

```
> document.getElementById('title')
< <h1 id="title" class="heading">DOM learning on Chai aur code
  </h1>

> document.getElementById('title').id
< 'title'

> document.getElementById('title').class
< undefined

> document.getElementById('title').className
< 'heading'
```

```
> document.getElementById('title').getAttribute('id')
< 'title'

> document.getElementById('title').getAttribute('class')
< 'heading'

> document.getElementById('title').setAttribute('class', 'test')
< undefined

> document.getElementById('title').setAttribute('class', 'test
  heading')
< undefined
```

```
> document.getElementById('title').getAttribute('id')
< 'title'

> document.getElementById('title').getAttribute('class')
< 'heading'

> document.getElementById('title').setAttribute('class', 'test')
< undefined

> document.getElementById('title').setAttribute('class', 'test
heading')
< undefined

> document.getElementById('title')
< <h1 id="title" class="test heading">DOM learning on Chai aur
code</h1>

> const title = document.getElementById('title')
< undefined
```

```
> title.innerText
```

```
< 'DOM learning on Chai aur code'
```

```
> title.textContent
```

```
< 'DOM learning on Chai aur code test text'
```

```
>
```

```
> title.innerHTML
```

```
< 'DOM learning on Chai aur code <span style="display: none;">test t  
  ext</span>'
```



```
> document.querySelector('h2')
```

```
< <h2>Lorem ipsum dolor sit.</h2>
```

```
> document.querySelector('#title')
```

```
< ▶<h1 id="title" class="heading">...</h1>
```

```
> document.querySelector('.heading')
```

```
< ▶<h1 id="title" class="heading">...</h1>
```

```
> |
```

```
> document.querySelector('input[type="password"]')
```

```
<< <input type="password" name id data-dashlane-rid="e252f14fa2c66f  
2f" data-form-type="password" data-kwimpalastatus="alive" data-  
kwimpalaid="1685620477907-0">
```

```
> document.querySelector('ul')
```

```
< ▶ <ul>...</ul>
```

```
> const myul = document.querySelector('ul')
```

```
< undefined
```

```
> myul.querySelector('li')
```

```
< ▶ <li>...</li>
```

```
> const turnGreen = myul.querySelector('li')
```

```
< undefined
```

```
> turnGreen.style.backgroundColor = "green"
```

```
< 'green'
```

```
> turnGreen.style.padding = "10px"
```

```
< '10px'
```

```
> |
```



# DOM learning on Chai aur code

Lorem ipsum dolor sit.

Lorem ipsum dolor sit.

Lorem ipsum dolor sit.

Lorem ipsum dolor sit amet.

- five
- two
- three

× Expression  
not available

> turnGreen.innerText

< 'one'

> turnGreen.innerText = "five"

< 'five'

>

## DOM learning on Chai aur code

Lorem ipsum dolor sit.

Lorem ipsum dolor sit.

Lorem ipsum dolor sit.

Lorem ipsum dolor sit amet.

• one

• two

• three

2 Issues: 2

× Expression  
not available

```
> const myH1 = document.querySelectorAll('h1')  
< undefined
```

```
> myH1  
< ▶ NodeList [h1#title.heading]
```

```
> myH1.style.color = 'green'
```

× ▶ Uncaught TypeError: Cannot set properties of undefined (setting 'color')  
at <anonymous>:1:18 VM2298:1

```
> myH1[0].style.color = 'green'  
< 'green'
```

```
> |
```

```
> document.querySelectorAll('li')
< ▶ NodeList(3) [li, li, li]

> cosnt tempLiList = document.querySelectorAll('li')
✖ Uncaught SyntaxError: Unexpected identifier 'tempLiList' VM1863:1

> const tempLiList = document.querySelectorAll('li')
< undefined

> tempLiList
< ▶ NodeList(3) [li, li, li]

> |
```

```
> const tempLiList = document.querySelectorAll('li')
```

```
< undefined
```

```
> tempLiList
```

```
< ▶ NodeList(3) [li, li, li]
```

```
> tempLiList.style.color = 'green'
```

```
✖ ▶ Uncaught TypeError: Cannot set properties of undefined  
   (setting 'color')  
     at <anonymous>:1:24
```

```
> tempLiList[0].style.color = 'green'
```

```
< 'green'
```

```
> tempLiList
```

```
< ▶ NodeList(3) [li, li, li]
```

```
> tempLiList.forEach(function (l) {})
```

```
< undefined
```

```
> tempLiList.forEach(function (l) {  
    l.style.backgroundColor = 'green'  
})
```

```
< undefined
```

```
> |
```



```
> document.getElementsByClassName('list-item')
< HTMLCollection(4) [li.list-item, li.list-item, li.list-item,
  ▼ li.list-item] ⓘ
  ▶ 0: li.list-item
  ▶ 1: li.list-item
  ▶ 2: li.list-item
  ▶ 3: li.list-item
    length: 4
  ▼ [[Prototype]]: HTMLCollection
    ▶ item: f item()
      length: (...)
    ▶ namedItem: f namedItem()
    ▶ constructor: f HTMLCollection()
    ▶ Symbol(Symbol.iterator): f values()
      Symbol(Symbol.toStringTag): "HTMLCollection"
    ▶ get length: f length()
    ▶ [[Prototype]]: Object
```

1. **document.getElementsByTagName():**

Returns an HTMLCollection of all elements with the specified tag name

```
let paragraphs = document.getElementsByTagName('p');
```

2. **document.getElementsByClassName():**

Returns an HTMLCollection of all elements with the specified class name

```
let elements = document.getElementsByClassName('my-class');
```

3. **document.forms:**

Returns an HTMLCollection of all <form> elements in the document.

```
let forms = document.forms;
```

4. **document.images:**

Returns an HTMLCollection of all <img> elements in the document.

```
let images = document.images;
```

5. **document.links:**

Returns an HTMLCollection of all anchor (<a>) elements with an href attribute.

```
let links = document.links;
```

6. **document.getElementsByName():**

Returns an HTMLCollection of all elements with a specified name attribute.

```
let radios = document.getElementsByName('gender');
```

### **Key Characteristics of HTMLCollection:**

It is live: This means that if the document structure changes (e.g., new elements are added or removed), the HTMLCollection is automatically updated to reflect those changes.

It does not support all array methods, but you can convert it into an array using `Array.from()` or the spread operator (...).

```
let paragraphsArray = Array.from(document.getElementsByTagName('p'));
```

```
// or
```

```
let paragraphsArray = [...document.getElementsByTagName('p')];
```

For modern development, you may prefer using `querySelectorAll()` because it returns a static `NodeList` (which does not auto-update) and supports more flexible CSS-style selectors.

✕ Expression  
not available

> tempClassList

◀ ▶ *HTMLCollection(4) [li.list-item, li.list-item, li.list-item, li.list-item]*

> Array.from(tempClassList)

◀ ▶ (4) [li.list-item, li.list-item, li.list-item, li.list-item]

> |

```
> const myH2 = document.querySelectorAll('.mw-headline')
```

```
< undefined
```

```
> myH2[0].innerText
```

```
< 'History'
```



```
> myH2.forEach(function(h){  
    h.style.color = 'red'  
})
```

```
< undefined
```

```
> myH2.forEach(function(h){  
    h.style.color = 'black';  
    h.style.backgroundColor = 'green';  
    h.style.padding = '10px'  
})
```

```
< undefined
```

```
> myH2.forEach(function(h){  
    h.style.color = 'black';  
    h.style.backgroundColor = 'green';  
    h.style.padding = '10px';  
    h.innerText = "Hitesh"  
})
```