



UNIVERSITÀ
DEGLI STUDI
DI PALERMO



Document Object Model

Corso di Programmazione Web e Mobile
a.a. 2021/2022

Prof. Roberto Pirrone

Sommario

- Struttura di un documento HTML (XML)
 - Nodi
 - Elementi
 - Oggetto document
- Metodi per ricercare gli elementi
- Metodi per ricercare/creare/rimuovere attributi
- Metodi per creare o rimuovere nodi/elementi
- Gestione dello stile
 - Proprietà style
 - ricerca di elementi sulla base di selettori CSS
- Gestione degli eventi

Struttura di un documento HTML (XML)

- Il Document Object Model (DOM) è una API definita per navigare e manipolare i documenti costruiti con linguaggi di marcatura XML o HTML accedendo alle proprietà dell'oggetto document
- Utilizza una struttura ad albero per gestire le relazioni di inclusione tra gli elementi del documento
- Consente di:
 - Accedere ad un nodo e gestirne gli ascendenti, discendenti e fratelli
 - Accedere all'elemento definito nel nodo con le sue proprietà e attributi

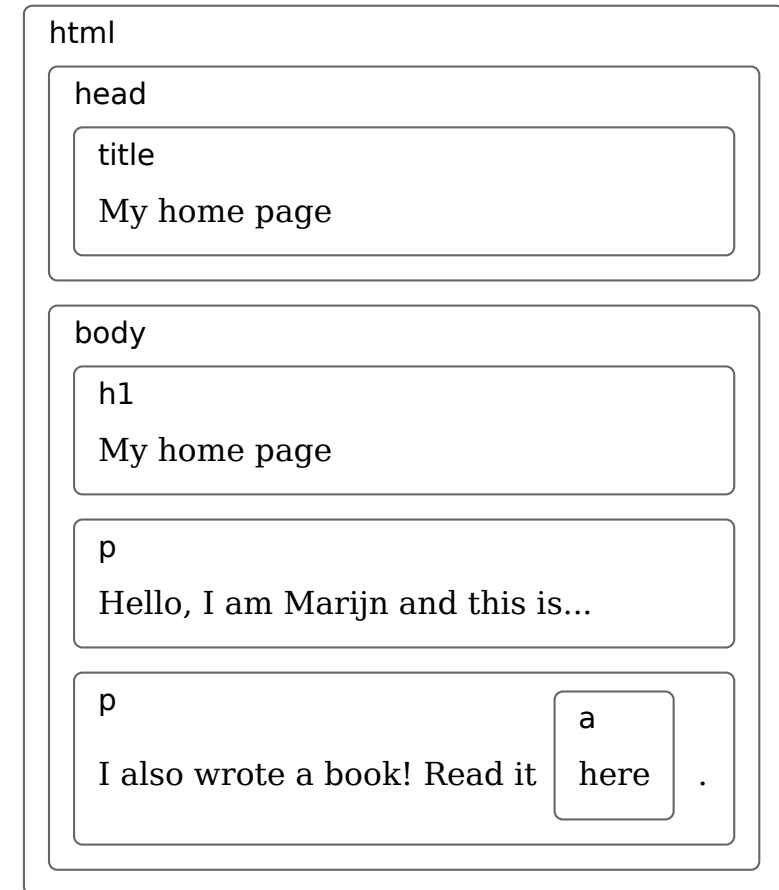
Struttura di un documento HTML (XML)

```
<!DOCTYPE html>
<html>

<head>
<title>My home page</title>
</head>

<body>
<h1>My home page</h1>
<p>Hello, I am Marijn and this is my home page.</p>
<p>I also wrote a book! Read it
<a href="http://eloquentjavascript.net">here</a>.</p>
</body>

</html>
```



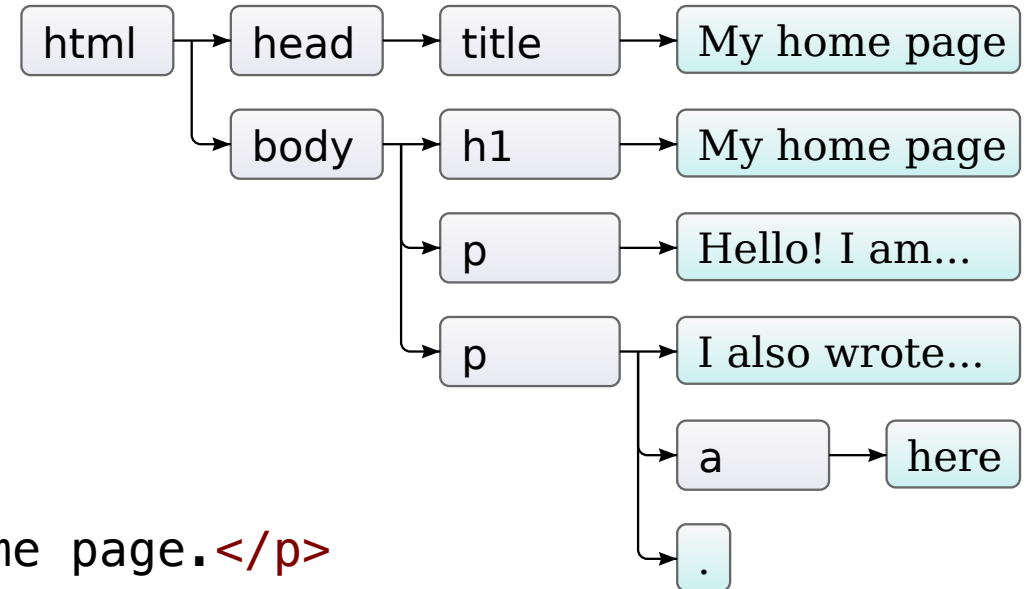
Struttura di un documento HTML (XML)

```
<!DOCTYPE html>
<html>

<head>
<title>My home page</title>
</head>

<body>
<h1>My home page</h1>
<p>Hello, I am Marijn and this is my home page.</p>
<p>I also wrote a book! Read it
<a href="http://eloquentjavascript.net">here</a>.</p>
</body>

</html>
```



Struttura di un documento HTML (XML)

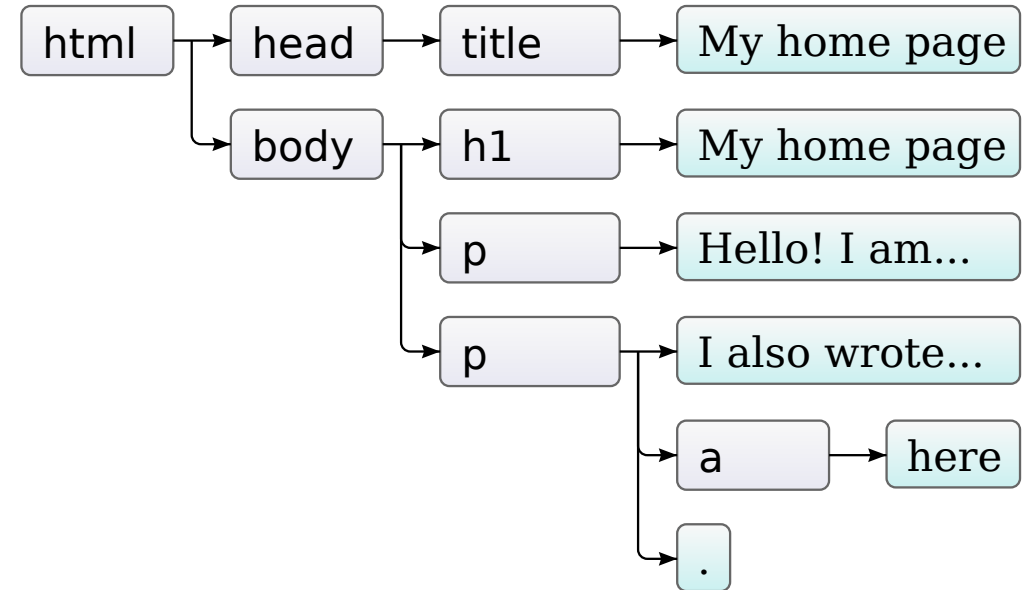
- Nel DOM sono definite due classi principali
 - Node: definisce le proprietà di in singolo nodo della struttura ad albero che rappresenta il documento
 - Element: definisce le proprietà di un elemento di marcatura, quali il suo nome, gli attributi e il testo contenuto
- Ogni oggetto di tipo Element è anche un oggetto di tipo Node, ma nell'albero ci sono anche *nodi che non sono elementi*

Struttura di un documento HTML (XML)

Tipi di nodo definiti nella classe Node:

Node.DOCUMENT_NODE → 9
Node.ELEMENT_NODE → 1
Node.ATTRIBUTE_NODE → 2
Node.TEXT_NODE → 8
Node.COMMENT_NODE → 3

...



Struttura di un documento HTML (XML)

```
<!DOCTYPE html>  
<html>
```

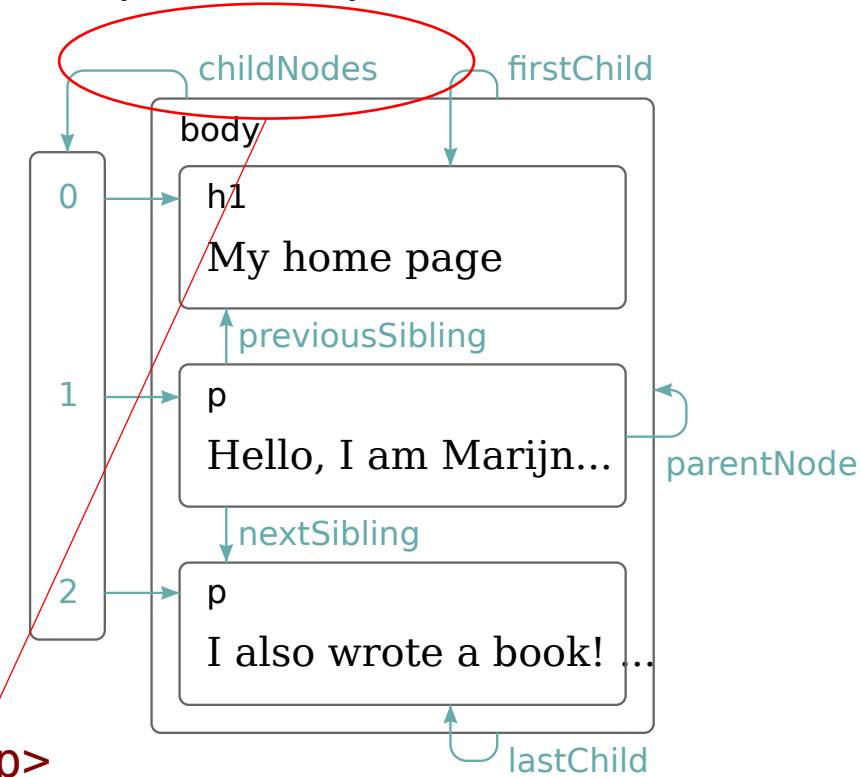
Alcune proprietà definite nella classe Node
per la navigazione tra nodi vicini

```
<head>  
<title>My home page</title>  
</head>
```

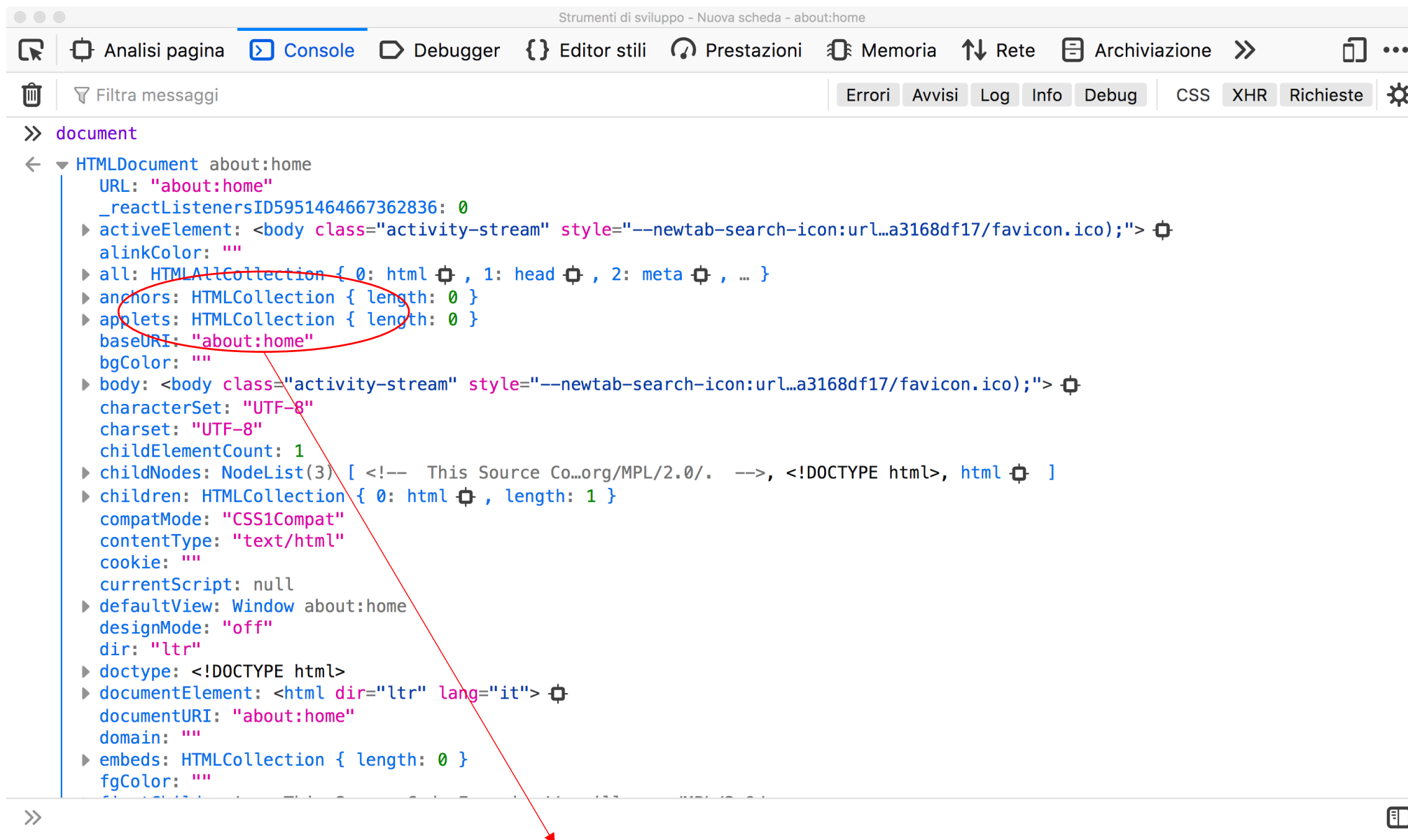
```
<body>  
<h1>My home page</h1>  
<p>Hello, I am Marijn and this is my home page.</p>  
<p>I also wrote a book! Read it  
<a href="http://eloquentjavascript.net">here</a>.</p>  
</body>
```

```
</html>
```

childNodes è una NodeList ovvero un
iterabile con proprietà length



DOM



Strumenti di sviluppo - Nuova scheda - about:home

Analisi pagina Console Debugger Editor stili Prestazioni Memoria Rete Archiviazione

Filtra messaggi Errori Avvisi Log Info Debug CSS XHR Richieste

```
>> document
< HTMLDocument about:home
  URL: "about:home"
  _reactListenersID5951464667362836: 0
  ▶ activeElement: <body class="activity-stream" style="--newtab-search-icon:url...a3168df17/favicon.ico);">
  alinkColor: ""
  ▶ all: HTMLAllCollection { 0: html , 1: head , 2: meta , ... }
  ▶ anchors: HTMLCollection { length: 0 }
  ▶ applets: HTMLCollection { length: 0 }
  baseURI: "about:home"
  bgColor: ""
  ▶ body: <body class="activity-stream" style="--newtab-search-icon:url...a3168df17/favicon.ico);">
  characterSet: "UTF-8"
  charset: "UTF-8"
  childElementCount: 1
  ▶ childNodes: NodeList(3) [ <!-- This Source Co...org/MPL/2.0/. -->, <!DOCTYPE html>, html ]
  ▶ children: HTMLCollection { 0: html , length: 1 }
  compatMode: "CSS1Compat"
  contentType: "text/html"
  cookie: ""
  currentScript: null
  ▶ defaultView: Window about:home
  designMode: "off"
  dir: "ltr"
  ▶ doctype: <!DOCTYPE html>
  ▶ documentElement: <html dir="ltr" lang="it">
  documentURI: "about:home"
  domain: ""
  ▶ embeds: HTMLCollection { length: 0 }
  fgColor: ""
```

Anche `HTMLCollection` è un iterabile con proprietà `length`, ma itera su *elementi*

DOM

Collezione delle form
del documento

Collezione delle
immagini del
documento

Collezione dei link
ipertestuali del
documento

Tipo del nodo

Strumenti di sviluppo - Nuova scheda - about:home

Analisi pagina Console Debugger Editor stili Prestazioni Memoria Rete Archiviazione

Filtra messaggi

Errori Avvisi Log Info Debug CSS XHR Richieste

```
> firstChild: <!-- This Source Code Form is...//mozilla.org/MPL/2.0/. -->
> firstElementChild: <html dir="ltr" lang="it">
> fonts: FontFaceSet { size: 0, onloading: null, status: "loaded", ... }
> forms: HTMLCollection { length: 0 }
fullscreen: false
fullscreenElement: null
fullscreenEnabled: true
> head: <head>
hidden: false
> images: HTMLCollection { 0: img , length: 1 }
> implementation: DOMImplementation { }
inputEncoding: "UTF-8"
isConnected: true
> l10n: DocumentL10n { ready: Promise { "fulfilled" } }
> lastChild: <html dir="ltr" lang="it">
> lastElementChild: <html dir="ltr" lang="it">
lastModified: "05/06/2020 11:31:45"
lastStyleSheetSet: null
linkColor: ""
> links: HTMLCollection { 0: a.top-site-button , 1: a.top-site-button , 2: a.top-site-button , ... }
location: Location about:home
mozFullScreen: false
mozFullScreenElement: null
mozFullScreenEnabled: true
nextSibling: null
nodeName: "#document"
nodeType: 9
nodeValue: null
onabort: null
onafterscriptexecute: null
onanimationcancel: null
```

DOM

Lista degli attributi di gestione degli eventi

Stato della risposta HTTP

Collezione degli script del documento

Lista dei fogli di stile

Strumenti di sviluppo - Nuova scheda - about:home

Analisi pagina Console Debugger Editor stili Prestazioni Memoria Rete Archiviazione

Filtra messaggi

Errori Avvisi Log Info Debug CSS XHR Richieste

```
onsuspend: null
ontimeupdate: null
ontoggle: null
ontransitioncancel: null
ontransitionend: null
ontransitionrun: null
ontransitionstart: null
onvisibilitychange: null
onvolumechange: null
onwaiting: null
onwebkitanimationend: null
onwebkitanimationiteration: null
onwebkitanimationstart: null
onwebkittransitionend: null
onwheel: null
ownerDocument: null
parentElement: null
parentNode: null
▶ plugins: HTMLCollection { length: 0 }
pointerLockElement: null
preferredStyleSheetSet: ""
previousSibling: null
readyState: "complete"
referrer: ""
rootElement: null
▶ scripts: HTMLCollection { 0: script , 1: script , 2: script , ... }
▶ scrollingElement: <html dir="ltr" lang="it">
selectedStyleSheetSet: null
▶ styleSheetSets: DOMStringList []
▶ styleSheets: StyleSheetList(5) [ StyleSheet chrome://browser/content/contentSearchUI.css, StyleSheet resource://activity-stream/css/activity-stream.css, StyleSheet , ... ]
```

DOM

Strumenti di sviluppo - Nuova scheda - about:home

Analisi pagina Console Debugger Editor stili Prestazioni Memoria Rete Archiviazione

Filtra messaggi

document.body

body.activity-stream

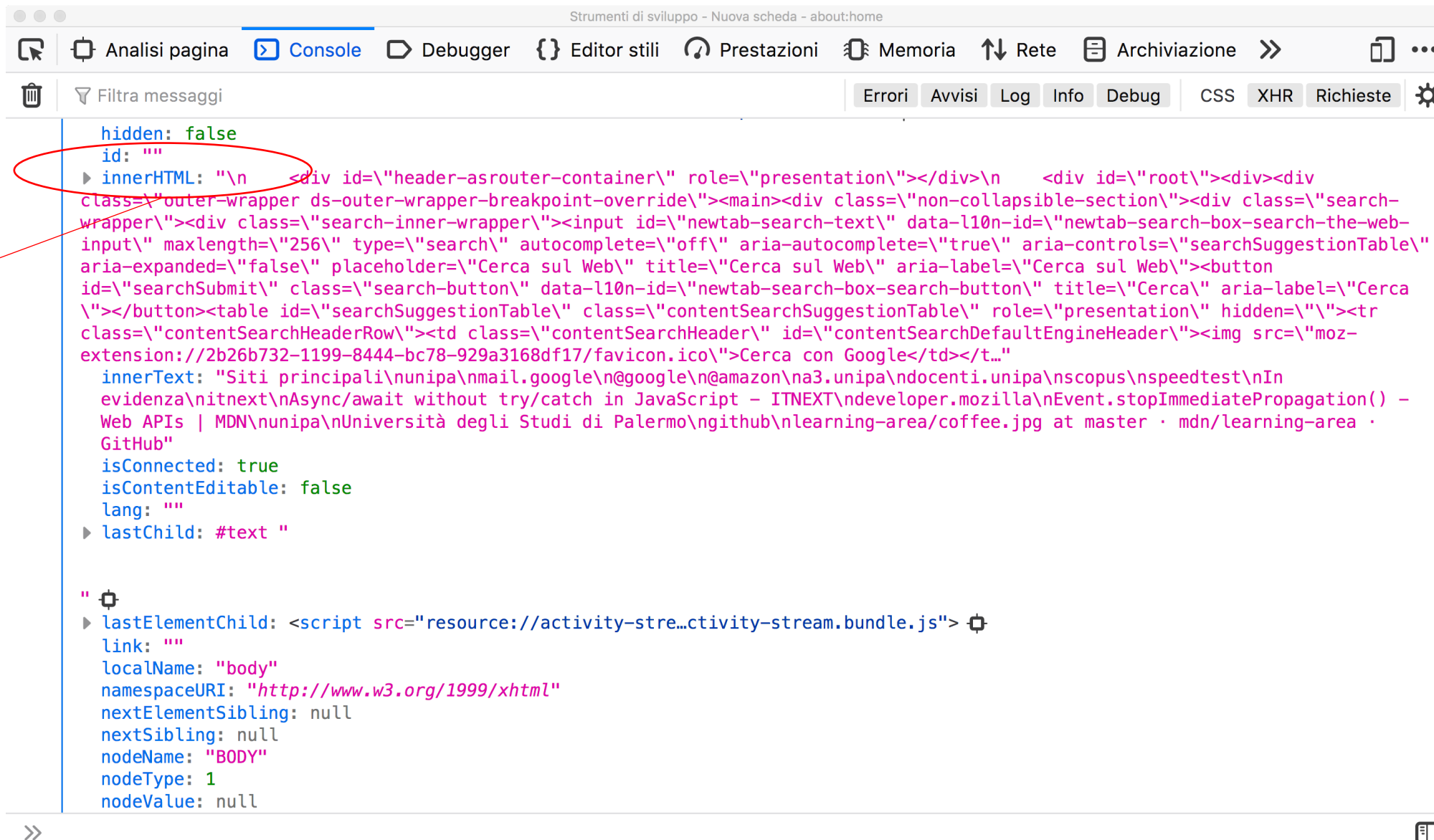
- aLink: ""
- accessKey: ""
- accessKeyLabel: ""
- assignedSlot: null
- attributes: NamedNodeMap [class="activity-stream", style="--newtab-search-icon:url(moz-extension://2b26b732-1199-8444-bc78-929a3168df17/favicon.ico);"]
- background: ""
- baseURI: "about:home"
- bgColor: ""
- childElementCount: 13
- childNodes: NodeList(27) [#text, div#header-asrouter-container, #text, ...]
- children: HTMLCollection { 0: div#header-asrouter-container, 1: div#root, 2: div#footer-asrouter-container, ... }
- classList: DOMTokenList ["activity-stream"]
- className: "activity-stream"
- clientHeight: 781
- clientLeft: 0
- clientTop: 0
- clientWidth: 1440
- contentEditable: "inherit"
- contextMenu: null
- dataset: DOMStringMap(0)
- dir: ""
- draggable: false
- firstChild: #text "
- firstElementChild: <div id="header-asrouter-container" role="presentation">
- hidden: false
- id: ""

Lista degli attributi
dell'elemento

Lista delle classi
dell'elemento e
nome completo
della(e) classe(i) di
appartenenza

DOM

id e contenuto
testuale
dell'elemento



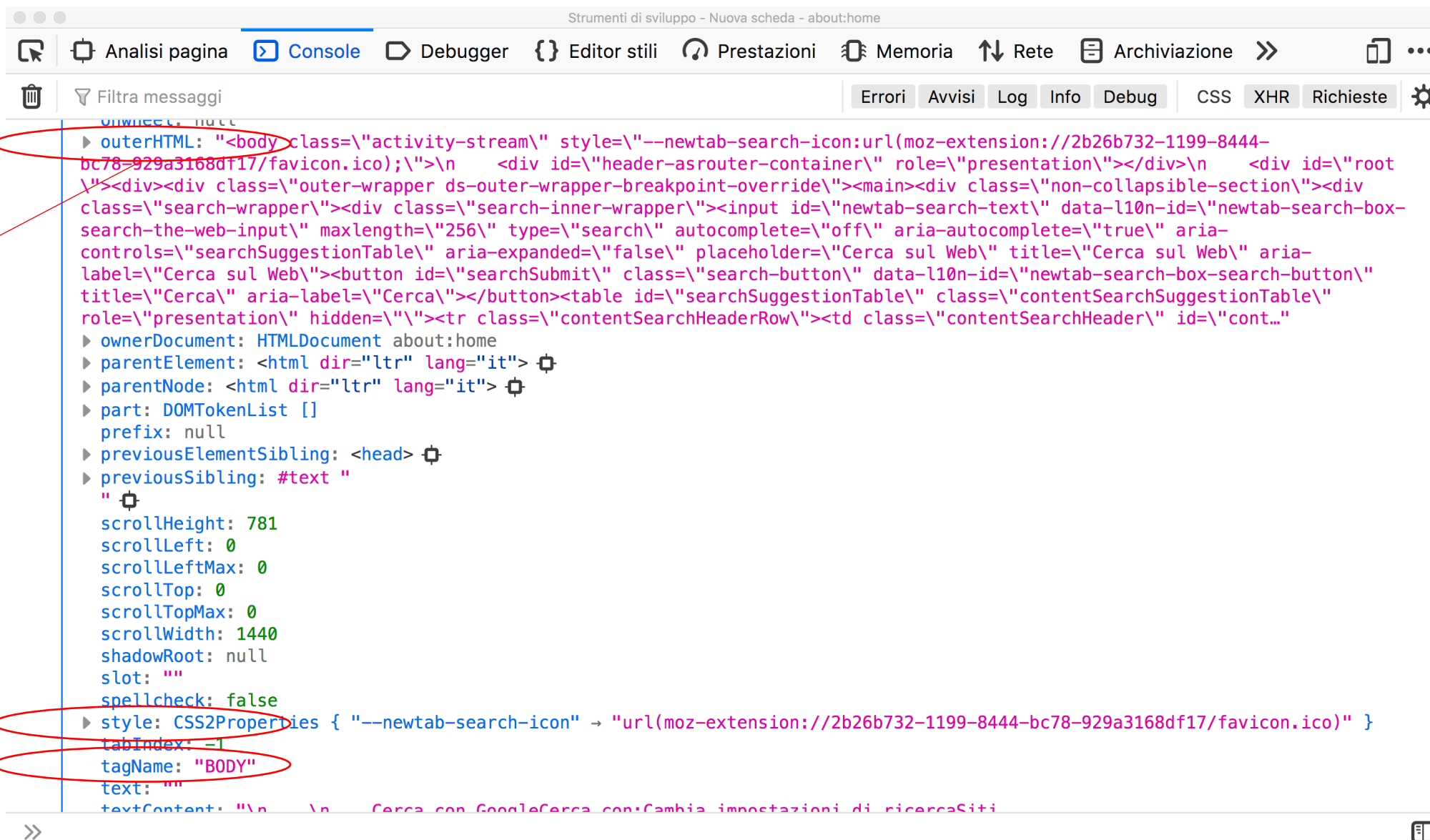
```
hidden: false
id: ""
▶ innerHTML: "\n <div id=\"header-asrouter-container\" role=\"presentation\"></div>\n <div id=\"root\"><div><div
class=\"outer-wrapper ds-outer-wrapper-breakpoint-override\"><main><div class=\"non-collapsible-section\"><div class=\"search-
wrapper\"><div class=\"search-inner-wrapper\"><input id=\"newtab-search-text\" data-l10n-id=\"newtab-search-box-search-the-web-
input\" maxlength=\"256\" type=\"search\" autocomplete=\"off\" aria-autocomplete=\"true\" aria-controls=\"searchSuggestionTable\"
aria-expanded=\"false\" placeholder=\"Cerca sul Web\" title=\"Cerca sul Web\" aria-label=\"Cerca sul Web\"><button
id=\"searchSubmit\" class=\"search-button\" data-l10n-id=\"newtab-search-box-search-button\" title=\"Cerca\" aria-label=\"Cerca
\"></button><table id=\"searchSuggestionTable\" class=\"contentSearchSuggestionTable\" role=\"presentation\" hidden=\"\"><tr
class=\"contentSearchHeaderRow\"><td class=\"contentSearchHeader\" id=\"contentSearchDefaultEngineHeader\"><img src=\"moz-
extension://2b26b732-1199-8444-bc78-929a3168df17/favicon.ico\">Cerca con Google</td></tr>
  innerText: "Siti principali\nunipa\nmail.google\n@google\n@amazon\nna3.unipa\ndocenti.unipa\nscopus\nspeedtest\nIn
evidenza\nitnext\nAsync/await without try/catch in JavaScript - ITNEXT\ndeveloper.mozilla\nEvent.stopImmediatePropagation() -
Web APIs | MDN\nunipa\nUniversità degli Studi di Palermo\ngithub\nlearning-area/coffee.jpg at master · mdn/learning-area ·
GitHub"
  isConnected: true
  isContentEditable: false
  lang: ""
  ▶ lastChild: #text "
```

DOM

Testo HTML
all'esterno
dell'elemento

Stile dell'elemento

Nome del tag

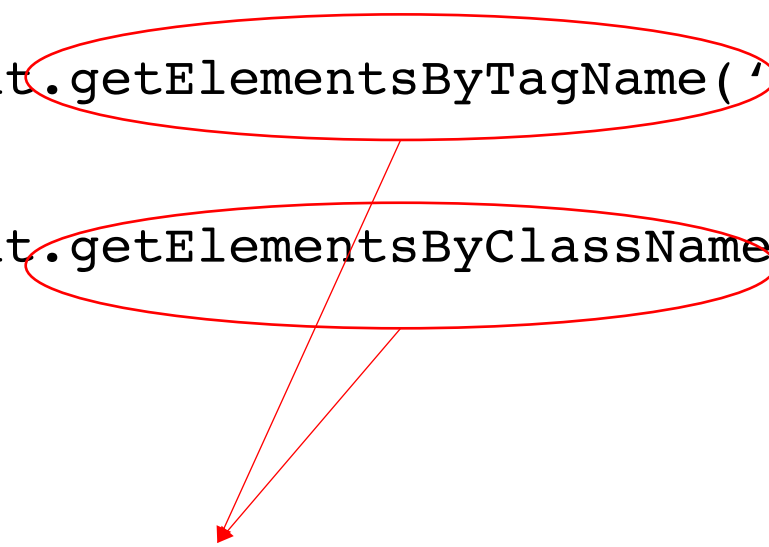


Metodi per ricercare gli elementi

```
let elemento = document.getElementById( 'id' )
```

```
let elemento = document.getElementsByTagName( 'nomeTag' ) [ indice ]
```

```
let elemento = document.getElementsByClassName( 'nomeClasse' ) [ indice ]
```



Restituiscono una `HTMLCollection`

Metodi per ricercare/creare/rimuovere attributi

```
let valoreAttributo = elemento.getAttribute( 'nomeAttributo' )
```

```
elemento.setAttribute( 'nomeAttributo', 'valoreAttributo' )
```

```
elemento.removeAttribute( 'nomeAttributo' )
```


Metodi per creare o rimuovere nodi/elementi

```
let nuovoElemento = document.createElement( 'nomeElemento' )
```

```
elmGenitore.insertBefore(nuovoElemento, elmDiRiferimento)
```

```
elmGenitore.append(... nuoviElementi) // appende in coda ai fratelli
```

```
elmGenitore.appendChild(nuovoElemento) // in coda ai fratelli
```

- L'elemento creato *non è direttamente inserito nell'albero* se non è invocato `insertBefore`, `append` o `appendChild`

Metodi per creare o rimuovere nodi/elementi

elemento.removeChild(*elementoFiglio*)

- Come consentire ad un elemento di «auto-rimuoversi»

elemFiglio.parentElement.removeChild(*elemFiglio*)

Gestione dello stile

- La proprietà `style` di un elemento è un oggetto che accede a tutte le proprietà CSS dell'elemento
- Essa esplicitamente scrive il contenuto dell'attributo `style` dell'elemento stesso
- Le proprietà dell'oggetto `style` hanno gli stessi nomi delle corrispondenti CSS, ma con le iniziali maiuscole

elemento.style.borderLeftColor

elemento.style.fontFamily

...

Gestione dello stile

- I selettori CSS hanno elevato potere discriminativo e possono essere usati in Javascript per selezionare singoli elementi o gruppi

elemento.querySelector('*selettoreCSS*')

elemento.querySelectorAll('*selettoreCSS*')

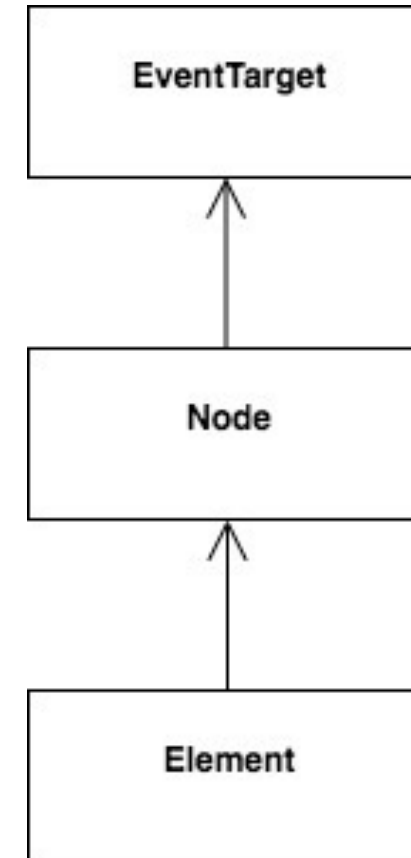
Restituisce il primo discendente di *elemento*
che soddisfa il selettore

Restituisce tutti i discendenti di *elemento*
che soddisfano il selettore

document.querySelector() estende la ricerca a tutto il documento

Gestione degli eventi

- L'esecuzione di Javascript sul front-end di una web application è legata alla gestione degli eventi
- Un evento si scatena su un `EventTarget` che può essere un nodo o un elemento



Gestione degli eventi

- Gli eventi Javascript sono codificati attraverso la classe `Event` e sono associabili a diversi ambiti:
 - Caricamento delle risorse (`load`, `unload`, `error`, `abort`, ...)
 - Eventi di rete (`online`, `offline`)
 - Focus ad un elemento (`focus`, `blur`, ...)
 - Eventi di Form (`reset`, `submit`)
 - Eventi di inserimento dati (`change`, `storage`)

Gestione degli eventi

- Gli eventi Javascript sono codificati attraverso la classe `Event` e sono associabili a diversi ambiti:
 - Eventi di mouse (`click`, `rightclick`, `contextmenu`, `mouseup`, `mousedown`, ...)
 - Eventi di tastiera (`keypress`, `keyup`, `keydown`)
 - Eventi di clipboard (`copy`, `cut`, `paste`)
 - Eventi di view (`resize`, `scroll`, ...)
 - Eventi di Websocket, di history, CSS

Gestione degli eventi

- Gli eventi sono gestiti attraverso la registrazione di una funzione di callback che è detta appunto *gestore dell'evento*
- Gestione diretta HTML:

```
<nomeTag onnomeEvento= ' codiceGestioneEvento '
... >...</nomeTag>
```


Gestione degli eventi

- Gestione interna Javascript:

```
elemento.onnomeEvento = function(e) {  
    codice gestore evento e  
}
```

```
elemento.addEventListener( 'nomeEvento', function(e) {  
    codice gestore evento e  
} ) ;
```

Gestione degli eventi

- Più di un gestore può essere registrato per lo stesso evento e per lo stesso elemento
- Un evento associato ad un certo *elemento può avere un comportamento di default* e il gestore dell'evento viene eseguito *prima* del comportamento di default
- Un evento *si propaga* (bubbling) dall'elemento in cui si verifica lungo la gerarchia dei suoi ascendenti *e innesca via via l'esecuzione di tutti i gestori per esso registrati negli ascendenti*

Gestione degli eventi

- Inibizione del comportamento di default (all'interno di un gestore di evento:

```
elemento.addEventListener( 'nomeEvento', function(e) {  
    codice gestore evento e  
    e.preventDefault();  
} );
```

Gestione degli eventi

- Interruzione della propagazione (all'interno di un gestore di evento:

```
elemento.addEventListener( 'nomeEvento', function(e) {  
    codice gestore evento e  
    e.stopPropagation();  
    // e.stopImmediatePropagation();  
    // blocca anche l'esecuzione degli altri  
    // gestori registrati per l'evento  
} );
```