Daftar Metode DOM JavaScript

1. Document Methods
Digunakan untuk mengambil elemen dari DOM.

getElementById	querySelectorAll	adoptNode
getElementsByClassName	createElement	open
getElementsByTagName	createTextNode	close
getElementsByName	createDocumentFragment	write
querySelector	importNode	writeln

2. Element Methods
Manipulasi elemen HTML secara langsung.

setAttribute	toggleAttribute	closest
getAttribute	insertAdjacentHTML	matches
removeAttribute	insertAdjacentElement	
hasAttribute	insertAdjacentText	

3. Node Methods
Digunakan untuk manipulasi struktur DOM.

appendChild	hasChildNodes	compareDocumentPosition
removeChild	normalize	contains
replaceChild	isEqualNode	
cloneNode	isSameNode	

4. Event Methods
Berhubungan dengan penanganan event.

addEventListener	removeEventListener	dispatchEvent

5. Traversal & Navigation
Navigasi antar node di dalam DOM.

parentNode	nextSibling	lastElementChild
childNodes	previousSibling	nextElementSibling
firstChild	children	previousElementSibling
lastChild	firstElementChild	

6. Attribute Methods

getAttribute	removeAttribute
setAttribute	hasAttribute

7. Style & Class Methods

Mengatur gaya dan class elemen.

style	classList.toggle	getComputedStyle
classList.add	classList.contains	
classList.remove	classList.replace	

8. Form & Input Methods

Khusus untuk elemen formulir.

submit	blur	checkValidity
reset	select	reportValidity
focus	setCustomValidity	

9. Selection & Range Methods

Digunakan untuk manipulasi teks terpilih.

getSelection	range.setEnd	range.insertNode
createRange	range.selectNode	
range.setStart	range.deleteContents	

10. Window & Screen Methods

Manipulasi jendela browser.

alert	setInterval	scrollBy
confirm	clearTimeout	print
prompt	clearInterval	open
setTimeout	scrollTo	close

Document Methods

getElementById - document.getElementById(id)

Penjelasan: Mengembalikan elemen dengan atribut id yang sesuai.
Syntax: document.getElementById(id)
Contoh:

```
document.getElementById('header');
```

Output:

```
<div id="header">...</div>
```

getElementsByClassName - document.getElementsByClassName(class)

Penjelasan: Mengembalikan koleksi semua elemen dengan nama class tertentu.
Syntax: document.getElementsByClassName(className)
Contoh:

```
document.getElementsByClassName('menu');
```

Output:

```
HTMLCollection [<ul class="menu">...</ul>]
```

getElementsByTagName - document.getElementsByTagName(tag)

Penjelasan: Mengembalikan koleksi semua elemen dengan tag tertentu.
Syntax: document.getElementsByTagName(tagName)
Contoh:

```
document.getElementsByTagName('p');
```

Output:

```
HTMLCollection [<p>...</p>, <p>...</p>]
```

querySelector - document.querySelector(selector)

Penjelasan: Mengembalikan elemen pertama yang cocok dengan selector CSS.
Syntax: document.querySelector(selectors)
Contoh:

```
document.querySelector('.menu');
```

Output:

```
<ul class="menu">...</ul>
```

querySelectorAll - document.querySelectorAll(selector)

Penjelasan: Mengembalikan semua elemen yang cocok dengan selector CSS.
Syntax: document.querySelectorAll(selectors)
Contoh:

```
document.querySelectorAll('li.active');
```

Output:

```
NodeList [<li class="active">Item</li>]
```

Document Methods

createElement - document.createElement(tagName)

Penjelasan: Membuat elemen HTML berdasarkan tag yang diberikan.
Syntax: document.createElement(tagName)
Contoh:

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

Output:

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

createTextNode - document.createTextNode(text)

Penjelasan: Membuat node teks dengan isi teks yang ditentukan.
Syntax: document.createTextNode(text)
Contoh:

```
let text = document.createTextNode('Hello');
```

Output:

```
#text "Hello"
```

createDocumentFragment - document.createDocumentFragment()

Penjelasan: Membuat node fragment yang dapat menyimpan node sebelum dimasukkan ke DOM.
Syntax: document.createDocumentFragment()
Contoh:

```
let fragment = document.createDocumentFragment();
```

Output:

```
DocumentFragment {}
```

importNode - document.importNode(node, deep)

Penjelasan: Mengimpor node dari dokumen lain ke dokumen saat ini.
Syntax: document.importNode(node, deep)
Contoh:

```
// assume xmlDoc is another document
let imported = document.importNode(xmlDoc.documentElement, true);
```

Output:

```
Node yang diimpor ke dokumen saat ini
```

adoptNode - document.adoptNode(node)

Penjelasan: Mengambil alih kepemilikan node dari dokumen lain ke dokumen saat ini.
Syntax: document.adoptNode(node)
Contoh:

```
// assume foreignNode from iframe
document.adoptNode(foreignNode);
```

Output:

```
Node sekarang dimiliki oleh dokumen saat ini
```

Document Methods

open - document.open()

Penjelasan: Membuka dokumen untuk ditulis ulang (jarang digunakan).
Syntax: document.open()
Contoh:

```
document.open(); document.write('Hello'); document.close();
```

Output:

```
Halaman menampilkan: Hello
```

close - document.close()

Penjelasan: Menutup stream dokumen yang dibuka dengan document.open().
Syntax: document.close()
Contoh:

```
document.open(); document.write('Hi'); document.close();
```

Output:

```
Halaman menampilkan: Hi
```

write - document.write(content)

Penjelasan: Menulis konten ke dokumen secara langsung.
Syntax: document.write(content)
Contoh:

```
document.write('<h1>Hello</h1>');
```

Output:

```
<h1>Hello</h1>
```

writeln - document.writeln(content)

Penjelasan: Menulis konten ke dokumen dan menambahkan newline di akhir.
Syntax: document.writeln(content)
Contoh:

```
document.writeln('Line 1');
```

Output:

```
Line 1
```

Element Methods

setAttribute - element.setAttribute(name, value)

Penjelasan: Menetapkan nilai atribut pada elemen.
Syntax: element.setAttribute(name, value)
Contoh:

```
document.querySelector('div').setAttribute('class', 'highlight');
```

Output:

```
<div class="highlight"></div>
```

getAttribute - element.getAttribute(name)

Penjelasan: Mengambil nilai dari atribut tertentu pada elemen.
Syntax: element.getAttribute(name)
Contoh:

```
<a href="https://example.com"></a>
let link = document.querySelector('a');
link.getAttribute('href');
```

Output:

```
"https://example.com"
```

removeAttribute - element.removeAttribute(name)

Penjelasan: Menghapus atribut dari elemen.
Syntax: element.removeAttribute(name)
Contoh:

```
document.querySelector('div').removeAttribute('style');
```

Output:

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

hasAttribute - element.hasAttribute(name)

Penjelasan: Mengembalikan true jika elemen memiliki atribut tertentu.
Syntax: element.hasAttribute(name)
Contoh:

```
<img alt="image" />
document.querySelector('img').hasAttribute('alt');
```

Output:

```
true
```

toggleAttribute - element.toggleAttribute(name)

Penjelasan: Menambah atau menghapus atribut tergantung kondisi keberadaan sebelumnya.
Syntax: element.toggleAttribute(name)
Contoh:

```
const el = document.querySelector('#box');
el.toggleAttribute('hidden');
```

Output:

```
<div id="box" hidden></div> atau <div id="box"></div>
```

Element Methods

insertAdjacentHTML - element.insertAdjacentHTML(position, text)

Penjelasan: Menyisipkan HTML pada posisi relatif terhadap elemen.
Syntax: element.insertAdjacentHTML(position, html)
Contoh:

```
el.insertAdjacentHTML('beforeend', '<p>Hello</p>');
```

Output:

```
<div>...<p>Hello</p></div>
```

insertAdjacentElement - element.insertAdjacentElement(position, element)

Penjelasan: Menyisipkan elemen ke posisi relatif terhadap elemen lain.
Syntax: element.insertAdjacentElement(position, element)
Contoh:

```
el.insertAdjacentElement('afterbegin', document.createElement('span'));
```

Output:

```
Elemen <span> ditambahkan sebagai anak pertama
```

insertAdjacentText - element.insertAdjacentText(position, text)

Penjelasan: Menyisipkan teks ke posisi relatif terhadap elemen.
Syntax: element.insertAdjacentText(position, text)
Contoh:

```
el.insertAdjacentText('beforeend', 'World');
```

Output:

```
teks 'World' disisipkan ke akhir isi elemen
```

closest - element.closest(selector)

Penjelasan: Mengembalikan elemen terdekat (teratas) yang cocok dengan selector.
Syntax: element.closest(selector)
Contoh:

```
el.closest('.container');
```

Output:

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

matches - element.matches(selector)

Penjelasan: Memeriksa apakah elemen cocok dengan selector yang diberikan.
Syntax: element.matches(selector)
Contoh:

```
el.matches('.selected');
```

Output:

```
true / false
```

Node Methods

appendChild - node.appendChild(child)

Penjelasan: Menambahkan node sebagai anak terakhir dari node induk.
Syntax: node.appendChild(child)
Contoh:

```
let p = document.createElement('p');
document.body.appendChild(p);
```

Output:

```
<body>...<p></p></body>
```

removeChild - node.removeChild(child)

Penjelasan: Menghapus anak node dari node induk.
Syntax: node.removeChild(child)
Contoh:

```
let el = document.getElementById('item');
el.parentNode.removeChild(el);
```

Output:

```
Elemen dihapus dari DOM
```

replaceChild - node.replaceChild(newChild, oldChild)

Penjelasan: Mengganti anak node lama dengan yang baru.
Syntax: node.replaceChild(newChild, oldChild)
Contoh:

```
let newP = document.createElement('p');
parent.replaceChild(newP, oldChild);
```

Output:

```
Node baru menggantikan node lama
```

cloneNode - node.cloneNode(deep)

Penjelasan: Mengkloning node, jika deep true termasuk semua anaknya.
Syntax: node.cloneNode(deep)
Contoh:

```
let clone = document.getElementById('box').cloneNode(true);
```

Output:

```
Salinan elemen dengan semua anak
```

hasChildNodes - node.hasChildNodes()

Penjelasan: Mengembalikan true jika node memiliki node anak.
Syntax: node.hasChildNodes()
Contoh:

```
document.body.hasChildNodes();
```

Output:

```
true
```

Node Methods

normalize - node.normalize()

Penjelasan: Menggabungkan node teks berurutan menjadi satu.
Syntax: node.normalize()
Contoh:

```
element.normalize();
```

Output:
Node teks digabungkan

isEqualNode - node.isEqualNode(otherNode)

Penjelasan: Memeriksa apakah dua node identik secara struktur dan konten.
Syntax: node.isEqualNode(otherNode)
Contoh:

```
node1.isEqualNode(node2);
```

Output:
true / false

isSameNode - node.isSameNode(otherNode)

Penjelasan: Memeriksa apakah dua node merujuk ke objek yang sama.
Syntax: node.isSameNode(otherNode)
Contoh:

```
node1.isSameNode(node2);
```

Output:
true / false

compareDocumentPosition - node.compareDocumentPosition(other)

Penjelasan: Menentukan posisi dokumen relatif terhadap node lain.
Syntax: node.compareDocumentPosition(other)
Contoh:

```
nodeA.compareDocumentPosition(nodeB);
```

Output:
2, 4, 8, 16, dst (bitmask)

contains - node.contains(other)

Penjelasan: Mengembalikan true jika node lain adalah keturunan dari node ini.
Syntax: node.contains(other)
Contoh:

```
parent.contains(child);
```

Output:
true / false

Event Methods

addEventListener - element.addEventListener(type, listener)

Penjelasan: Menambahkan fungsi event listener untuk menangani event tertentu.
Syntax: element.addEventListener(event, function)
Contoh:

```
document.querySelector('button').addEventListener('click', () => alert('Clicked!'));
```

Output:
Menampilkan alert saat tombol diklik

removeEventListener - element.removeEventListener(type, listener)

Penjelasan: Menghapus fungsi event listener yang telah ditambahkan sebelumnya.
Syntax: element.removeEventListener(event, function)
Contoh:

```
const handler = () => alert('Hi');
button.addEventListener('click', handler);
button.removeEventListener('click', handler);
```

Output:
Fungsi handler tidak akan dipanggil saat tombol diklik

dispatchEvent - element.dispatchEvent(event)

Penjelasan: Memicu event secara manual pada elemen.
Syntax: element.dispatchEvent(event)
Contoh:

```
const event = new Event('click');
document.querySelector('button').dispatchEvent(event);
```

Output:
Memicu semua handler yang terkait event tersebut

Traversal & Navigation Methods

parentNode - node.parentNode

Penjelasan: Mengembalikan node induk dari node saat ini.
Syntax: node.parentNode
Contoh:

```
let parent = document.querySelector('li').parentNode;
```

Output:

```
<ul>...</ul>
```

childNodes - node.childNodes

Penjelasan: Mengembalikan NodeList dari semua anak (termasuk teks dan komentar).
Syntax: node.childNodes
Contoh:

```
let children = document.body.childNodes;
```

Output:

```
NodeList(…) [text, div, text, script, ...]
```

firstChild - node.firstChild

Penjelasan: Mengembalikan node anak pertama dari node.
Syntax: node.firstChild
Contoh:

```
let first = document.body.firstChild;
```

Output:

```
#text atau <div>
```

lastChild - node.lastChild

Penjelasan: Mengembalikan node anak terakhir dari node.
Syntax: node.lastChild
Contoh:

```
let last = document.body.lastChild;
```

Output:

```
#text atau <script>
```

nextSibling - node.nextSibling

Penjelasan: Mengembalikan node saudara setelah node ini (termasuk teks).
Syntax: node.nextSibling
Contoh:

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

Output:

```
#text atau elemen
```

Traversal & Navigation Methods

previousSibling - node.previousSibling

Penjelasan: Mengembalikan node saudara sebelum node ini.
Syntax: node.previousSibling
Contoh:

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

Output:

```
#text atau elemen
```

children - element.children

Penjelasan: Mengembalikan HTMLCollection dari anak elemen saja.
Syntax: element.children
Contoh:

```
let kids = document.body.children;
```

Output:

```
HTMLCollection [header, main, footer]
```

firstElementChild - element.firstElementChild

Penjelasan: Mengembalikan elemen anak pertama.
Syntax: element.firstElementChild
Contoh:

```
let firstEl = document.body.firstElementChild;
```

Output:

```
<header>...</header>
```

lastElementChild - element.lastElementChild

Penjelasan: Mengembalikan elemen anak terakhir.
Syntax: element.lastElementChild
Contoh:

```
let lastEl = document.body.lastElementChild;
```

Output:

```
<footer>...</footer>
```

nextElementSibling - element.nextElementSibling

Penjelasan: Mengembalikan elemen setelah elemen saat ini.
Syntax: element.nextElementSibling
Contoh:

```
let nextEl = document.querySelector('li').nextElementSibling;
```

Output:

```
<li>Item berikutnya</li>
```

Traversal & Navigation Methods

previousElementSibling - element.previousElementSibling

Penjelasan: Mengembalikan elemen sebelum elemen saat ini.
Syntax: element.previousElementSibling
Contoh:

```
let prevEl = document.querySelector('li').previousElementSibling;
```

Output:

```
<li>Item sebelumnya</li>
```

Attribute Methods

getAttribute - element.getAttribute(name)

Penjelasan: Mengambil nilai atribut dari elemen.
Syntax: element.getAttribute(name)
Contoh:

```
<div id="box"></div>
document.getElementById('box').getAttribute('id');
```

Output:

```
"box"
```

setAttribute - element.setAttribute(name, value)

Penjelasan: Menetapkan atau memperbarui nilai atribut.
Syntax: element.setAttribute(name, value)
Contoh:

```
let el = document.getElementById('box');
el.setAttribute('data-type', 'info');
```

Output:

```
<div id="box" data-type="info"></div>
```

removeAttribute - element.removeAttribute(name)

Penjelasan: Menghapus atribut dari elemen.
Syntax: element.removeAttribute(name)
Contoh:

```
let el = document.getElementById('box');
el.removeAttribute('id');
```

Output:

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

hasAttribute - element.hasAttribute(name)

Penjelasan: Mengembalikan true jika elemen memiliki atribut tertentu.
Syntax: element.hasAttribute(name)
Contoh:

```
let el = document.getElementById('box');
el.hasAttribute('id');
```

Output:

```
true
```

Style & Class Methods

style - element.style.property

Penjelasan: Digunakan untuk mengatur atau mengambil gaya inline pada elemen.
Syntax: element.style.property
Contoh:

```
document.getElementById('box').style.backgroundColor = 'red';
```

Output:

```
<div id="box" style="background-color: red;"></div>
```

getComputedStyle - getComputedStyle(element)

Penjelasan: Mengembalikan semua gaya CSS yang dihitung dari elemen.
Syntax: getComputedStyle(element)
Contoh:

```
let styles = getComputedStyle(document.getElementById('box'));
styles.backgroundColor;
```

Output:

```
"rgb(255, 0, 0)"
```

classList.add - element.classList.add(className)

Penjelasan: Menambahkan class ke elemen.
Syntax: element.classList.add(className)
Contoh:

```
document.getElementById('box').classList.add('active');
```

Output:

```
<div id="box" class="active"></div>
```

classList.remove - element.classList.remove(className)

Penjelasan: Menghapus class dari elemen.
Syntax: element.classList.remove(className)
Contoh:

```
document.getElementById('box').classList.remove('active');
```

Output:

```
<div id="box"></div>
```

classList.toggle - element.classList.toggle(className)

Penjelasan: Menambahkan class jika belum ada, dan menghapusnya jika sudah ada.
Syntax: element.classList.toggle(className)
Contoh:

```
document.getElementById('box').classList.toggle('hidden');
```

Output:

```
Menambah atau menghapus class "hidden"
```

Style & Class Methods

classList.contains - element.classList.contains(className)

Penjelasan: Memeriksa apakah class tertentu ada pada elemen.
Syntax: element.classList.contains(className)
Contoh:

```
document.getElementById('box').classList.contains('active');
```

Output:

```
true / false
```

classList.replace - element.classList.replace(oldClass, newClass)

Penjelasan: Mengganti class lama dengan class baru.
Syntax: element.classList.replace(oldClass, newClass)
Contoh:

```
document.getElementById('box').classList.replace('old', 'new');
```

Output:

```
<div id="box" class="new"></div>
```

Form & Input Methods

submit - form.submit()

Penjelasan: Mengirim form secara langsung tanpa trigger event submit.
Syntax: form.submit()
Contoh:

```
document.querySelector('form').submit();
```

Output:
Form dikirim ke server

reset - form.reset()

Penjelasan: Mengatur ulang form ke nilai defaultnya.
Syntax: form.reset()
Contoh:

```
document.querySelector('form').reset();
```

Output:
Semua input form direset

focus - element.focus()

Penjelasan: Memberikan fokus pada elemen input.
Syntax: element.focus()
Contoh:

```
document.getElementById('name').focus();
```

Output:
Kursor berpindah ke input dengan id 'name'

blur - element.blur()

Penjelasan: Menghapus fokus dari elemen.
Syntax: element.blur()
Contoh:

```
document.getElementById('name').blur();
```

Output:
Fokus hilang dari input

select - input.select()

Penjelasan: Menyeleksi semua teks dalam input.
Syntax: input.select()
Contoh:

```
document.getElementById('text').select();
```

Output:
Semua teks di dalam input disorot

Form & Input Methods

setCustomValidity - input.setCustomValidity(message)

Penjelasan: Mengatur pesan validasi kustom untuk input.
Syntax: input.setCustomValidity(message)
Contoh:

```
input.setCustomValidity('Kolom ini wajib diisi');
```

Output:
Validasi akan gagal dan pesan ditampilkan

checkValidity - form.checkValidity()

Penjelasan: Memeriksa apakah seluruh form valid.
Syntax: form.checkValidity()
Contoh:

```
document.querySelector('form').checkValidity();
```

Output:
true / false

reportValidity - form.reportValidity()

Penjelasan: Menampilkan pesan validasi jika form tidak valid.
Syntax: form.reportValidity()
Contoh:

```
document.querySelector('form').reportValidity();
```

Output:
true jika valid, jika tidak maka browser akan menampilkan pesan validasi

Selection & Range Methods

getSelection - window.getSelection()

Penjelasan: Mengambil objek selection saat ini (teks yang dipilih pengguna).
Syntax: window.getSelection()
Contoh:

```
let selection = window.getSelection();
```

Output:

```
Selection { anchorNode: ..., focusNode: ..., ... }
```

createRange - document.createRange()

Penjelasan: Membuat objek Range kosong untuk manipulasi DOM.
Syntax: document.createRange()
Contoh:

```
let range = document.createRange();
```

Output:

```
Range {}
```

range.setStart - range.setStart(node, offset)

Penjelasan: Mengatur titik awal range.
Syntax: range.setStart(node, offset)
Contoh:

```
range.setStart(paragraph, 0);
```

Output:

```
Range dimulai dari awal paragraf
```

range.setEnd - range.setEnd(node, offset)

Penjelasan: Mengatur titik akhir range.
Syntax: range.setEnd(node, offset)
Contoh:

```
range.setEnd(paragraph, 1);
```

Output:

```
Range berakhir setelah node pertama
```

range.selectNode - range.selectNode(node)

Penjelasan: Memilih seluruh node sebagai range.
Syntax: range.selectNode(node)
Contoh:

```
range.selectNode(document.querySelector('p'));
```

Output:

```
Range mencakup seluruh elemen <p>
```

Selection & Range Methods

range.deleteContents - range.deleteContents()

Penjelasan: Menghapus semua konten dalam range.
Syntax: range.deleteContents()
Contoh:

```
range.deleteContents();
```

Output:
Konten yang dipilih terhapus

range.insertNode - range.insertNode(node)

Penjelasan: Menyisipkan node ke posisi range.
Syntax: range.insertNode(node)
Contoh:

```
range.insertNode(document.createTextNode('Hello'));
```

Output:
Teks 'Hello' disisipkan pada range

Window & Screen Methods

alert - window.alert(message)

Penjelasan: Menampilkan kotak dialog peringatan dengan pesan.
Syntax: alert(message)
Contoh:

```
alert('Hello, world!');
```

Output:
Kotak dialog dengan pesan 'Hello, world!'

confirm - window.confirm(message)

Penjelasan: Menampilkan kotak dialog konfirmasi dengan tombol OK dan Cancel.
Syntax: confirm(message)
Contoh:

```
confirm('Apakah Anda yakin?');
```

Output:
true jika OK ditekan, false jika Cancel ditekan

prompt - window.prompt(message, default)

Penjelasan: Menampilkan kotak dialog input teks kepada pengguna.
Syntax: prompt(message, defaultText)
Contoh:

```
prompt('Masukkan nama:', 'Anonim');
```

Output:
Nilai yang diketik pengguna, atau null jika dibatalkan

setTimeout - window.setTimeout(function, delay)

Penjelasan: Menjalankan fungsi setelah jeda tertentu (ms).
Syntax: setTimeout(function, milliseconds)
Contoh:

```
setTimeout(() => alert('Delay 2 detik'), 2000);
```

Output:
Menampilkan alert setelah 2 detik

setInterval - window.setInterval(function, interval)

Penjelasan: Menjalankan fungsi berulang tiap interval waktu tertentu.
Syntax: setInterval(function, milliseconds)
Contoh:

```
setInterval(() => console.log('Ulang'), 1000);
```

Output:
Menulis 'Ulang' ke konsol setiap 1 detik

Window & Screen Methods

clearTimeout - window.clearTimeout(id)

Penjelasan: Membatalkan fungsi yang dijadwalkan dengan setTimeout.
Syntax: clearTimeout(id)
Contoh:

```
let id = setTimeout(fn, 3000);
clearTimeout(id);
```

Output:
Fungsi tidak akan dijalankan

clearInterval - window.clearInterval(id)

Penjelasan: Membatalkan eksekusi berulang yang dijadwalkan dengan setInterval.
Syntax: clearInterval(id)
Contoh:

```
let id = setInterval(fn, 1000);
clearInterval(id);
```

Output:
Interval dihentikan

scrollTo - window.scrollTo(x, y)

Penjelasan: Menggulir halaman ke koordinat tertentu.
Syntax: scrollTo(x, y)
Contoh:

```
window.scrollTo(0, 500);
```

Output:
Halaman menggulir ke bawah 500px

scrollBy - window.scrollBy(x, y)

Penjelasan: Menggulir halaman relatif terhadap posisi saat ini.
Syntax: scrollBy(x, y)
Contoh:

```
window.scrollBy(0, 100);
```

Output:
Halaman menggulir turun 100px dari posisi sekarang

print - window.print()

Penjelasan: Membuka dialog cetak untuk halaman saat ini.
Syntax: print()
Contoh:

```
window.print();
```

Output:
Menampilkan dialog cetak browser

Window & Screen Methods

open - window.open(url, name, specs)

Penjelasan: Membuka jendela atau tab browser baru dengan URL.
Syntax: open(url, name, specs)
Contoh:

```
window.open('https://example.com', '_blank');
```

Output:
Tab baru membuka https://example.com

close - window.close()

Penjelasan: Menutup jendela browser yang dibuka oleh script.
Syntax: close()
Contoh:

```
window.close();
```

Output:
Jendela script saat ini akan tertutup