

widows

```
object.style.widows
object.style.widows = "number|initial|inherit"
```

zIndex

```
object.style.zIndex
object.style.zIndex = "auto|number|initial|inherit"

document.getElementById("img1").style.zIndex = "1";
```

AJAX BAB XXI

Asynchronous JavaScript And XML (AJAX) merupakan script programming yang digunakan untuk membuat interaksi antara web page dengan server berjalan secara asinkron artinya AJAX dapat berjalan di latar belakang terpisah dari web page. Dengan demikian web page tidak perlu harus melakukan reload untuk melakukan update.

AJAX pada dasarnya menggunakan built-in object dari browser yaitu `XMLHttpRequest` untuk mengirimkan http request ke Server kemudian response dari Server akan digunakan oleh JavaScript dan HTML DOM untuk ditampilkan di web page atau diproses lebih lanjut.

Tahapan dalam melakukan request ke server adalah sebagai berikut :

1. Create object `XMLHttpRequest`

```
variable = new XMLHttpRequest();
```

2. Mendefinisikan callback function (function yang digunakan untuk menjadi parameter pada function yang lain)

```
xhttp = new XMLHttpRequest();
xhttp.onload = function() {
      // to do action
}
```

3. Mendefinisikan request yang akan dikirimkan ke server

```
xhttp = new XMLHttpRequest();
xhttp.onload = function() {
      // to do action
}
xhttp.open("GET", "ajax_info.txt");
```

4. Mengirimkan request ke server

```
xhttp = new XMLHttpRequest();
xhttp.onload = function() {
      // to do action
}
xhttp.open("GET", "ajax_info.txt");
xhttp.send();
```

Jika terdapat beberapa AJAX yang diproses maka sebaiknya dibuatkan satu function yang digunakan hanya untuk membuat object XMLHttpRequest dan kemudian membuat callback function untuk setiap AJAX yang akan diproses seperti contoh di bawah ini :

```
loadDoc("url-1", myFunction1);

loadDoc("url-2", myFunction2);

function loadDoc(url, cFunction) {
   const xhttp = new XMLHttpRequest();
   xhttp.onload = function() {cFunction(this);}
   xhttp.open("GET", url);
   xhttp.send();
}
```

```
function myFunction1(xhttp) {
   // action to do
}
function myFunction2(xhttp) {
   // action to do
}
```

A. XMLHttpRequest Object Method

Method	Keterangan
`new XMLHttpRequest()`	Create XMLHttpRequest object
`abort()`	Cancel current request
`getAllResponseHeaders()`	Return header information
`getResponseHeader()`	Return specific header information
`open(`*method, url, async, user, psw*`)`	Mendefinisikan request *method*: request type GET atau POST *url*: file location *async*: true (asynchronous - default) atau false (synchronous) *user*: optional user name *psw*: optional password

`send()`	Mengirimkan GET request ke server
	`xhttp.open("GET", "demo_get.asp");`
	`xhttp.send();`
	Pada script di atas akan menghasilkan cache result, untuk menghindarinya gunakan contoh di bawah ini :
	`xhttp.open("GET", "demo_get.asp?t=" + Math.random());`
	`xhttp.send();`
	atau seperti ini :
	`xhttp.open("GET", "demo_get2.asp?fname=Henry&lname=Ford");`
	`xhttp.send();`

`send(`*string*`)`	Mengirimkan POST request ke server
	`xhttp.open("POST", "demo_post.asp");`
	`xhttp.send();`
	Untuk mengirimkan data seperti HTML Form :
	`xhttp.open("POST", "ajax_test.asp");`
	`xhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");`
	`xhttp.send("fname=Henry&lname=Ford");`

`setRequestHeader(`*header, value*`)`	Menambahkan label pada request header yang akan dikirim

B. XMLHttpRequest Object Property

Property	Keterangan
`onload`	Mendefinisikan function yang akan digunakan ketika request sudah selesai diproses server.
`onreadystatechange`	Mendefinisikan function yang digunakan ketika terjadi perubahan pada properti `readyState`
`readyState`	Status dari XMLHttpRequest. 0: request belum dibuat. 1: koneksi dengan server terbentuk 2: request diterima server 3: request sedang diproses 4: request selesai diproses dan ready
`responseText`	Return response data dalam bentuk string
`responseXML`	Return response data dalam bentuk XML data
`status`	Return status-number dari request 200: "OK" 403: "Forbidden" 404: "Not Found" Selengkapnya : Http Status Messages Bab IX
`statusText`	Return status-text (contoh: "OK" atau "Not Found")

Contoh :

```
function loadDoc() {
   const xhttp = new XMLHttpRequest();
   xhttp.onreadystatechange = function() {
      if (this.readyState == 4 && this.status == 200) {
         document.getElementById("demo").innerHTML =
         this.responseText;
      }
   };
   xhttp.open("GET", "ajax_info.txt");
   xhttp.send();
}
```

C. Server Response Property

Property	Keterangan
`responseText`	response data dari server dalam bentuk JavaScript string
`responseXML`	response data dari server dalam bentuk XML data

`XMLHttpRequest` memiliki built-in parser. `responseXML` akan memberikan response dalam bentuk XML DOM object sehingga file XML seperti contoh di bawah ini dapat langsung diproses.

`cd_catalog.xml`

```
<CATALOG>
   <CD>
```

```
      <TITLE>Empire Burlesque</TITLE>
      <ARTIST>Bob Dylan</ARTIST>
      <COUNTRY>USA</COUNTRY>
      <COMPANY>Columbia</COMPANY>
      <PRICE>10.90</PRICE>
      <YEAR>1985</YEAR>
   </CD>
   <CD>
      <TITLE>Hide your heart</TITLE>
      <ARTIST>Bonnie Tyler</ARTIST>
      <COUNTRY>UK</COUNTRY>
      <COMPANY>CBS Records</COMPANY>
      <PRICE>9.90</PRICE>
      <YEAR>1988</YEAR>
   </CD>
   <CD>
      <TITLE>Greatest Hits</TITLE>
      <ARTIST>Dolly Parton</ARTIST>
      <COUNTRY>USA</COUNTRY>
      <COMPANY>RCA</COMPANY>
      <PRICE>9.90</PRICE>
      <YEAR>1982</YEAR>
   </CD>
<CATALOG>
```

```
const xhttp = new XMLHttpRequest();
xhttp.onload = function() {
  const xmlDoc = this.responseXML;
  const x = xmlDoc.getElementsByTagName("ARTIST");
  let txt = "";
  for (let i = 0; i < x.length; i++) {
    txt = txt + x[i].childNodes[0].nodeValue + "<br>";
  }
  document.getElementById("demo").innerHTML = txt;
}
xhttp.open("GET", "cd_catalog.xml");
xhttp.send();
```

Output :

```
Bob Dylan
Bonnie Tyler
Dolly Parton
```

D. Server Response Method

Method — **Description**

`getResponseHeader()` — Return specific header information dari server resource

```
const xhttp = new XMLHttpRequest();
xhttp.onload = function() {
   document.getElementById("demo").innerHTML =
   this.getResponseHeader("Last-Modified");
}
xhttp.open("GET", "ajax_info.txt");
xhttp.send();
```

`getAllResponseHeaders()` — Return seluruh header information dari server resource

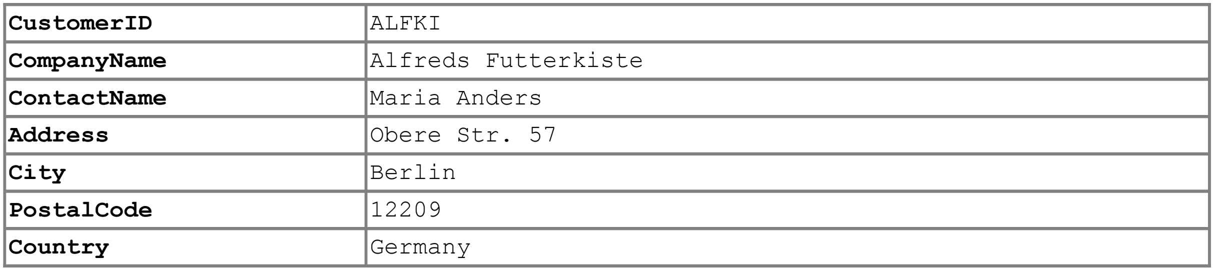
```
const xhttp = new XMLHttpRequest();
xhttp.onload = function() {
   document.getElementById("demo").innerHTML =
   this.getAllResponseHeaders();
}
xhttp.open("GET", "ajax_info.txt");
xhttp.send();
```

Contoh :

customer.html

```
<!DOCTYPE html>
<html>
<style>
th,td {
  padding: 5px;
}
</style>
<body>

<h2>The XMLHttpRequest Object</h2>

<form action="">
  <select name="customers" onchange="showCustomer(this.value)">
    <option value="">Select a customer:</option>
    <option value="ALFKI">Alfreds Futterkiste</option>
    <option value="NORTS ">North/South</option>
    <option value="WOLZA">Wolski Zajazd</option>
  </select>
</form>
<br>
<div id="txtHint">Customer info will be listed here...</div>

<script>
function showCustomer(str) {
  if (str == "") {
    document.getElementById("txtHint").innerHTML = "";
    return;
  }
  const xhttp = new XMLHttpRequest();
  xhttp.onload = function() {
    document.getElementById("txtHint").innerHTML = this.responseText;
  }
  xhttp.open("GET", "getcustomer.php?q="+str);
  xhttp.send();
}
</script>
</body>
</html>
```

Output :



CustomerID	ALFKI
CompanyName	Alfreds Futterkiste
ContactName	Maria Anders
Address	Obere Str. 57
City	Berlin
PostalCode	12209
Country	Germany

getcustomer.php

```
<?php
$mysqli = new mysqli("servername", "username", "password", "dbname");
if($mysqli->connect_error) {
   exit('Could not connect');
}

$sql = "SELECT customerid, companyname, contactname, address, city, postalcode, country
FROM customers WHERE customerid = ?";

$stmt = $mysqli->prepare($sql);
$stmt->bind_param("s", $_GET['q']);
$stmt->execute();
$stmt->store_result();
$stmt->bind_result($cid, $cname, $name, $adr, $city, $pcode, $country);
```

```
$stmt->fetch();
$stmt->close();

echo "<table>";
echo "<tr>";
echo "<th>CustomerID</th>";
echo "<td>" . $cid . "</td>";
echo "<th>CompanyName</th>";
echo "<td>" . $cname . "</td>";
echo "<th>ContactName</th>";
echo "<td>" . $name . "</td>";
echo "<th>Address</th>";
echo "<td>" . $adr . "</td>";
echo "<th>City</th>";
echo "<td>" . $city . "</td>";
echo "<th>PostalCode</th>";
echo "<td>" . $pcode . "</td>";
echo "<th>Country</th>";
echo "<td>" . $country . "</td>";
echo "</tr>";
echo "</table>";
?>
```

JSON BAB XXII

JavaScript Object Notation (JSON) adalah format data yang digunakan untuk menyimpan maupun melakukan transfer data. JSON hanya berupa text yang diformat seperti halnya notasi pada JavaScript sehingga JSON data dapat dengan mudah dikonversi menjadi data JavaScript.

JavaScript memiliki fungsi `JSON.parse()` yang digunakan untuk merubah JSON string menjadi JavaScript object dan fungsi `JSON.stringify()` digunakan untuk merubah JavaScript object menjadi JSON string.

Syntax yang digunakan :

1. Data merupakan pasangan `key` dan `value` → "name":"John" dan `key` harus selalu dalam bentuk string. Sedangkan `value` dapat berupa :
   - string `{"name":"John"}`
   - number `{"age":30}`
   - object
     ```
     {
       "employee":{"name":"John", "age":30, "city":"New York"}
     }
     ```
   - array
     ```
     {
       "employees":["John", "Anna", "Peter"]
     }
     ```
   - boolean `{"sale":true}`
   - null `{"middlename":null}`

   String harus selalu diapit dengan double quotes " "
2. Data dipisahkan dengan tanda comma.
3. Data yang diapit oleh curly brace { } merupakan suatu object.
4. Data yang diapit oleh square bracket [ ] merupakan suatu array.

A. `JSON.parse()`

Contoh :

Data JSON yang diterima dari server selalu berbentuk string, misalnya seperti ini :

```
'{"name":"John", "age":30, "city":"New York"}'
```

untuk merubahnya menjadi JavaScript object :

```
<script>
   const obj = JSON.parse('{"name":"John", "age":30, "city":"New York"}');
   document.getElementById("demo").innerHTML = obj.name;
</script>
```

Jika berupa array :

```
const text = '["Ford", "BMW", "Audi", "Fiat"]';
const myArr = JSON.parse(text);
```

A.1. Date Parsing

Khusus untuk Date (tanggal) maka date tersebut harus dalam bentuk string dan kemudian dikonversi menjadi date format.

Contoh :

```
const text = '{"name":"John", "birth":"1986-12-14", "city":"New York"}';
const obj = JSON.parse(text);
obj.birth = new Date(obj.birth);

document.getElementById("demo").innerHTML = obj.name + ", " + obj.birth;
```

Atau menggunakan fungsi parameter *reviver* dari JSON.parse()

```
const text = '{"name":"John", "birth":"1986-12-14", "city":"New York"}';
const obj = JSON.parse(text, function (key, value) {
   if (key == "birth") {
      return new Date(value);
   } else {
      return value;
   }
});

document.getElementById("demo").innerHTML = obj.name + ", " + obj.birth;
```

A.2. Function Parsing

Jika suatu function harus dimasukkan kedalam JSON maka function tersebut harus dalam bentuk string

```
const text = '{"name":"John", "age":"function () {return 30;}", "city":"New
York"}';
const obj = JSON.parse(text);
obj.age = eval("(" + obj.age + ")");

document.getElementById("demo").innerHTML = obj.name + ", " + obj.age();
```

A.3. Object Loop

Loop property :

```
const myJSON = '{"name":"John", "age":30, "car":null}';
const myObj = JSON.parse(myJSON);
let text = "";
for (const x in myObj) {
   text += x + ", ";
}
```

Loop property value :

```
const myJSON = '{"name":"John", "age":30, "car":null}';
const myObj = JSON.parse(myJSON);
let text = "";
for (const x in myObj) {
   text += myObj[x] + ", ";
}
```

Loop Array :

```
const myJSON = '{"name":"John", "age":30, "cars":["Ford", "BMW", "Fiat"]}';
const myObj = JSON.parse(myJSON);

document.getElementById("demo").innerHTML = myObj.cars[0];

for (let i in myObj.cars) {
   x += myObj.cars[i];
}

// atau menggunakan for loop
for (let i = 0; i < myObj.cars.length; i++) {
   x += myObj.cars[i];
}
```

B. `JSON.stringify()`

Ketika data yang dikirimkan ke server berupa string maka `JSON.stringify()` akan merubah JavaScript object menjadi string :

```
const obj = {name: "John", age: 30, city: "New York"};
const myJSON = JSON.stringify(obj);
```

demikian juga dengan array :

```
const arr = ["John", "Peter", "Sally", "Jane"];
const myJSON = JSON.stringify(arr);
```

Untuk menyimpan dan menarik kembali JavaScript object :

```
// Storing data:
const myObj = {name: "John", age: 31, city: "New York"};
const myJSON = JSON.stringify(myObj);
localStorage.setItem("testJSON", myJSON);

// Retrieving data:
let text = localStorage.getItem("testJSON");
let obj = JSON.parse(text);
document.getElementById("demo").innerHTML = obj.name;
```

Untuk Date format :

```
const obj = {name: "John", today: new Date(), city : "New York"};
const myJSON = JSON.stringify(obj);
```

C. JSON Server

Data yang dikirimkan maupun diterima dari server selalu berbentuk string.

C.1. Sending Data

```
const myObj = {name: "John", age: 31, city: "New York"};
const myJSON = JSON.stringify(myObj);
window.location = "demo_json.php?x=" + myJSON;
```

C.2. Receiving Data

```
const myJSON = '{"name":"John", "age":31, "city":"New York"}';
const myObj = JSON.parse(myJSON);
document.getElementById("demo").innerHTML = myObj.name;
```

C.3. Menggunakan AJAX request

```
const xmlhttp = new XMLHttpRequest();
xmlhttp.onload = function() {
  const myObj = JSON.parse(this.responseText);
  document.getElementById("demo").innerHTML = myObj.name;
};
xmlhttp.open("GET", "json_demo.txt");
xmlhttp.send();
```

D. PHP

PHP memiliki built-in function `json_encode()` untuk merubah PHP object menjadi JSON :

demo_file.php

```
<?php
$myObj->name = "John";
$myObj->age = 30;
$myObj->city = "New York";

$myJSON = json_encode($myObj);

echo $myJSON;
?>
```

```
const xmlhttp = new XMLHttpRequest();
xmlhttp.onload = function() {
  const myObj = JSON.parse(this.responseText);
  document.getElementById("demo").innerHTML = myObj.name;
}
xmlhttp.open("GET", "demo_file.php");
xmlhttp.send();
```

E. HTML

1. Membuat HTML table dari data JSON

```
const dbParam = JSON.stringify({table:"customers",limit:20});
const xmlhttp = new XMLHttpRequest();
xmlhttp.onload = function() {
   myObj = JSON.parse(this.responseText);
   let text = "<table border='1'>"
   for (let x in myObj) {
       text += "<tr><td>" + myObj[x].name + "</td></tr>";
   }
   text += "</table>"
   document.getElementById("demo").innerHTML = text;
}
xmlhttp.open("POST", "json_demo_html_table.php");
xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xmlhttp.send("x=" + dbParam);
```

2. Dynamic HTML Table

```
<select id="myselect" onchange="change_myselect(this.value)">
<option value="">Choose an option:</option>
<option value="customers">Customers</option>
<option value="products">Products</option>
<option value="suppliers">Suppliers</option>
</select>

<script>
function change_myselect(sel) {
   const dbParam = JSON.stringify({table:sel,limit:20});
   const xmlhttp = new XMLHttpRequest();
   xmlhttp.onload = function() {
      const myObj = JSON.parse(this.responseText);
      let text = "<table border='1'>"
      for (let x in myObj) {
          text += "<tr><td>" + myObj[x].name + "</td></tr>";
      }
      text += "</table>"
      document.getElementById("demo").innerHTML = text;
   }
   xmlhttp.open("POST", "json_demo_html_table.php");
   xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
   xmlhttp.send("x=" + dbParam);
}
</script>
```

3. HTML Drop Down List

```
const dbParam = JSON.stringify({table:"customers",limit:20});
const xmlhttp = new XMLHttpRequest();
xmlhttp.onload = function() {
   const myObj = JSON.parse(this.responseText);
   let text = "<select>"
   for (let x in myObj) {
      text += "<option>" + myObj[x].name + "</option>";
   }
   text += "</select>"
   document.getElementById("demo").innerHTML = text;
   }
}
xmlhttp.open("POST", "json_demo_html_table.php", true);
xmlhttp.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
xmlhttp.send("x=" + dbParam);
```

www.w3.org

262.ecma-international.org/12.0/

tc39.es/ecma262/

developer.mozilla.org

getbootstrap.com/docs/5.0/

json.org

jquery.com

www.w3schools.com

Stack Overflow Contributors
Learning HTML - eBook

Dr. Axel Rauschmayer
JavaScript for impatient programmers (ES2022 edition)

Ed Tittel and Mary C. Burmeister
HTML 4 For Dummies, 5th Edition,
Wiley Publishing, Inc. 111 River Street Hoboken, NJ 07030-5774 www.wiley.com

Lucinda Dykes and Ed Tittel
XML For Dummies®, 4th Edition
Wiley Publishing, Inc. 111 River Street, Hoboken, NJ 07030-5774 www.wiley.com

Jonathan Chaffer and Karl Swedberg
Learning jQuery
Packt Publishing Ltd.32 Lincoln RoadOltonBirmingham, B27 6PA, UK.

Russ Ferguson and Christian Heitmann
Beginning JavaScript with DOM Scripting and Ajax
Springer Science+Business Media New York, 233 Spring Street, 6th Floor,
New York, NY 10013.