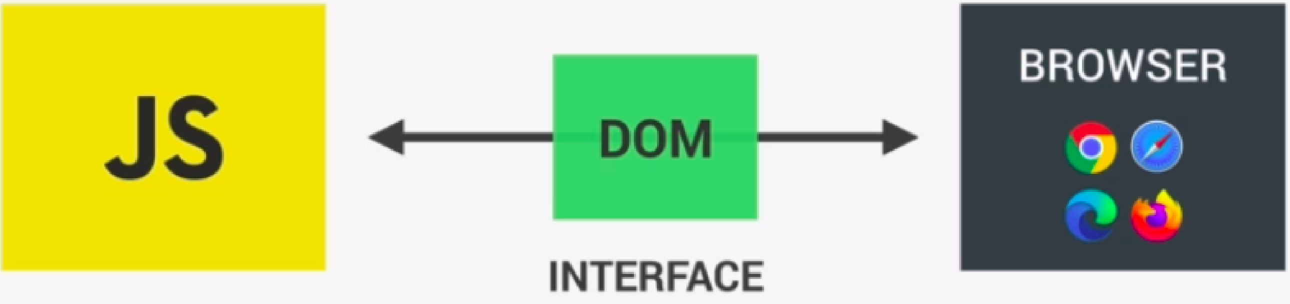
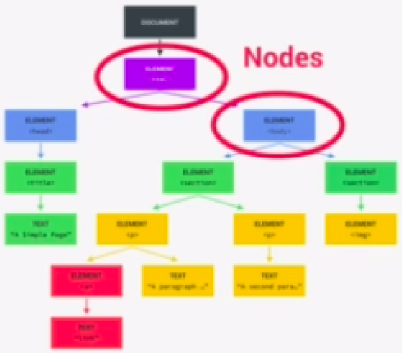
REVIEW: WHAT IS THE DOM?



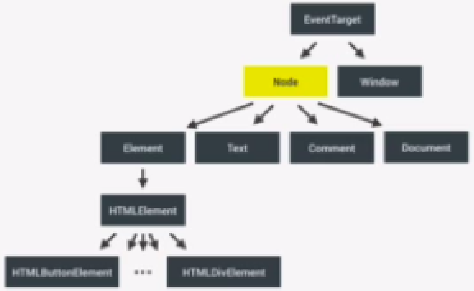


DOM tree

- Allows us to make JavaScript interact with the browser;
- We can write JavaScript to create, modify and delete HTML elements; set styles, classes and attributes; and listen and respond to events;
- DOM tree is generated from an HTML document, which we can then interact with;
- DOM is a very complex API that contains lots of methods and properties to interact with the DOM tree

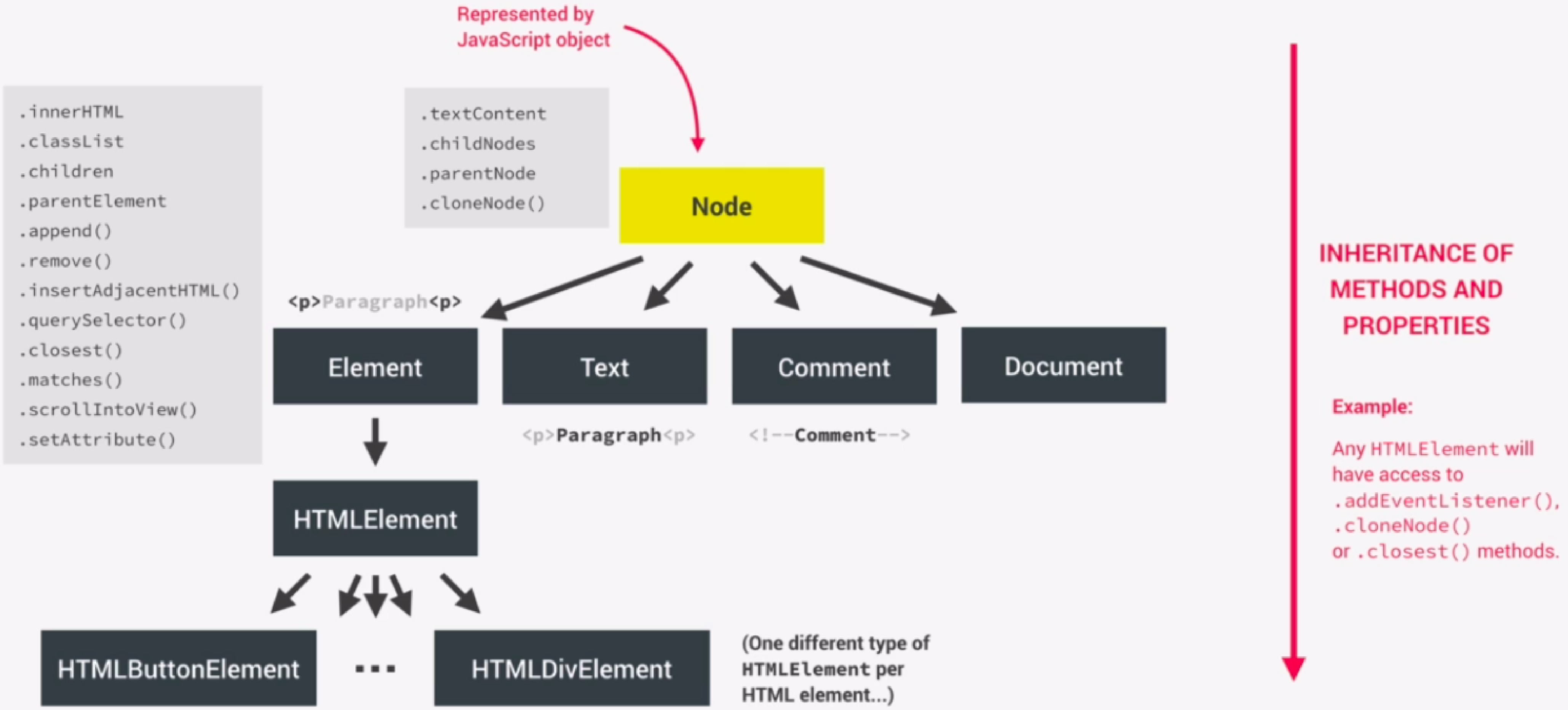
Application Programming Interface

```
.querySelector() / .addEventListener() / .createElement() /
.innerHTML / .textContent / .children / etc ...
```



"Types" of DOM objects (next slide)

HOW THE DOM API IS ORGANIZED BEHIND THE SCENES



Represented by JavaScript object

```
.innerHTML
.classList
.children
.parentElement
.append()
.remove()
.insertAdjacentHTML()
.querySelector()
.closest()
.matches()
.scrollIntoView()
.setAttribute()
```

```
.textContent
.childNodes
.parentNode
.cloneNode()
```

`<p>Paragraph<p>`

`<p>Paragraph<p>`

`<!--Comment-->`

(One different type of `HTMLElement` per HTML element...)

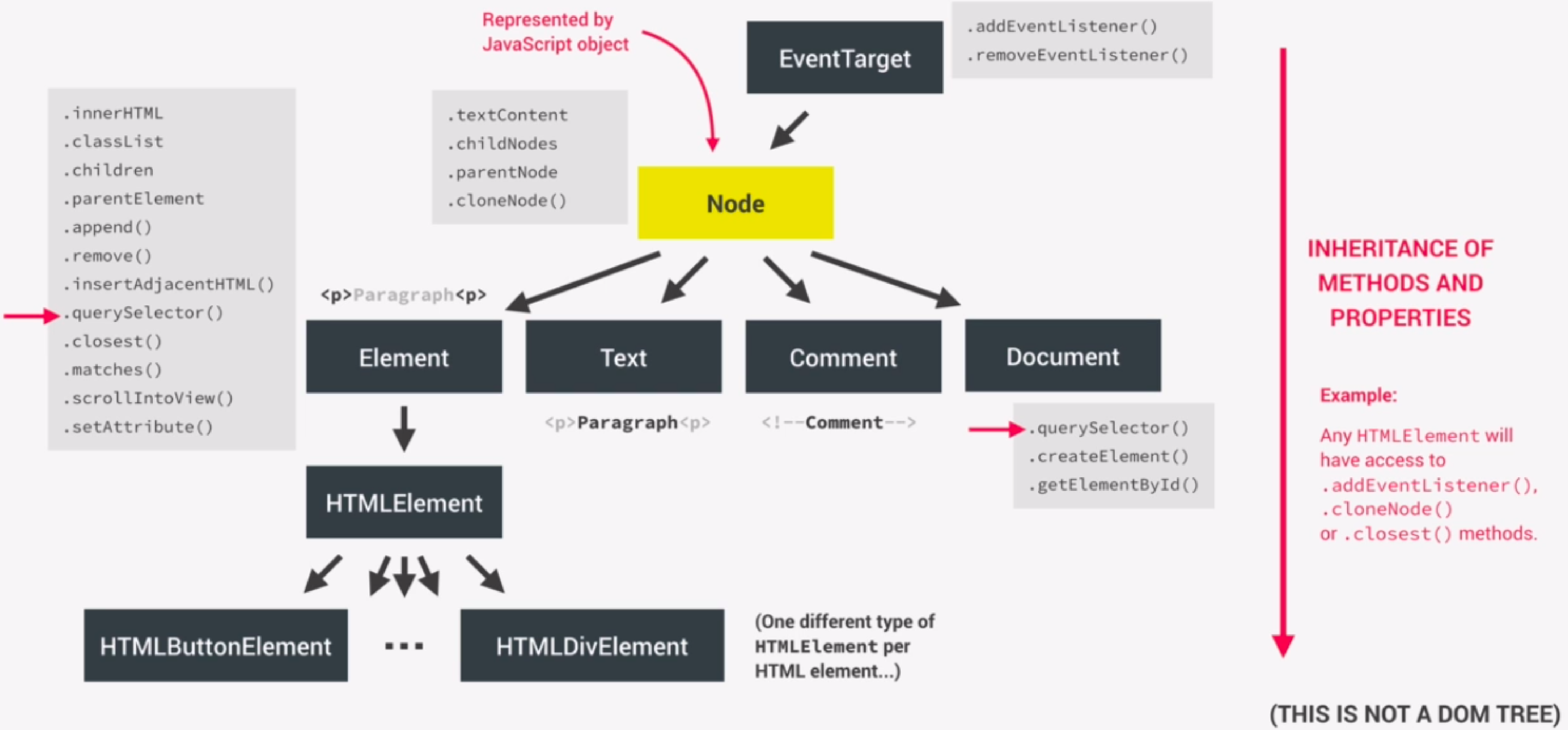
INHERITANCE OF METHODS AND PROPERTIES

Example:

Any `HTMLElement` will have access to `.addEventListener()`, `.cloneNode()` or `.closest()` methods.

(THIS IS NOT A DOM TREE)

HOW THE DOM API IS ORGANIZED BEHIND THE SCENES



Represented by JavaScript object

EventTarget

```
.addEventListener()
.removeEventListener()
```

Node

```
.textContent
.childNodes
.parentNode
.cloneNode()
```

Element

`<p>Paragraph<p>`

```
.innerHTML
.classList
.children
.parentElement
.append()
.remove()
.insertAdjacentHTML()
.querySelector()
.closest()
.matches()
.scrollIntoView()
.setAttribute()
```

Text

`<p>Paragraph<p>`

Comment

`<!--Comment-->`

Document

```
.querySelector()
.createElement()
.getElementById()
```

HTMLElement

HTMLButtonElement ... HTMLDivElement

(One different type of HTMLElement per HTML element...)

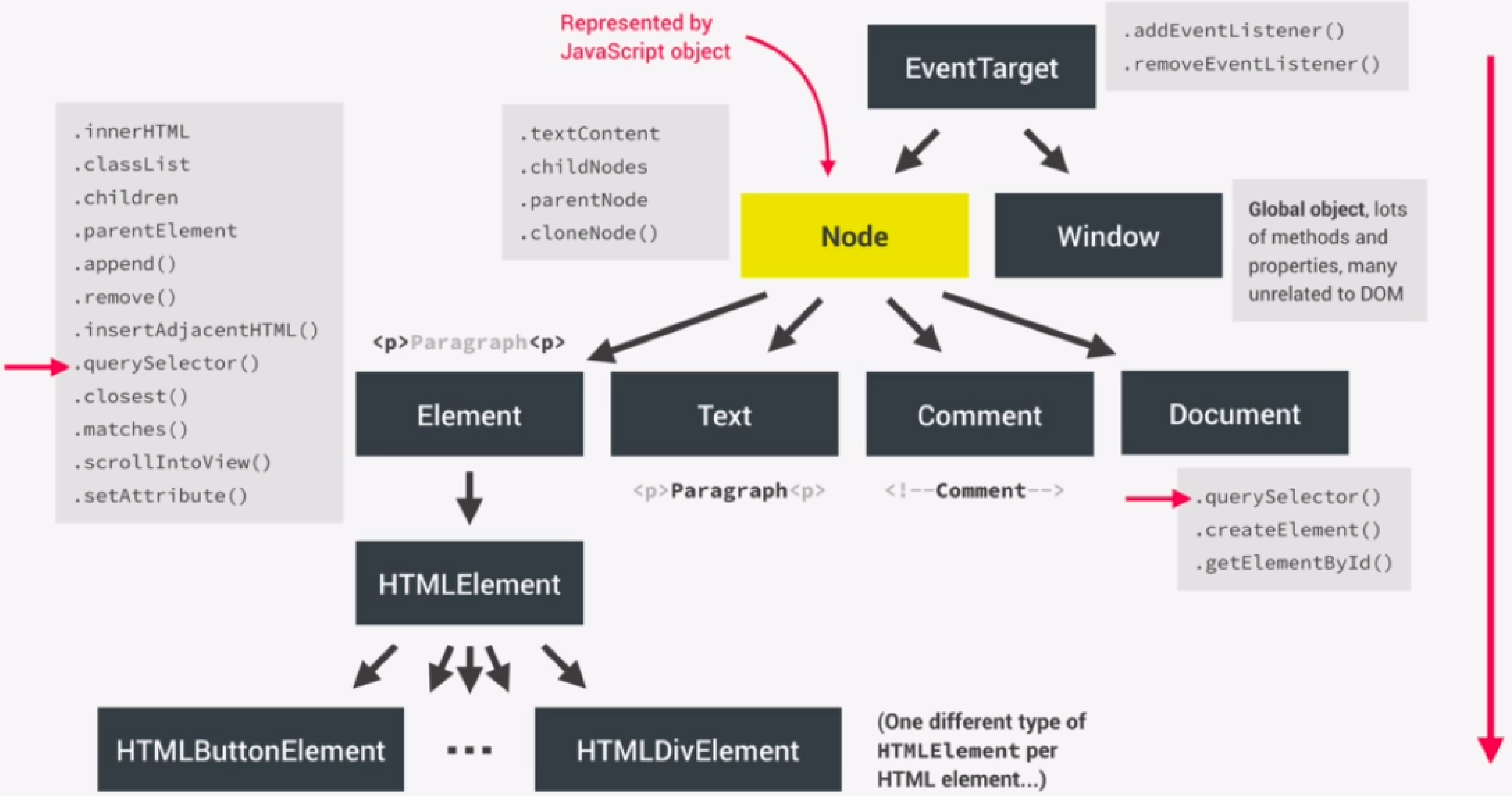
INHERITANCE OF METHODS AND PROPERTIES

Example:

Any HTMLElement will have access to .addEventListener(), .cloneNode() or .closest() methods.

(THIS IS NOT A DOM TREE)

HOW THE DOM API IS ORGANIZED BEHIND THE SCENES



Represented by JavaScript object

EventTarget

.addEventListener()
.removeEventListener()

Node

.textContent
.childNodes
.parentNode
.cloneNode()

Window

Global object, lots of methods and properties, many unrelated to DOM

Element

<p>Paragraph<p>

.innerHTML
.classList
.children
.parentElement
.append()
.remove()
.insertAdjacentHTML()
.querySelector()
.closest()
.matches()
.scrollIntoView()
.setAttribute()

Text

<p>**Paragraph**<p>

Comment

<!--**Comment**-->

Document

.querySelector()
.createElement()
.getElementById()

HTMLElement

HTMLButtonElement

...

HTMLDivElement

(One different type of HTMLElement per HTML element...)

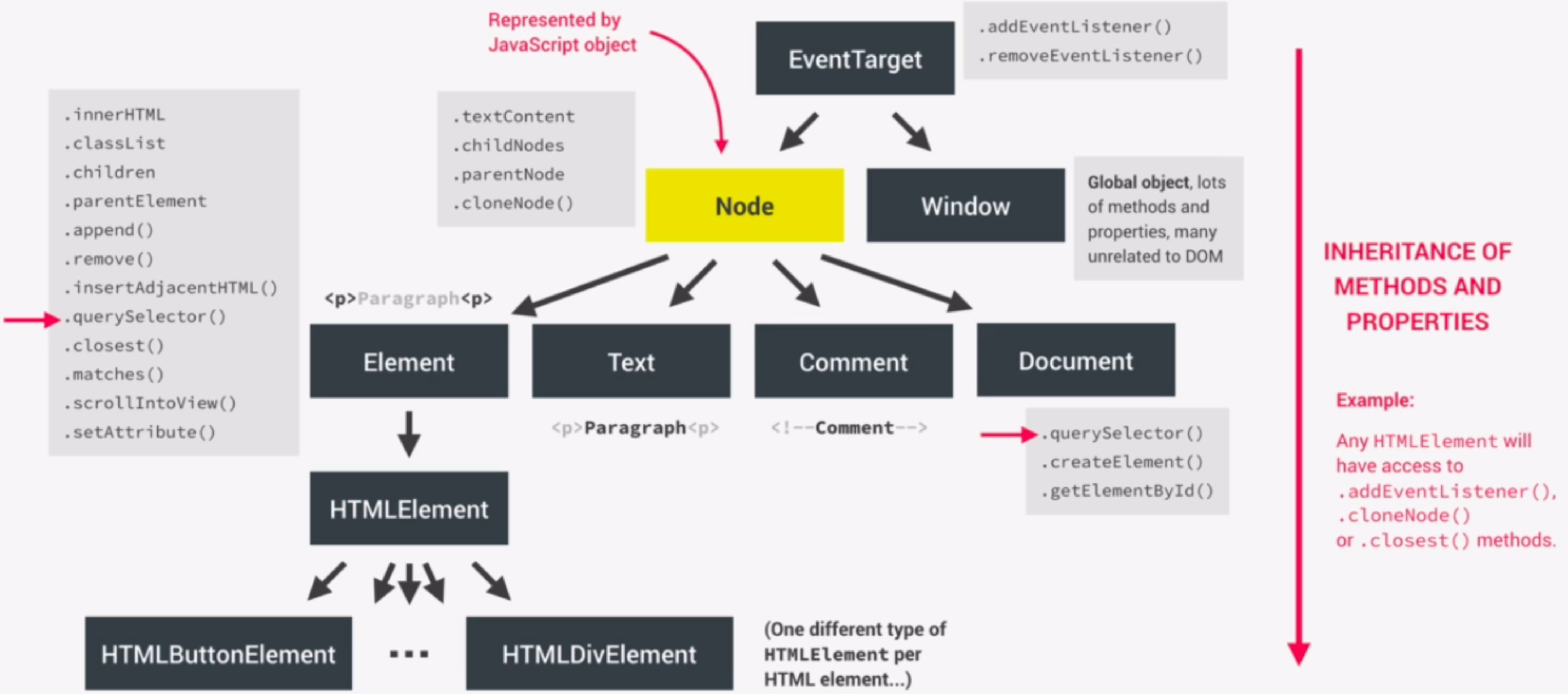
INHERITANCE OF METHODS AND PROPERTIES

Example:

Any HTMLElement will have access to .addEventListener(), .cloneNode() or .closest() methods.

(THIS IS NOT A DOM TREE)

HOW THE DOM API IS ORGANIZED BEHIND THE SCENES



Represented by JavaScript object

EventTarget

.addEventListener()
.removeEventListener()

Node

.textContent
.childNodes
.parentNode
.cloneNode()

Window

Global object, lots of methods and properties, many unrelated to DOM

<p>Paragraph<p>

Element

.innerHTML
.classList
.children
.parentElement
.append()
.remove()
.insertAdjacentHTML()
.querySelector()
.closest()
.matches()
.scrollIntoView()
.setAttribute()

Text

<p>**Paragraph**<p>

Comment

<!--**Comment**-->

Document

.querySelector()
.createElement()
.getElementById()

HTMLElement

HTMLButtonElement

...

HTMLDivElement

(One different type of HTMLElement per HTML element...)

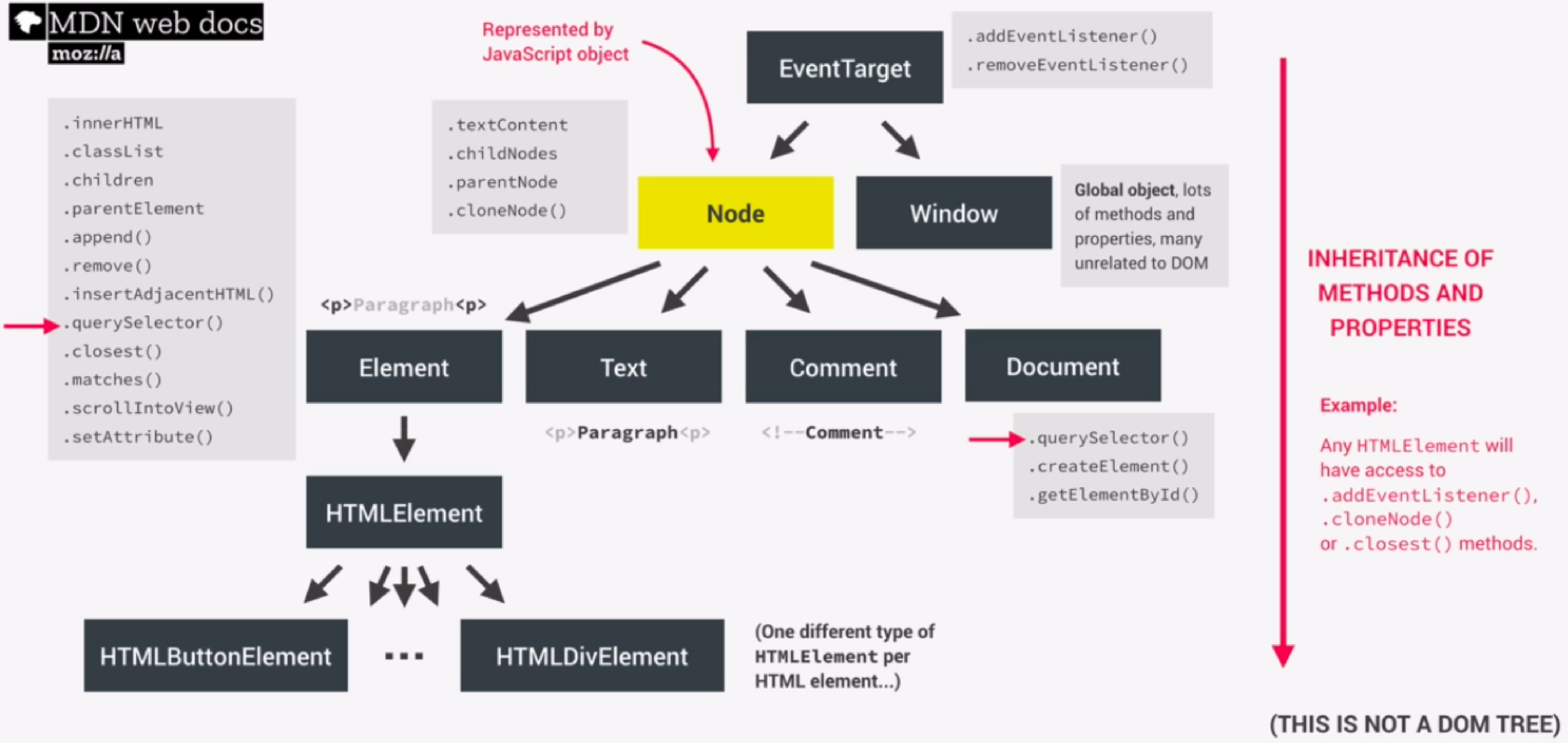
INHERITANCE OF METHODS AND PROPERTIES

Example:

Any HTMLElement will have access to .addEventListener(), .cloneNode() or .closest() methods.

(THIS IS NOT A DOM TREE)

HOW THE DOM API IS ORGANIZED BEHIND THE SCENES



MDN web docs
moz://a

.innerHTML
.classList
.children
.parentElement
.append()
.remove()
.insertAdjacentHTML()
.querySelector()
.closest()
.matches()
.scrollIntoView()
.setAttribute()

Represented by JavaScript object

EventTarget

.addEventListener()
.removeEventListener()

.textContent
.childNodes
.parentNode
.cloneNode()

Node

Window

Global object, lots of methods and properties, many unrelated to DOM

<p>Paragraph<p>

Element

Text

Comment

Document

<p>Paragraph<p>

<!--Comment-->

.querySelector()
.createElement()
.getElementById()

HTMLElement

HTMLButtonElement

...

HTMLDivElement

(One different type of HTMLElement per HTML element...)

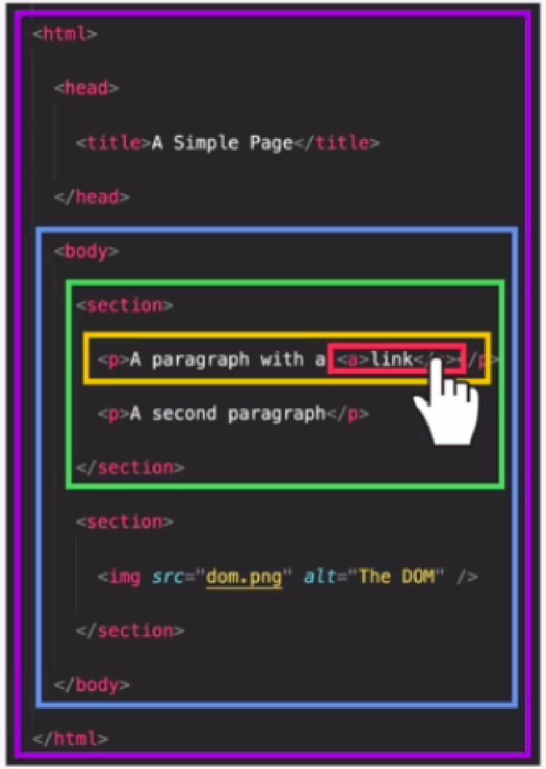
INHERITANCE OF METHODS AND PROPERTIES

Example:

Any HTMLElement will have access to .addEventListener(), .cloneNode() or .closest() methods.

(THIS IS NOT A DOM TREE)

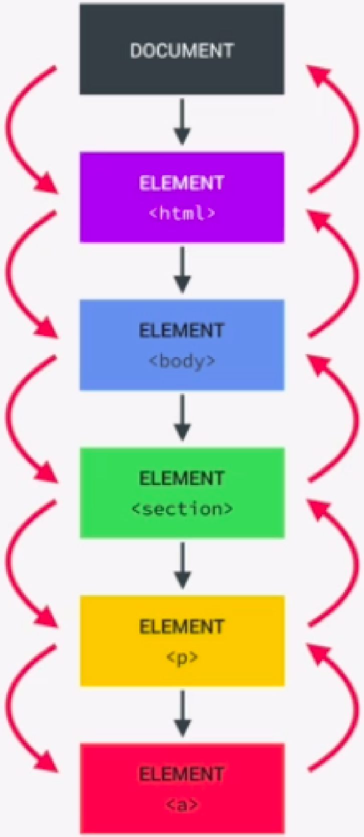
BUBBLING AND CAPTURING



```
<html>

  <head>

    <title>A Simple Page</title>

  </head>

  <body>

    <section>

      <p>A paragraph with a <a>link</a></p>

      <p>A second paragraph</p>

    </section>

    <section>

      <img src="dom.png" alt="The DOM" />

    </section>

  </body>

</html>
```

1

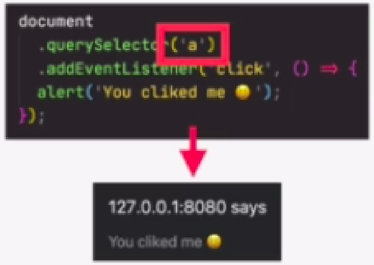
CAPTURING PHASE



Click event

3

BUBBLING PHASE



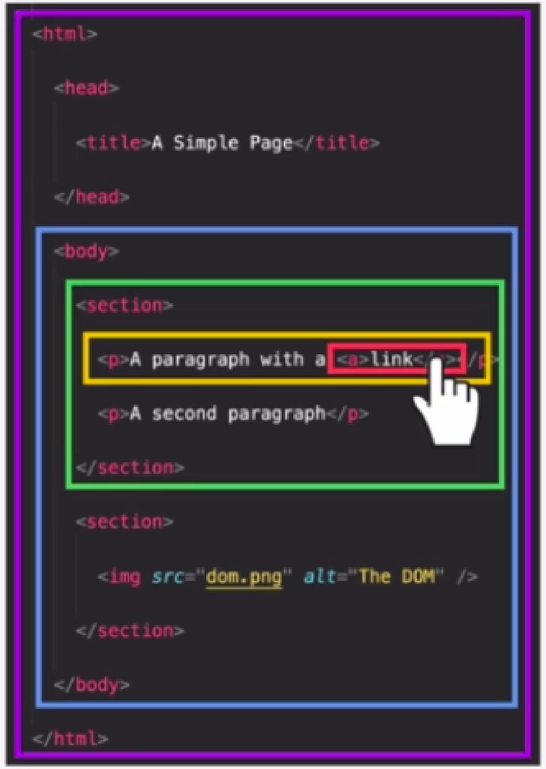
```
document
  .querySelector('a')
  .addEventListener('click', () => {
  alert('You cliked me 😀');
});
```

```
127.0.0.1:8080 says
You cliked me 😀
```

2

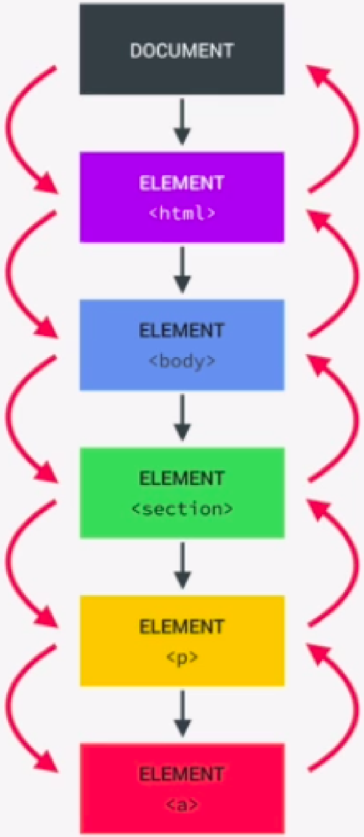
TARGET PHASE

BUBBLING AND CAPTURING



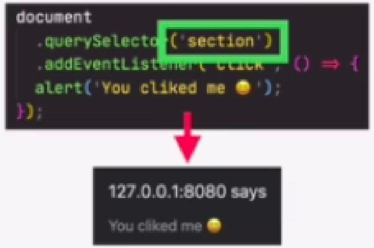
```
<html>

  <head>

    <title>A Simple Page</title>

  </head>

  <body>

    <section>

      <p>A paragraph with a <a>link</a></p>

      <p>A second paragraph</p>

    </section>

    <section>

      <img src="dom.png" alt="The DOM" />

    </section>

  </body>

</html>
```

1 CAPTURING PHASE



Click event

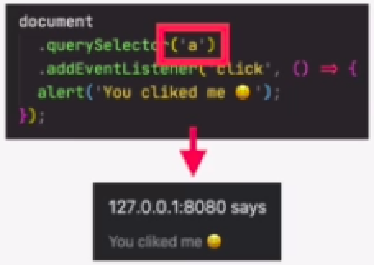
3 BUBBLING PHASE



```
document
  .querySelector('section')
  .addEventListener('click', () => {
  alert('You cliked me 😀');
});
```

127.0.0.1:8080 says

You cliked me 😀



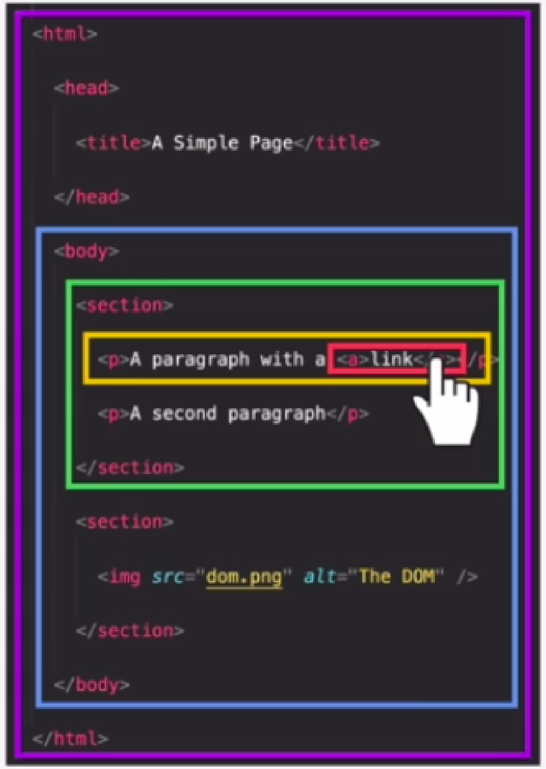
```
document
  .querySelector('a')
  .addEventListener('click', () => {
  alert('You cliked me 😀');
});
```

127.0.0.1:8080 says

You cliked me 😀

2 TARGET PHASE

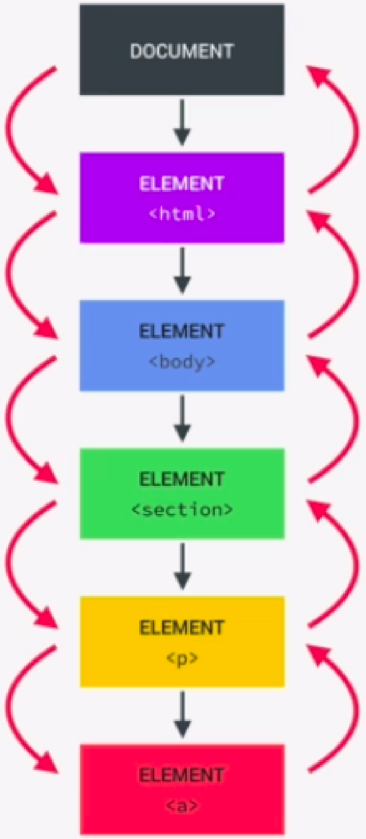
BUBBLING AND CAPTURING



```
<html>
  <head>
    <title>A Simple Page</title>
  </head>
  <body>
    <section>
      <p>A paragraph with a <a>link</a></p>
      <p>A second paragraph</p>
    </section>
    <section>
      <img src="dom.png" alt="The DOM" />
    </section>
  </body>
</html>
```

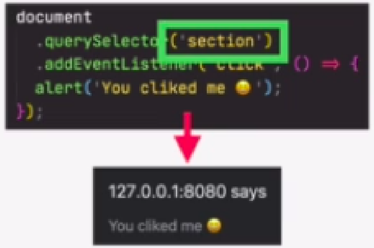
(THIS DOES NOT HAPPEN ON *ALL* EVENTS)

1 CAPTURING PHASE



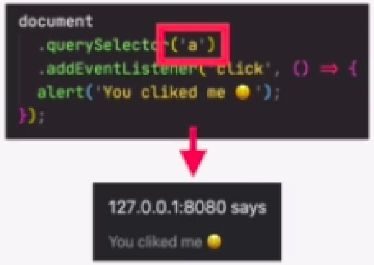
Click event

3 BUBBLING PHASE



```
document
  .querySelector('section')
  .addEventListener('click', () => {
  alert('You cliked me 😀');
});
```

```
127.0.0.1:8080 says
You cliked me 😀
```



```
document
  .querySelector('a')
  .addEventListener('click', () => {
  alert('You cliked me 😀');
});
```

```
127.0.0.1:8080 says
You cliked me 😀
```

2 TARGET PHASE