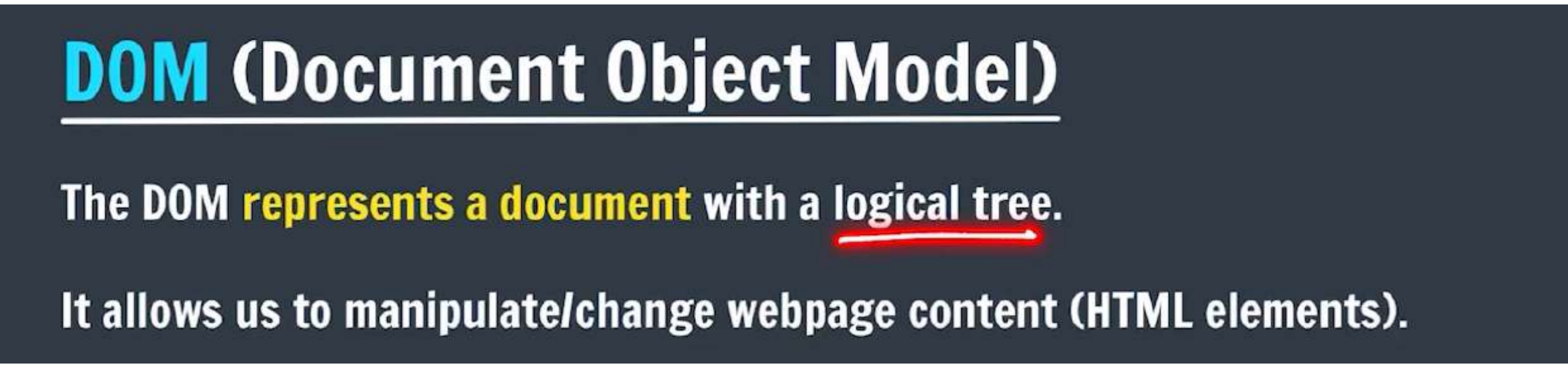
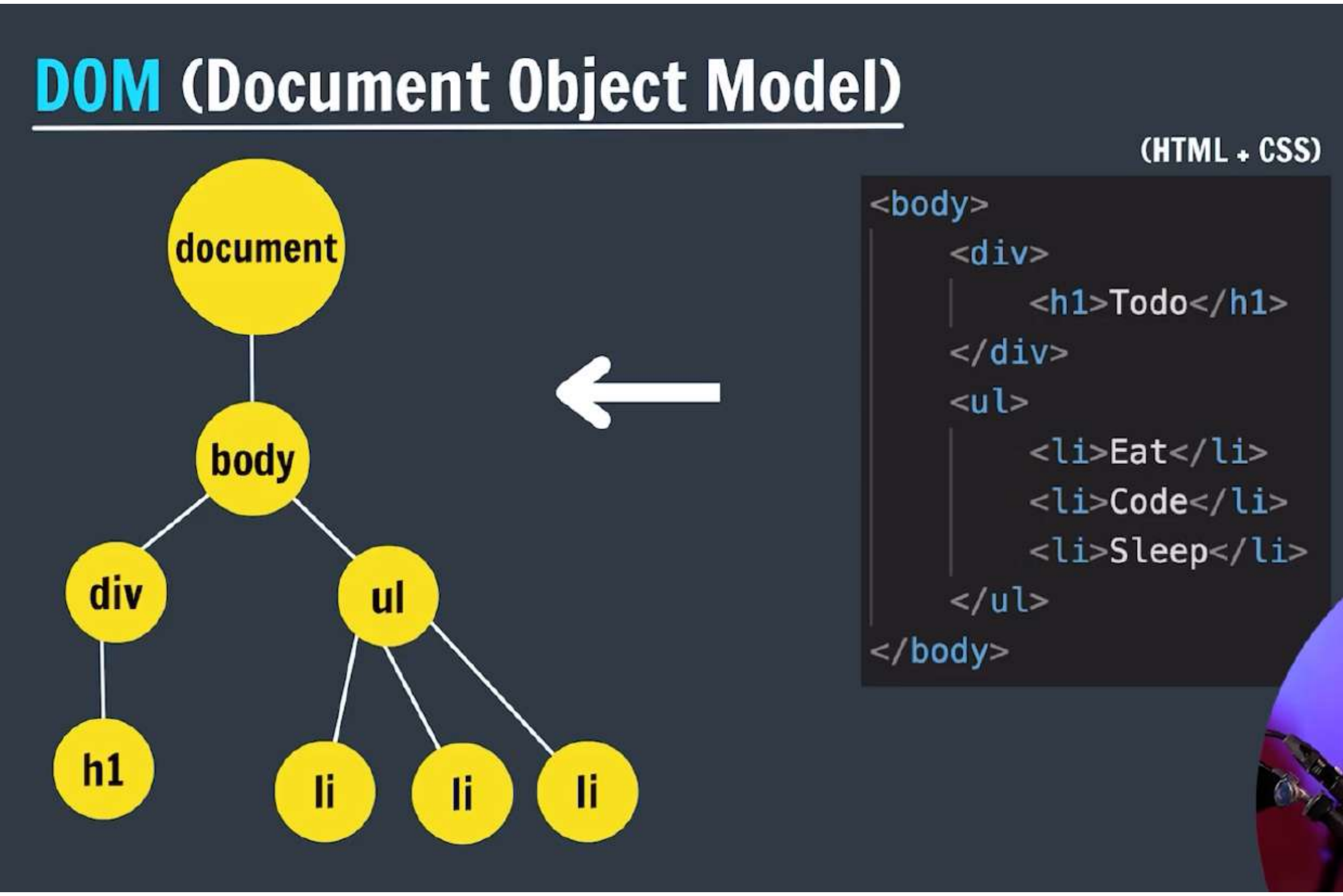


DOM (Document Object Model)

The DOM **represents a document** with a logical tree.

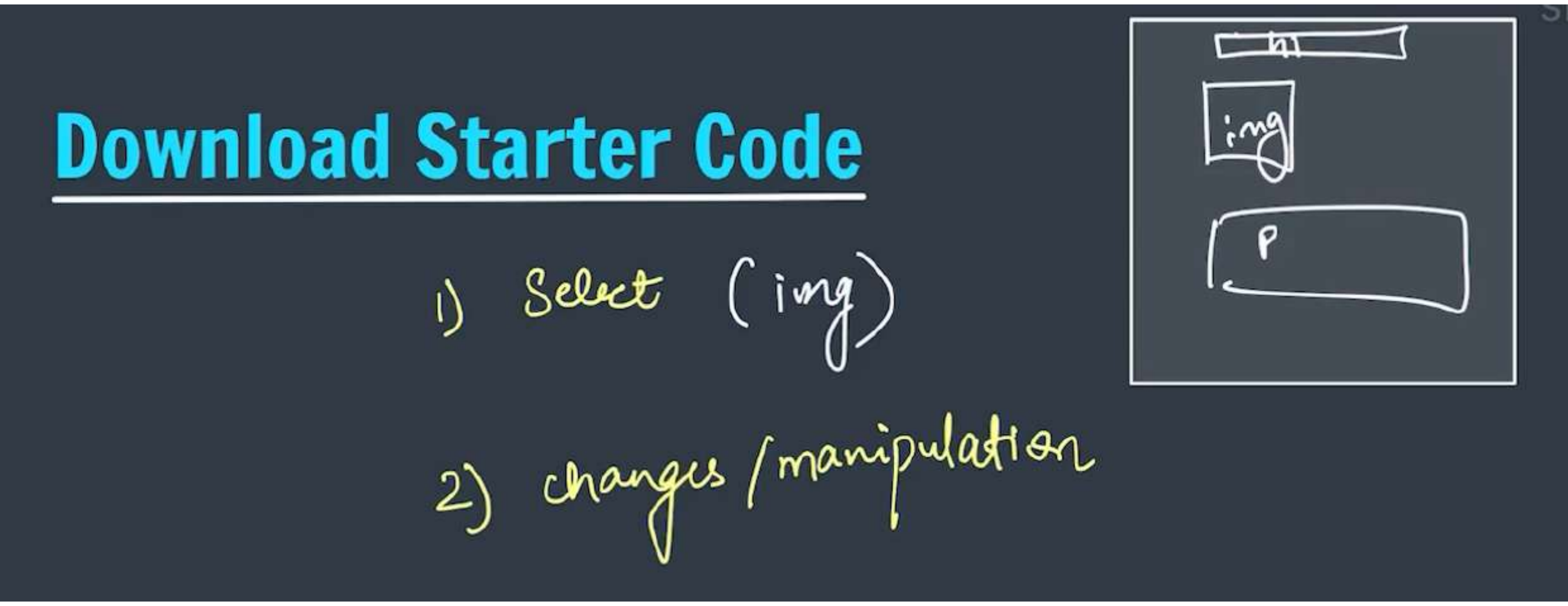
It allows us to manipulate/change webpage content (HTML elements).



DOM (Document Object Model)

(HTML + CSS)

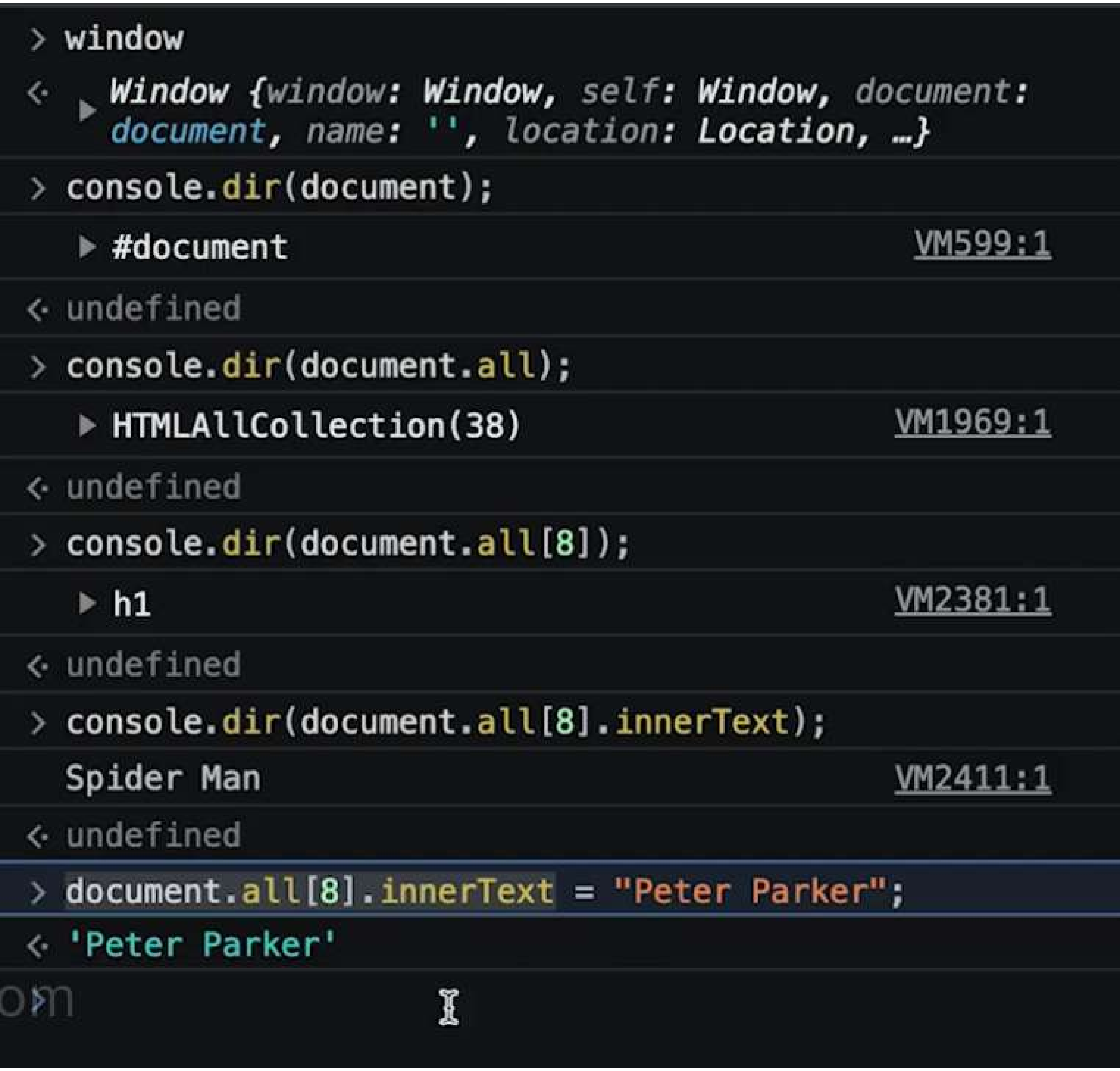
```
<body>
    <div>
        <h1>Todo</h1>
    </div>
    <ul>
        <li>Eat</li>
        <li>Code</li>
        <li>Sleep</li>
    </ul>
</body>
```



Download Starter Code

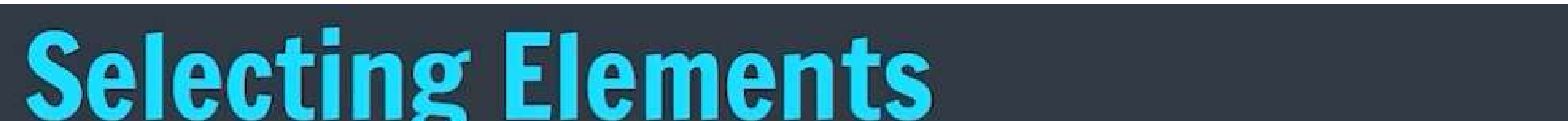
1) Select (img)

2) changes / manipulation



```
> window
<· ▶ Window {window: Window, self: Window, document:
      document, name: '', location: Location, …}
> console.dir(document);
    ▶ #document                                      VM599:1
<· undefined
> console.dir(document.all);
    ▶ HTMLAllCollection(38)                         VM1969:1
<· undefined
> console.dir(document.all[8]);
    ▶ h1                                            VM2381:1
<· undefined
> console.dir(document.all[8].innerText);
  Spider Man                                        VM2411:1
<· undefined
> document.all[8].innerText = "Peter Parker";
<· 'Peter Parker'
>
```

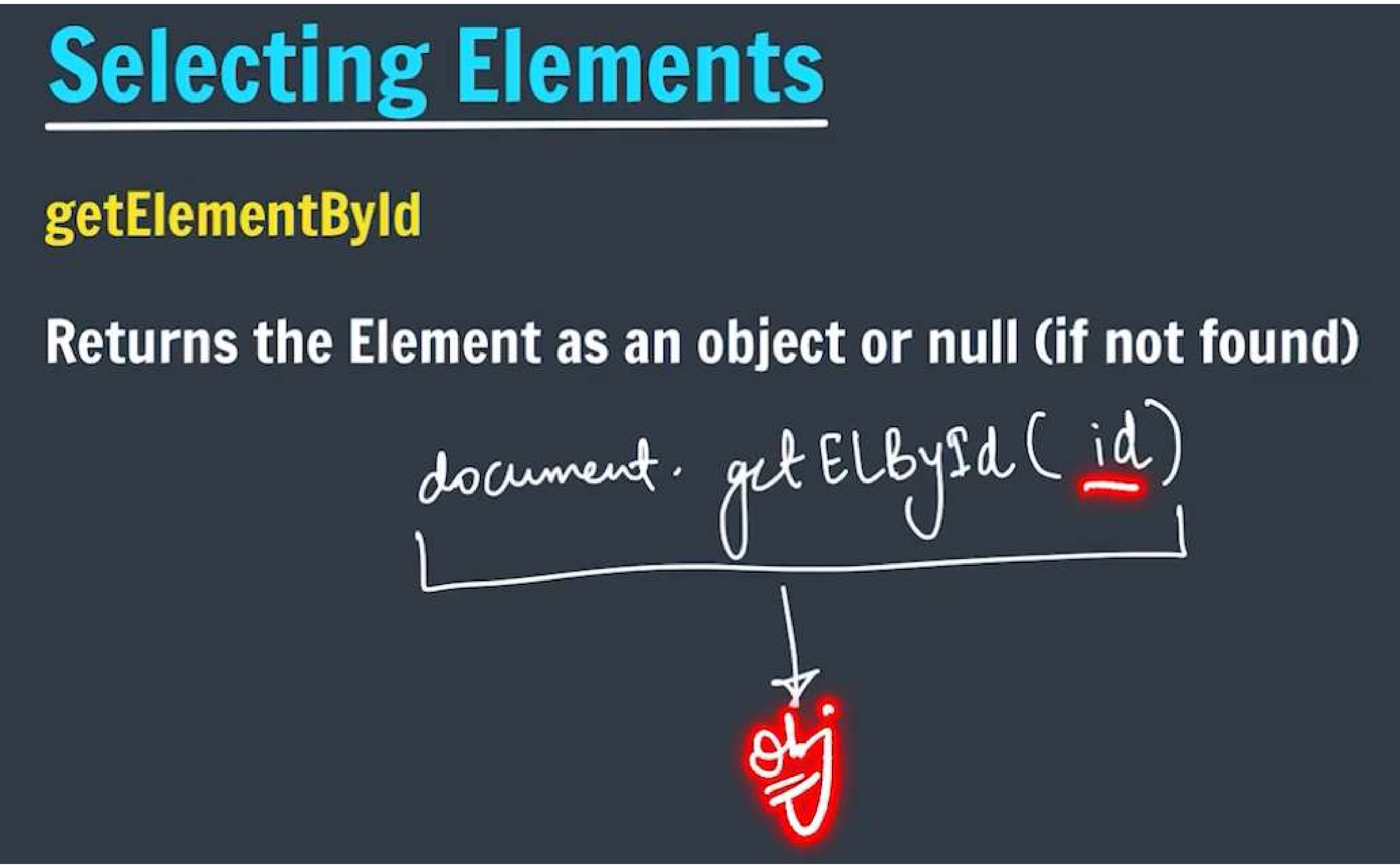
Peter Parker



Selecting Elements

getElementById

Returns the Element as an object or null (if not found)



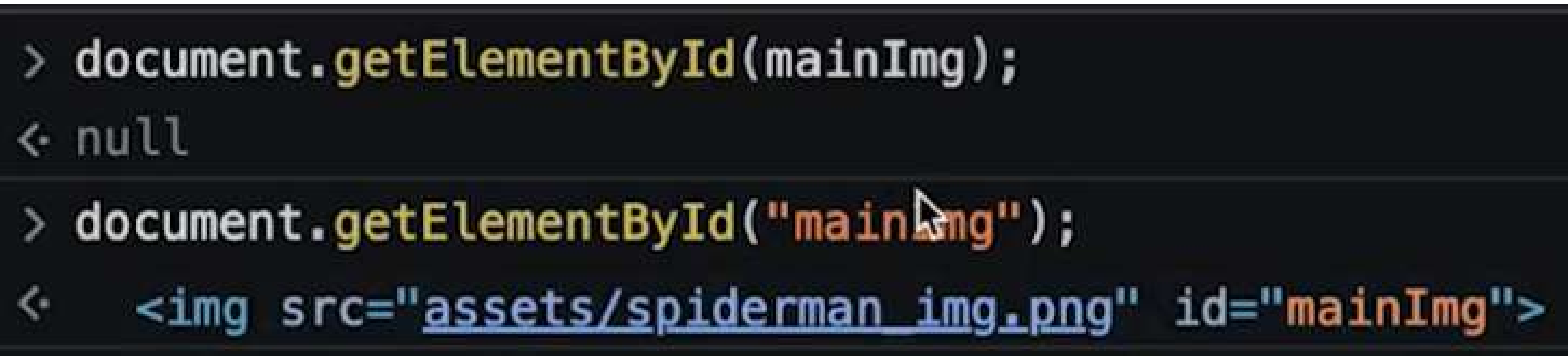
Selecting Elements

getElementById

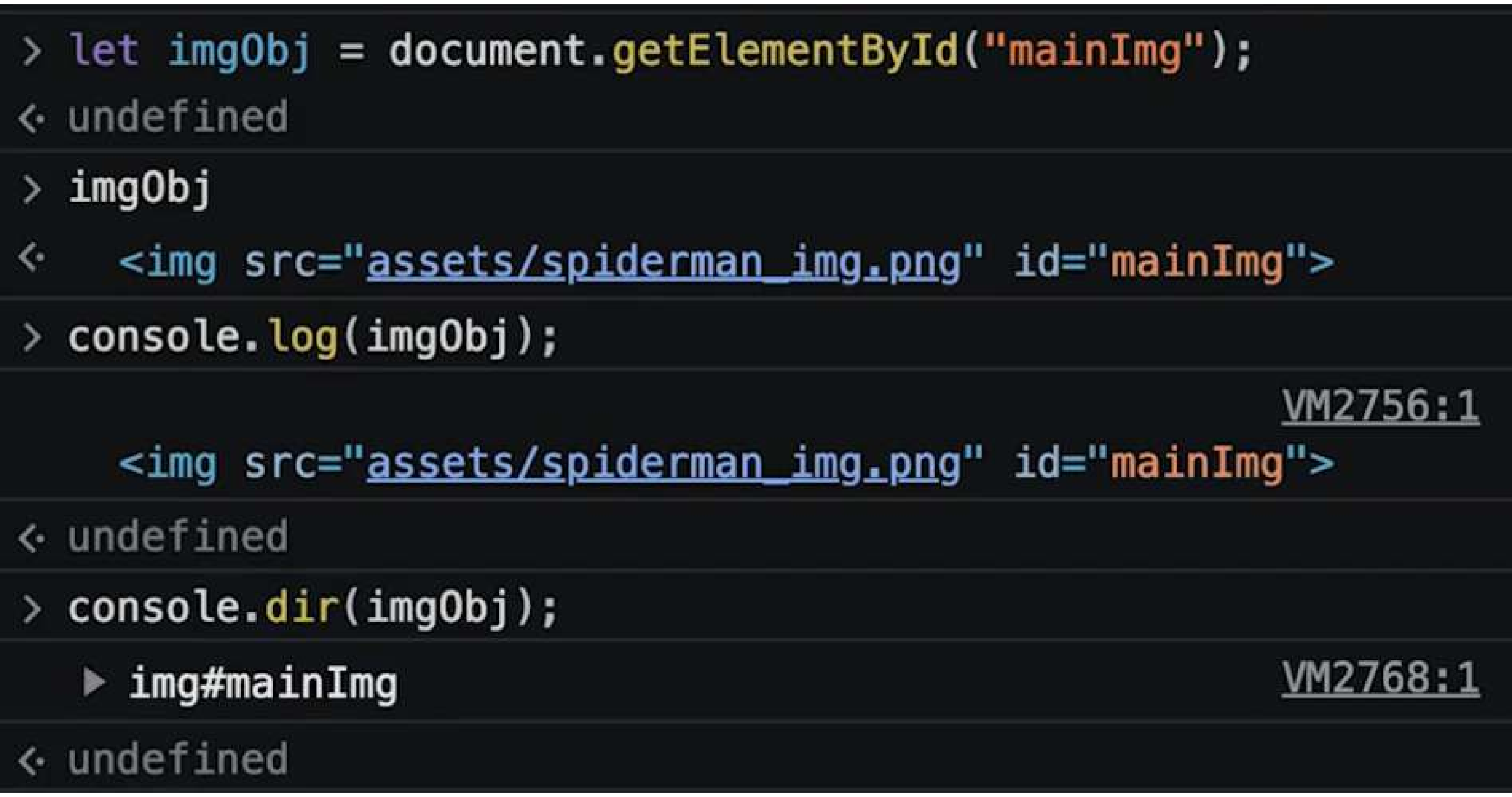
Returns the Element as an object or null (if not found)

document. getElById(id)

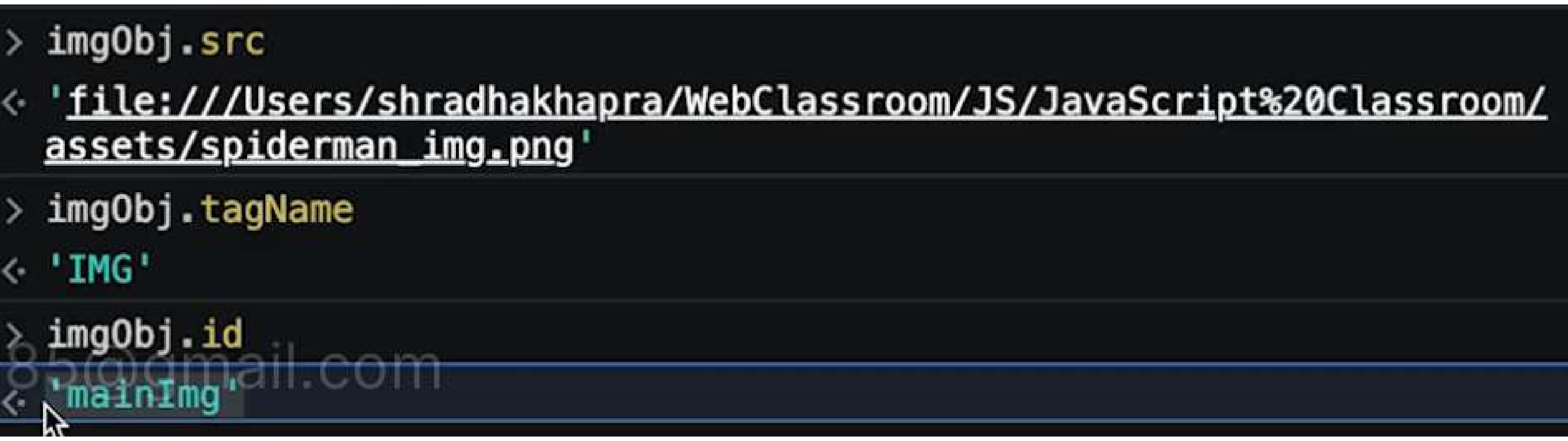
obj



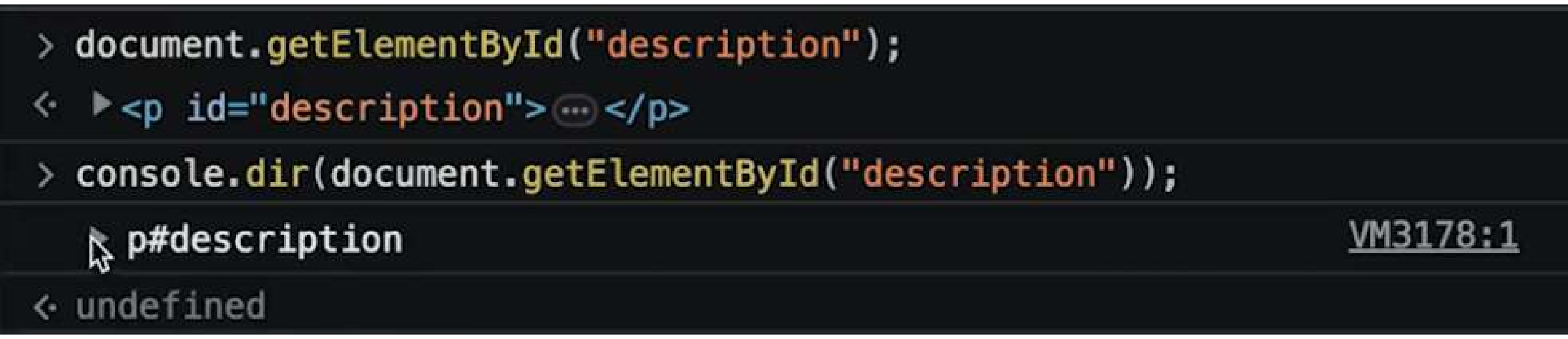
```
> document.getElementById(mainImg);
<· null
> document.getElementById("mainImg");
<·   <img src="assets/spiderman_img.png" id="mainImg">
```



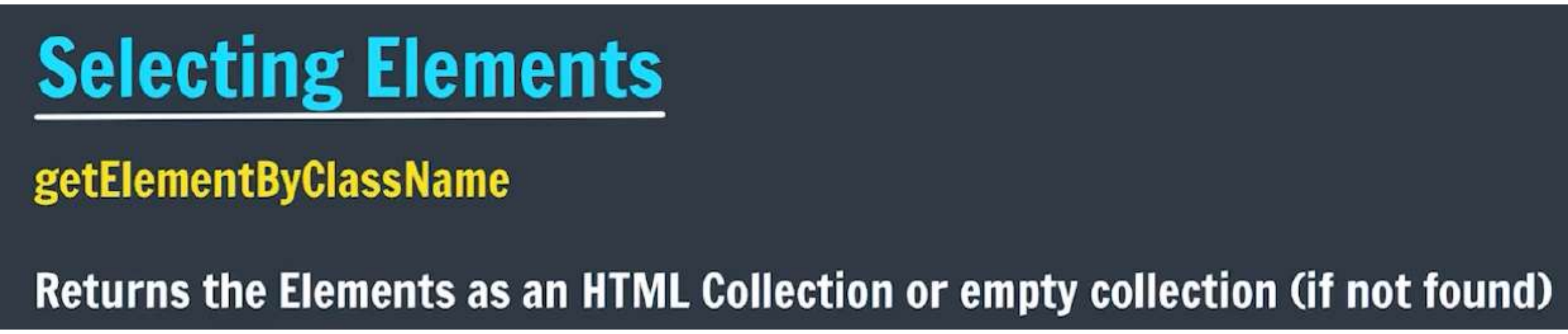
```
> let imgObj = document.getElementById("mainImg");
<· undefined
> imgObj
<·   <img src="assets/spiderman_img.png" id="mainImg">
> console.log(imgObj);
                                                      VM2756:1
     <img src="assets/spiderman_img.png" id="mainImg">
<· undefined
> console.dir(imgObj);
   ▶ img#mainImg                                      VM2768:1
<· undefined
```



```
> imgObj.src
<· 'file:///Users/shradhakhapra/WebClassroom/JS/JavaScript%20Classroom/
  assets/spiderman_img.png'
> imgObj.tagName
<· 'IMG'
> imgObj.id
<· 'mainImg'
```



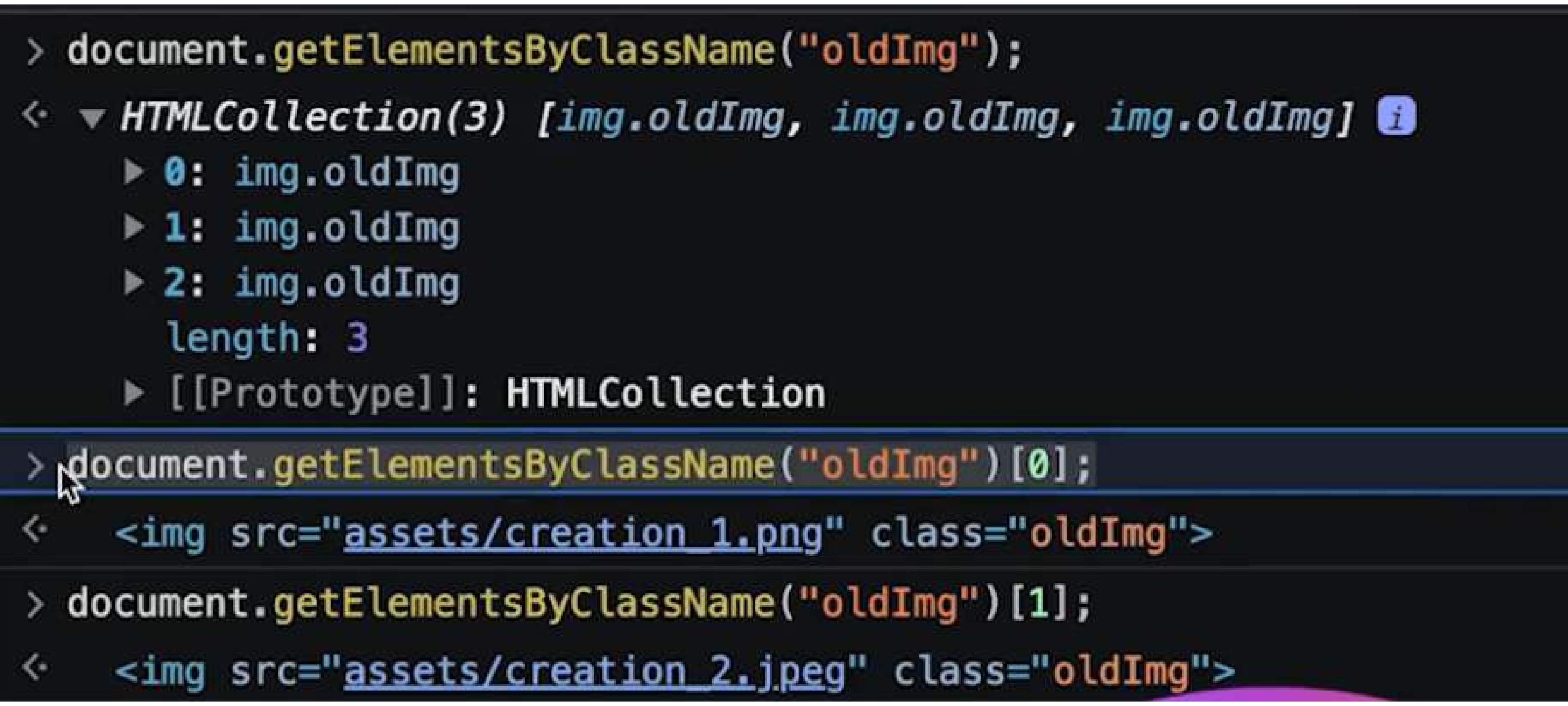
```
> document.getElementById("description");
<  ▶<p id="description">⋯</p>
> console.dir(document.getElementById("description"));
   ▶ p#description                                            VM3178:1
< undefined
```



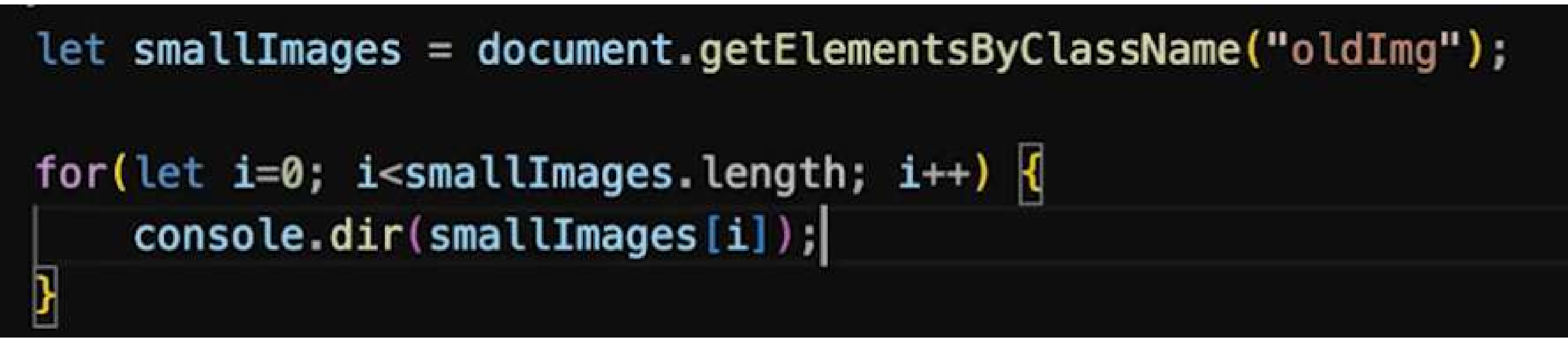
Selecting Elements

getElementByClassName

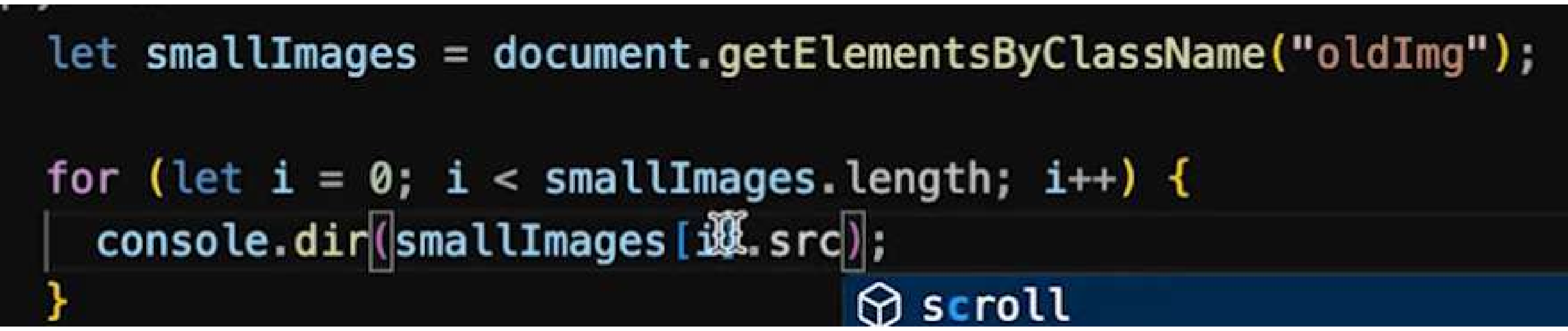
Returns the Elements as an HTML Collection or empty collection (if not found)



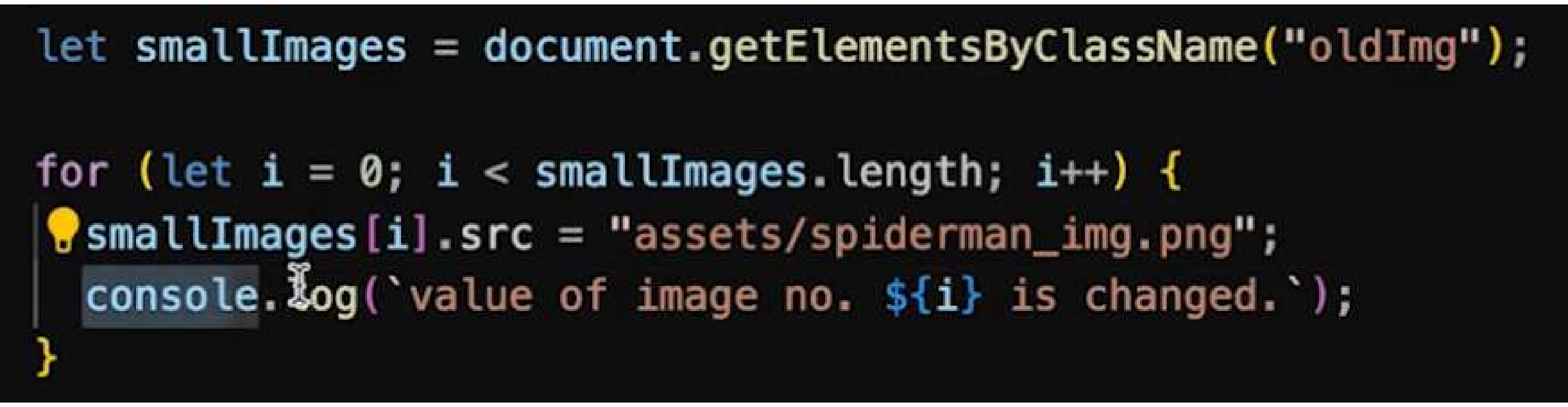
```
> document.getElementsByClassName("oldImg");
< ▼ HTMLCollection(3) [img.oldImg, img.oldImg, img.oldImg] i
    ▶ 0: img.oldImg
    ▶ 1: img.oldImg
    ▶ 2: img.oldImg
      length: 3
    ▶ [[Prototype]]: HTMLCollection
> document.getElementsByClassName("oldImg")[0];
<   <img src="assets/creation_1.png" class="oldImg">
> document.getElementsByClassName("oldImg")[1];
<   <img src="assets/creation_2.jpeg" class="oldImg">
```



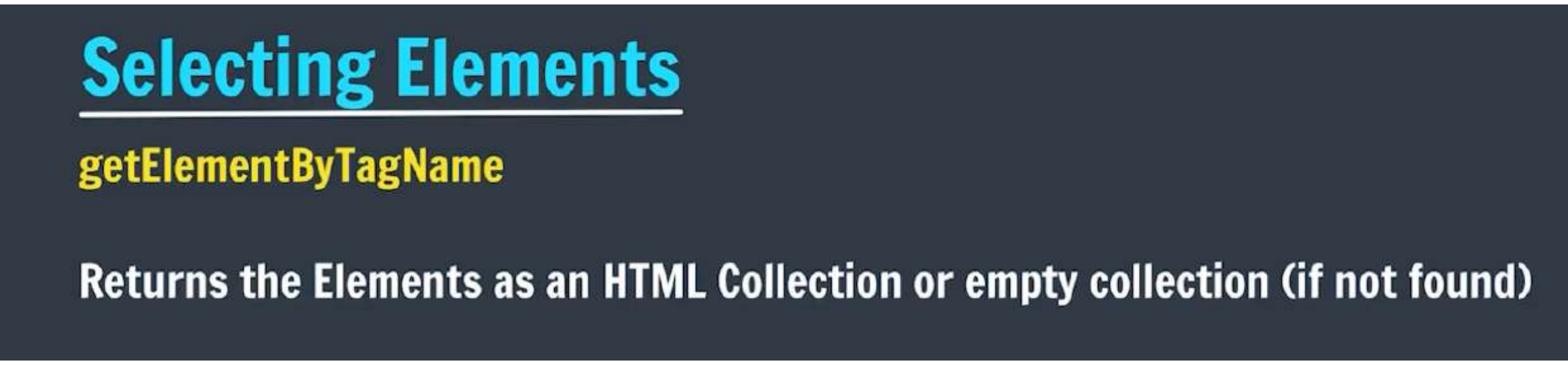
```
let smallImages = document.getElementsByClassName("oldImg");

for(let i=0; i<smallImages.length; i++) {
    console.dir(smallImages[i]);
}
```



```
let smallImages = document.getElementsByClassName("oldImg");

for (let i = 0; i < smallImages.length; i++) {
  console.dir(smallImages[i].src);
}
```



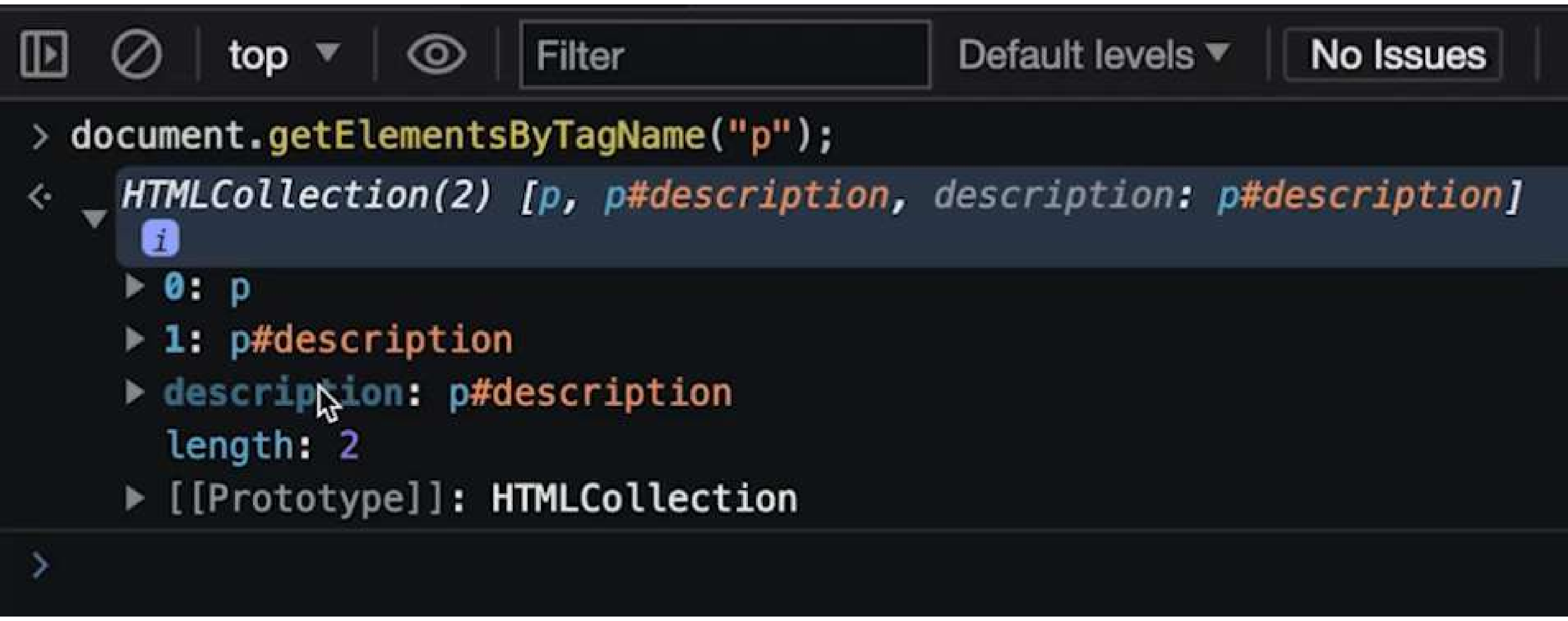
```
let smallImages = document.getElementsByClassName("oldImg");

for (let i = 0; i < smallImages.length; i++) {
  smallImages[i].src = "assets/spiderman_img.png";
  console.log(`value of image no. ${i} is changed.`);
}
```



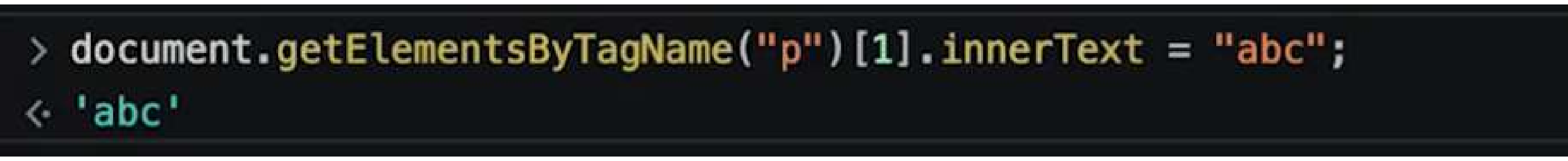
Selecting Elements

getElementByTagName

Returns the Elements as an HTML Collection or empty collection (if not found)

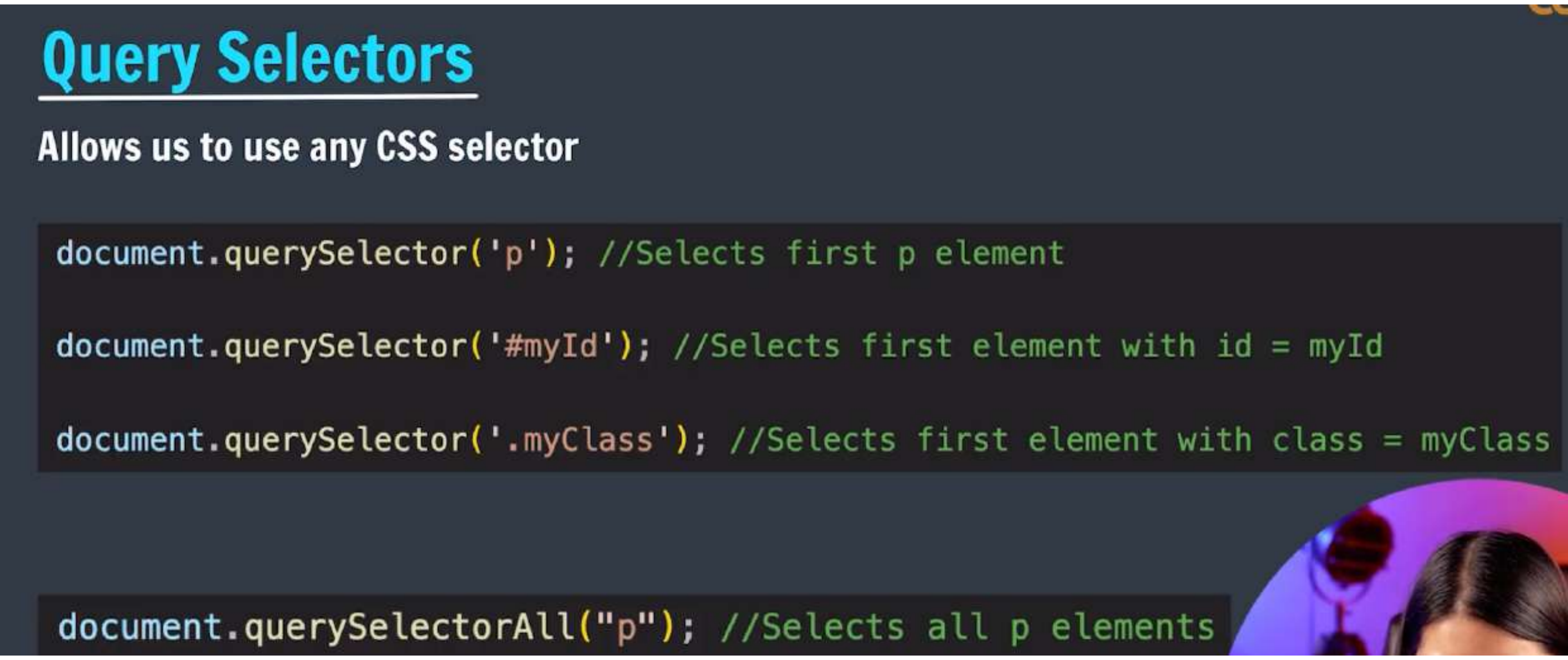


```
> document.getElementsByTagName("p");
<· ▼ HTMLCollection(2) [p, p#description, description: p#description]
     ▶ 0: p
     ▶ 1: p#description
     ▶ description: p#description
       length: 2
     ▶ [[Prototype]]: HTMLCollection
>
```



```
> document.getElementsByTagName("p")[1].innerText = "abc";
<· 'abc'
```

abc

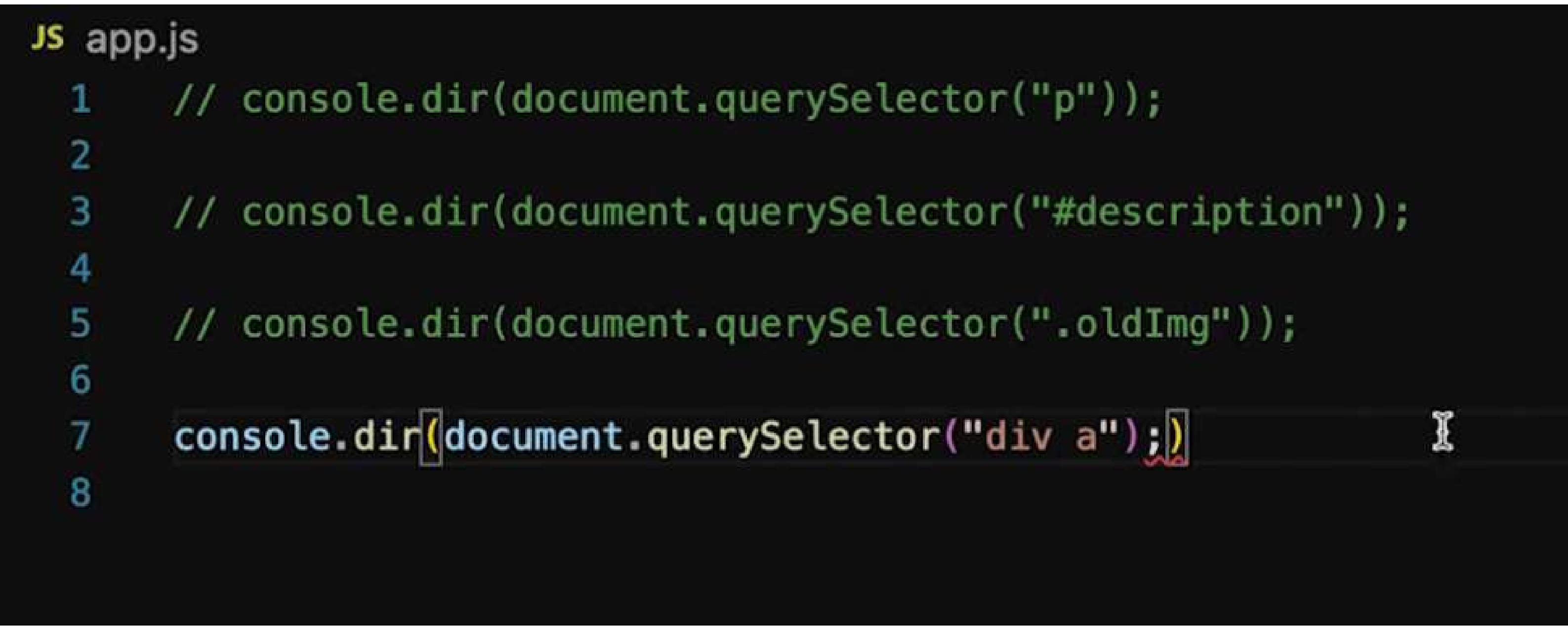


Query Selectors

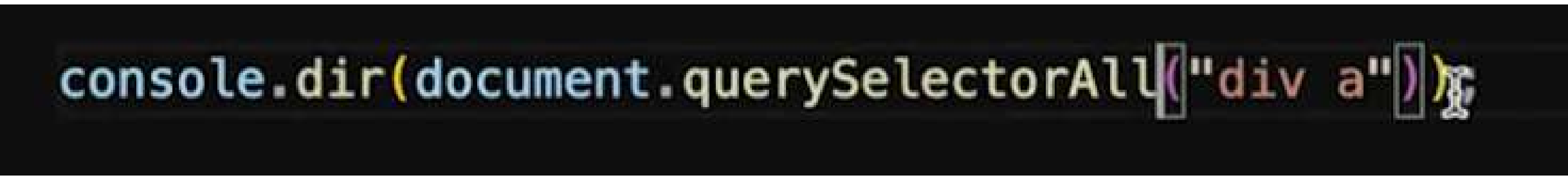
Allows us to use any CSS selector

```
document.querySelector('p'); //Selects first p element

document.querySelector('#myId'); //Selects first element with id = myId

document.querySelector('.myClass'); //Selects first element with class = myClass
```

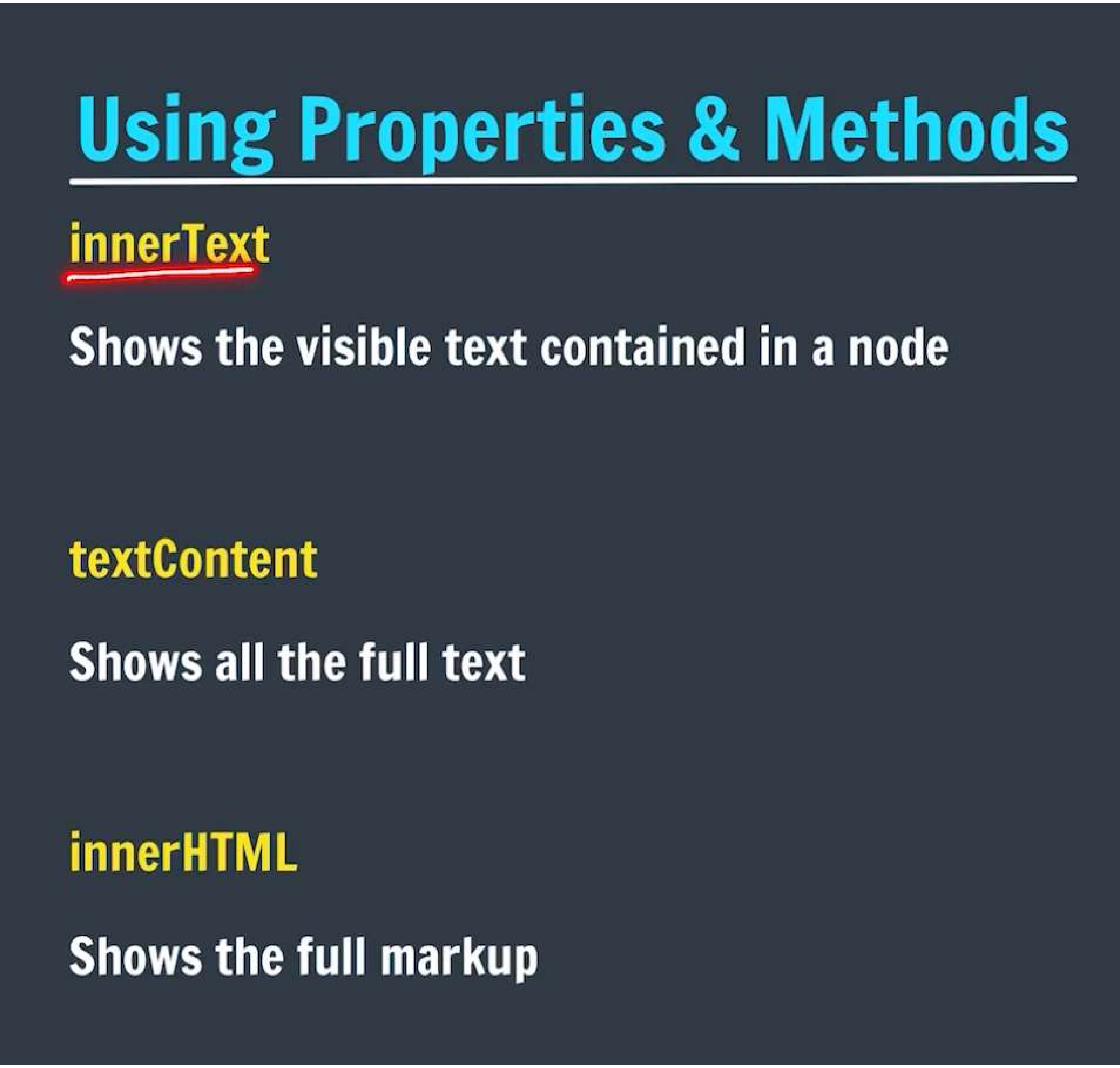
```
document.querySelectorAll("p"); //Selects all p elements
```



```
// console.dir(document.querySelector("p"));

// console.dir(document.querySelector("#description"));

// console.dir(document.querySelector(".oldImg"));

console.dir(document.querySelector("div a");)

```



```
console.dir(document.querySelectorAll("div a"));
```



Using Properties & Methods

innerText

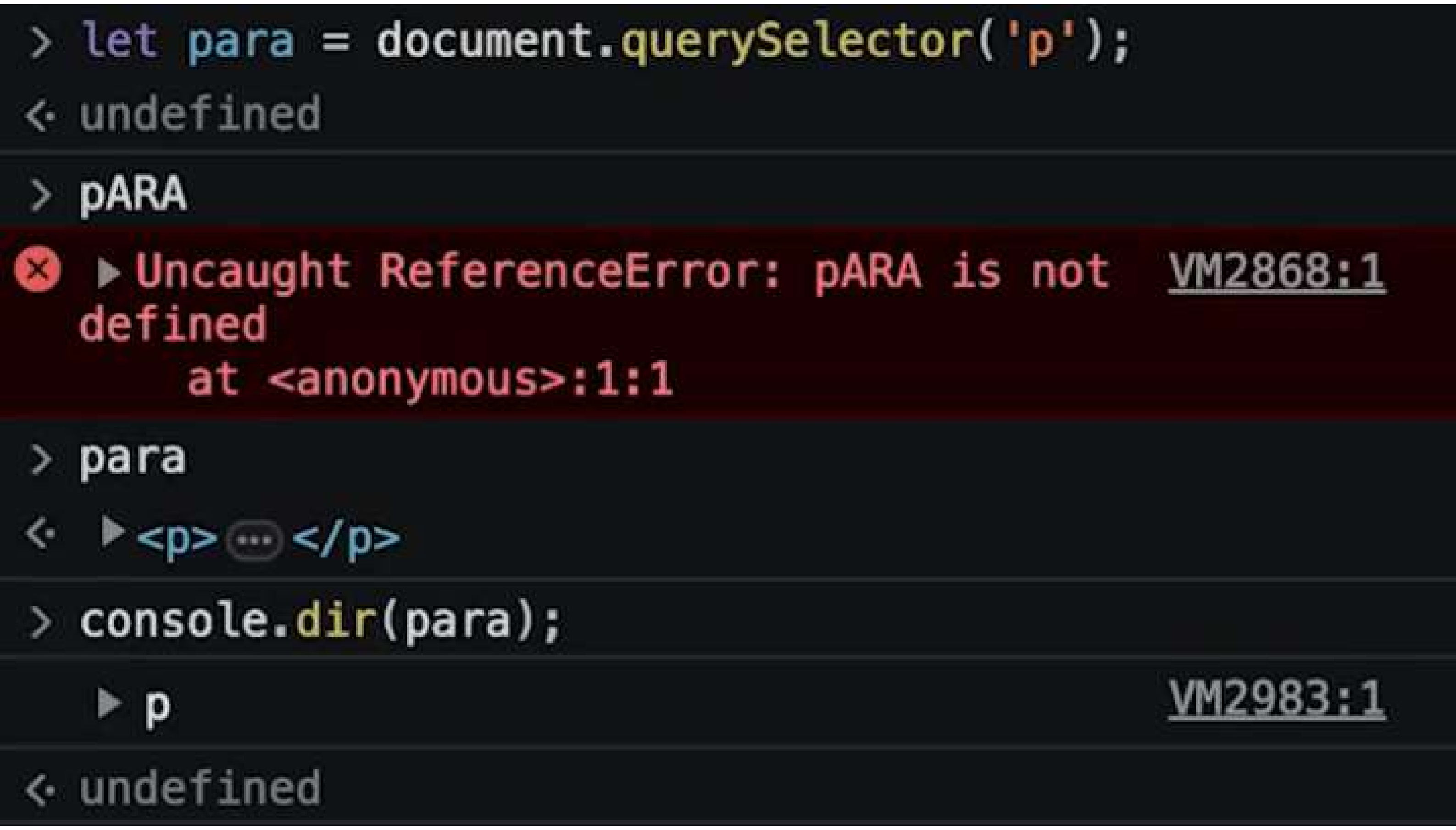
Shows the visible text contained in a node

textContent

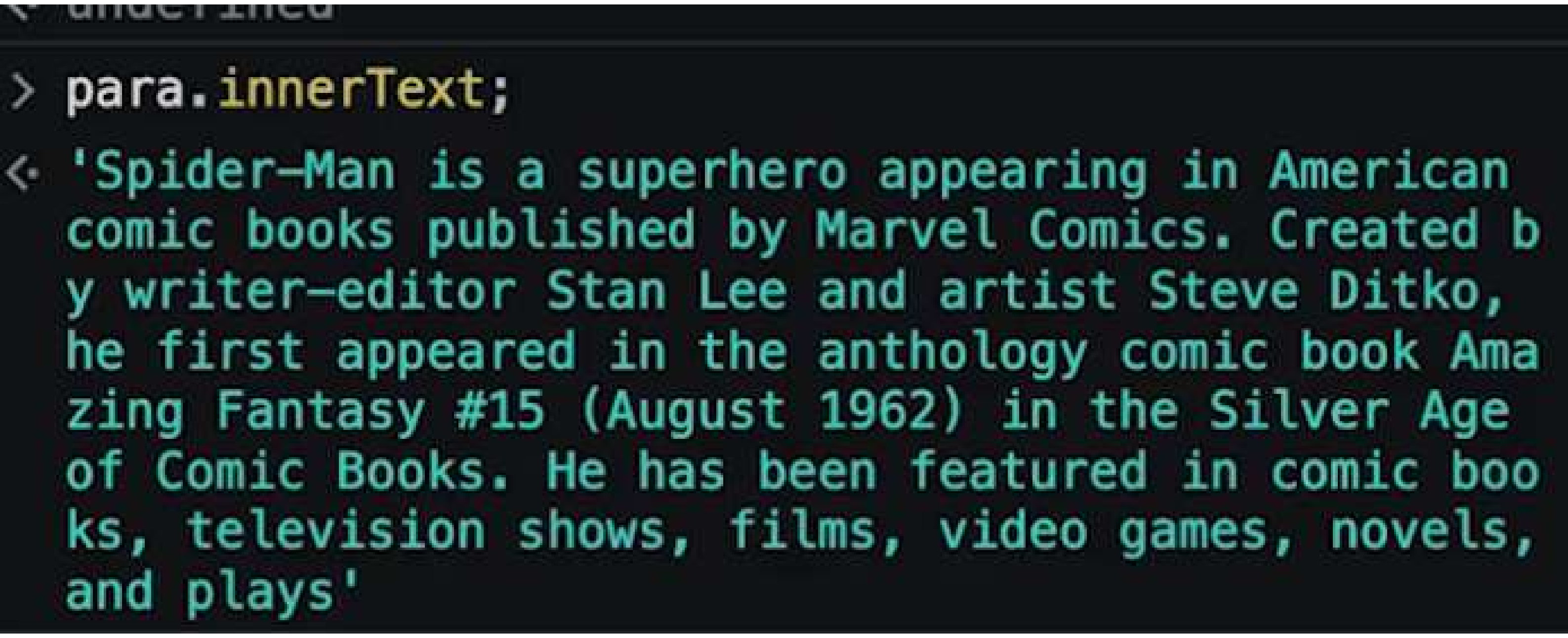
Shows all the full text

innerHTML

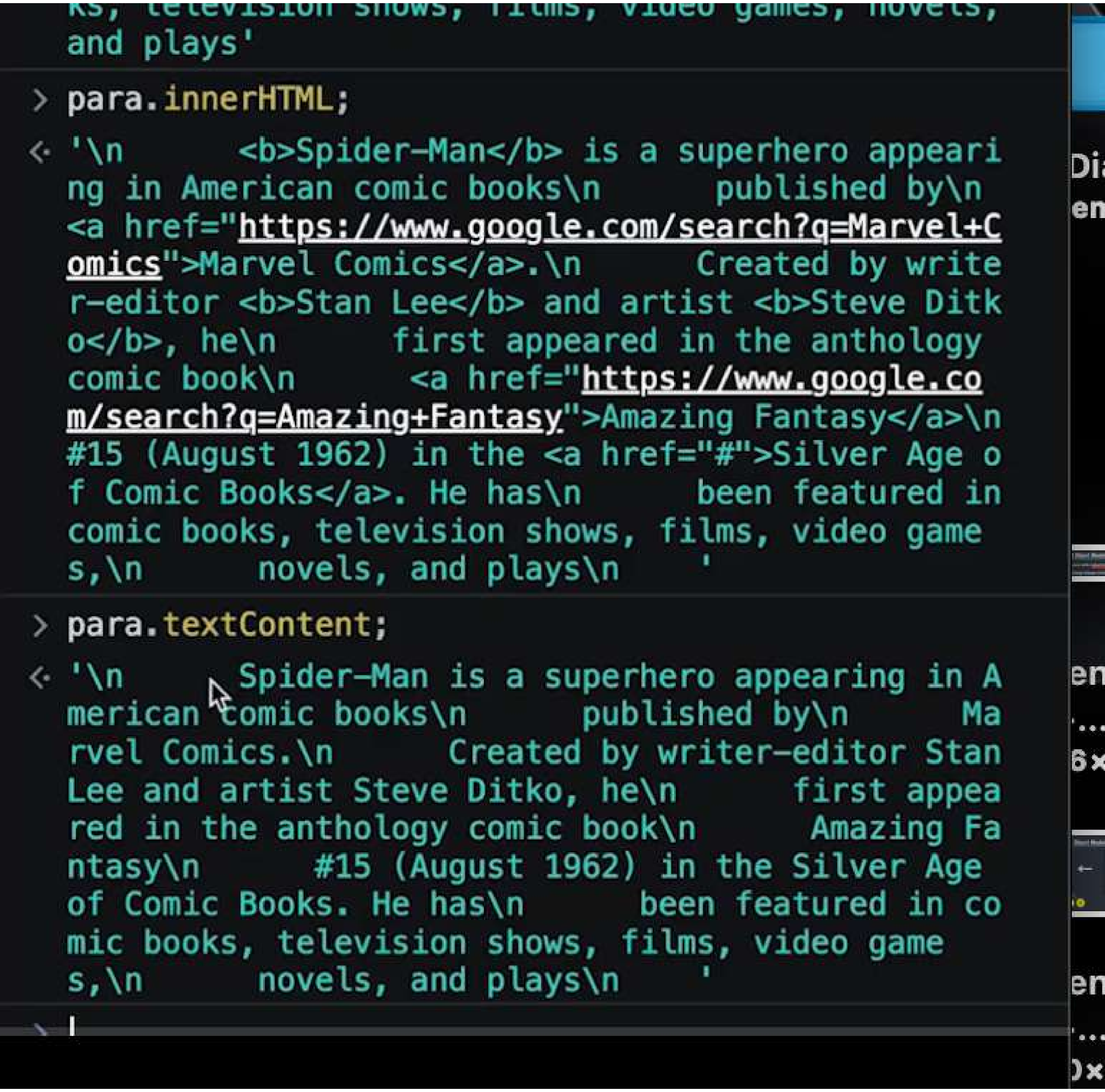
Shows the full markup



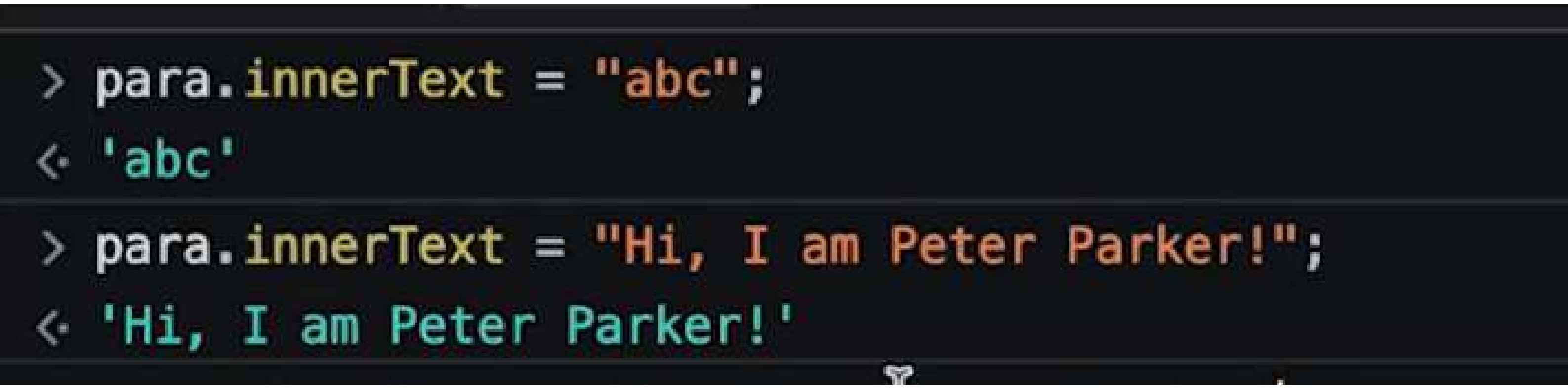
```
> let para = document.querySelector('p');
<· undefined
> pARA
✕  ▶Uncaught ReferenceError: pARA is not   VM2868:1
 defined
     at <anonymous>:1:1
> para
<·  ▶<p>⋯</p>
> console.dir(para);
    ▶ p                                     VM2983:1
<· undefined
```



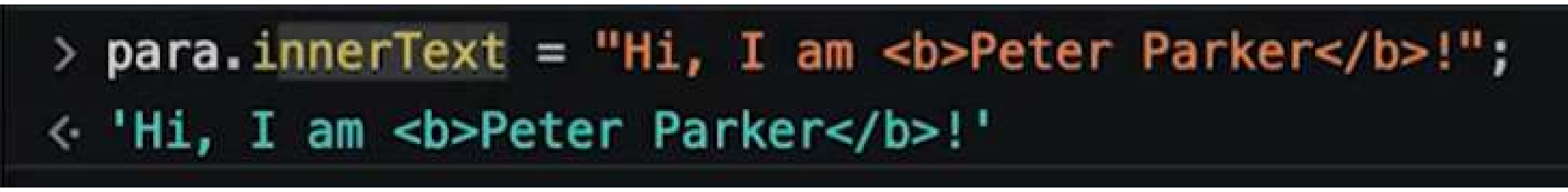
```
<· undefined
> para.innerText;
<· 'Spider—Man is a superhero appearing in American
   comic books published by Marvel Comics. Created b
   y writer—editor Stan Lee and artist Steve Ditko,
   he first appeared in the anthology comic book Ama
   zing Fantasy #15 (August 1962) in the Silver Age
   of Comic Books. He has been featured in comic boo
   ks, television shows, films, video games, novels,
   and plays'
```



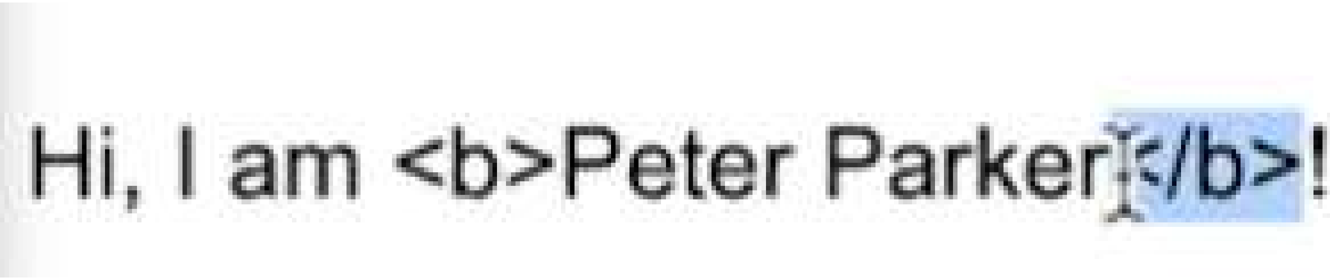
```
  ks, television shows, films, video games, novels,
  and plays'
> para.innerHTML;
<· '\n      <b>Spider-Man</b> is a superhero appeari
  ng in American comic books\n      published by\n
  <a href="https://www.google.com/search?q=Marvel+C
  omics">Marvel Comics</a>.\n      Created by write
  r-editor <b>Stan Lee</b> and artist <b>Steve Ditk
  o</b>, he\n      first appeared in the anthology
  comic book\n      <a href="https://www.google.co
  m/search?q=Amazing+Fantasy">Amazing Fantasy</a>\n
  #15 (August 1962) in the <a href="#">Silver Age o
  f Comic Books</a>. He has\n      been featured in
  comic books, television shows, films, video game
  s,\n      novels, and plays\n    '
> para.textContent;
<· '\n      Spider-Man is a superhero appearing in A
  merican comic books\n      published by\n      Ma
  rvel Comics.\n      Created by writer-editor Stan
  Lee and artist Steve Ditko, he\n      first appea
  red in the anthology comic book\n      Amazing Fa
  ntasy\n      #15 (August 1962) in the Silver Age
  of Comic Books. He has\n      been featured in co
  mic books, television shows, films, video game
  s,\n      novels, and plays\n    '
> |
```



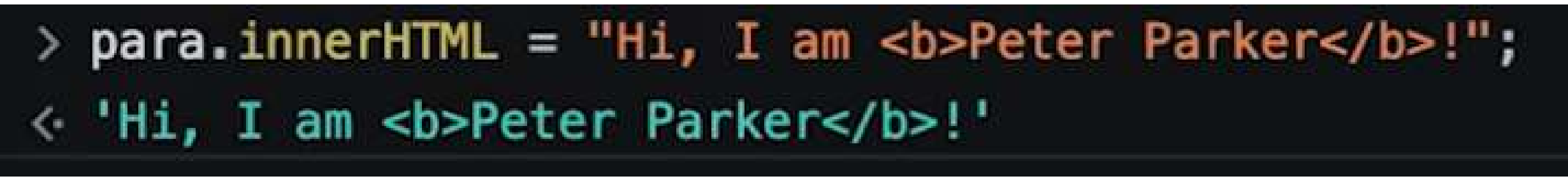
```
> para.innerText = "abc";
<· 'abc'
> para.innerText = "Hi, I am Peter Parker!";
<· 'Hi, I am Peter Parker!'
```



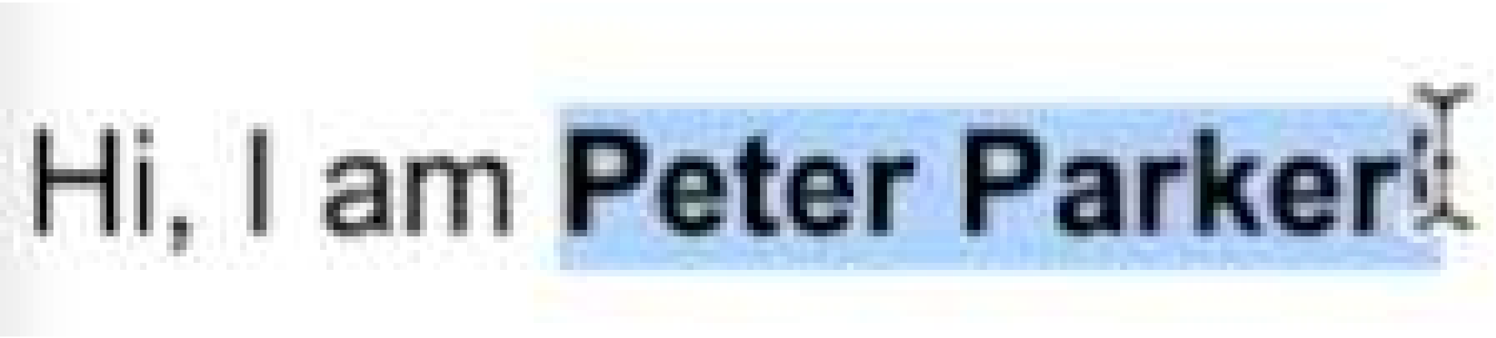
```
> para.innerText = "Hi, I am <b>Peter Parker</b>!";
< 'Hi, I am <b>Peter Parker</b>!'
```



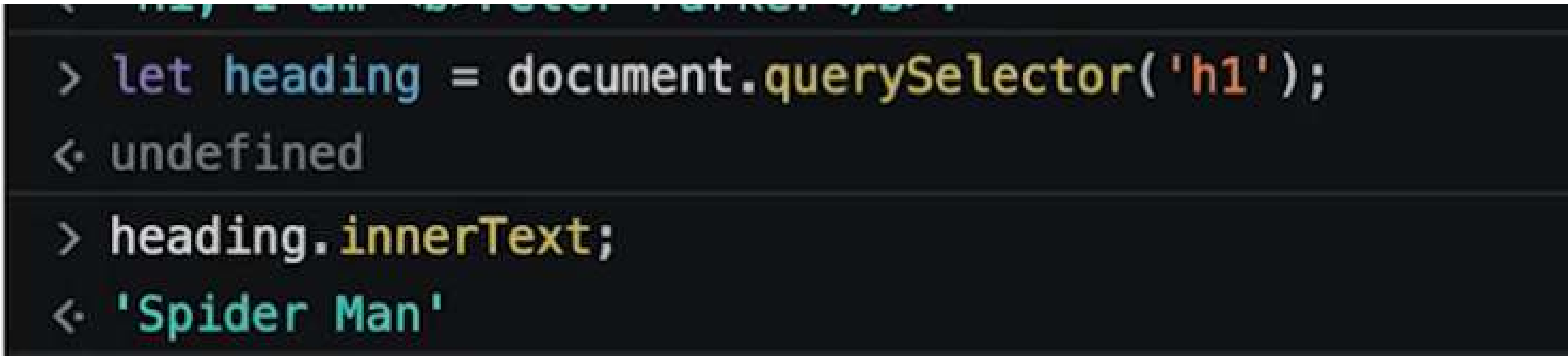
```
Hi, I am <b>Peter Parker</b>!
```



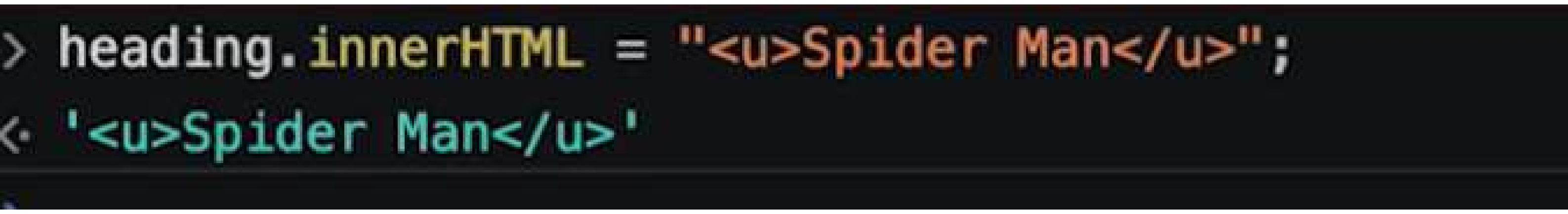
```
> para.innerHTML = "Hi, I am <b>Peter Parker</b>!";
<· 'Hi, I am <b>Peter Parker</b>!'
```



Hi, I am **Peter Parker**

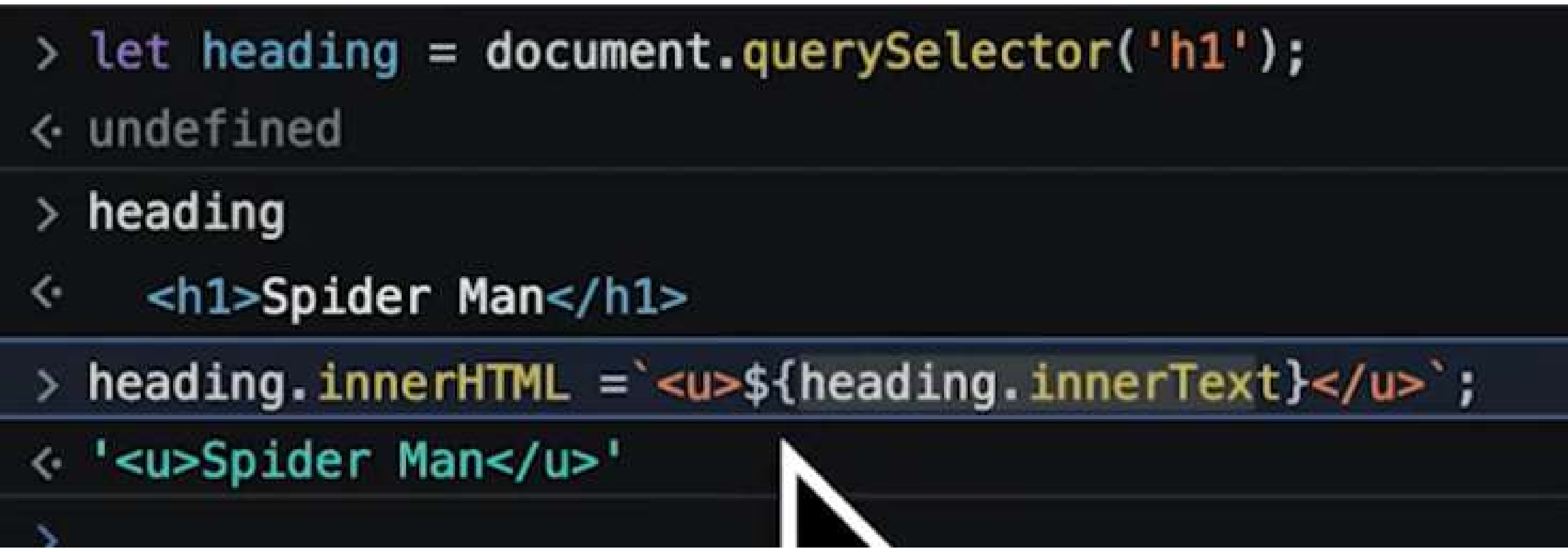


```
<· 'Hi, I am <b>Peter Parker</b>'
> let heading = document.querySelector('h1');
<· undefined
> heading.innerText;
<· 'Spider Man'
```

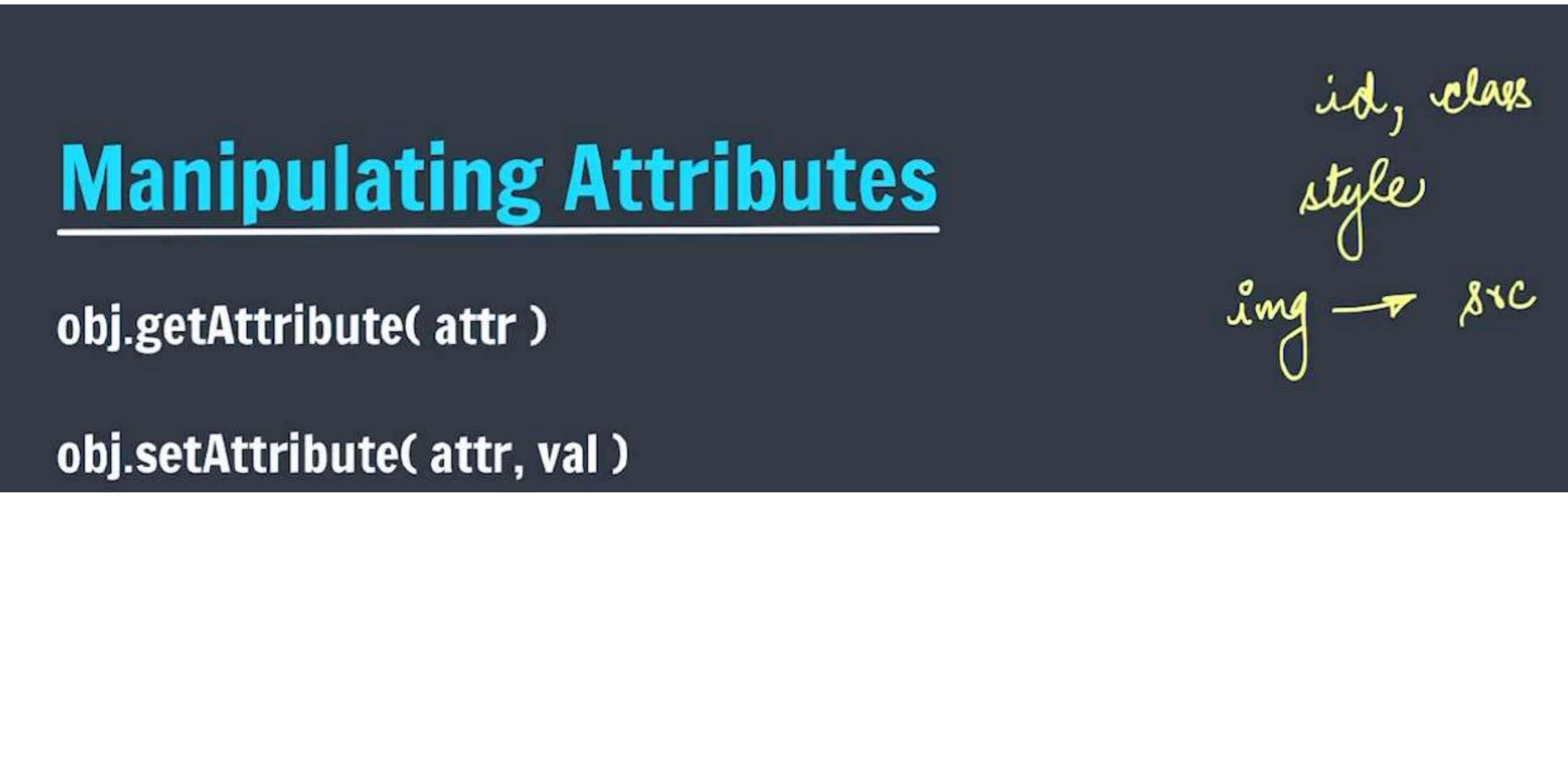


```
> heading.innerHTML = "<u>Spider Man</u>";
<· '<u>Spider Man</u>'
```

Spider Man



```
> let heading = document.querySelector('h1');
<· undefined
> heading
<·   <h1>Spider Man</h1>
> heading.innerHTML =`<u>${heading.innerText}</u>`;
<· '<u>Spider Man</u>'
>
```



Manipulating Attributes

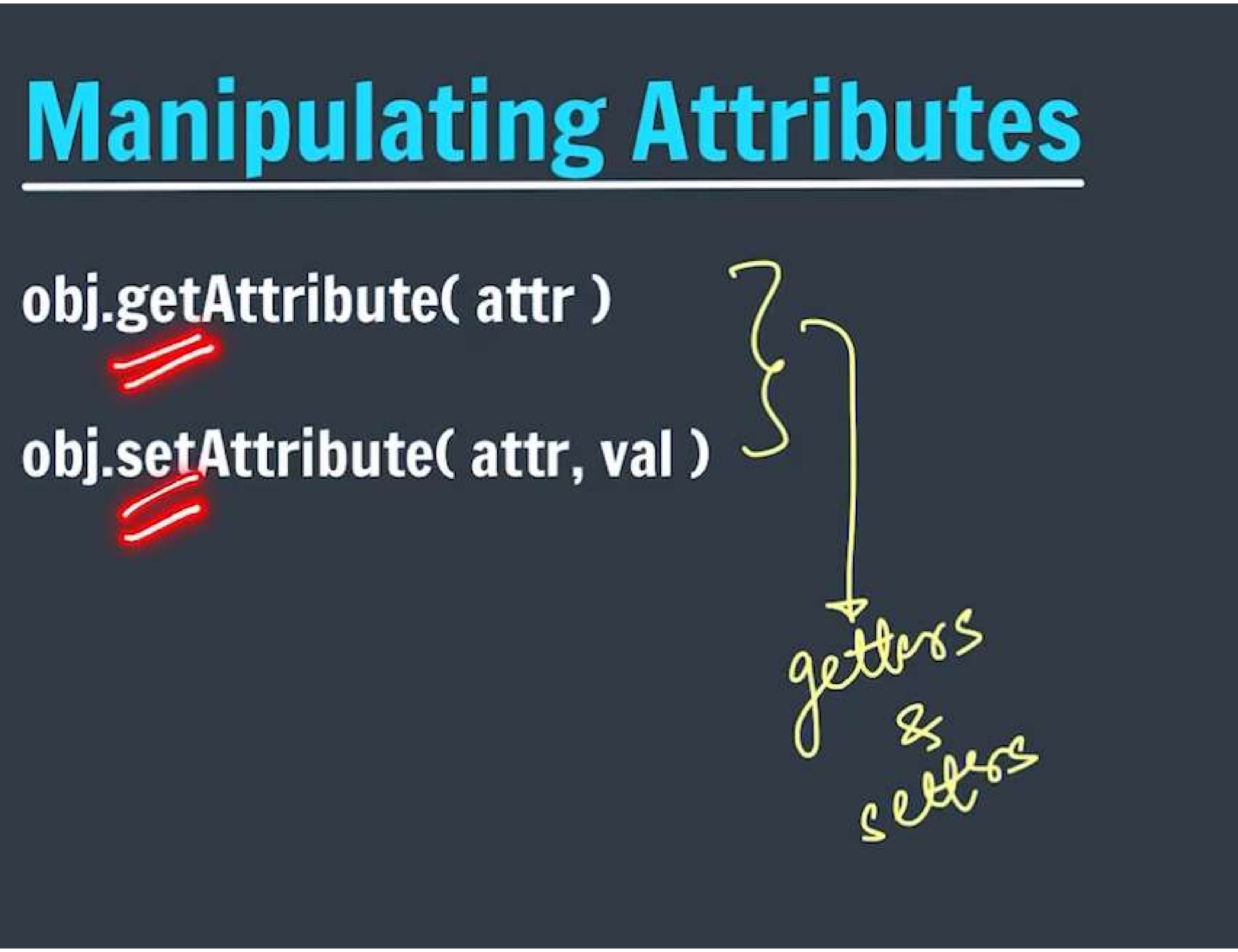
obj.getAttribute(attr)

obj.setAttribute(attr, val)

id, class

style

img → src

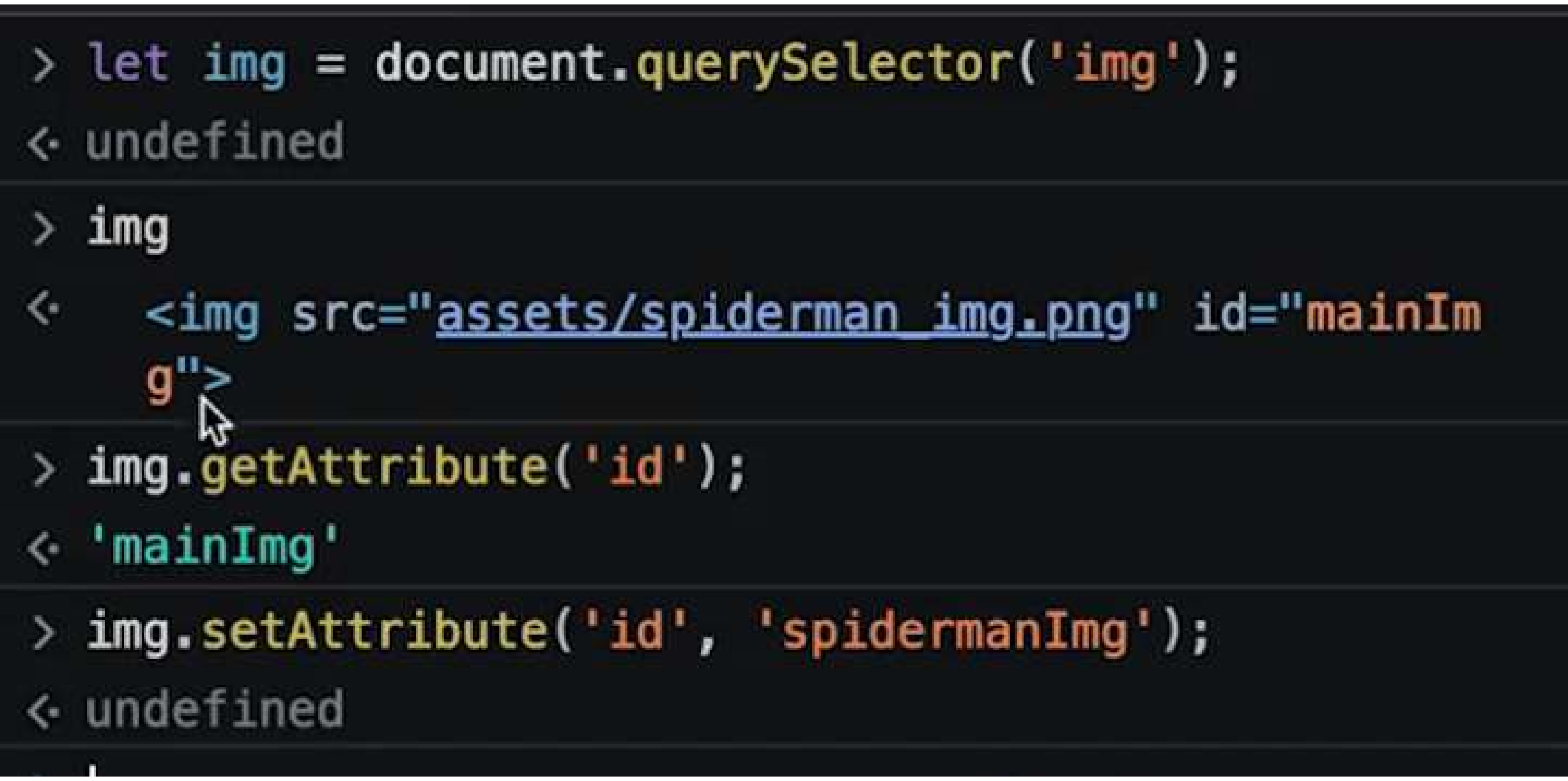


Manipulating Attributes

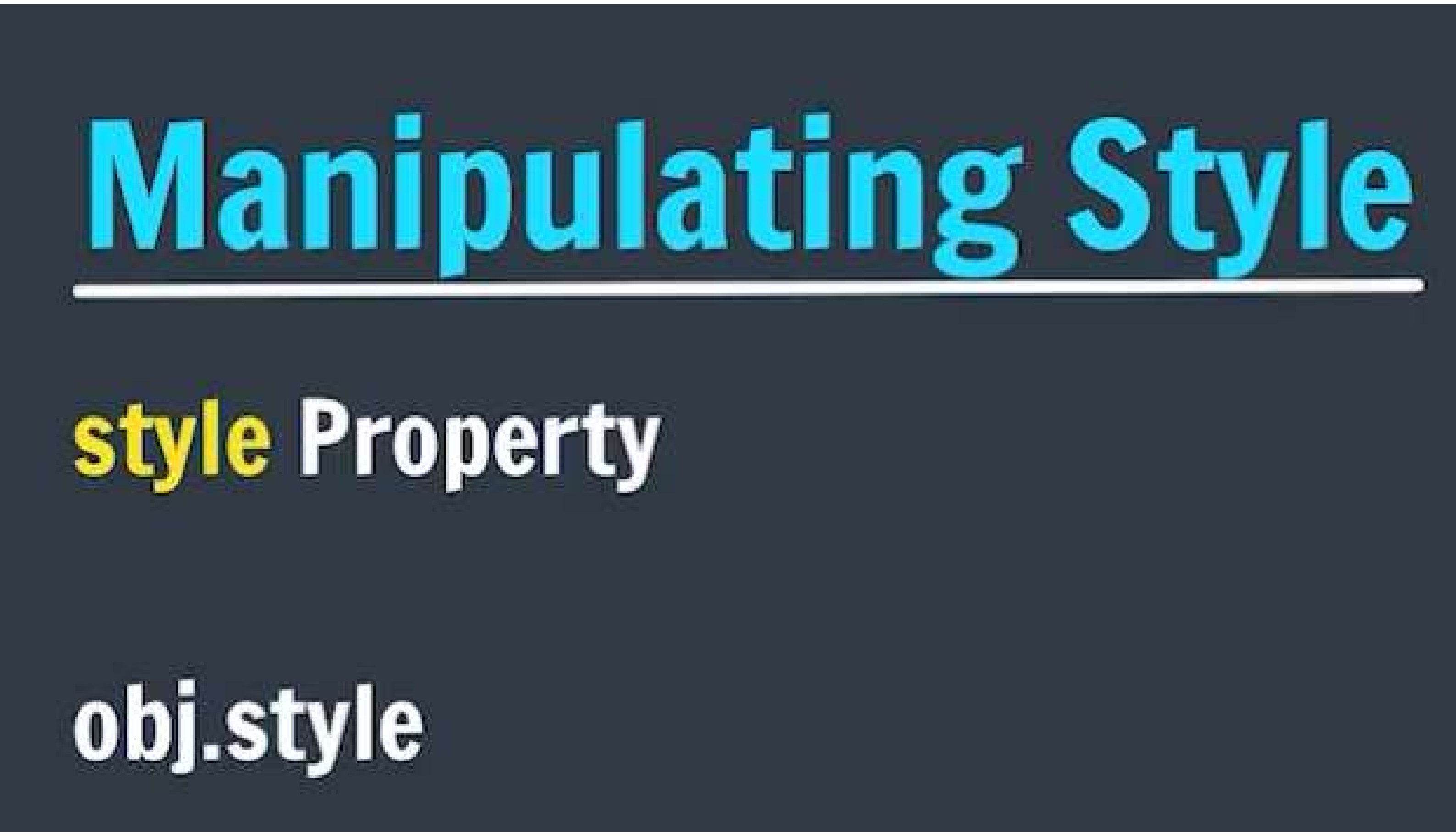
obj.getAttribute(attr)

obj.setAttribute(attr, val)

getters
&
setters



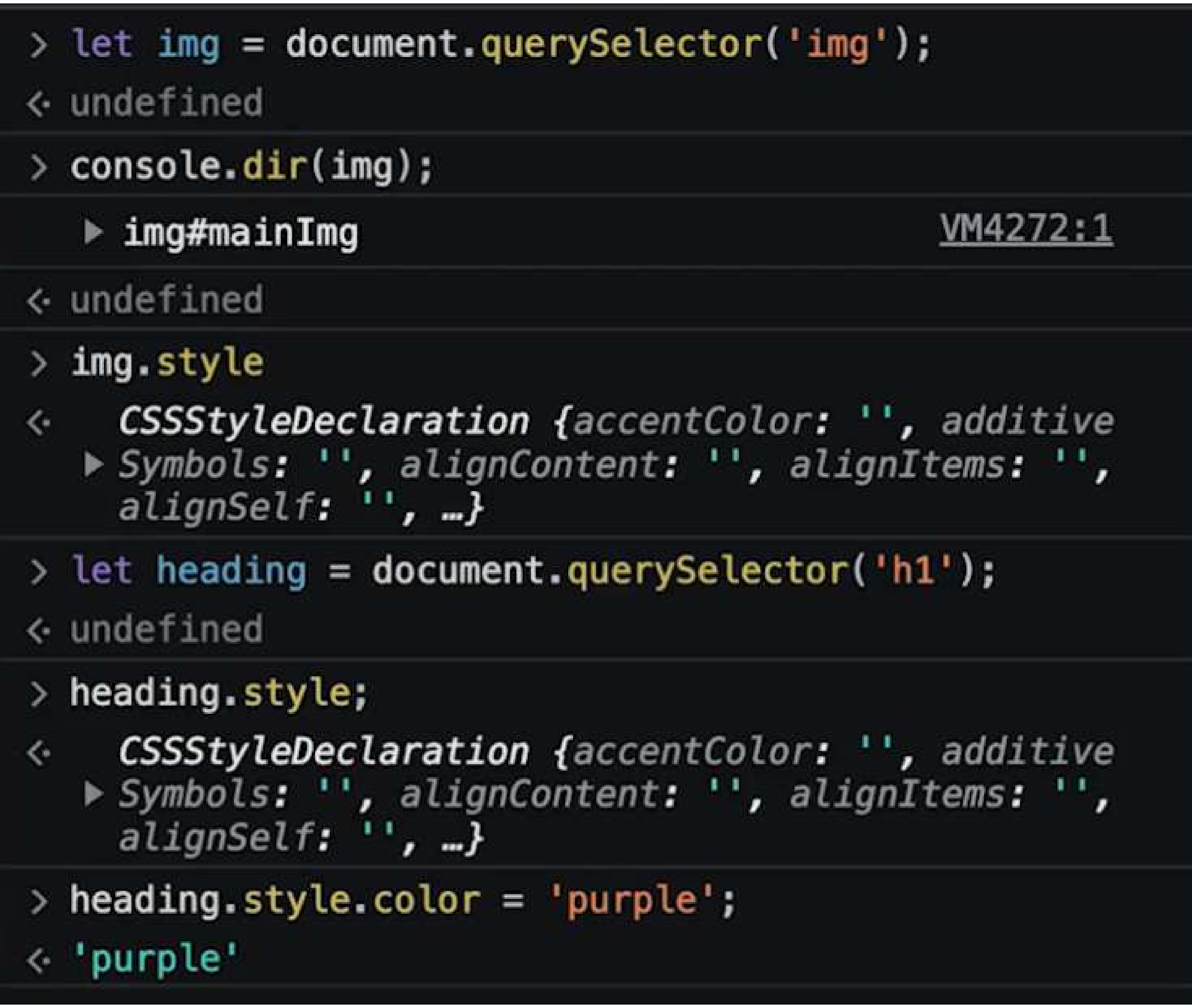
```
> let img = document.querySelector('img');
<· undefined
> img
<·   <img src="assets/spiderman_img.png" id="mainIm
     g">
> img.getAttribute('id');
<· 'mainImg'
> img.setAttribute('id', 'spidermanImg');
<· undefined
```



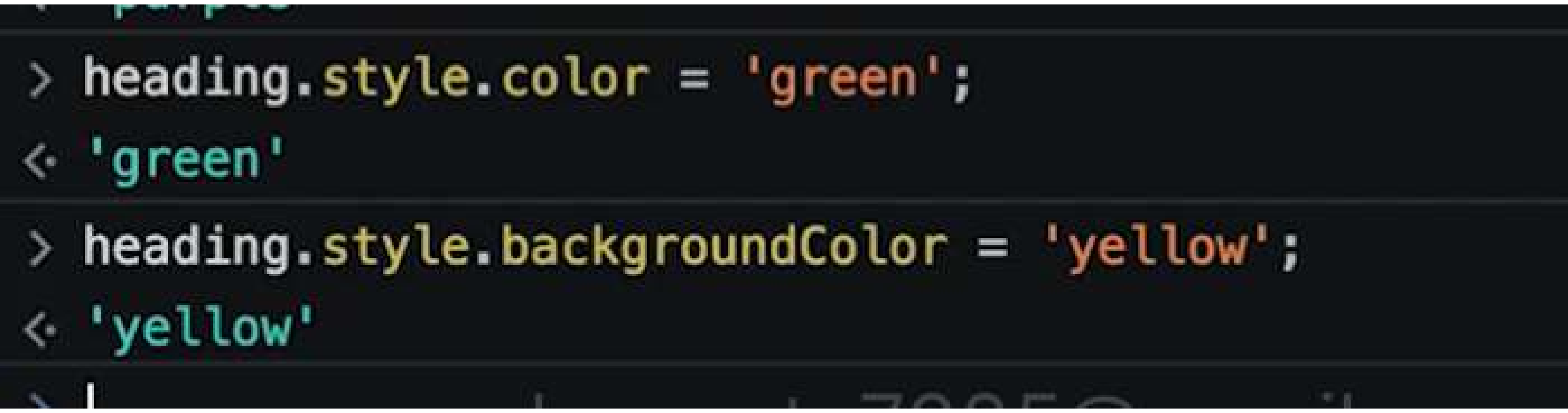
Manipulating Style

style Property

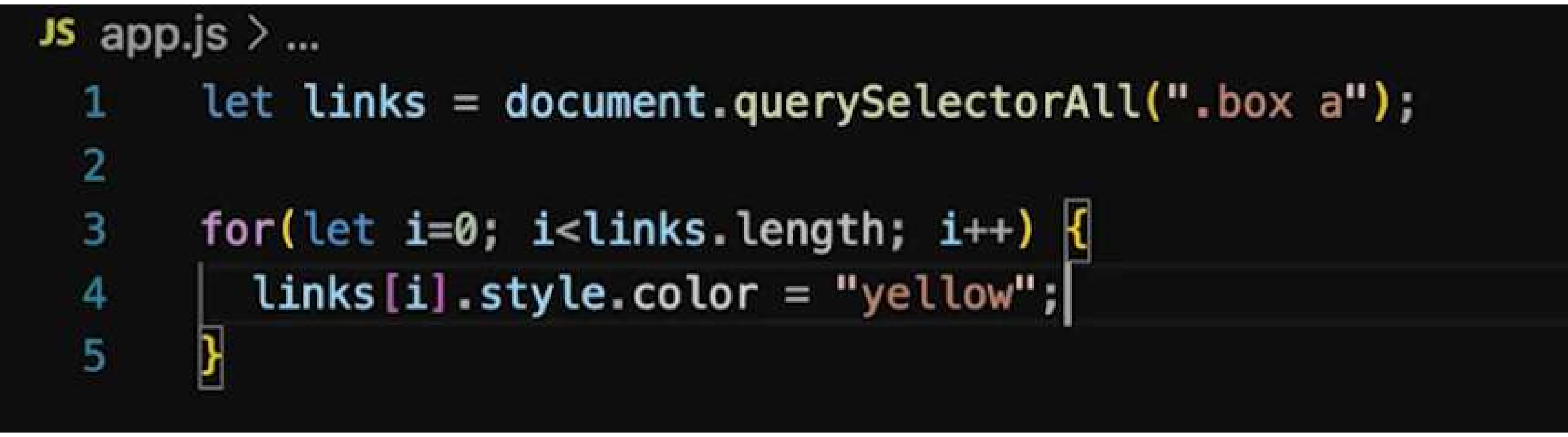
obj.style



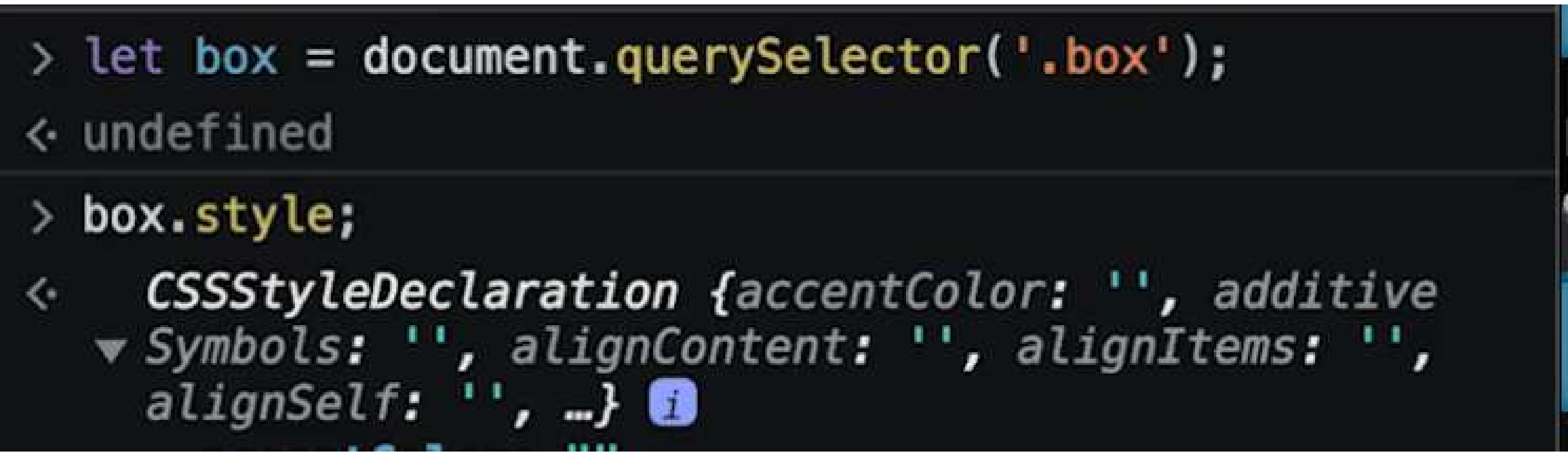
```
> let img = document.querySelector('img');
<· undefined
> console.dir(img);
    ▶ img#mainImg                                       VM4272:1
<· undefined
> img.style
<·    CSSStyleDeclaration {accentColor: '', additive
    ▶ Symbols: '', alignContent: '', alignItems: '',
      alignSelf: '', …}
> let heading = document.querySelector('h1');
<· undefined
> heading.style;
<·    CSSStyleDeclaration {accentColor: '', additive
    ▶ Symbols: '', alignContent: '', alignItems: '',
      alignSelf: '', …}
> heading.style.color = 'purple';
<· 'purple'
```



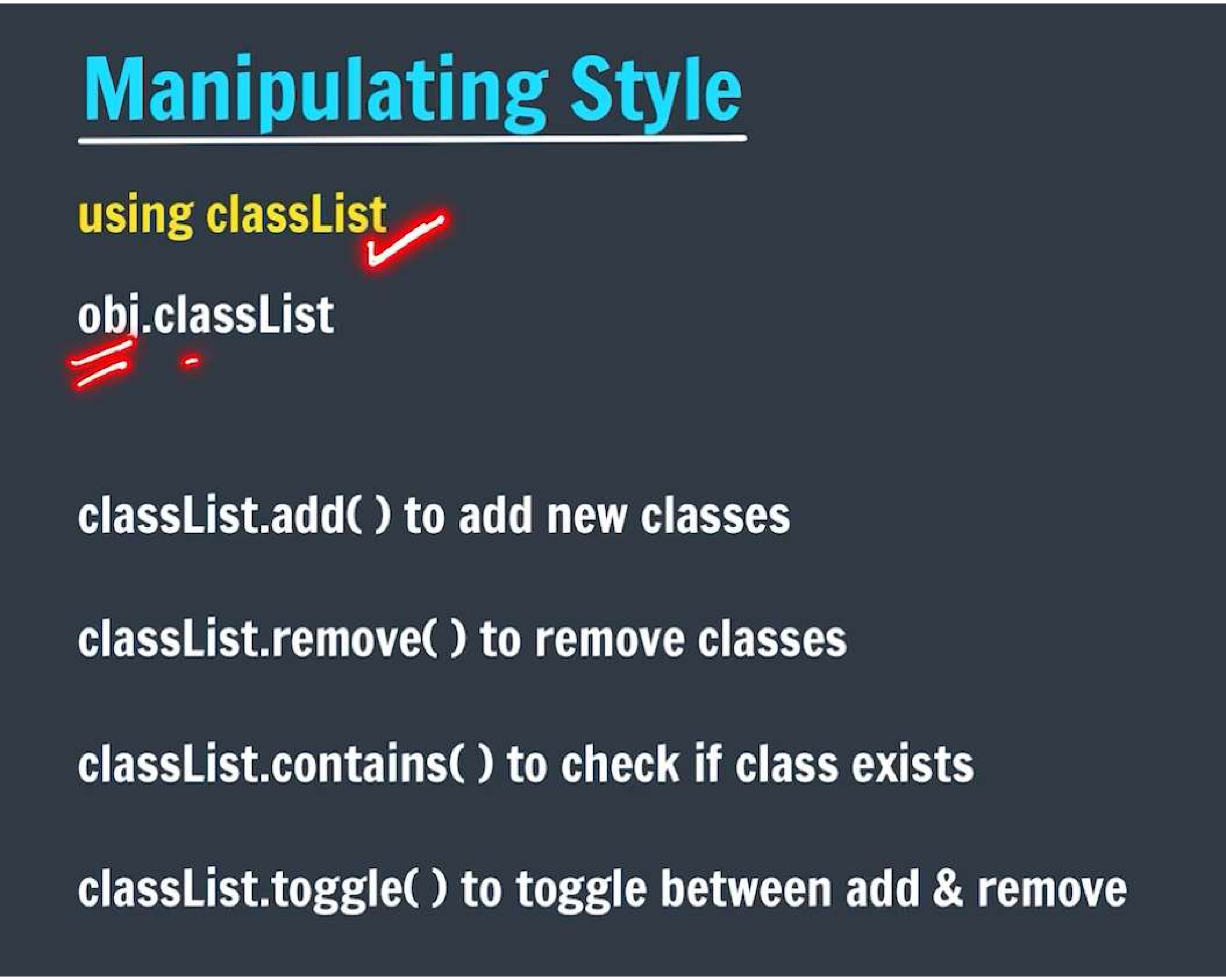
```
> heading.style.color = 'green';
<· 'green'
> heading.style.backgroundColor = 'yellow';
<· 'yellow'
>
```



```
let links = document.querySelectorAll(".box a");

for(let i=0; i<links.length; i++) {
  links[i].style.color = "yellow";
}
```



```
> let box = document.querySelector('.box');
<· undefined
> box.style;
<·   CSSStyleDeclaration {accentColor: '', additive
   ▼ Symbols: '', alignContent: '', alignItems: '',
     alignSelf: '', …} i
```



Manipulating Style

using classList

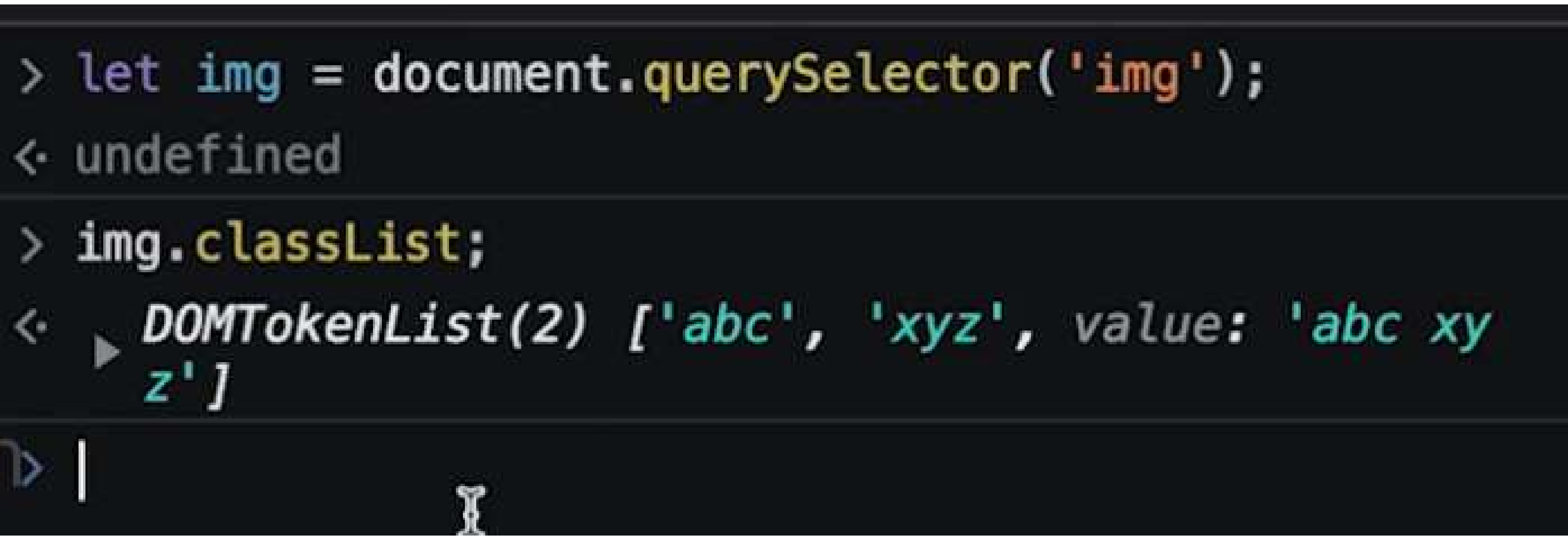
obj.classList

classList.add( ) to add new classes

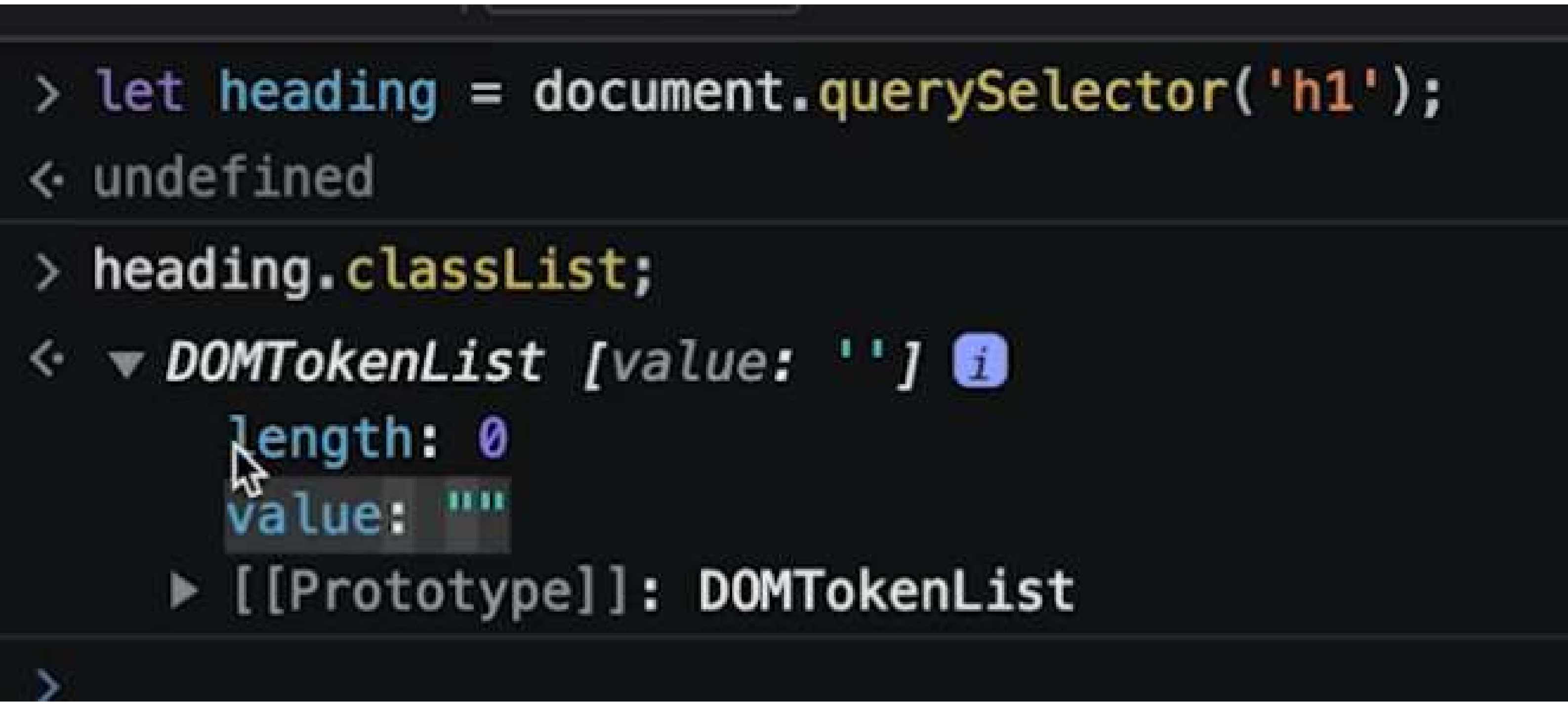
classList.remove( ) to remove classes

classList.contains( ) to check if class exists

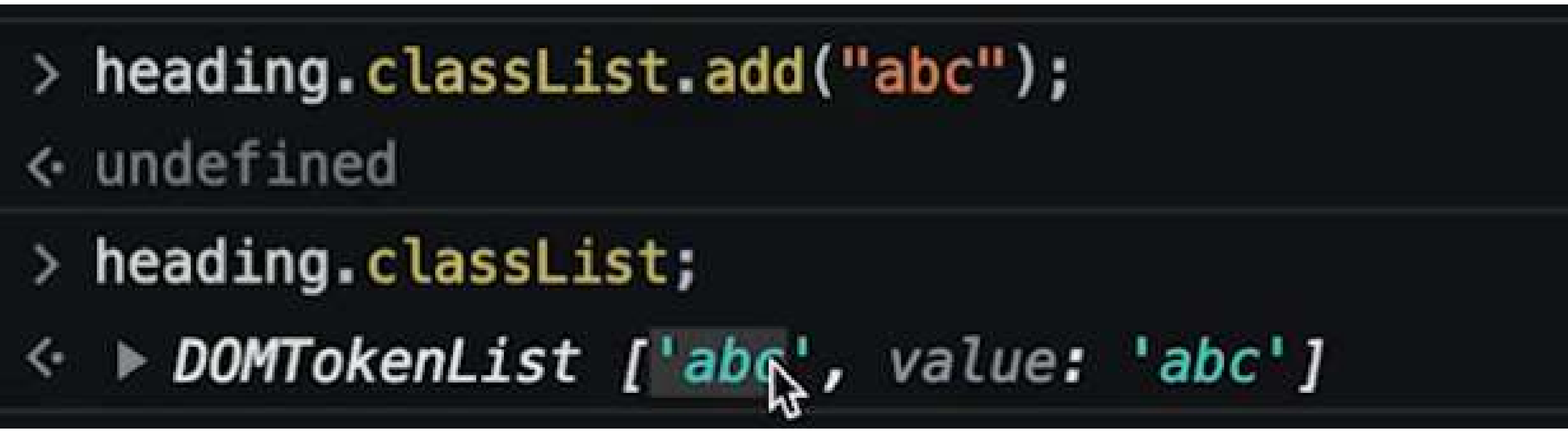
classList.toggle( ) to toggle between add & remove



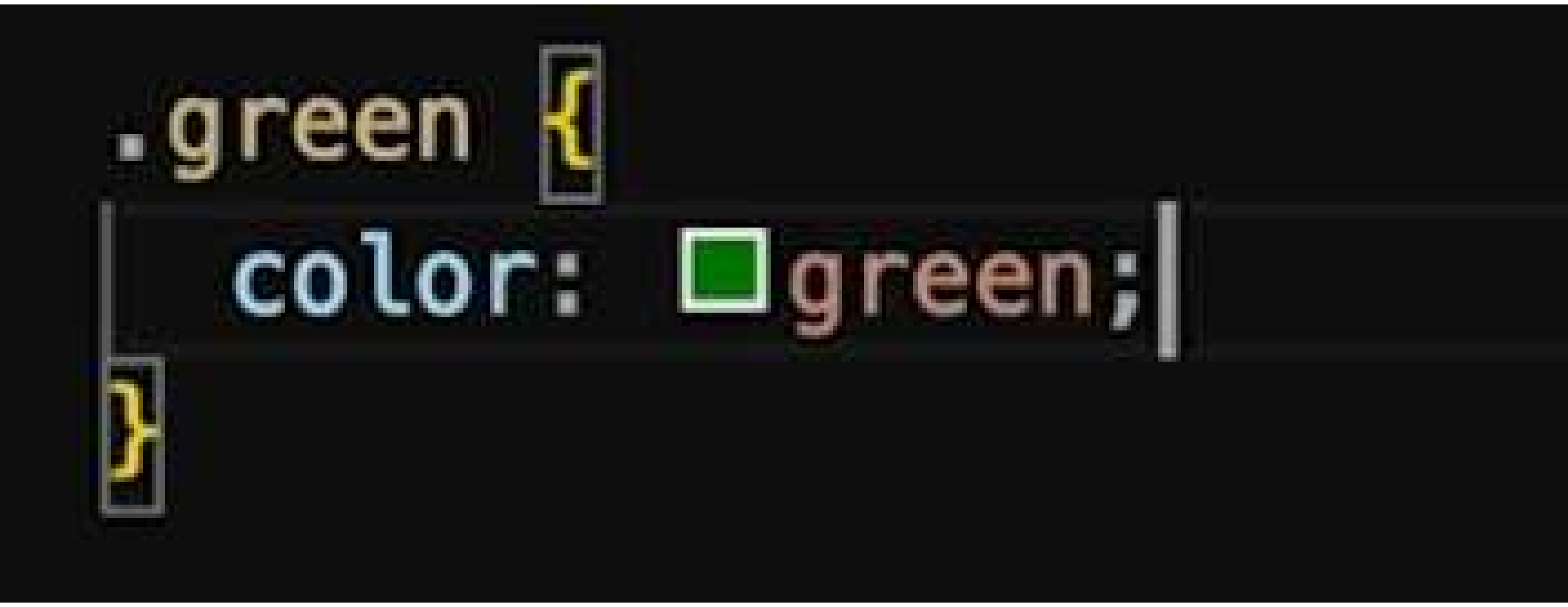
```
> let img = document.querySelector('img');
<· undefined
> img.classList;
<· ▶ DOMTokenList(2) ['abc', 'xyz', value: 'abc xy
     z']
> |
```



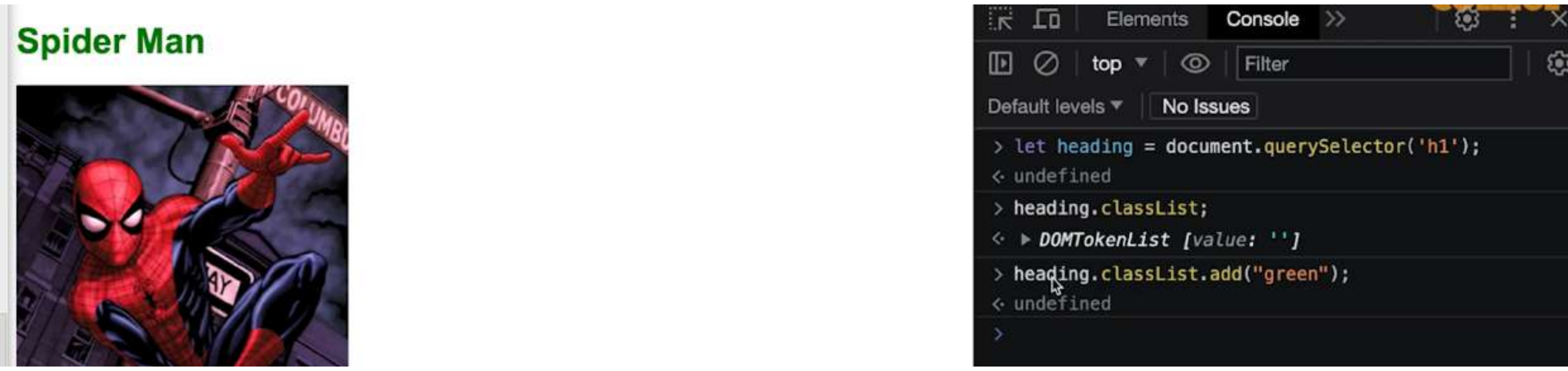
```
> let heading = document.querySelector('h1');
<· undefined
> heading.classList;
<· ▼ DOMTokenList [value: ''] i
        length: 0
        value: ""
      ▶ [[Prototype]]: DOMTokenList
>
```



```
> heading.classList.add("abc");
<· undefined
> heading.classList;
<· ▶ DOMTokenList ['abc', value: 'abc']
```

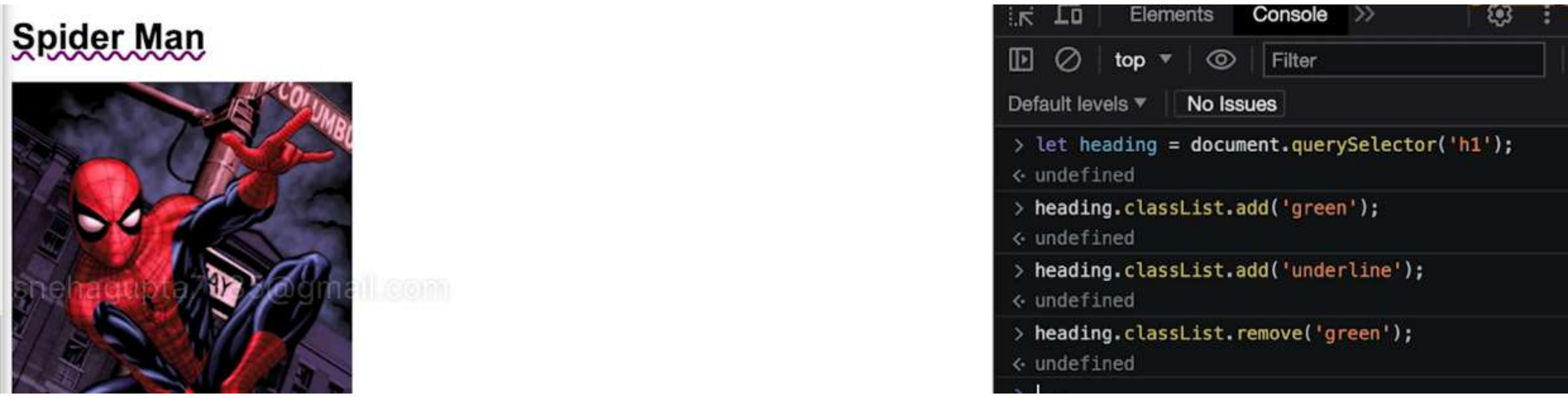


```
.green {
  color: green;
}
```



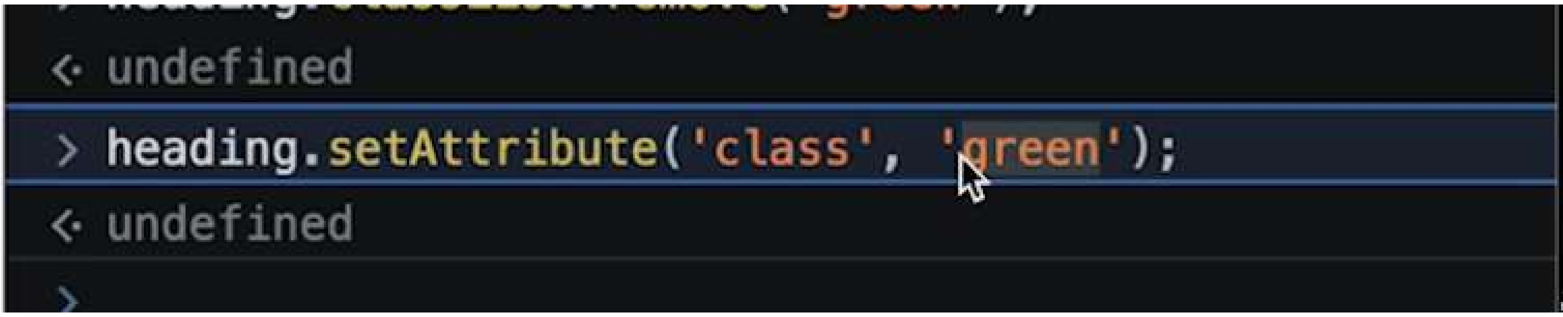
Spider Man

```
> let heading = document.querySelector('h1');
<· undefined
> heading.classList;
<· ▶ DOMTokenList [value: '']
> heading.classList.add("green");
<· undefined
>
```

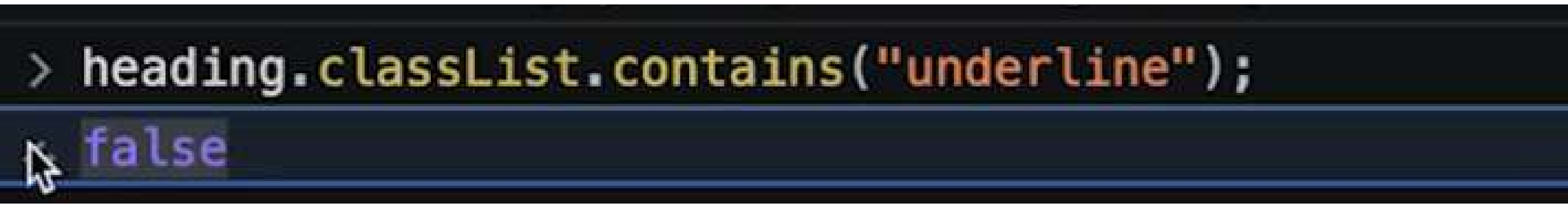


Spider Man

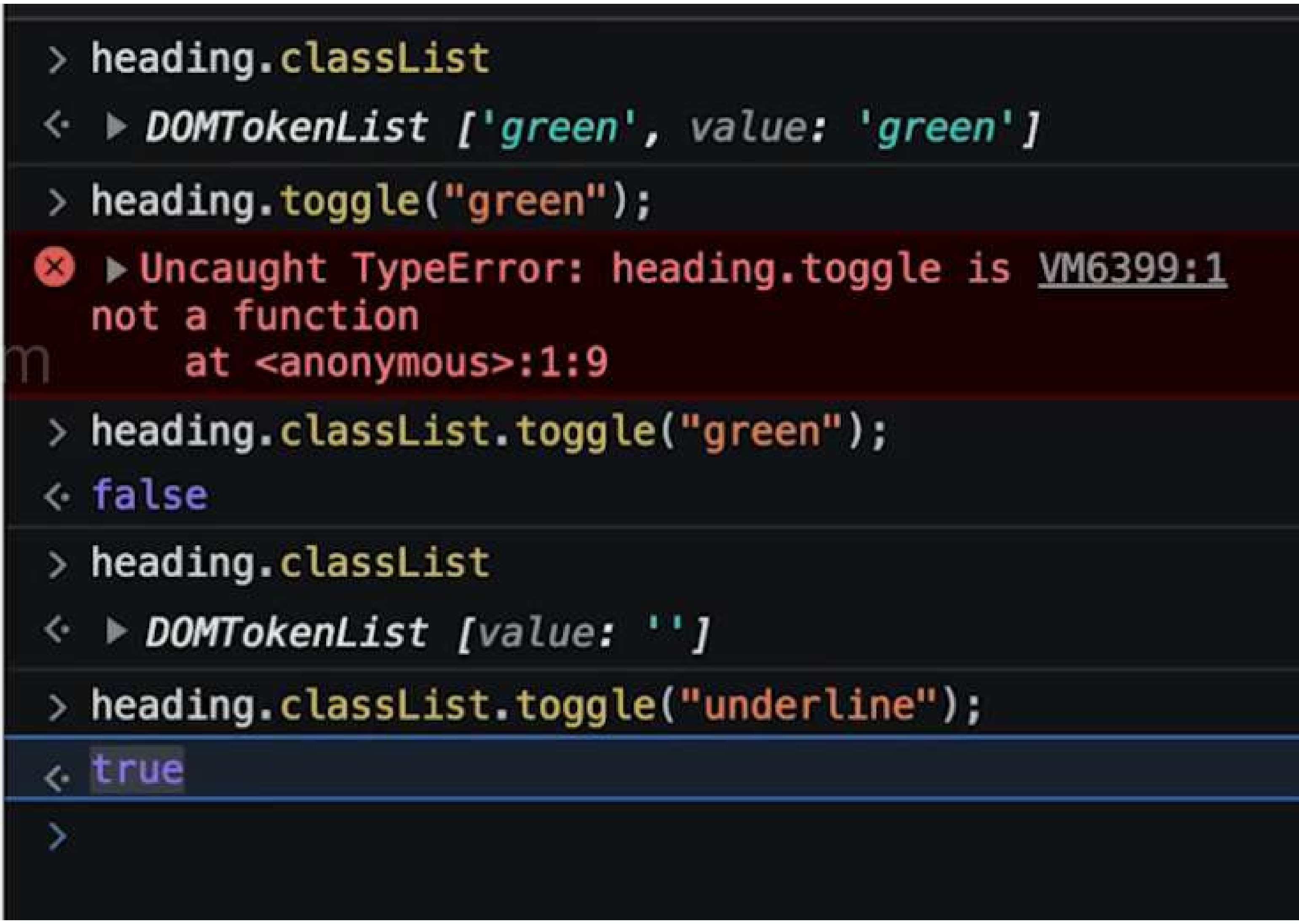
```
> let heading = document.querySelector('h1');
<· undefined
> heading.classList.add('green');
<· undefined
> heading.classList.add('underline');
<· undefined
> heading.classList.remove('green');
<· undefined
```



```
> heading.classList.remove('green');
<· undefined
> heading.setAttribute('class', 'green');
<· undefined
>
```

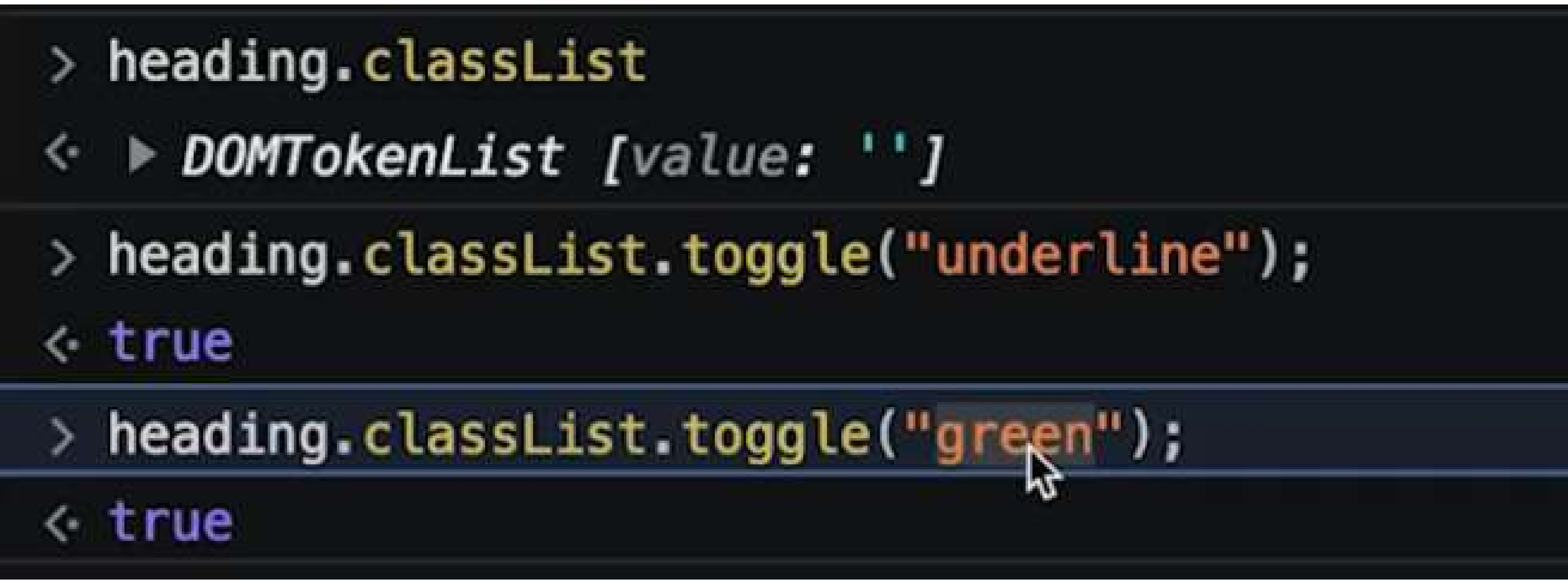


```
> heading.classList.contains("underline");
false
```

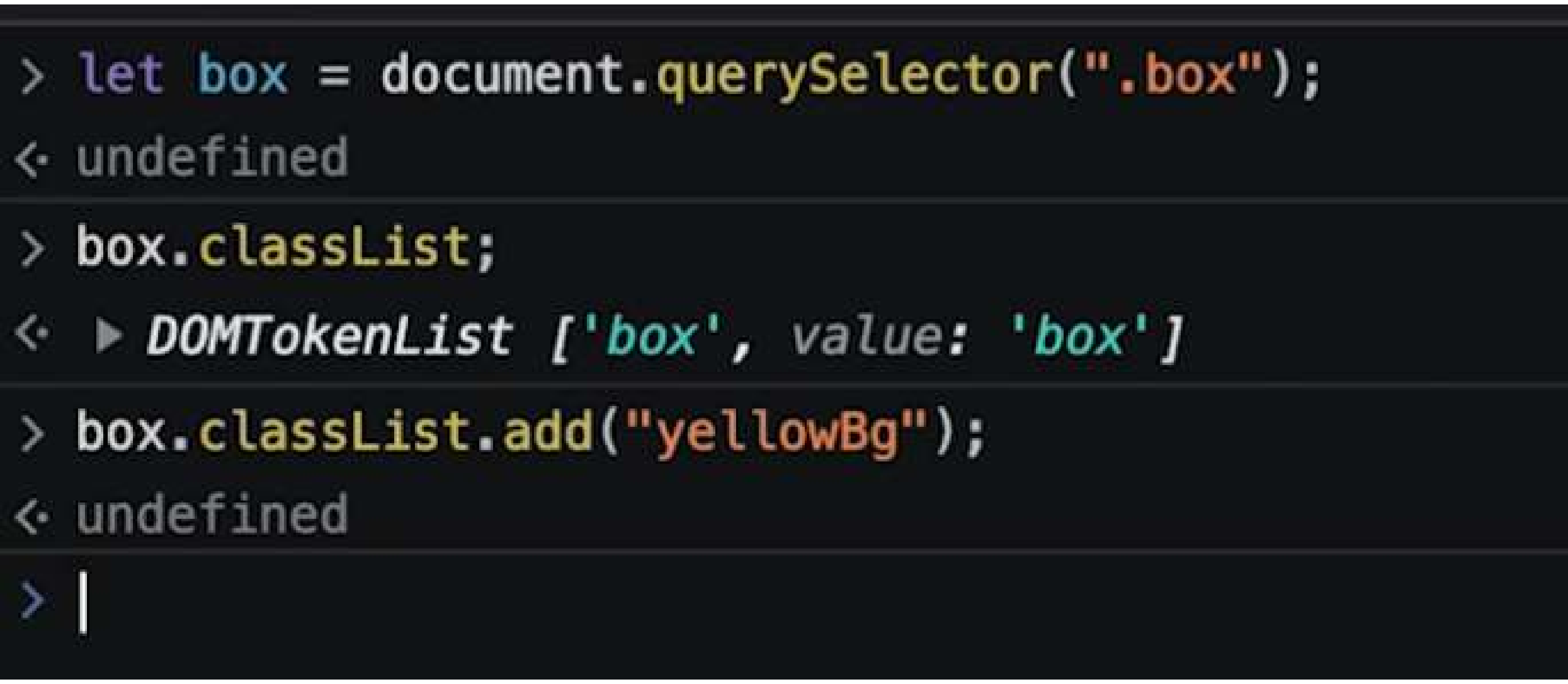


```
> heading.classList
<· ▶ DOMTokenList ['green', value: 'green']
> heading.toggle("green");
✕ ▶ Uncaught TypeError: heading.toggle is VM6399:1
  not a function
      at <anonymous>:1:9
> heading.classList.toggle("green");
<· false
> heading.classList
<· ▶ DOMTokenList [value: '']
> heading.classList.toggle("underline");
<· true
>
```

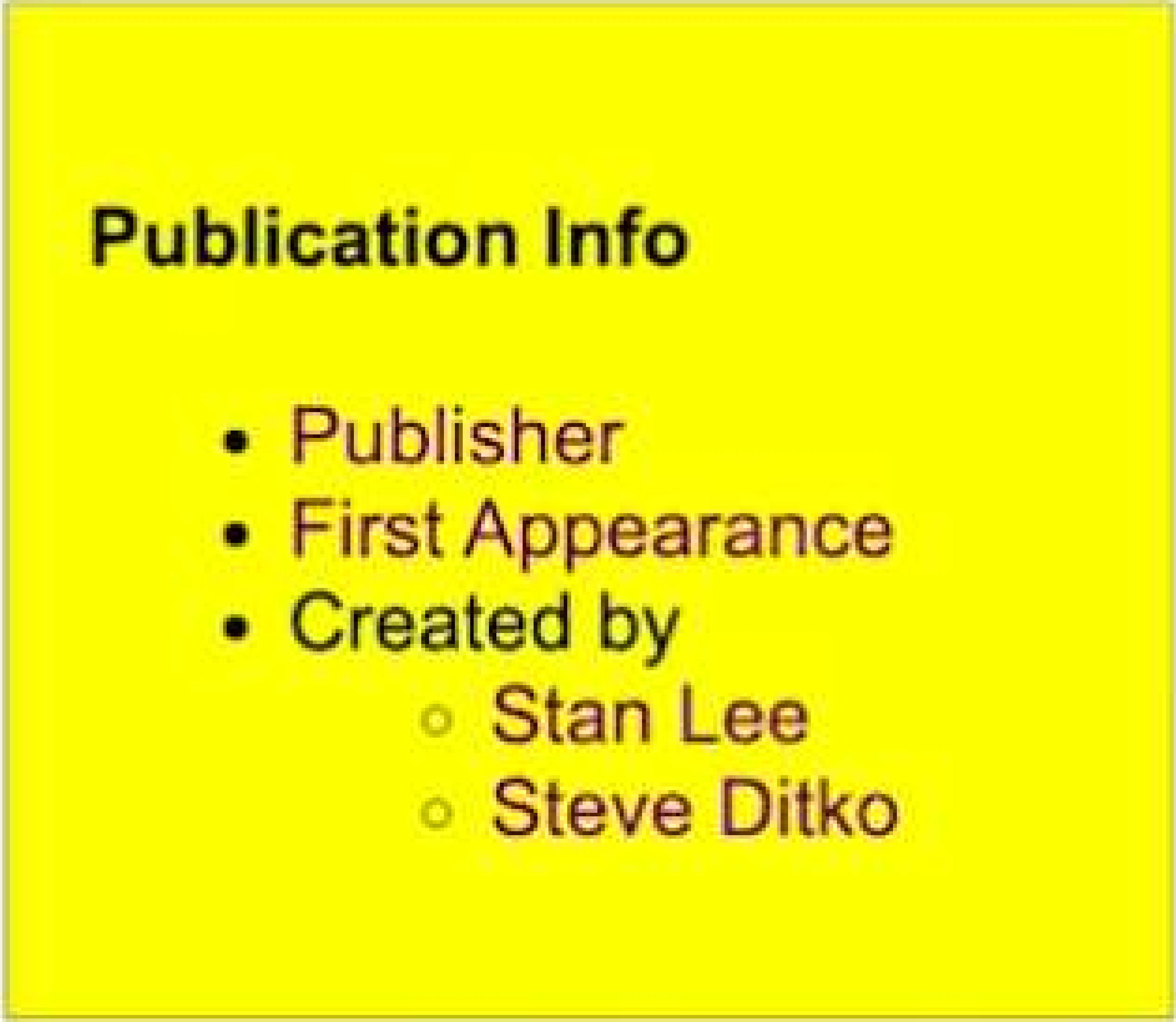
Spider Man



```
> heading.classList
<· ▶ DOMTokenList [value: '']
> heading.classList.toggle("underline");
<· true
> heading.classList.toggle("green");
<· true
```

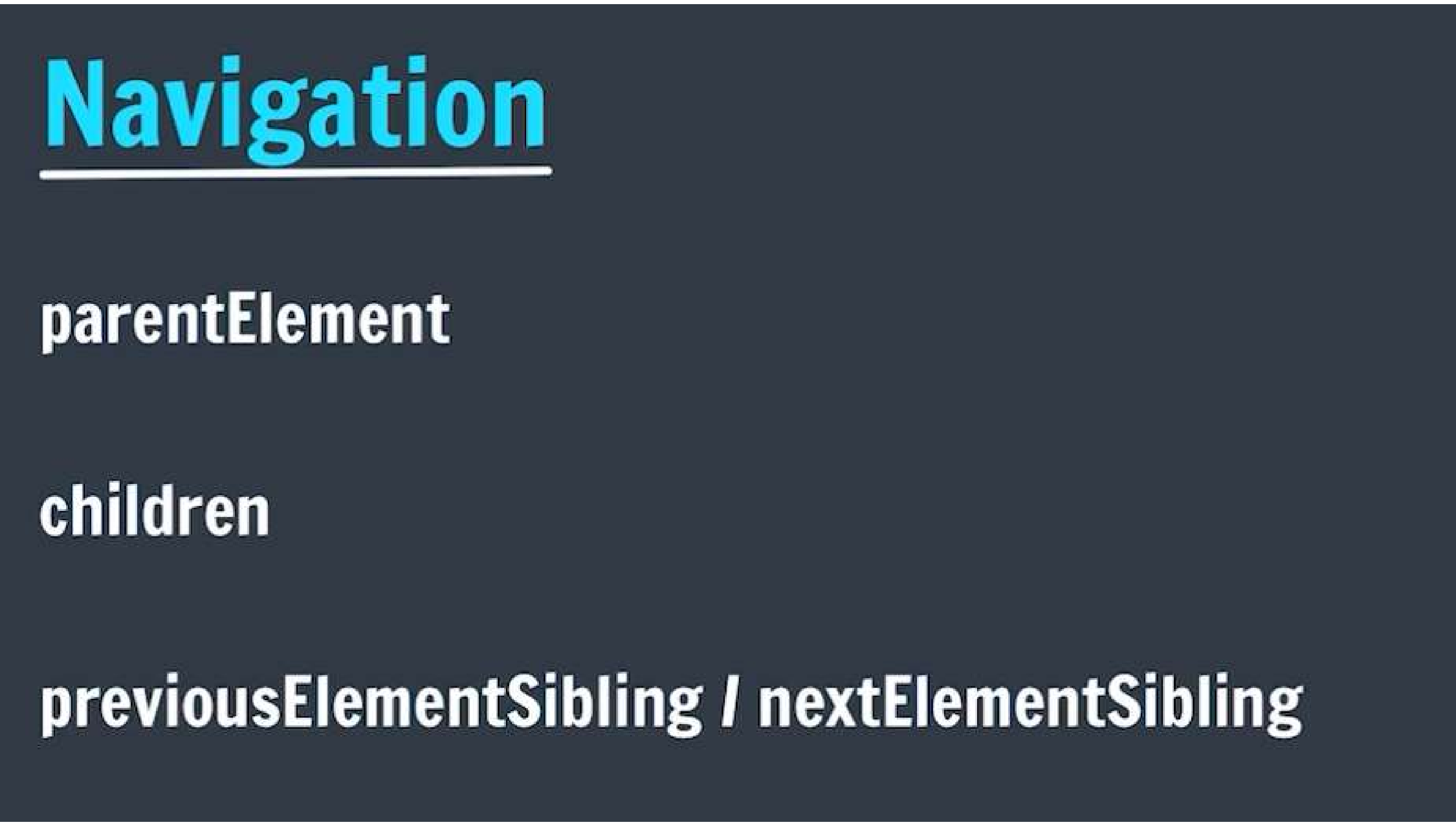


```
> let box = document.querySelector(".box");
<· undefined
> box.classList;
<·  ▶ DOMTokenList ['box', value: 'box']
> box.classList.add("yellowBg");
<· undefined
> |
```



Publication Info

- Publisher
- First Appearance
- Created by
  - Stan Lee
  - Steve Ditko

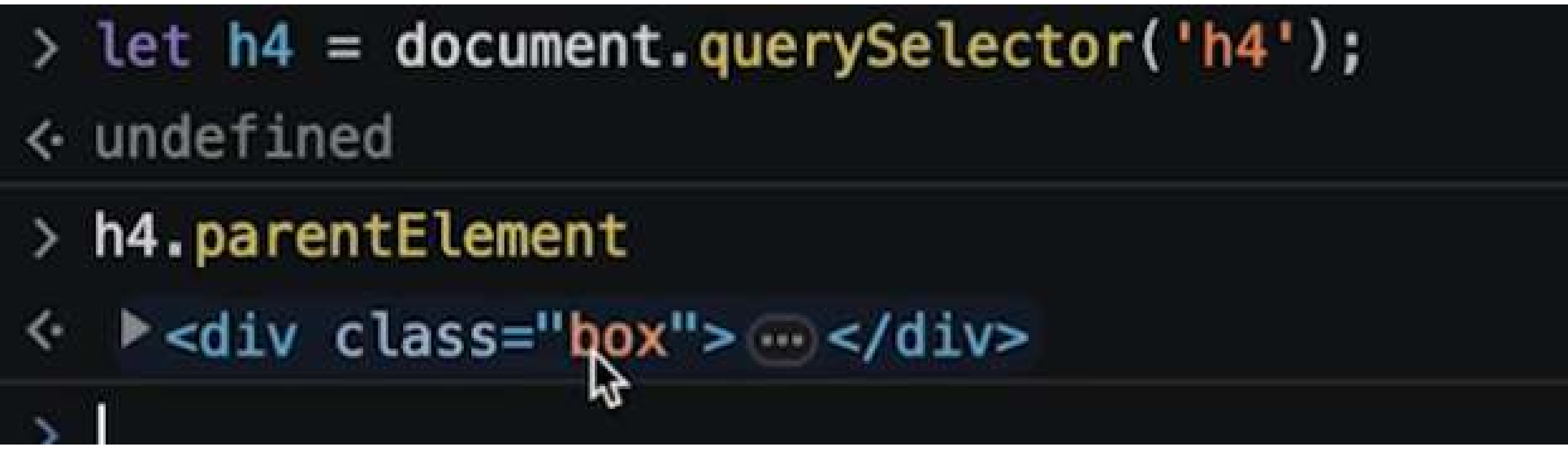


Navigation

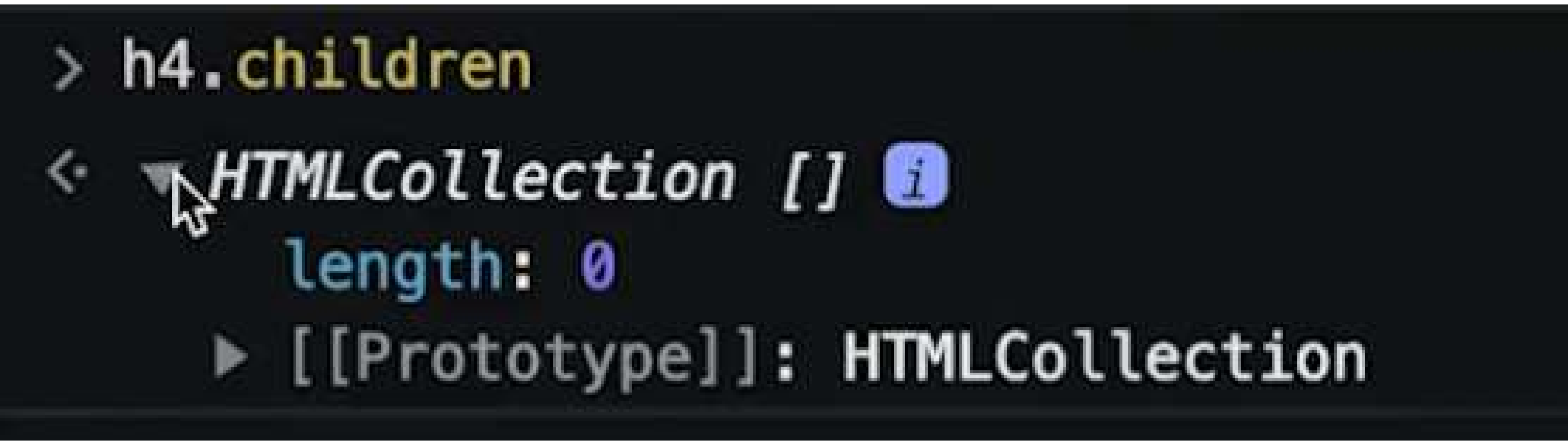
parentElement

children

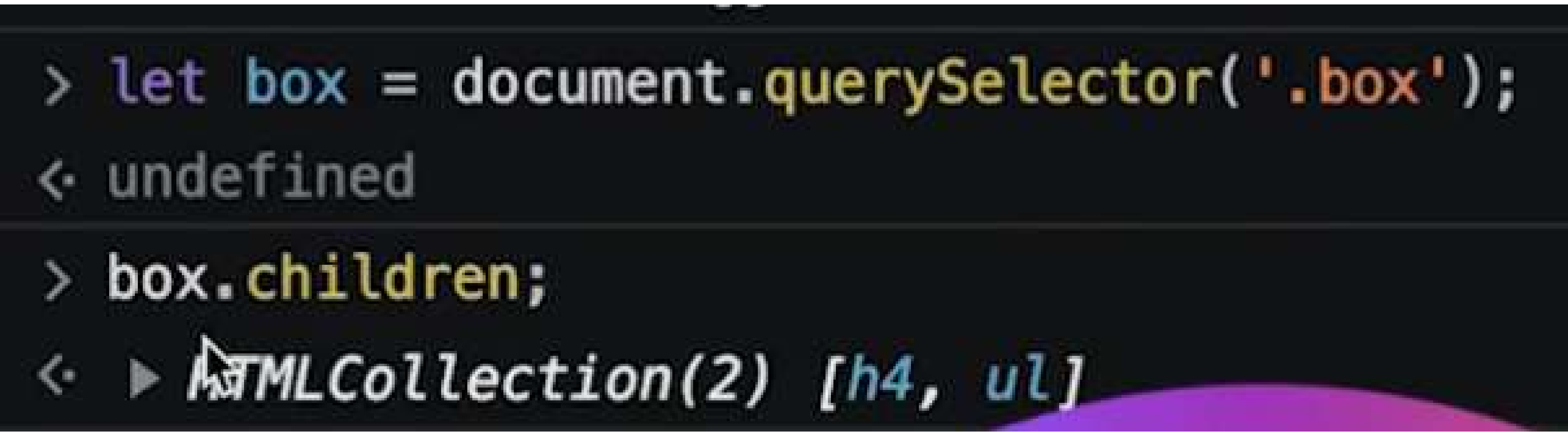
previousElementSibling / nextElementSibling



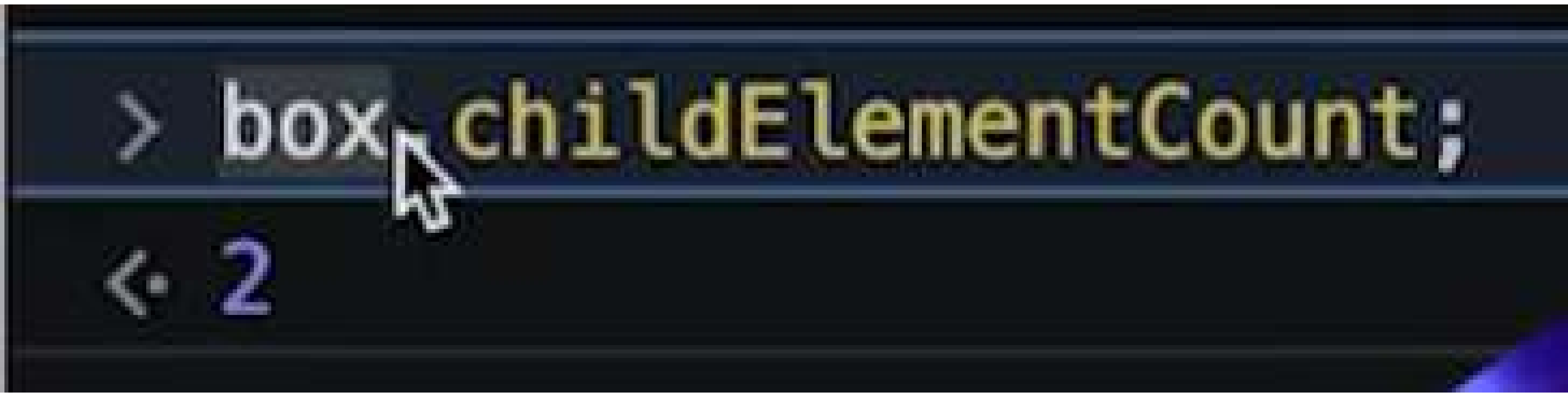
```
> let h4 = document.querySelector('h4');
<· undefined
> h4.parentElement
<·  ▶<div class="box">…</div>
>
```



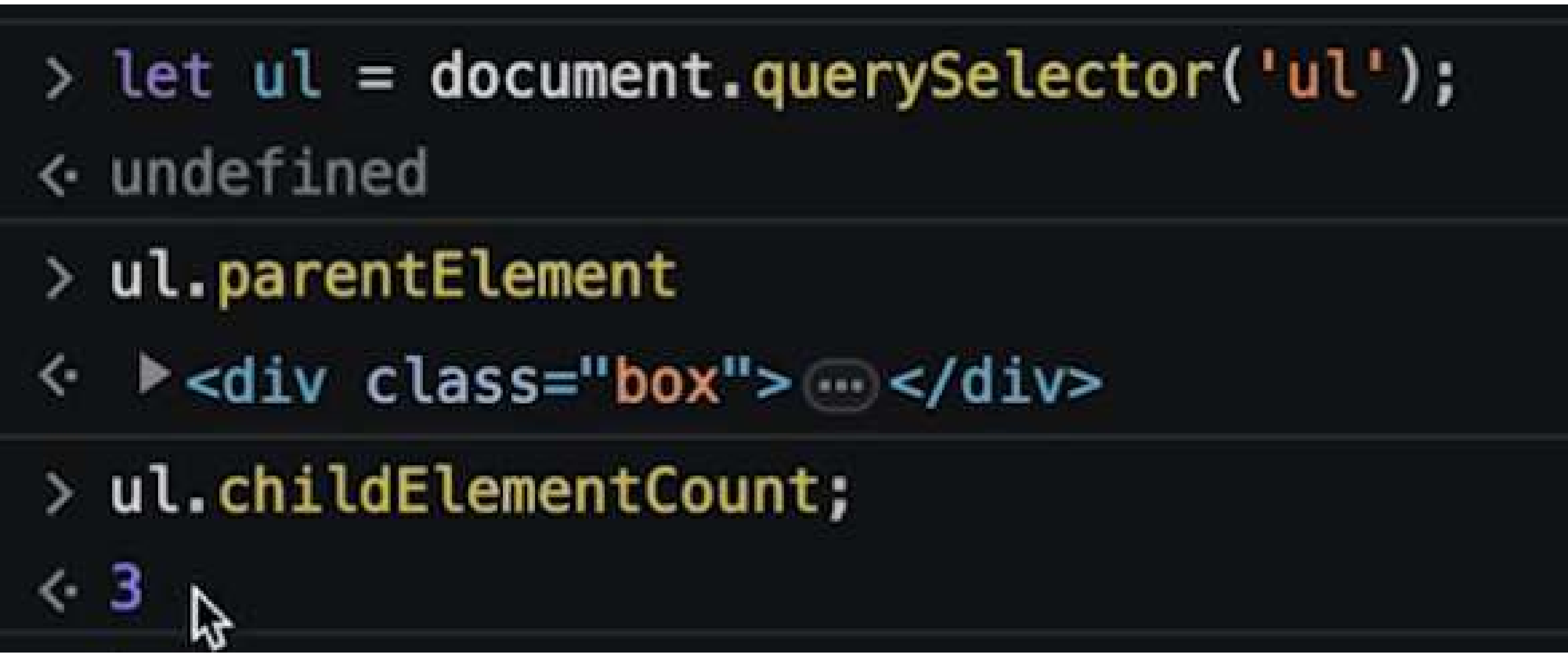
```
> h4.children
<· ▼HTMLCollection [] i
      length: 0
    ▶ [[Prototype]]: HTMLCollection
```



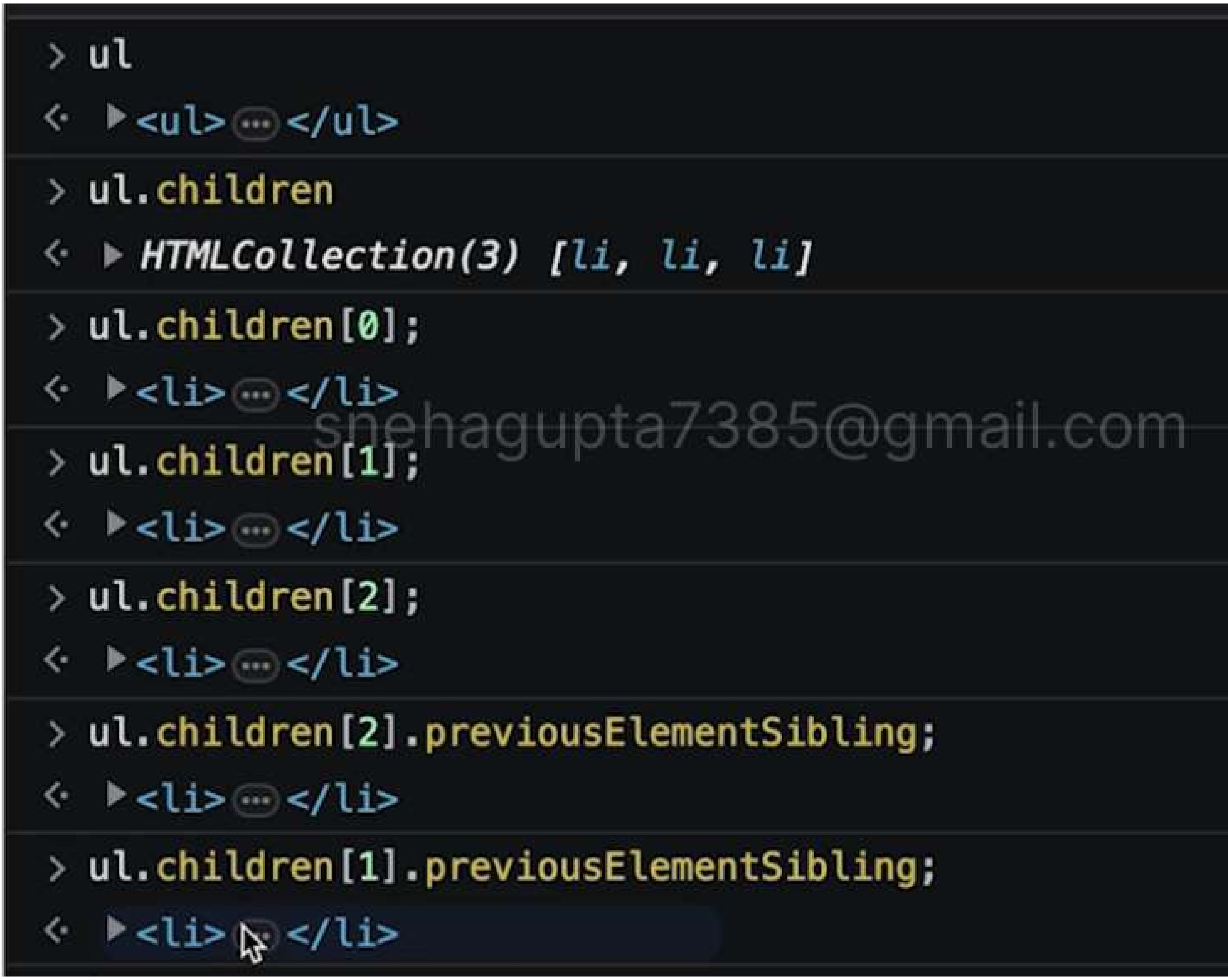
```
> let box = document.querySelector('.box');
<· undefined
> box.children;
<· ▶ HTMLCollection(2) [h4, ul]
```



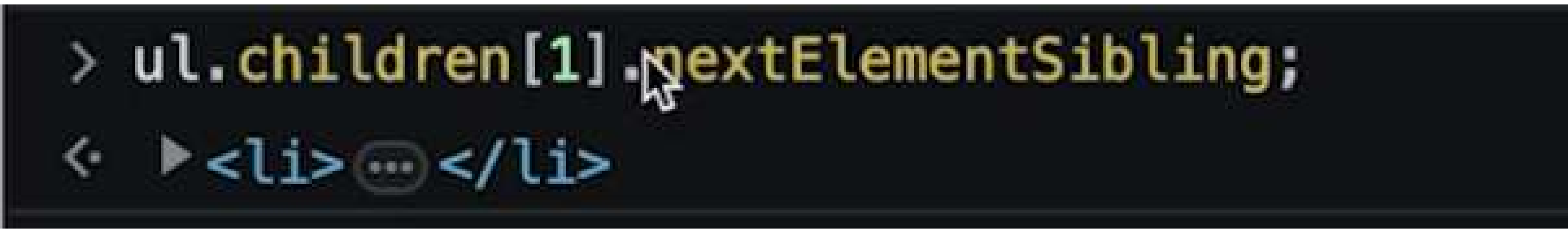
```
> box.childElementCount;
<· 2
```



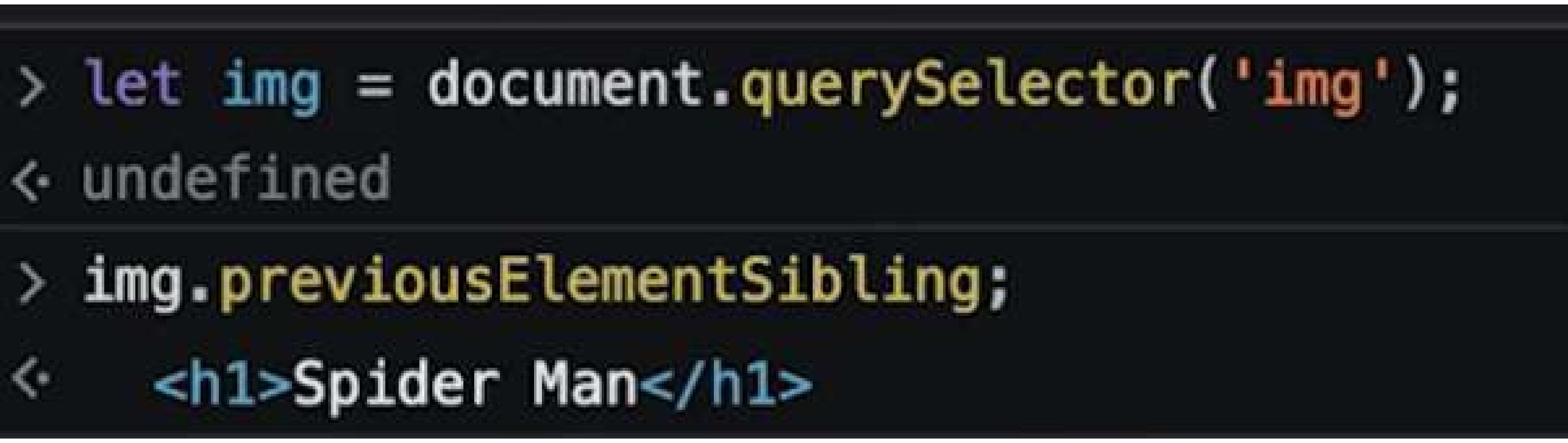
```
> let ul = document.querySelector('ul');
<· undefined
> ul.parentElement
<·  ▶<div class="box">…</div>
> ul.childElementCount;
<· 3
```



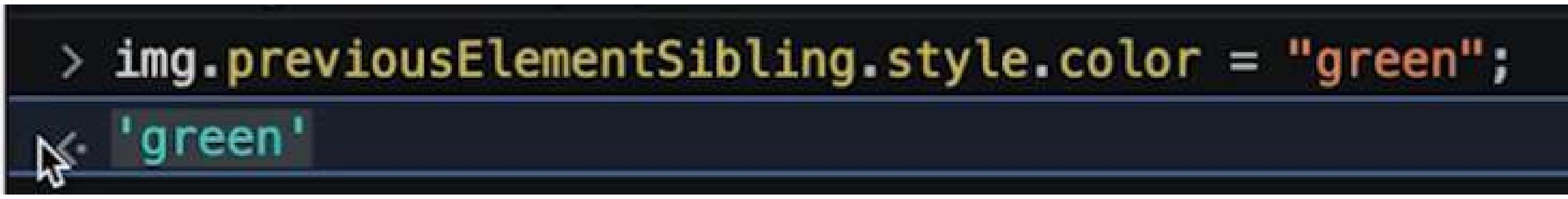
```
> ul
<· ▶<ul>…</ul>
> ul.children
<· ▶HTMLCollection(3) [li, li, li]
> ul.children[0];
<· ▶<li>…</li>
> ul.children[1];
<· ▶<li>…</li>
> ul.children[2];
<· ▶<li>…</li>
> ul.children[2].previousElementSibling;
<· ▶<li>…</li>
> ul.children[1].previousElementSibling;
<· ▶<li>…</li>
```



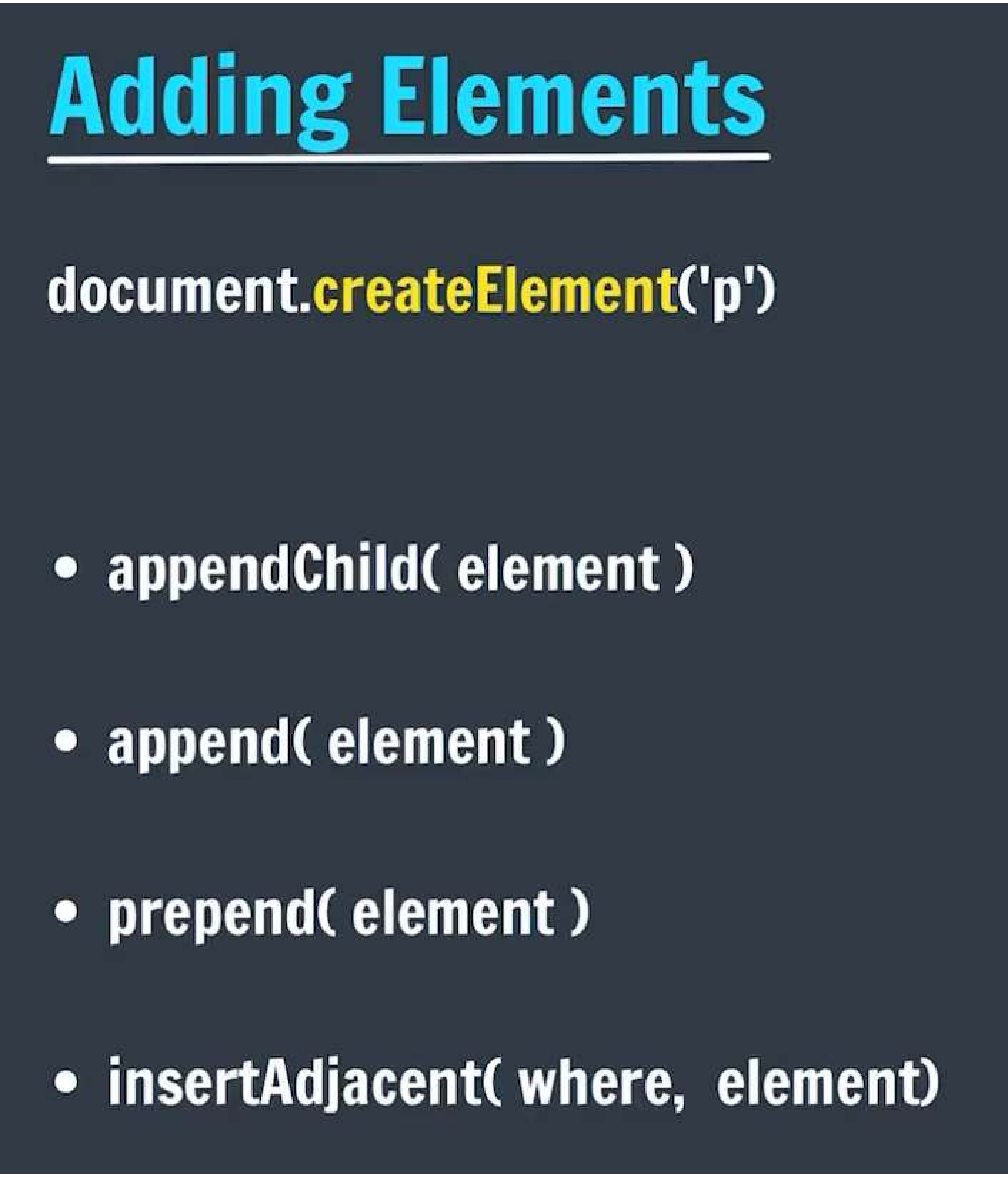
```
> ul.children[1].nextElementSibling;
<· ▶<li>…</li>
```



```
> let img = document.querySelector('img');
<· undefined
> img.previousElementSibling;
<·   <h1>Spider Man</h1>
```



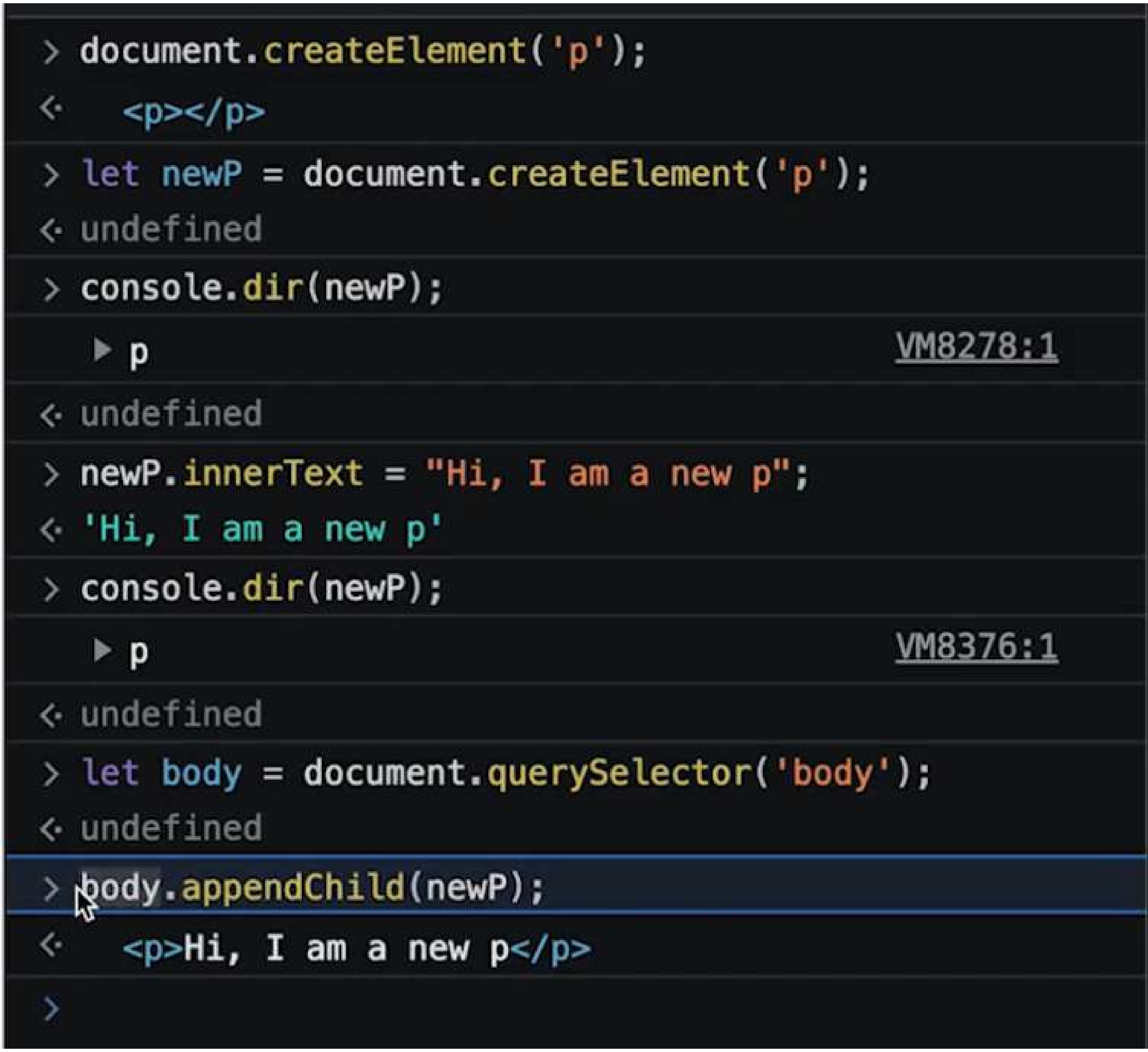
```
> img.previousElementSibling.style.color = "green";
<· 'green'
```



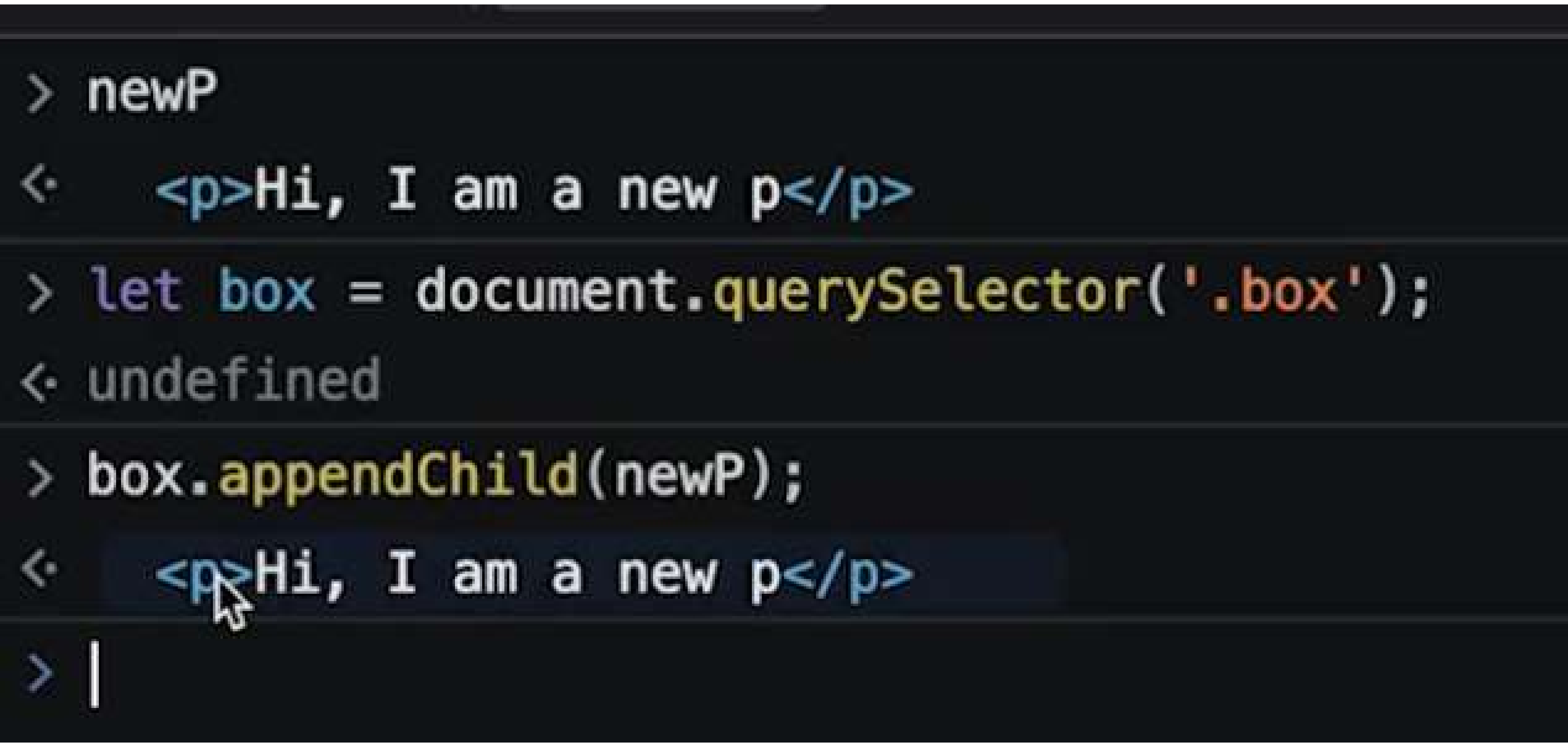
Adding Elements

```
document.createElement('p')
```

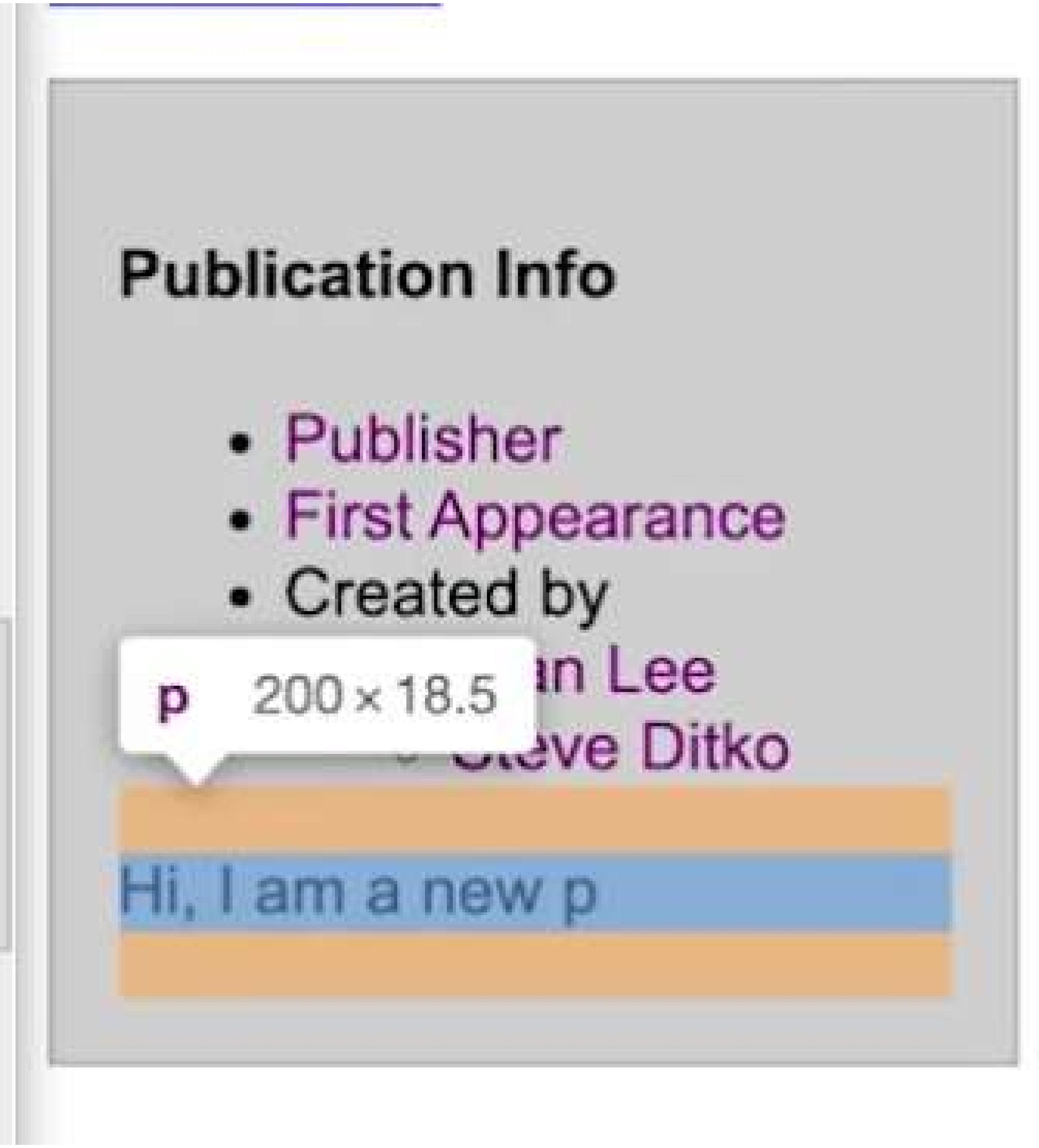
- appendChild(element)
- append(element)
- prepend(element)
- insertAdjacent(where, element)



```
> document.createElement('p');
<·   <p></p>
> let newP = document.createElement('p');
<· undefined
> console.dir(newP);
    ▶ p                                                   VM8278:1
<· undefined
> newP.innerText = "Hi, I am a new p";
<· 'Hi, I am a new p'
> console.dir(newP);
    ▶ p                                                   VM8376:1
<· undefined
> let body = document.querySelector('body');
<· undefined
> body.appendChild(newP);
<·   <p>Hi, I am a new p</p>
>
```



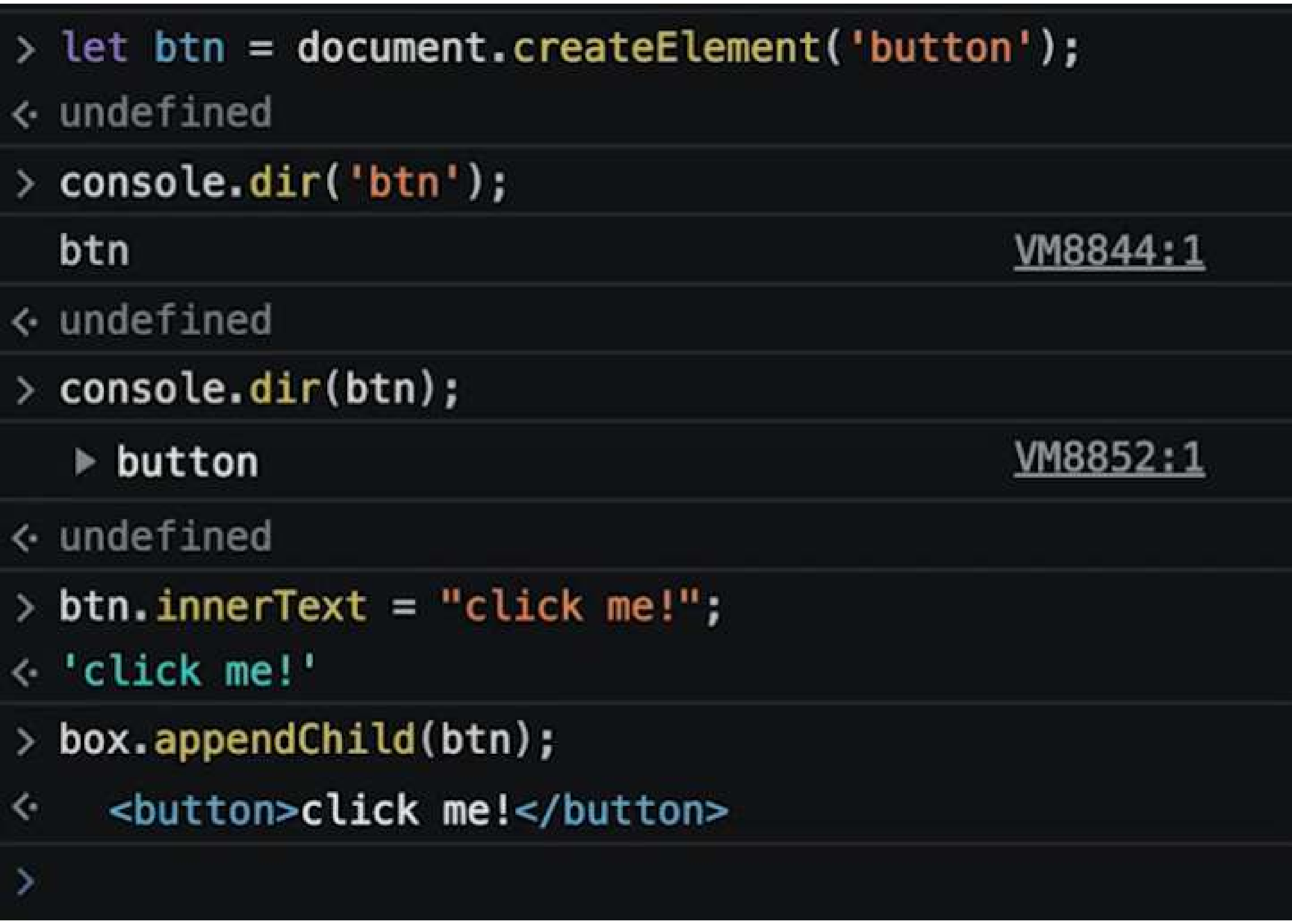
```
> newP
<·   <p>Hi, I am a new p</p>
> let box = document.querySelector('.box');
<· undefined
> box.appendChild(newP);
<·   <p>Hi, I am a new p</p>
> |
```



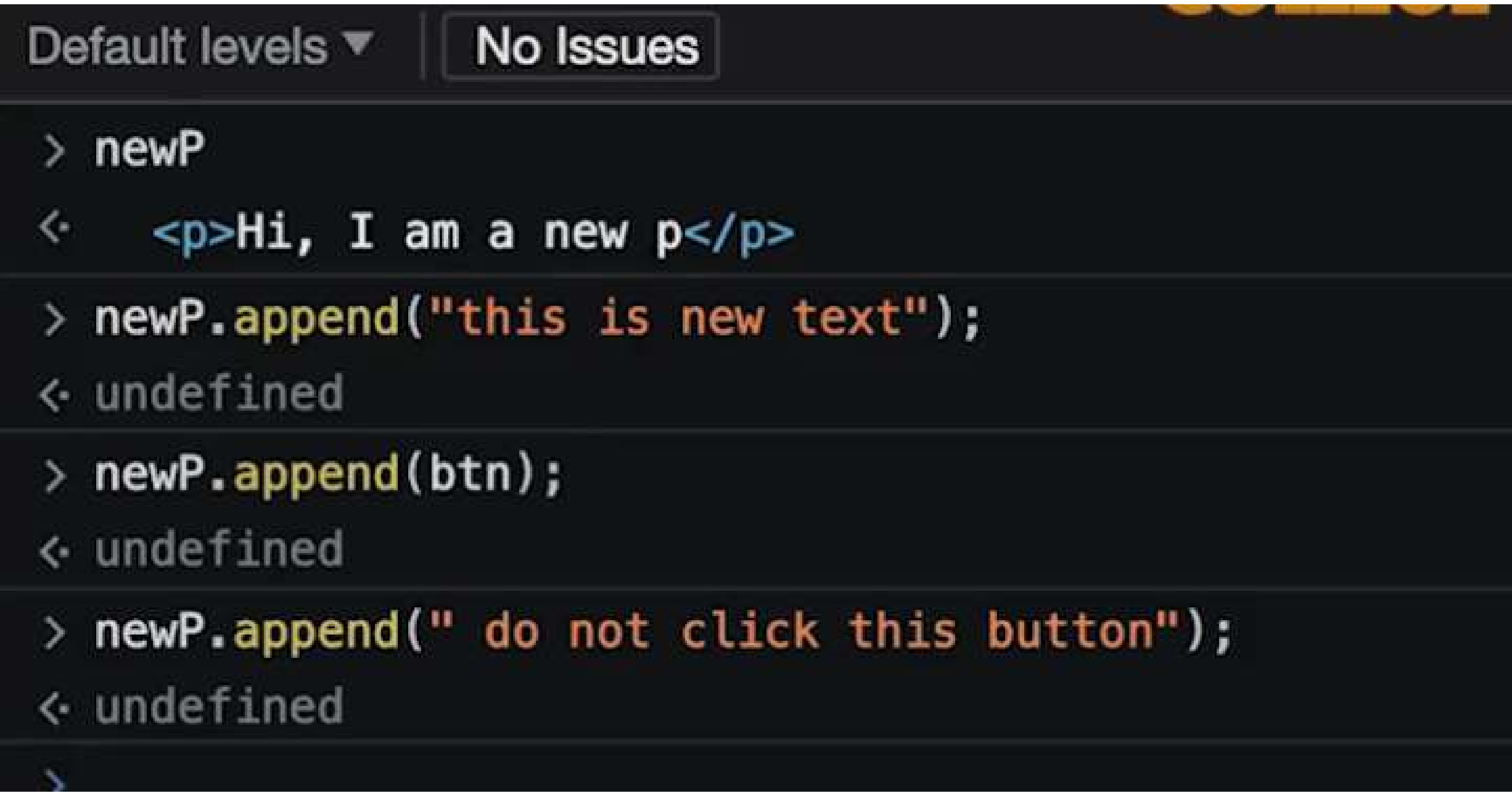
Publication Info

- Publisher
- First Appearance
- Created by
  - Stan Lee
  - Steve Ditko

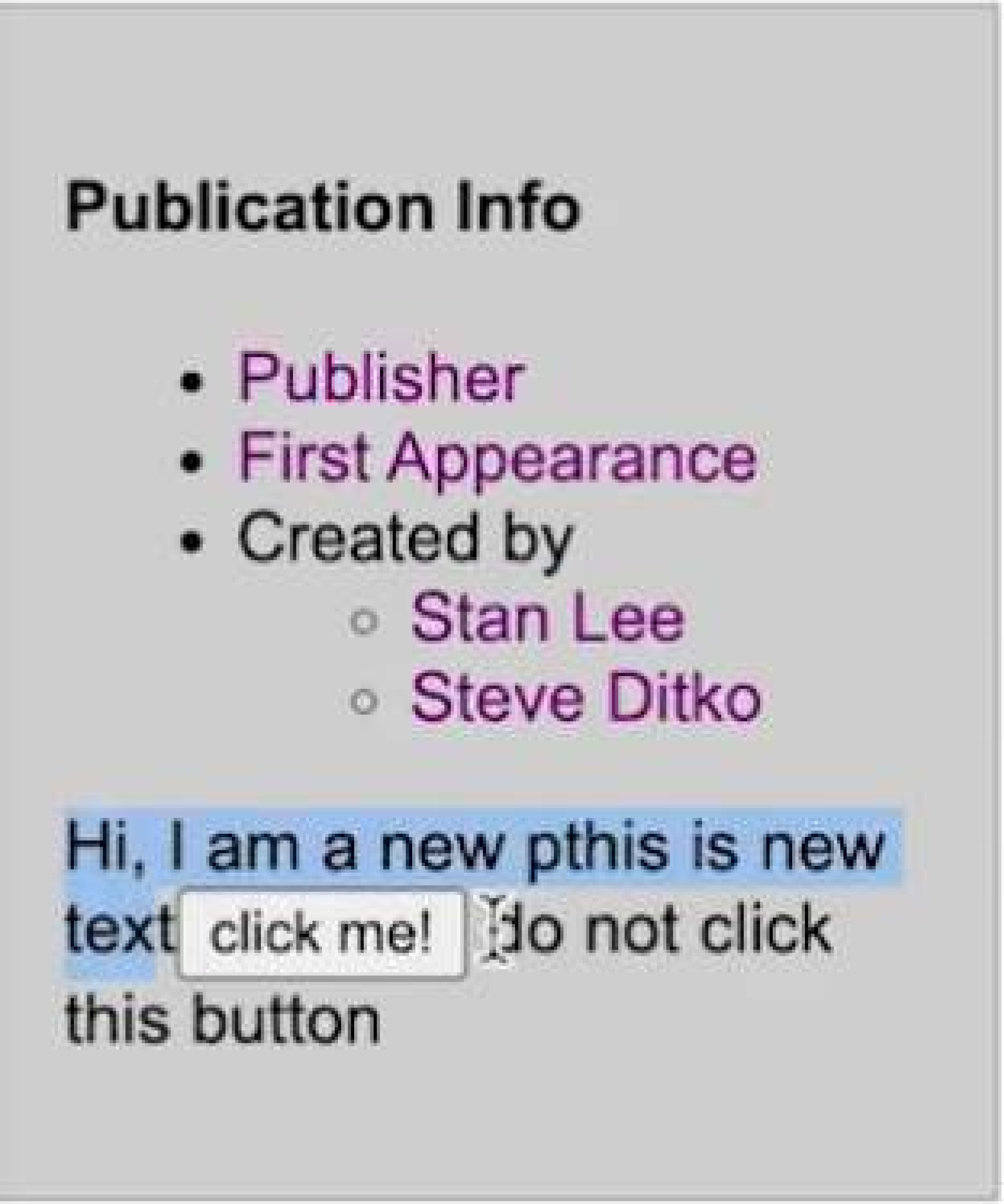
Hi, I am a new p



```
> let btn = document.createElement('button');
<· undefined
> console.dir('btn');
  btn                                                    VM8844:1
<· undefined
> console.dir(btn);
    ▶ button                                             VM8852:1
<· undefined
> btn.innerText = "click me!";
<· 'click me!'
> box.appendChild(btn);
<·   <button>click me!</button>
>
```



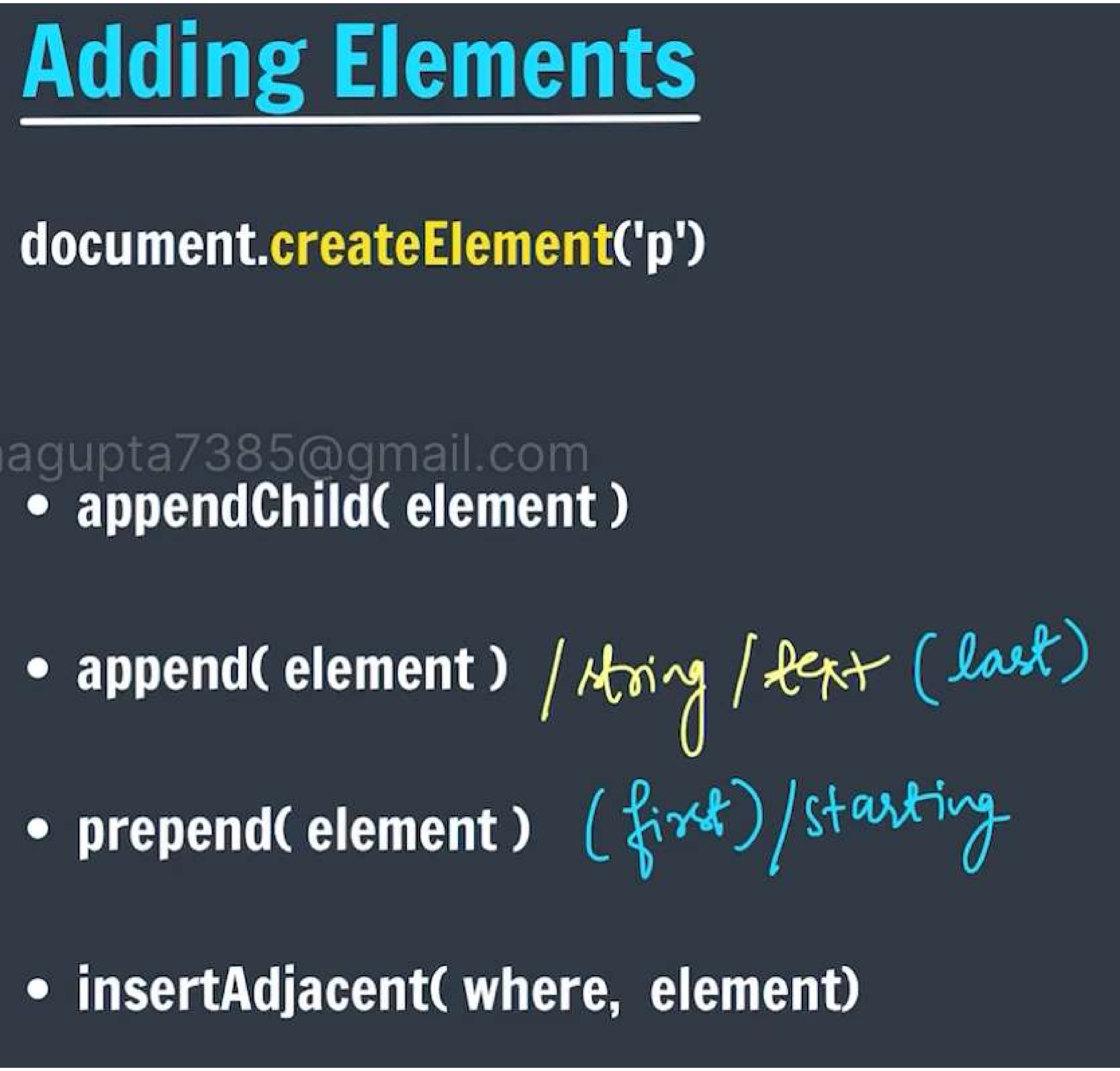
```
> newP
<·   <p>Hi, I am a new p</p>
> newP.append("this is new text");
<· undefined
> newP.append(btn);
<· undefined
> newP.append(" do not click this button");
<· undefined
>
```



Publication Info

- Publisher
- First Appearance
- Created by
  - Stan Lee
  - Steve Ditko

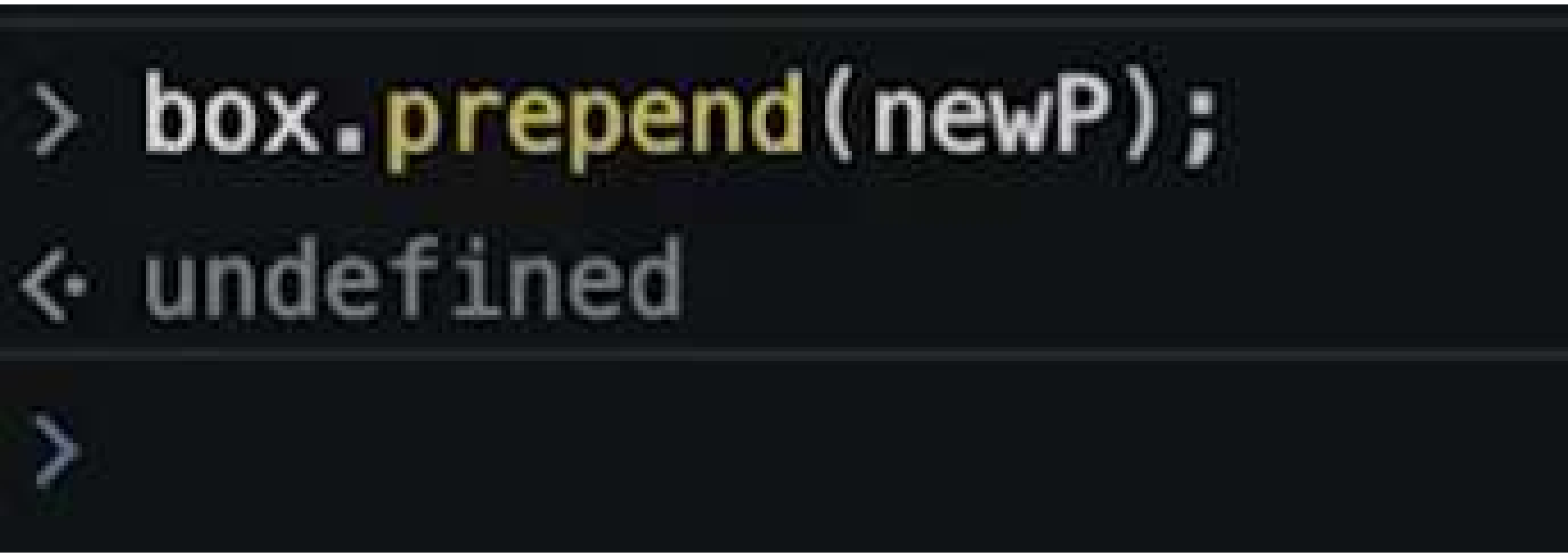
Hi, I am a new pthis is new text do not click this button



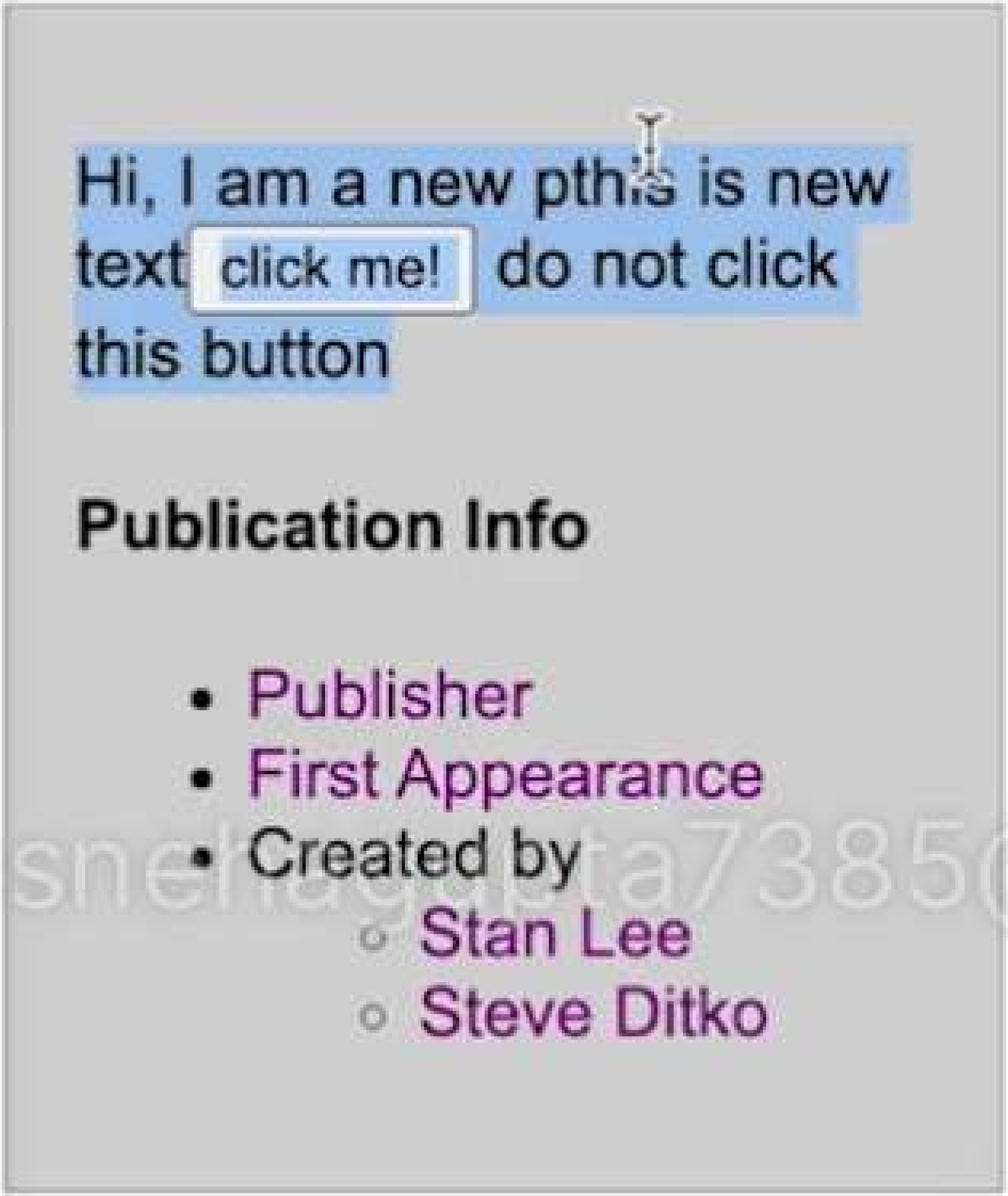
Adding Elements

```
document.createElement('p')
```

- appendChild(element)
- append(element) / string / text (last)
- prepend(element) (first) / starting
- insertAdjacent(where, element)



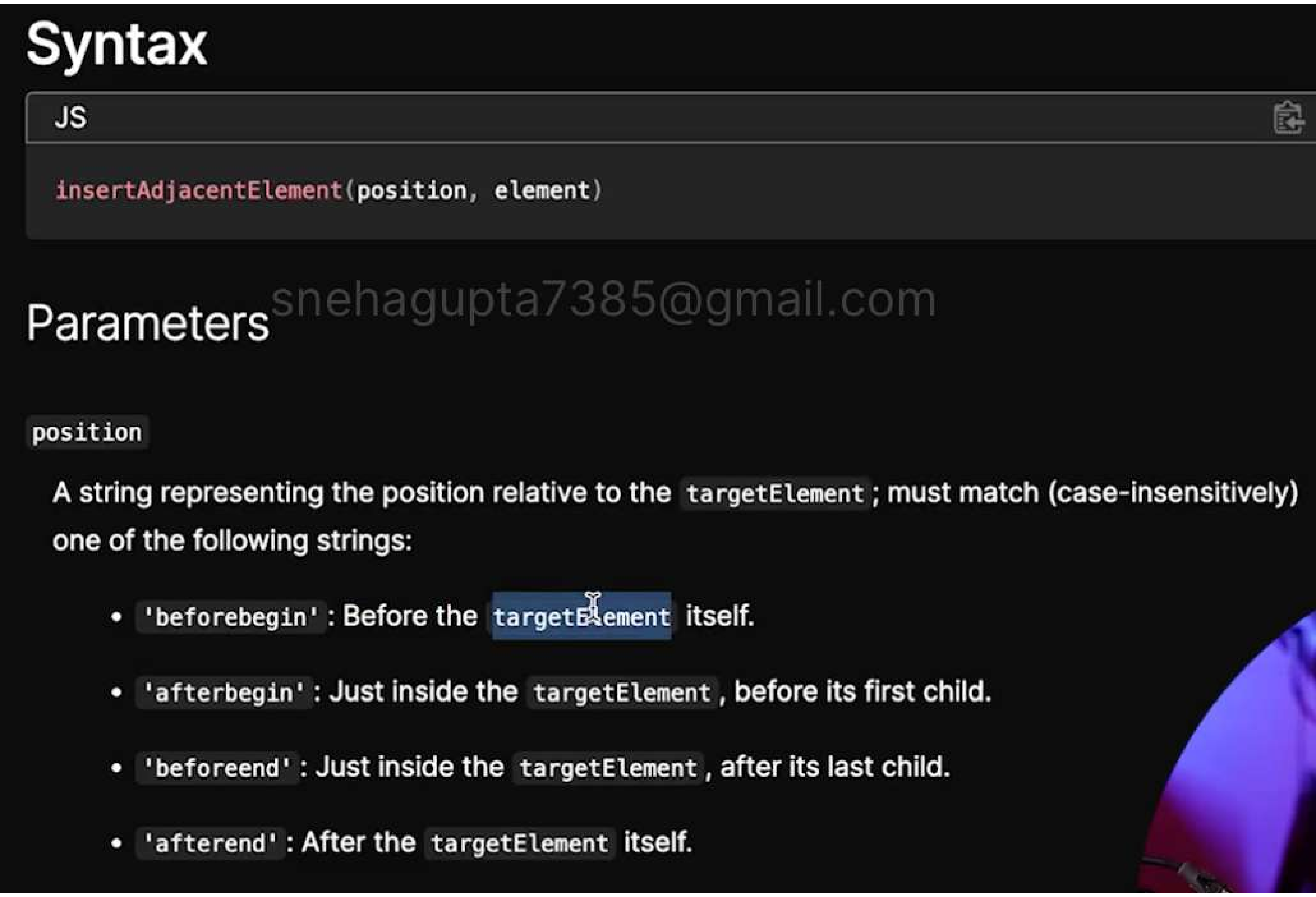
```
> box.prepend(newP);
<· undefined
>
```



Hi, I am a new pthis is new text do not click this button

Publication Info

- Publisher
- First Appearance
- Created by
  - Stan Lee
  - Steve Ditko



Syntax

JS

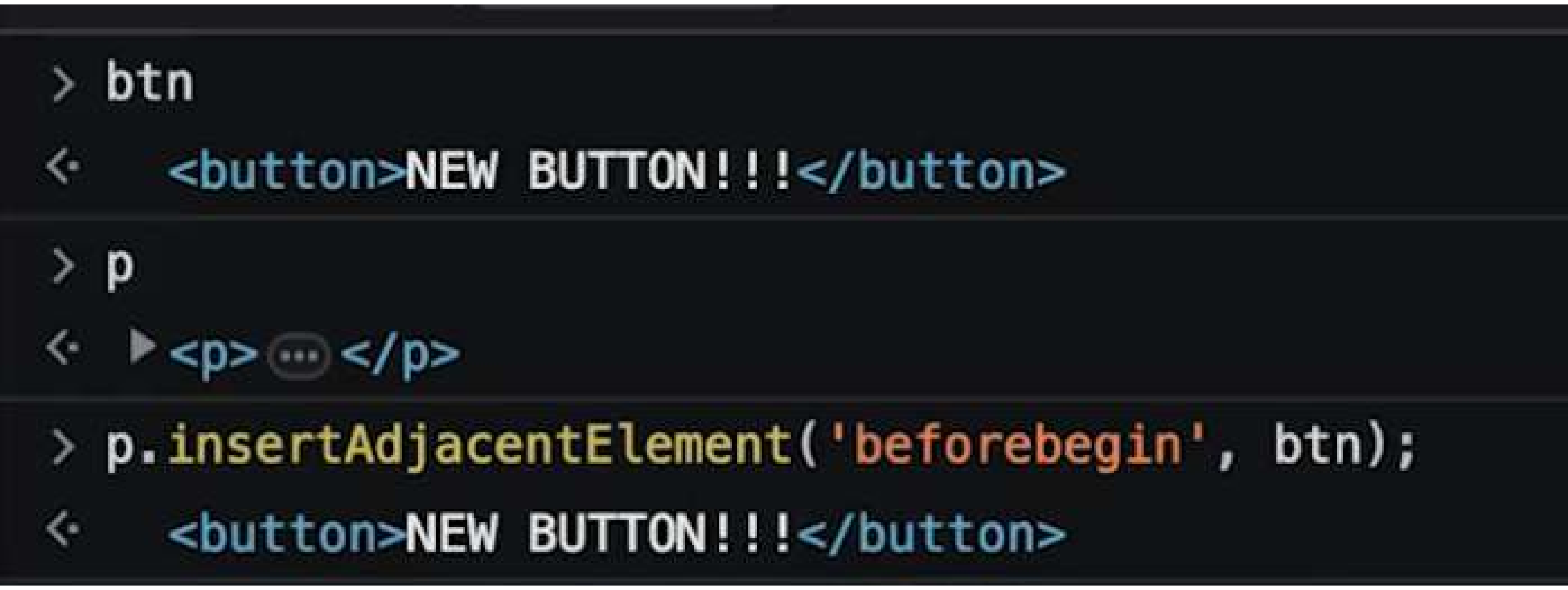
```js
insertAdjacentElement(position, element)
```

Parameters

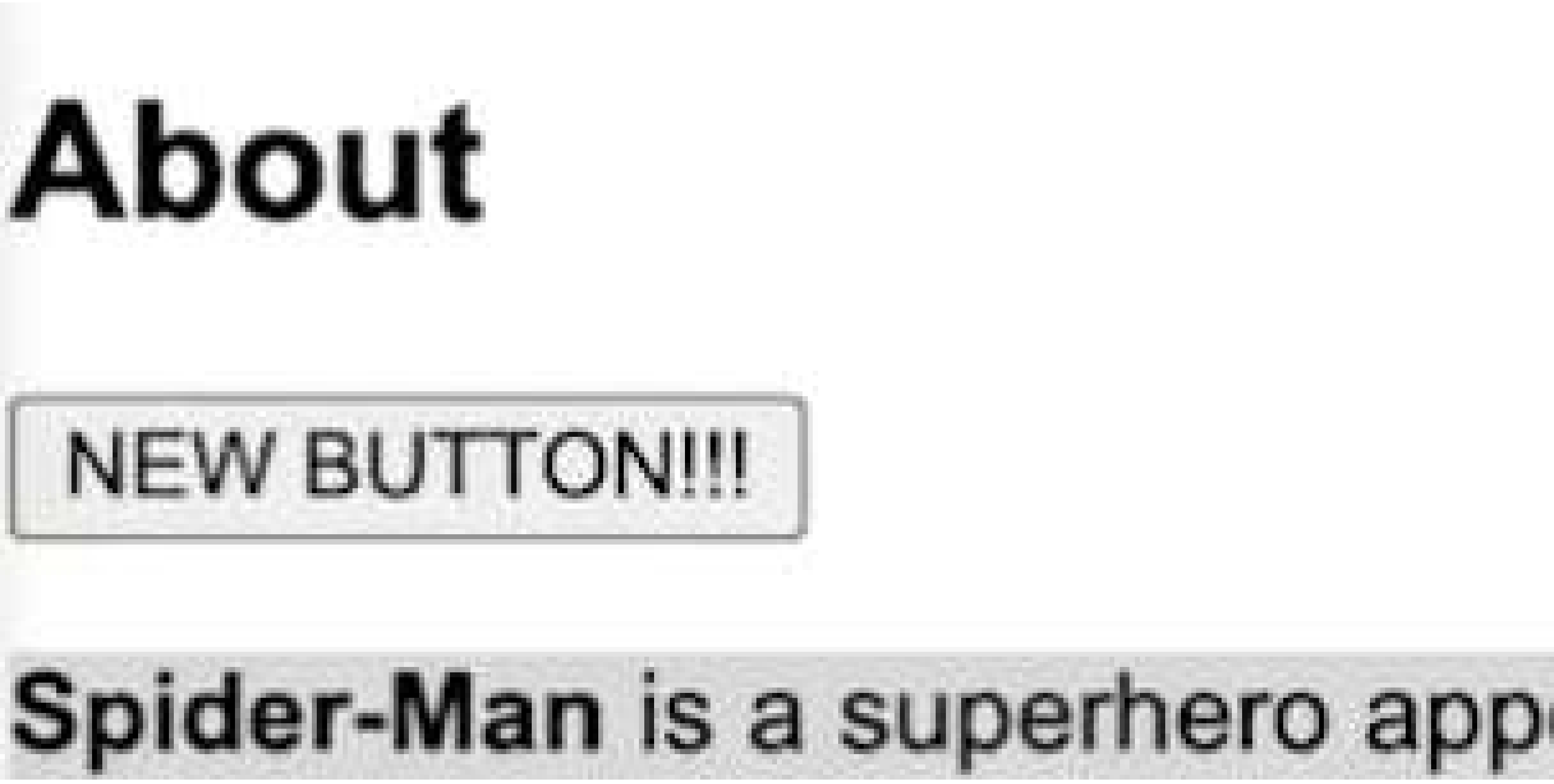
`position`

A string representing the position relative to the `targetElement`; must match (case-insensitively) one of the following strings:

- `'beforebegin'`: Before the `targetElement` itself.
- `'afterbegin'`: Just inside the `targetElement`, before its first child.
- `'beforeend'`: Just inside the `targetElement`, after its last child.
- `'afterend'`: After the `targetElement` itself.

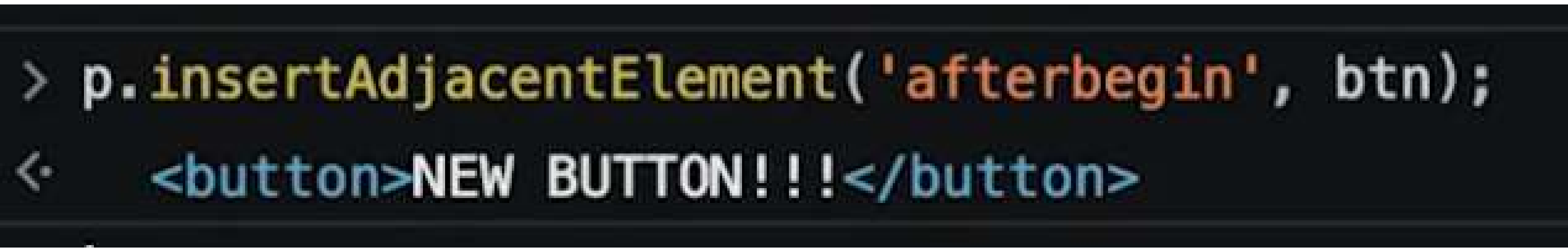


```
> btn
<·   <button>NEW BUTTON!!!</button>
> p
<·  ▶<p>…</p>
> p.insertAdjacentElement('beforebegin', btn);
<·   <button>NEW BUTTON!!!</button>
```

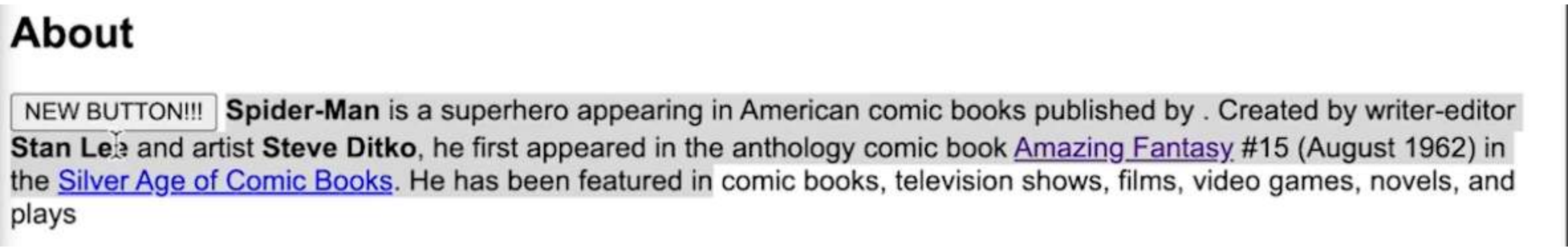


About

Spider-Man is a superhero appe

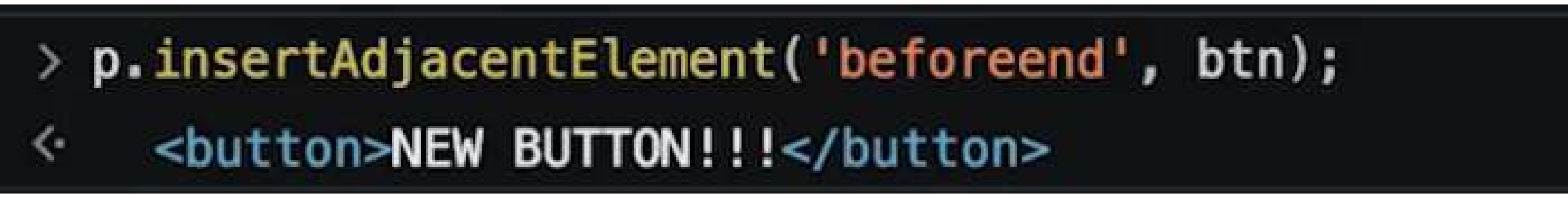


```
> p.insertAdjacentElement('afterbegin', btn);
<·   <button>NEW BUTTON!!!</button>
```

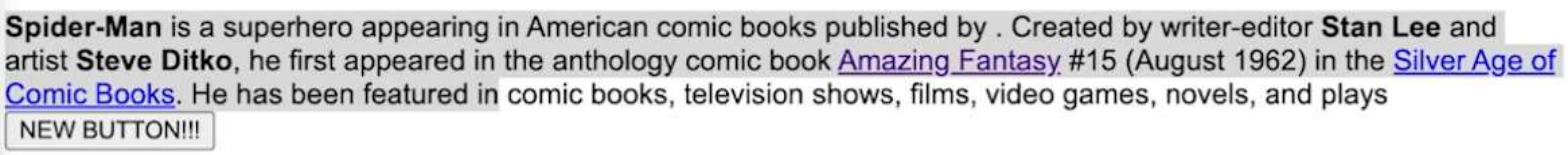


About

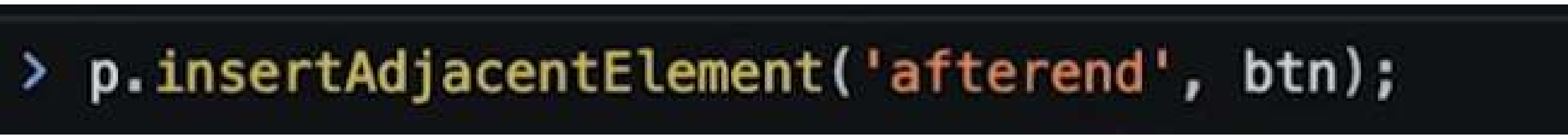
Spider-Man is a superhero appearing in American comic books published by . Created by writer-editor **Stan Lee** and artist **Steve Ditko**, he first appeared in the anthology comic book Amazing Fantasy #15 (August 1962) in the Silver Age of Comic Books. He has been featured in comic books, television shows, films, video games, novels, and plays



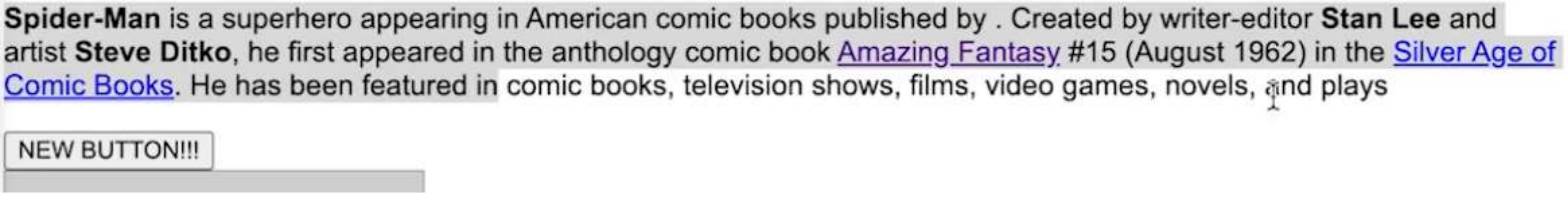
```
> p.insertAdjacentElement('beforeend', btn);
<·   <button>NEW BUTTON!!!</button>
```



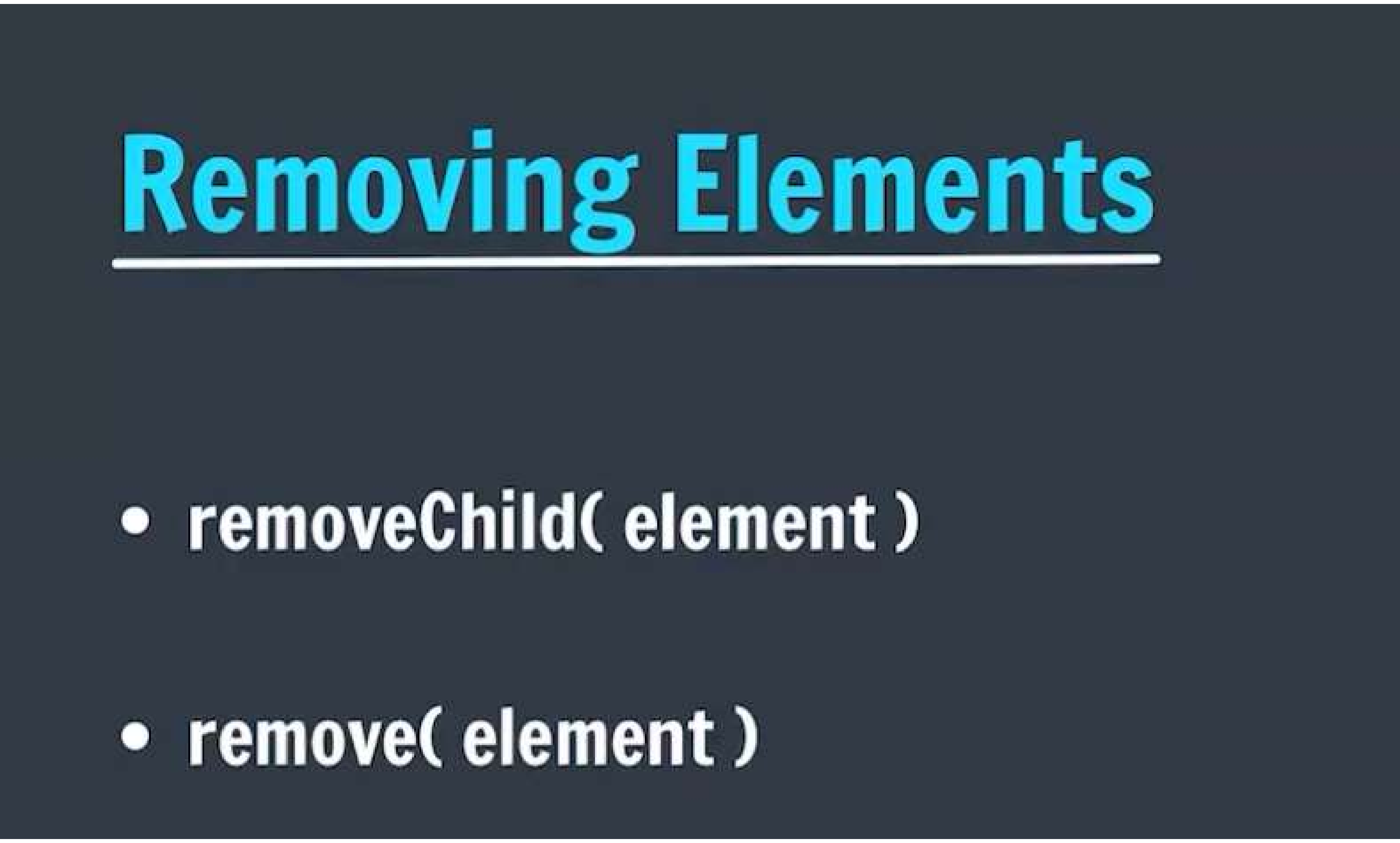
Spider-Man is a superhero appearing in American comic books published by . Created by writer-editor **Stan Lee** and artist **Steve Ditko**, he first appeared in the anthology comic book Amazing Fantasy #15 (August 1962) in the Silver Age of Comic Books. He has been featured in comic books, television shows, films, video games, novels, and plays



```
> p.insertAdjacentElement('afterend', btn);
```

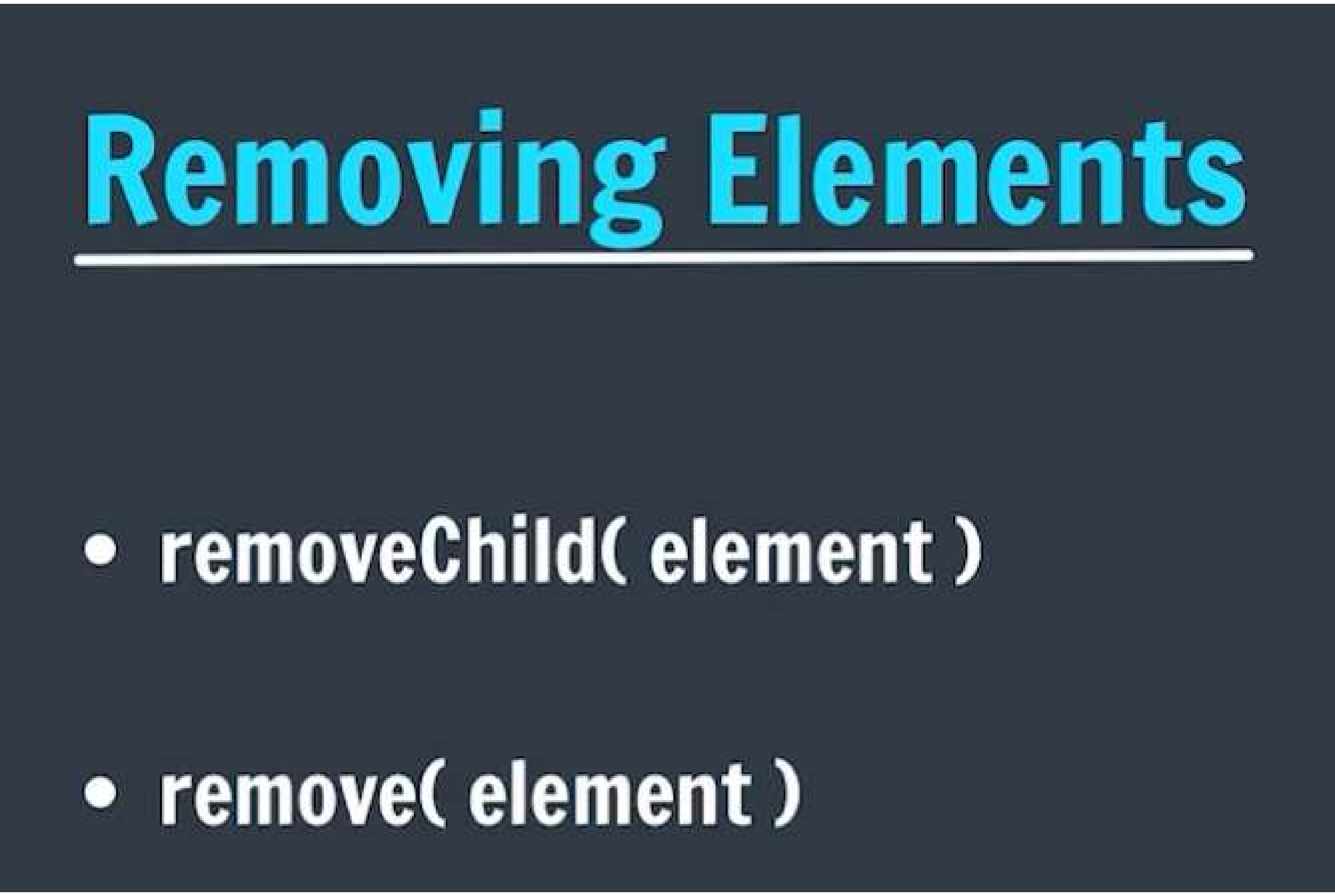


Spider-Man is a superhero appearing in American comic books published by . Created by writer-editor **Stan Lee** and artist **Steve Ditko**, he first appeared in the anthology comic book Amazing Fantasy #15 (August 1962) in the Silver Age of Comic Books. He has been featured in comic books, television shows, films, video games, novels, and plays



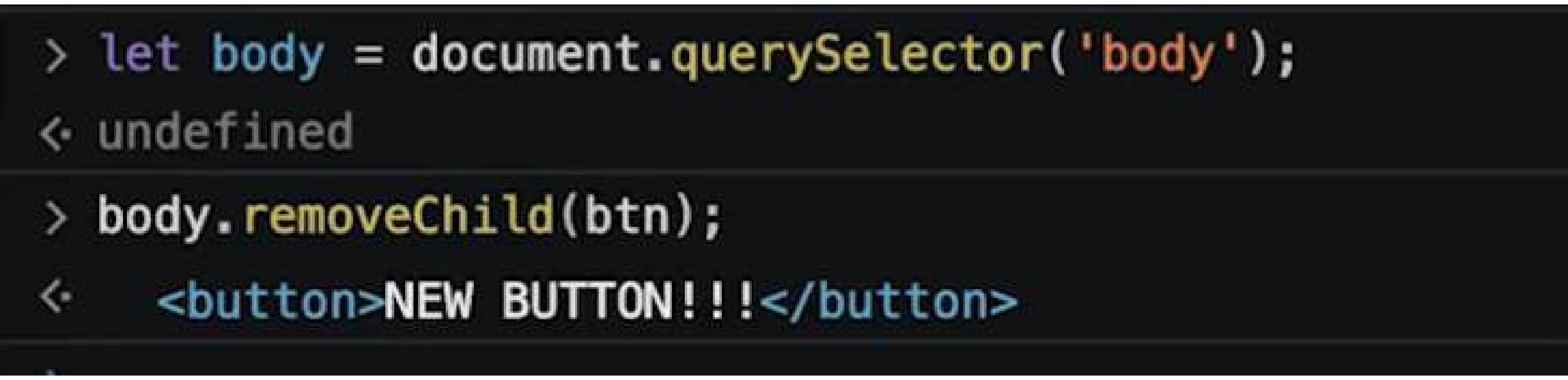
Removing Elements

- removeChild(element)
- remove(element)

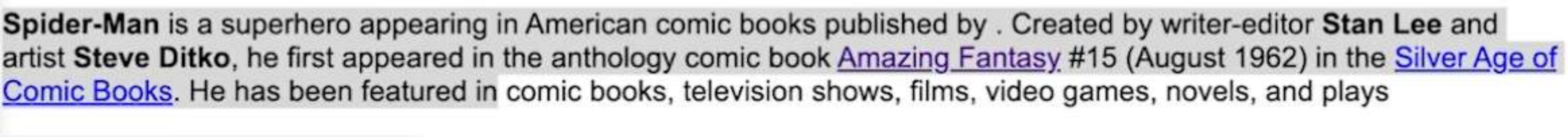


Removing Elements

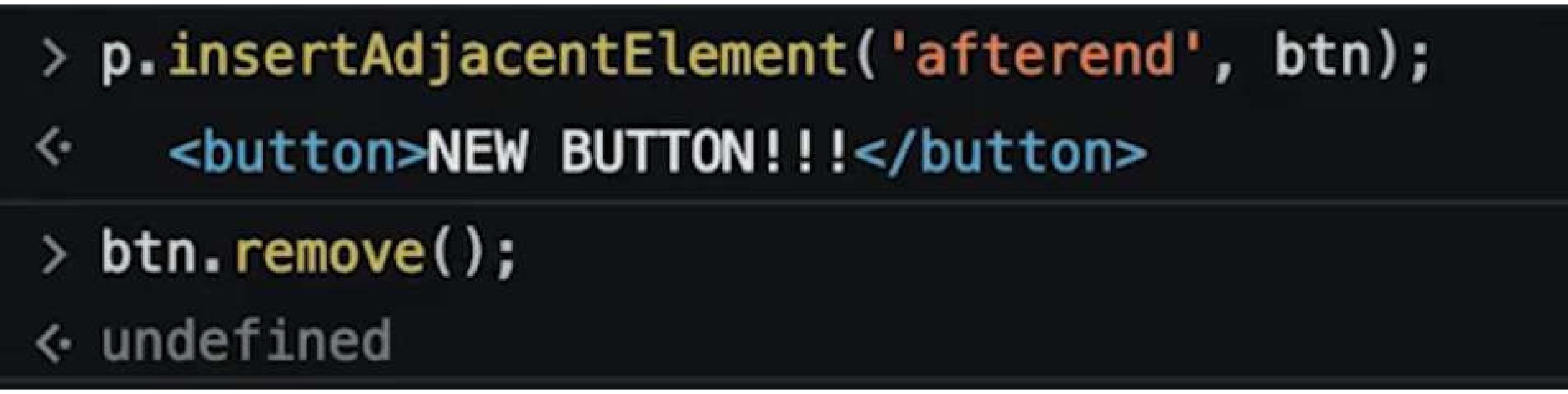
- removeChild(element)
- remove(element)



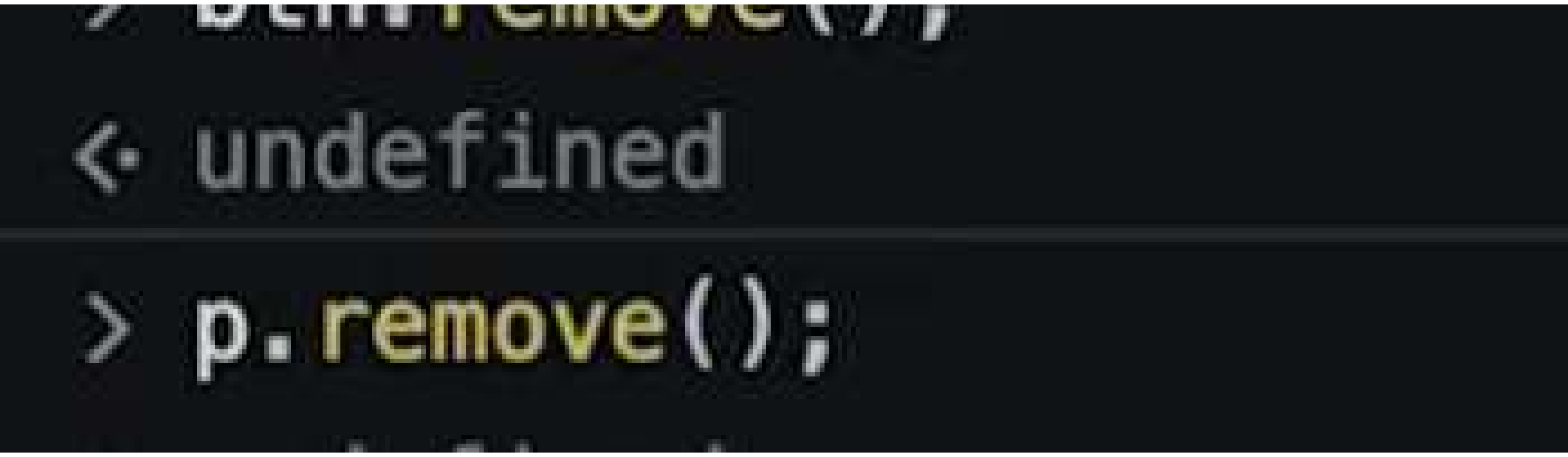
```
> let body = document.querySelector('body');
<· undefined
> body.removeChild(btn);
<·   <button>NEW BUTTON!!!</button>
```



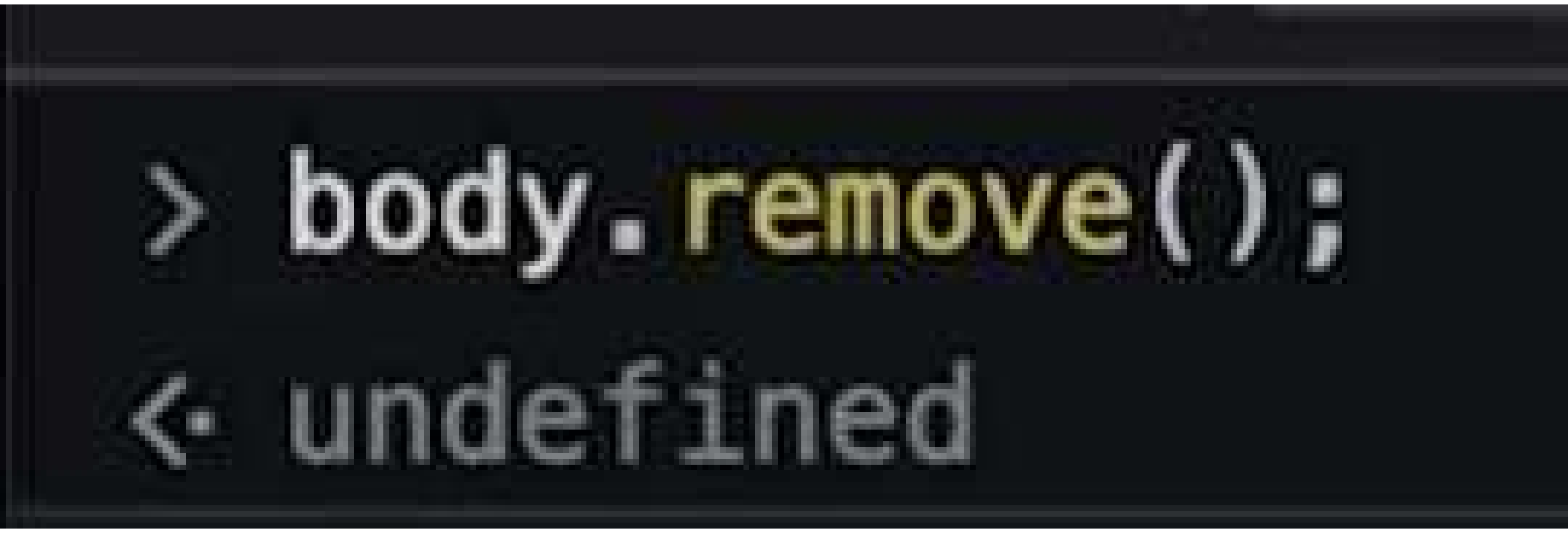
Spider-Man is a superhero appearing in American comic books published by . Created by writer-editor **Stan Lee** and artist **Steve Ditko**, he first appeared in the anthology comic book Amazing Fantasy #15 (August 1962) in the Silver Age of Comic Books. He has been featured in comic books, television shows, films, video games, novels, and plays



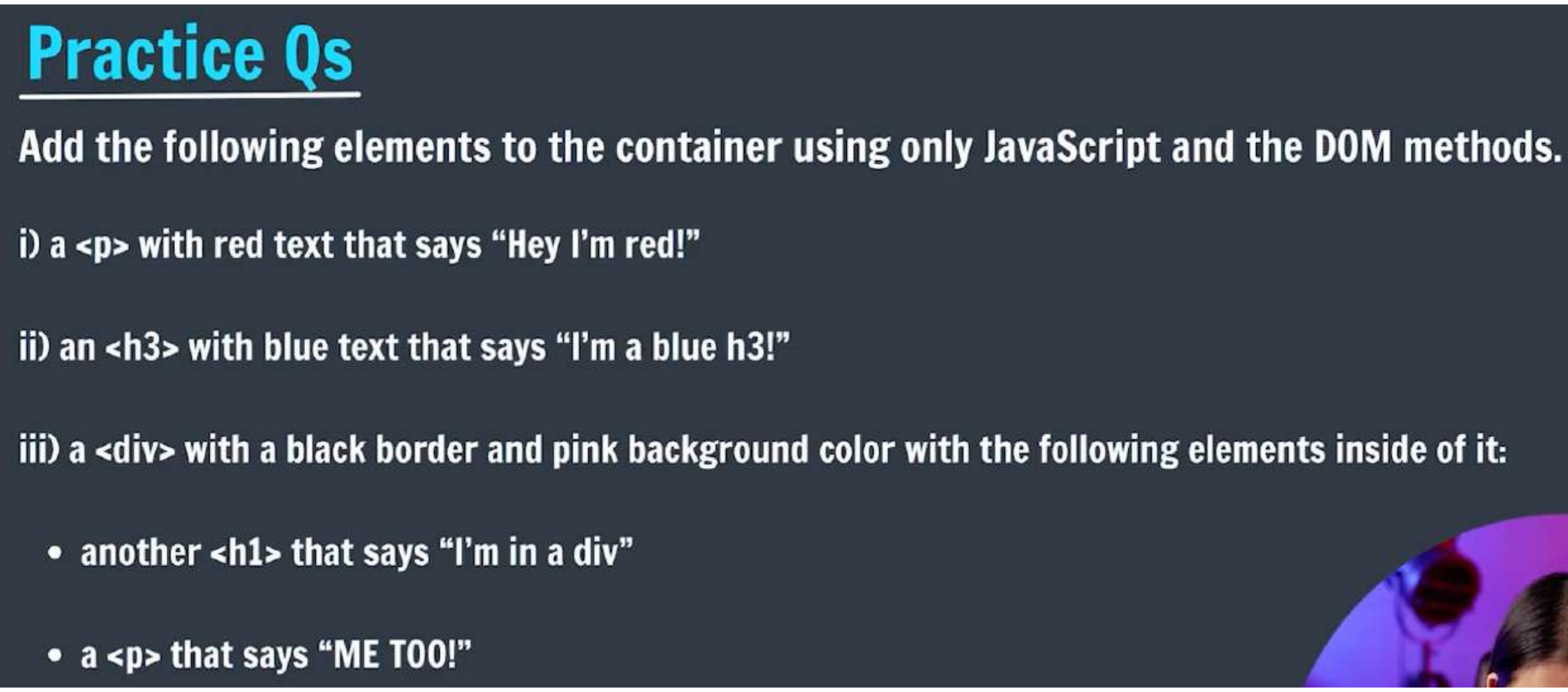
```
> p.insertAdjacentElement('afterend', btn);
<·   <button>NEW BUTTON!!!</button>
> btn.remove();
<· undefined
```



```
> btn.remove();
<· undefined
> p.remove();
```



```
> body.remove();
<· undefined
```



Practice Qs

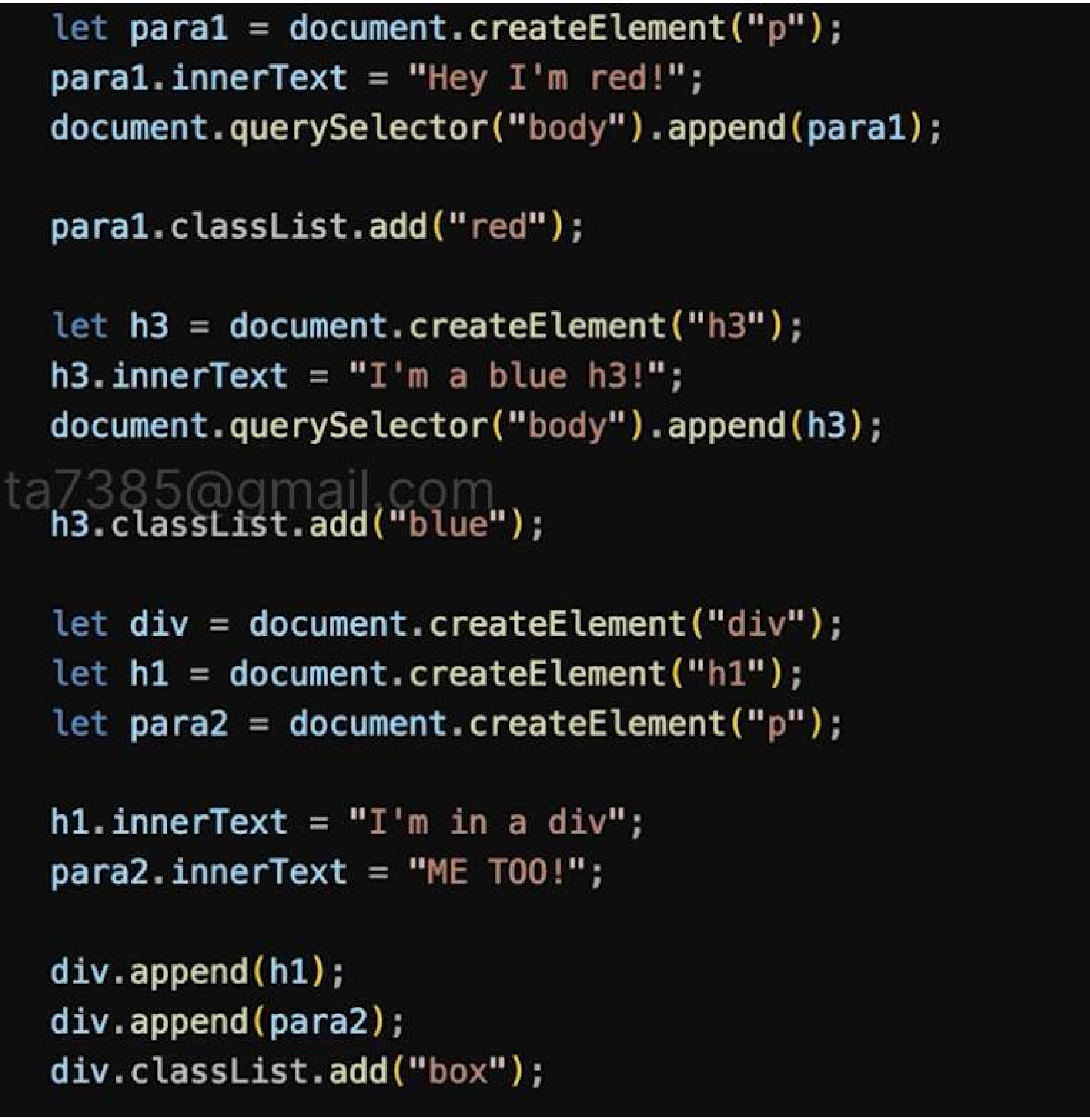
Add the following elements to the container using only JavaScript and the DOM methods.

i) a <p> with red text that says “Hey I’m red!”

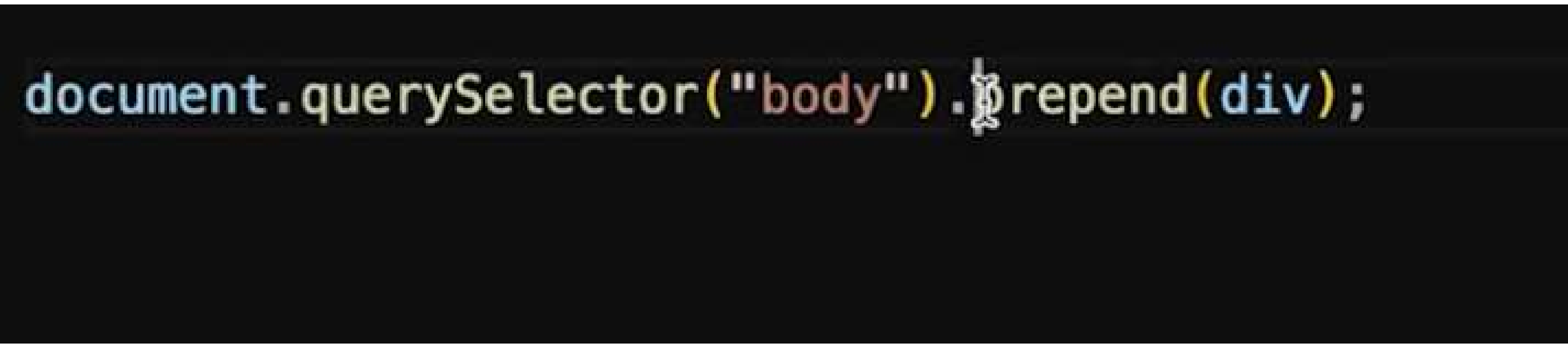
ii) an <h3> with blue text that says “I’m a blue h3!”

iii) a <div> with a black border and pink background color with the following elements inside of it:

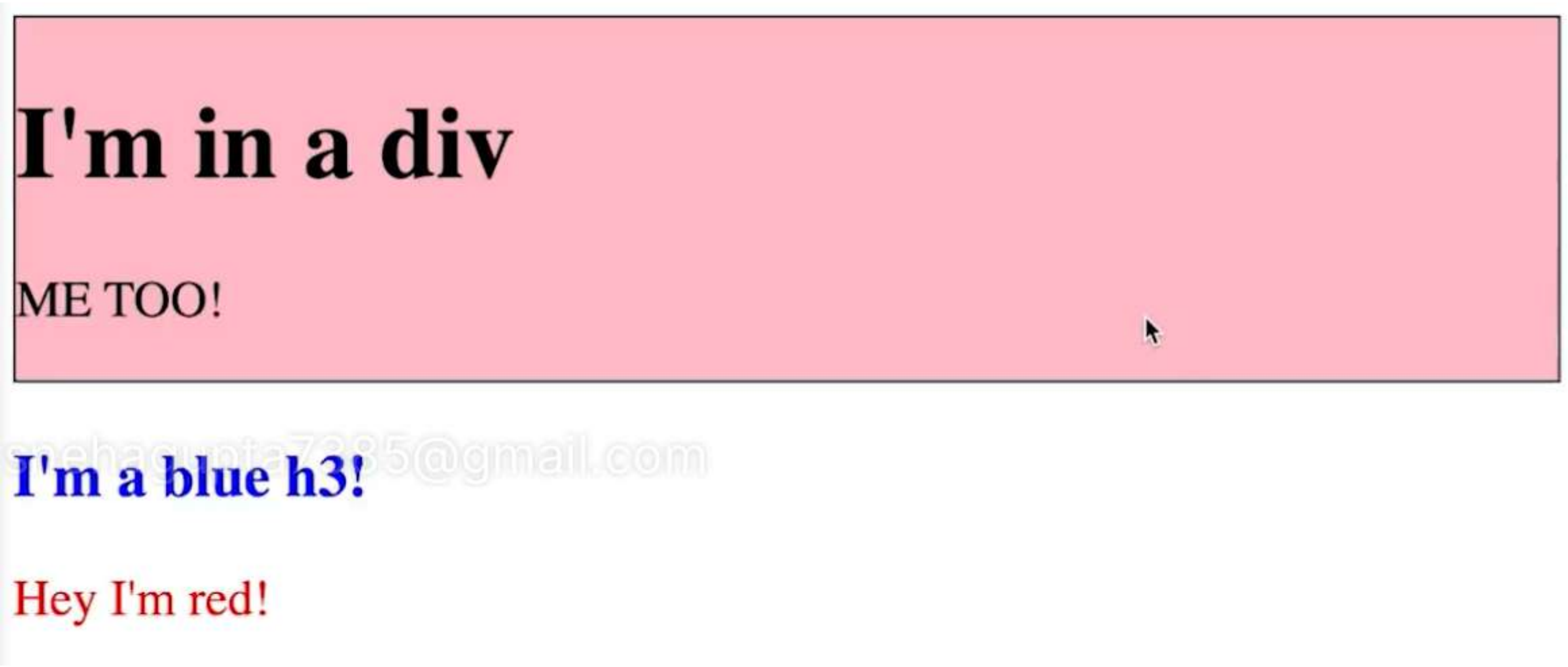
- another <h1> that says “I’m in a div”
- a <p> that says “ME TOO!”



```
let para1 = document.createElement("p");
para1.innerText = "Hey I'm red!";
document.querySelector("body").append(para1);

para1.classList.add("red");

let h3 = document.createElement("h3");
h3.innerText = "I'm a blue h3!";
document.querySelector("body").append(h3);

h3.classList.add("blue");

let div = document.createElement("div");
let h1 = document.createElement("h1");
let para2 = document.createElement("p");

h1.innerText = "I'm in a div";
para2.innerText = "ME TOO!";

div.append(h1);
div.append(para2);
div.classList.add("box");
```



```
document.querySelector("body").prepend(div);
```

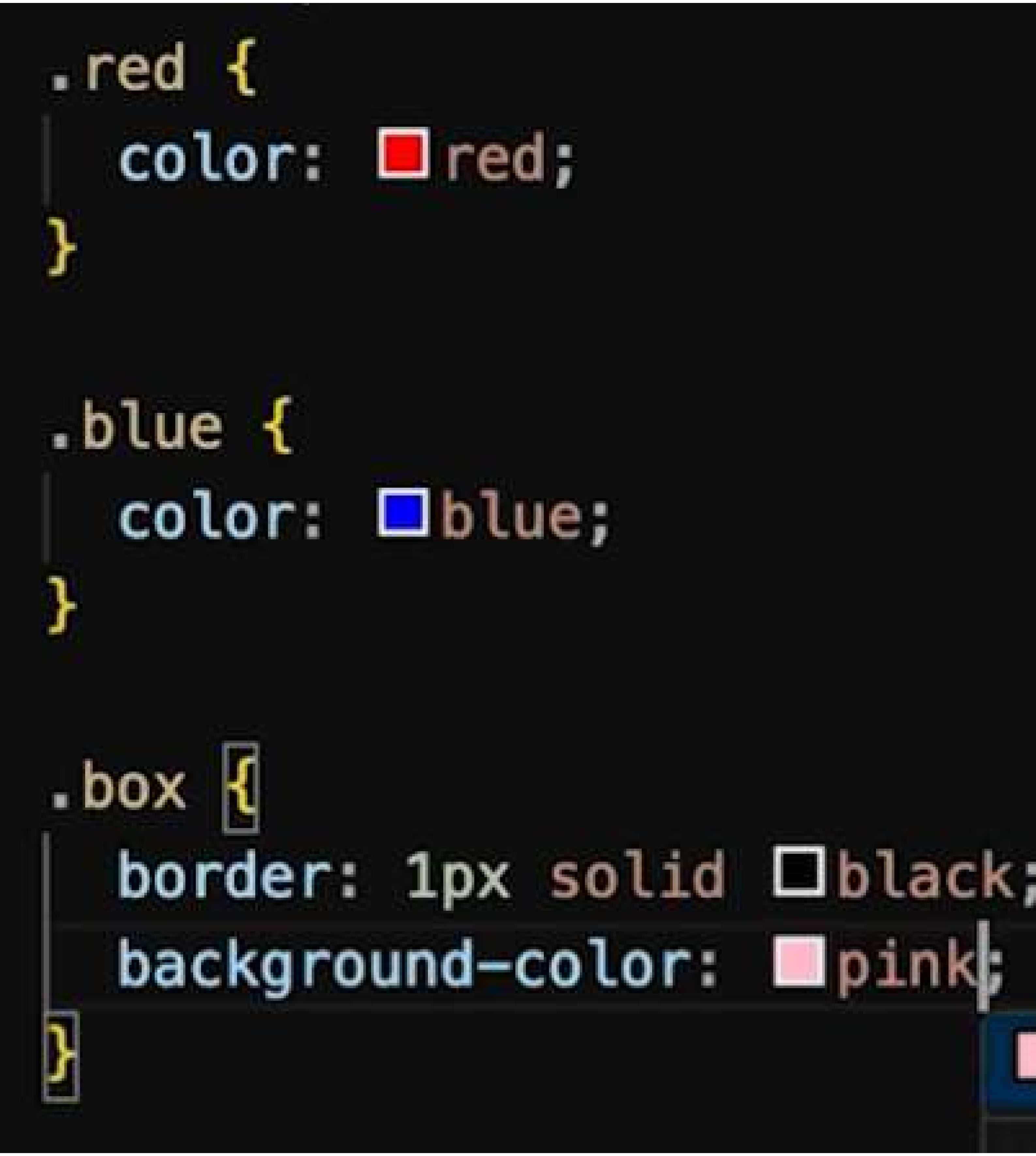


I'm in a div

ME TOO!

I'm a blue h3!

Hey I'm red!



```
.red {
  color: red;
}

.blue {
  color: blue;
}

.box {
  border: 1px solid black;
  background-color: pink;
}
```