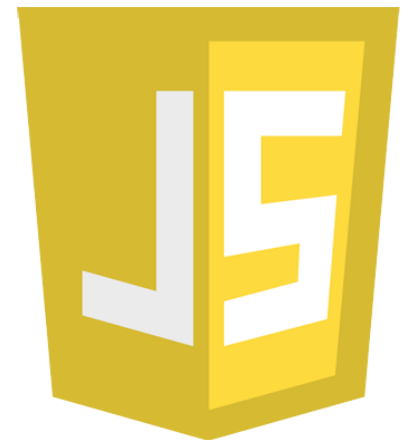


JAVASCRIPT

Riccardo Cattaneo

Lezione 1



Che cos'è Javascript

Javascript è un linguaggio di programmazione orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione **web lato client** (**esteso poi anche al lato server**) per la creazione, nei siti web e applicazioni web, di effetti dinamici interattivi tramite funzioni di script invocate da eventi innescati a loro volta in vari modi dall'utente sulla pagina web in uso (mouse, tastiera ecc...).

Originariamente sviluppato da Netscape con il nome di LiveScript, in seguito è stato rinominato "Javascript" ed è stato Standardizzato per la prima volta nel 1997 dalla ECMA con il nome **ECMAScript**.

ECMAScript

ECMAScript (o ES) è la specifica tecnica di un linguaggio di scripting, standardizzata e mantenuta da ECMA International.

Le implementazioni più conosciute di questo linguaggio (spesso definite come dialetti) sono **Javascript**, JScript e Actionscript che sono entrati largamente in uso, inizialmente, come linguaggi lato client nello sviluppo web.

La sesta edizione, conosciuta inizialmente come ECMAScript (ES6) ma rinominata successivamente **ECMAScript 2015** fu pronta nel 2015. Questa edizione portò importanti ampliamenti alla sintassi, pensati per rendere più agevole la scrittura di applicazioni complesse, tra cui le nuove dichiarazioni delle classi e i moduli attraverso le istruzioni import e export.

Alcune altre novità sintattiche furono il nuovo iteratore for ... of, le funzioni a freccia (() => {...}), le parole chiave let e const per dichiarare le variabili, array tipizzati, nuove collections, promises ecc.

GLI STRUMENTI DI LAVORO

Per iniziare a lavorare con Javascript c'è bisogno di 2 strumenti :

1. Un editor di testo
2. Un debugger

Visual Studio Code

Visual Studio code è l'editor di testo che useremo per questo corso. E' un editor adatto per scrivere codice di molti linguaggi, tra cui il nostro Javascript.

Per scaricarlo andiamo su google e cerchiamo Visual Studio Code, andiamo sul sito ufficiale e procediamo con il download e con l'installazione del programma (disponibile per Windows, Linux e Mac) :



Visual Studio Code

[Docs](#)

[Updates](#)

[Blog](#)

[API](#)

[Extensions](#)

[FAQ](#)

[Learn](#)



Search

Code editing. Redefined.

Free. Built on open source. Runs everywhere.

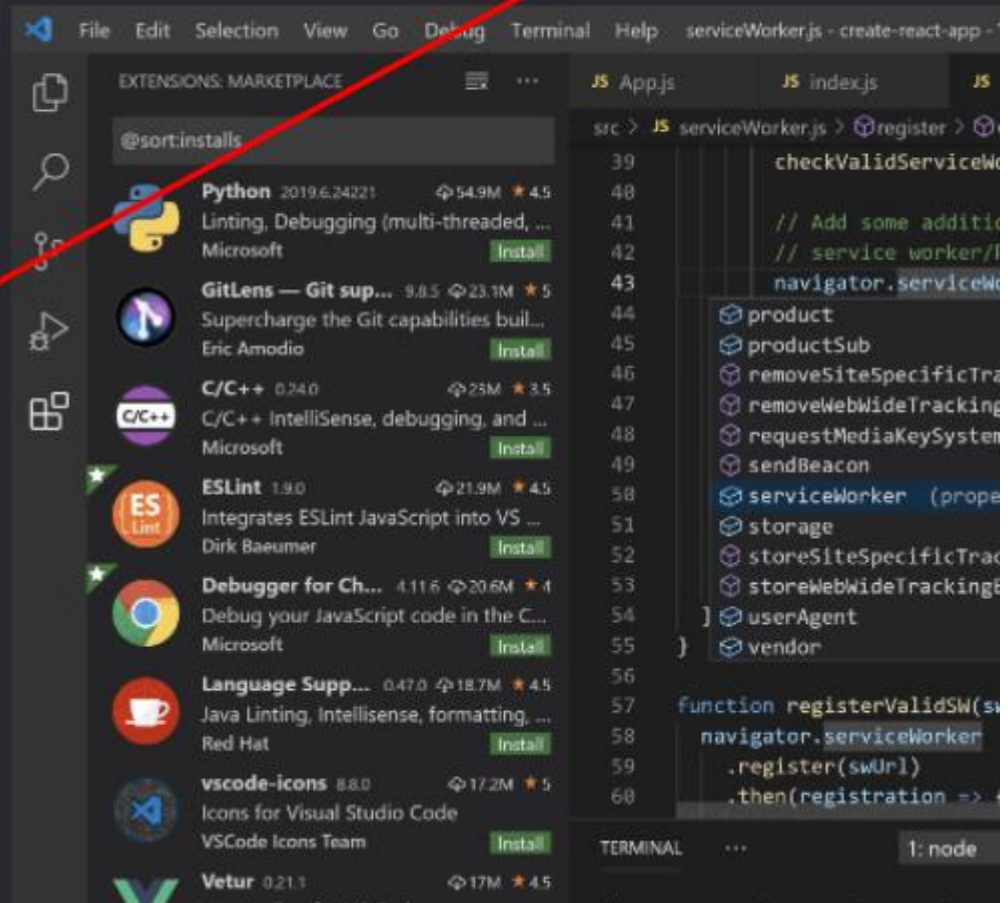
Download for Windows

Stable Build



[Other platforms](#) and [Insiders Edition](#)

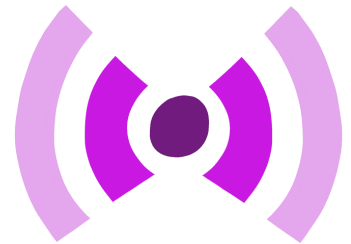
By using VS Code, you agree to its
[license and privacy statement](#).



Estensione Live Server

Un altro compito molto noioso durante lo sviluppo Web è che quando si modifica il codice, è necessario aggiornare il browser per vedere l'effetto delle modifiche. L'estensione "Live Server" può aiutarti aggiornando automaticamente il tuo browser quando qualcosa viene cambiato.

Live Server



FileEditSelectionViewGoRunTerminalHelp

EXTENSIONS: MARKETPL...

live

Live Server 5.6.1

Launch a development local Server with...

Ritwick Dey

Install

Live Share 1.0.4498

Real-time collaborative development fr...

Microsoft

Install

Live Share Audio 0.1.91

Adds audio calling capabilities to Visual ...

Microsoft

Install

Live Share Extension Pack 0.4.0

Collection of extensions that enable real...

Microsoft

Install

Live Sass Compiler 3.0.0

Compile Sass or Scss to CSS at realtime ...

Ritwick Dey

Install

Live HTML Previewer 0.3.0

Edit and preview HTML documents in V...

Harshdeep Gupta

Install

Live Server Preview 0.1.4

Preview your HTML file with localhost se...


negokaz

Install

Emmet Live 1.0.0

Expand your Emmet abbreviations dyna...

Extension: Live Server X




Live Server

ritwickdey.liveserver

Ritwick Dey | 12.879.247 | ★★★★★ | Repository | Live

Launch a development local Server with live reload feature for static & dyn

Install



Details

Feature Contributions

Changelog

[I'm sorry but I'm super busy now. If you want to be a maintainer of the project, please feel free to co

Live Server

Live Server loves ❤️ your multi-root workspace

Live Server for server side pages like PHP. [Check Here](#)

[For 'command not found error' #78]

vscode marketplace

v5.6.1

downloads

16M

rating

4.4/5 (312)

travis branch

passing

appveyor branch

failing

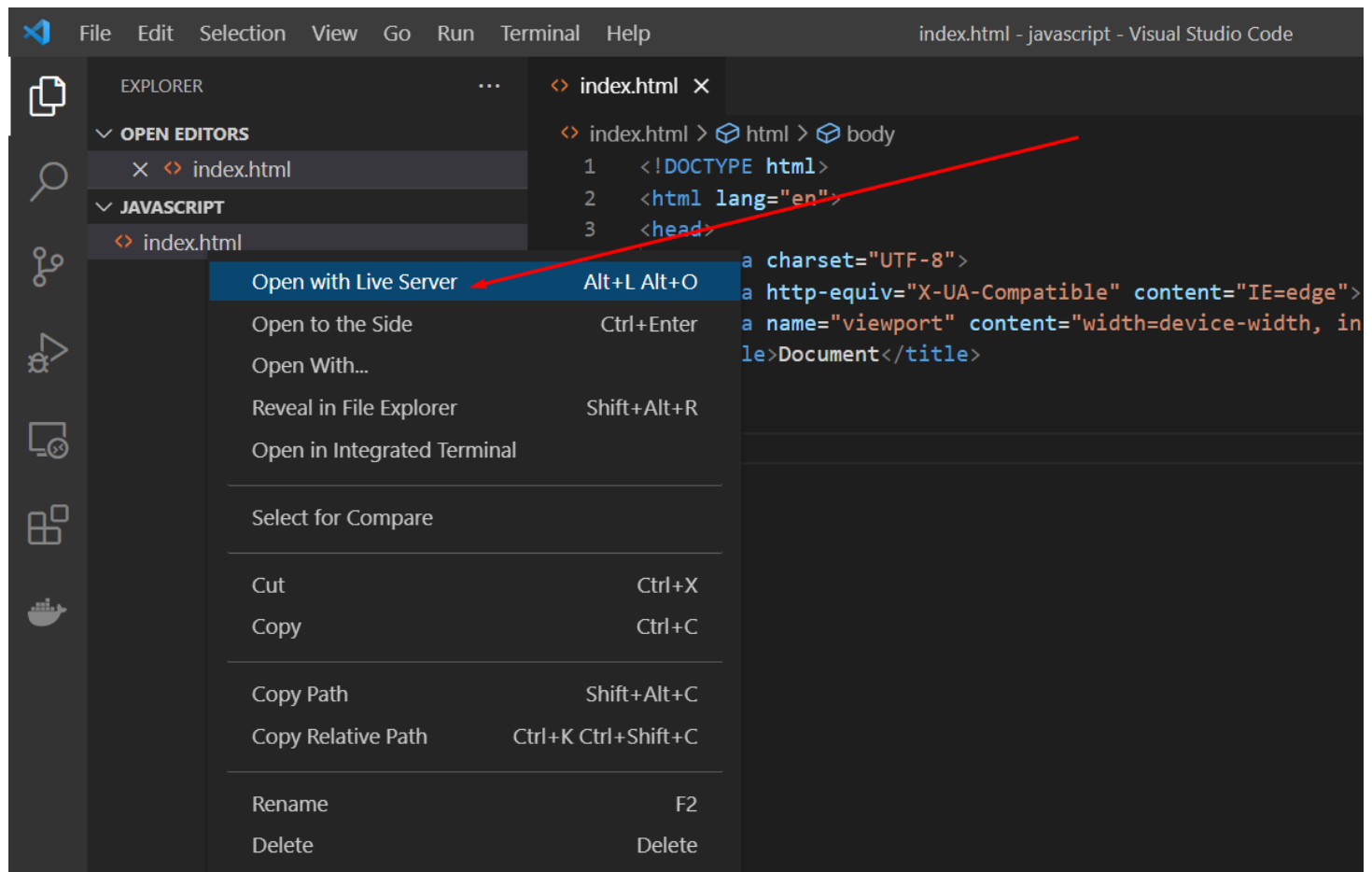
license

MIT

Launch a local development server with live reload feature for static & dynamic pages.

JavaScript

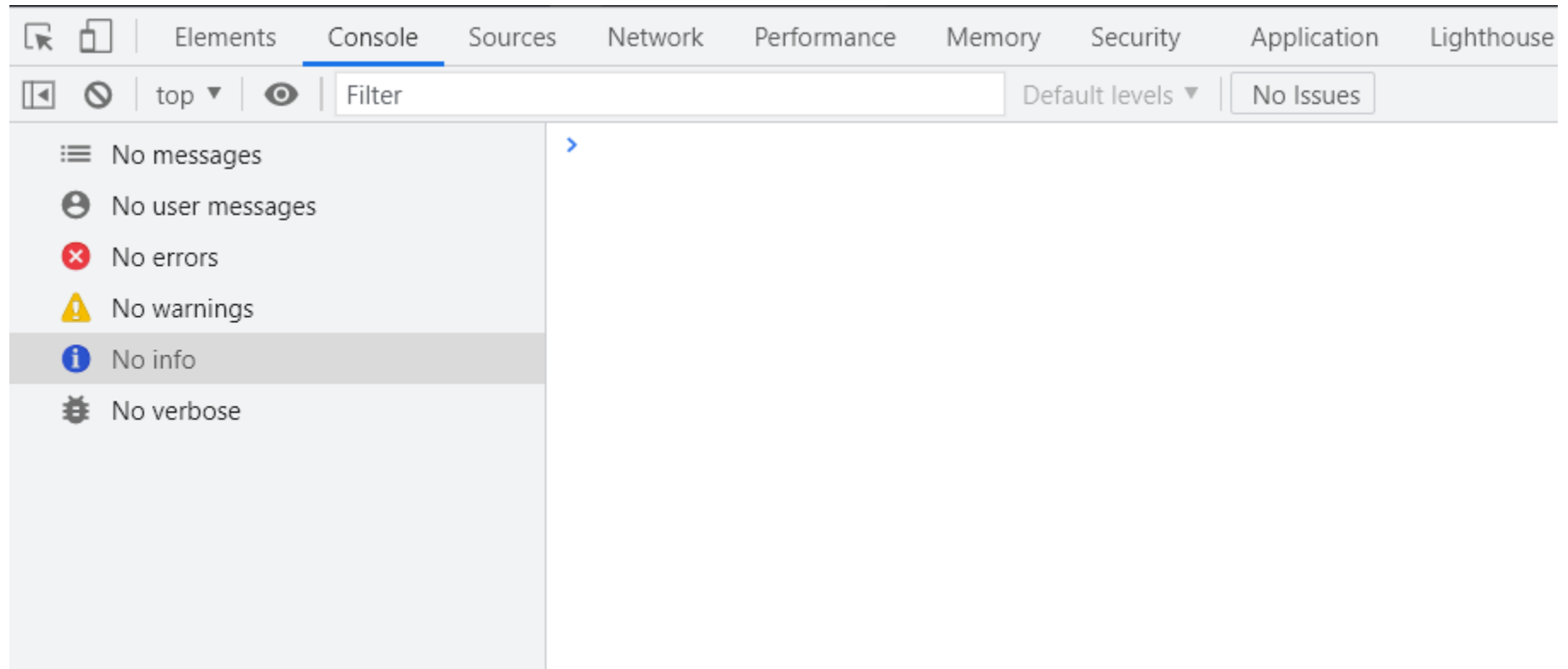
Il plugin è molto comodo in quanto simula la visualizzazione dei file come se venissero interpretati ed inviati da un Server.



DEBUGGER

Gli ambienti di sviluppo hanno in genere un debugger integrato che consente di analizzare il codice durante l'esecuzione.

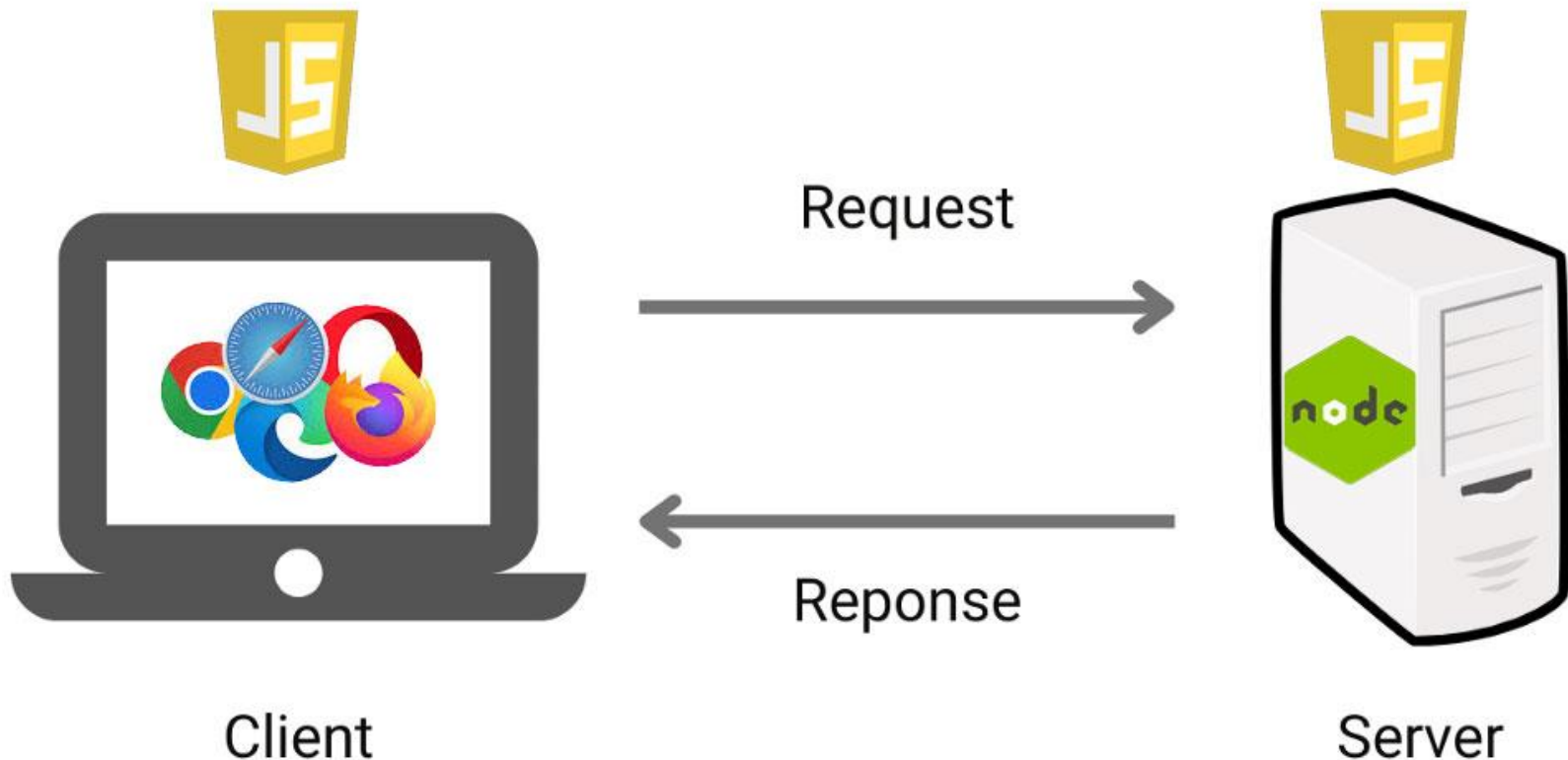
In assenza di un ambiente di sviluppo integrato, possiamo comunque ricorrere a un comune browser, dal momento che i più diffusi prevedono ormai un ambiente di debugging. Ad esempio, la seguente figura mostra una schermata del Tools di Chrome (tasto F12):



JavaScript

Client vs Server

Javascript è un linguaggio che può essere eseguito sia lato client che lato server :



La differenza nell'eseguire Javascript lato client o lato server sta nell'ambiente in cui viene eseguito. Il linguaggio è uno solo, ma a differenza di altri linguaggi, invece di avere librerie interne, sfrutta le librerie dell'ambiente in cui si trova...



```
function add() {  
  return 2 + 2;  
}
```



V8 Engine



011011101

Ogni browser ha il suo «motore» Javascript specifico. Ad esempio FireFox ha «SpiderMonkey», mentre Chrome ha «V8». Inizialmente Javascript è nato come linguaggio lato client e quindi eseguito dal «motore» residente all'interno del browser.

JS lato Client

Lato client Javascript viene utilizzato in «collaborazione» con l'HTML, mentre lato server viene utilizzato singolarmente senza nessuna «collaborazione». Ed è per questo che lato server lavoreremo con dei semplici file.js. Mentre lato client, dovendosi interfacciare con l'HTML, possiamo scrivere il codice in 3 modi diversi :

- Inline
- Blocchi di codice
- File esterno

Codice InLine

Consiste nell'inserire direttamente le istruzioni Javascript nel codice di un elemento HTML, assegnandolo ad un attributo che rappresenta un evento. Chiariamo il concetto con un esempio :

```
<button type="button" onclick="window.alert('x')">Vai</button>
```

Abbiamo assegnato all'attributo onclick dell'elemento button la stringa alert('Ciao!'). L'attributo onclick rappresenta l'evento del clic sul pulsante del mouse, quindi in corrispondenza di questo evento verrà analizzato ed eseguito il codice JavaScript assegnato. Nel caso specifico verrà visualizzato un box con la scritta Ciao.

Blocchi di codice, il tag `<script>`

L'approccio inline può risultare immediato perché mette direttamente in relazione il codice da eseguire con un elemento HTML. Risulta però scomodo quando il codice da eseguire è più complesso o abbiamo necessità di definire variabili e funzioni. In questi casi possiamo ricorrere al tag `<script>` per inserire blocchi di codice in una pagina HTML, come nel seguente esempio:

```
<script>alert('Ciao!')</script>
```

Possiamo inserire blocchi di codice (e i relativi tag `<script>`) nella sezione `<head>` o nella sezione `<body>` della pagina HTML.

Javascript esterno


Il terzo approccio, quello più consigliato, consiste nel collegare alla pagina HTML, codice Javascript presente in un file esterno. Questa tecnica permette di agganciare script e librerie in modo non intrusivo, con il vantaggio di una separazione netta tra la struttura del documento e il codice, come accade per i fogli di stile CSS, che separano struttura e presentazione. Per inserire un file Javascript esterno ci serviamo sempre del tag `<script>` in cui si specifica l'attributo **src**, come mostrato dal seguente esempio:

```
<script src="codice.js"></script>
```

Il riferimento al file Javascript può essere relativo alla pagina HTML corrente oppure assoluto:

```
<script src="http://www.server.com/codice.js"></script>
```


Esempio inline

lezione2 > <> index.html >  html

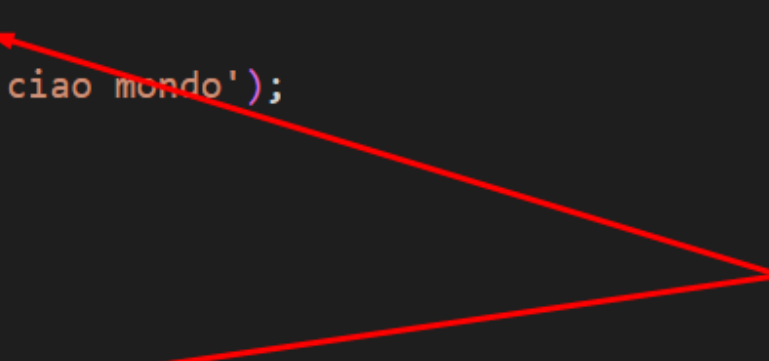
```
1  <!DOCTYPE html>
2  <html lang="it">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Prova inline</title>
8  </head>
9  <body>
10     <h1>Benvenuti</h1>
11     <button onclick="window.alert('ciao mondo')">clicca qui</button>
12 </body>
13 </html>
```



Esempio blocco

lezione2 > <> index.html >  html


```
1  <!DOCTYPE html>
2  <html lang="it">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Prova inline</title>
8      <script>
9          function prova(){
10              window.alert('ciao mondo');
11          }
12      </script>
13 </head>
14 <body>
15     <h1>Benvenuti</h1>
16     <button onclick="prova()">clicca qui</button>
17 </body>
18 </html>
```



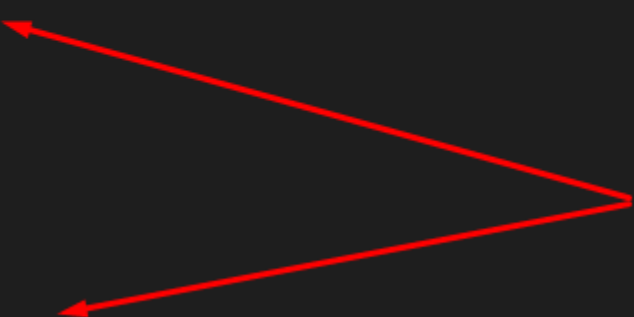
ATTENZIONE : Da notare che, nel caso in cui utilizzo il blocco (o il file esterno), in qualche modo devo avere un «riferimento» al codice da eseguire, questo riferimento si realizza attraverso le funzioni (che vedremo più avanti).

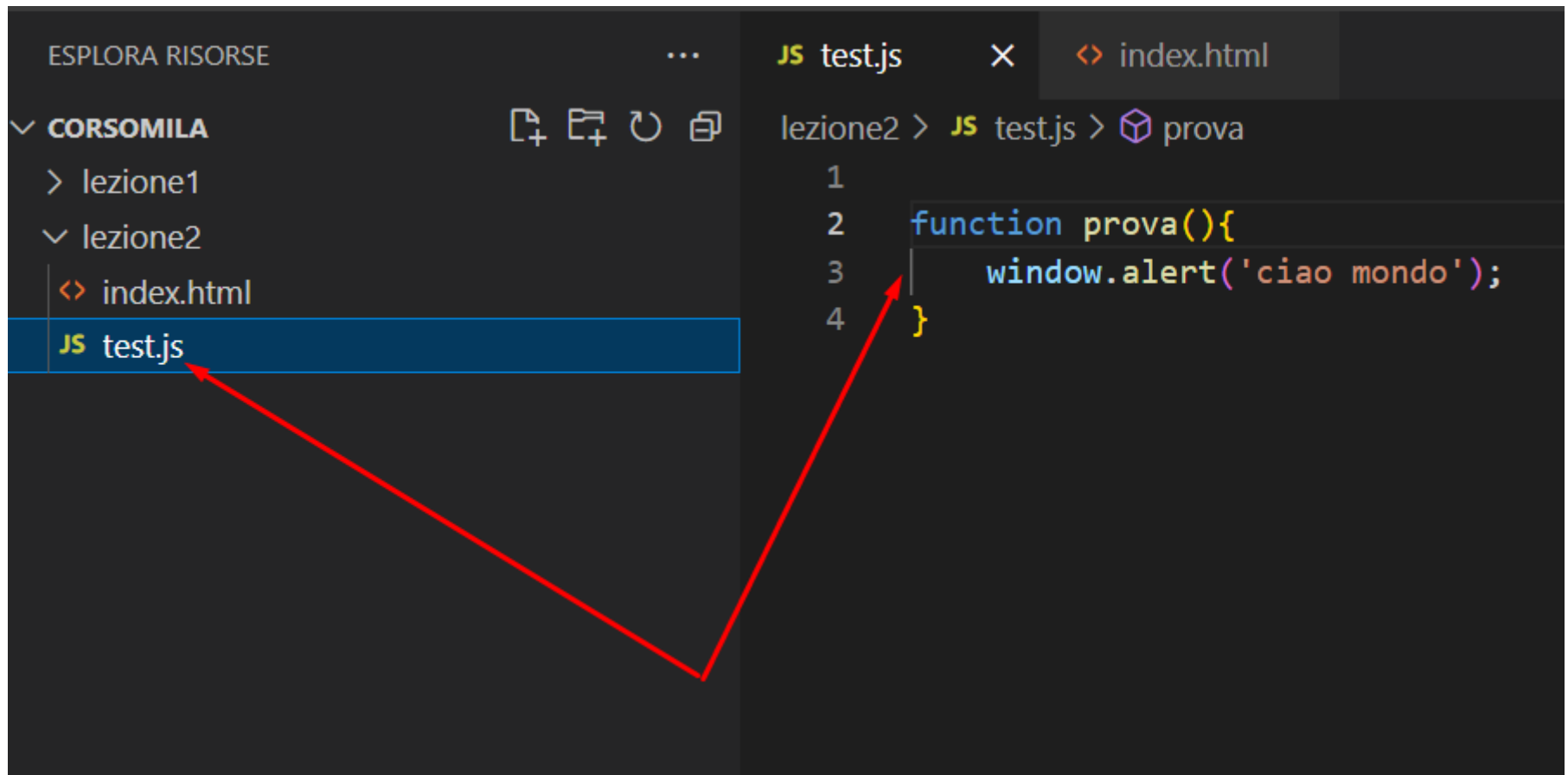
Per il momento pensiamo ad una funzione come ad un blocco di codice che viene racchiuso tra parentesi graffe e a questo blocco viene dato un nome (nel nostro caso `prova()`) e le parentesi tonde stanno ad indicare all'engine V8 che si tratta di una funzione.

Esempio file esterno

lezione2 > <> index.html >  html

```
1  <!DOCTYPE html>
2  <html lang="it">
3  <head>
4      <meta charset="UTF-8">
5      <meta http-equiv="X-UA-Compatible" content="IE=edge">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Prova inline</title>
8      <script src="test.js">
9
10     </script>
11 </head>
12 <body>
13     <h1>Benvenuti</h1>
14     <button onclick="prova()">clicca qui</button>
15 </body>
16 </html>
```





JS lato Server

Lato server Javascript viene eseguito senza nessuna «collaborazione». Ed è per questo che lato server lavoreremo con dei semplici file.js scrivendo il codice in un unico modo : tramite file esterni.

Come abbiamo detto in precedenza se lato client Javascript «sfrutta» e viene eseguito sul browser, lato server abbiamo bisogno di Node.js.

Node.js

Node.js è una runtime di JavaScript Open source multiplatforma orientato agli eventi per l'esecuzione di codice JavaScript, costruita sul motore JavaScript V8 di Google Chrome.

In origine JavaScript veniva utilizzato principalmente **lato client**. In questo scenario gli script JavaScript, generalmente incorporati all'interno dell'HTML di una pagina web, vengono interpretati da un motore di esecuzione incorporato direttamente all'interno di un Browser. Node.js consente invece di utilizzare JavaScript anche per scrivere codice da eseguire **lato server**, ad esempio per la produzione del contenuto delle pagine web dinamiche prima che la pagina venga inviata al Browser dell'utente.

Nel 2009 l'idea «rivoluzionaria» del progettista di Node.js Ryan Dahl è stata quella di prendere il motore V8 ed inserirlo all'interno di un nuovo ambiente in cui eseguire codice JavaScript. Questo ambiente prende il nome di Node.js che non opera lato client ma opera lato server.



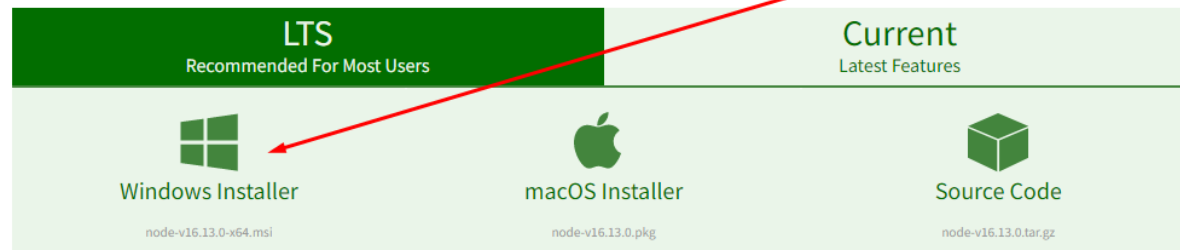
Installazione Node.js

Per installare localmente Node.js bisogna andare sul sito ufficiale www.nodejs.org e scaricare il pacchetto per il nostro sistema operativo.

Downloads

Latest LTS Version: **16.13.0** (includes npm 8.1.0)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.



Windows Installer (.msi)

Windows Binary (.zip)

macOS Installer (.pkg)

macOS Binary (.tar.gz)

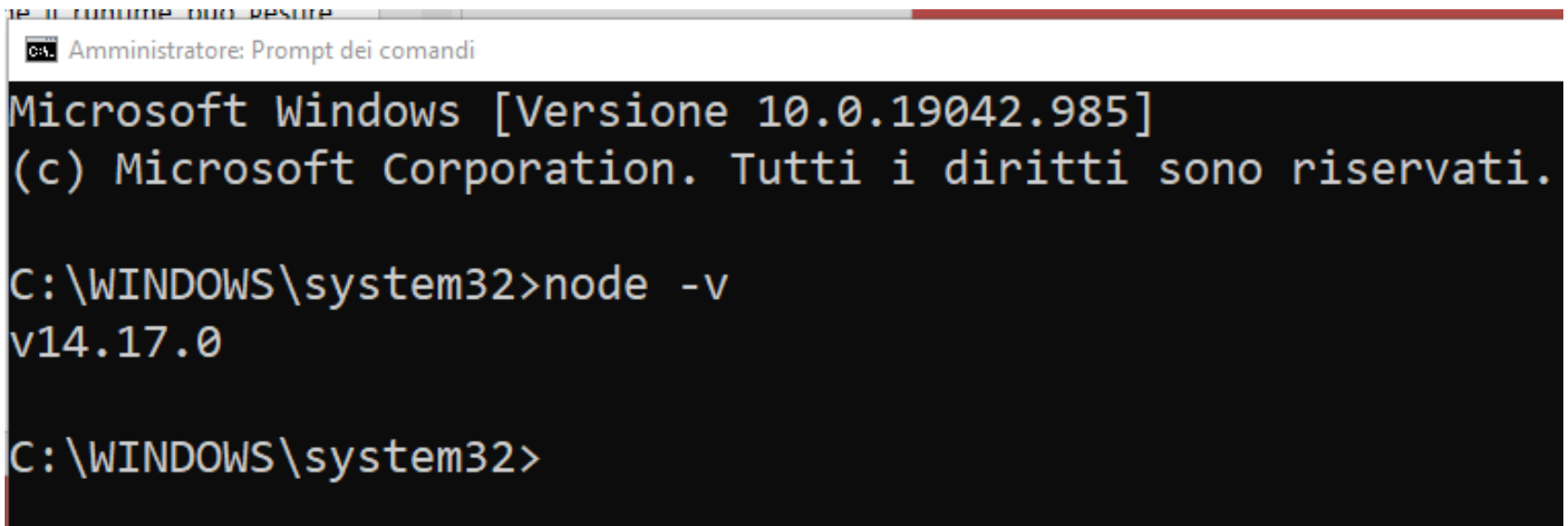
Linux Binaries (x64)

Linux Binaries (ARM)

Source Code

32-bit	64-bit
32-bit	64-bit
64-bit / ARM64	
64-bit	ARM64
64-bit	
ARMv7	ARMv8
node-v16.13.0.tar.gz	

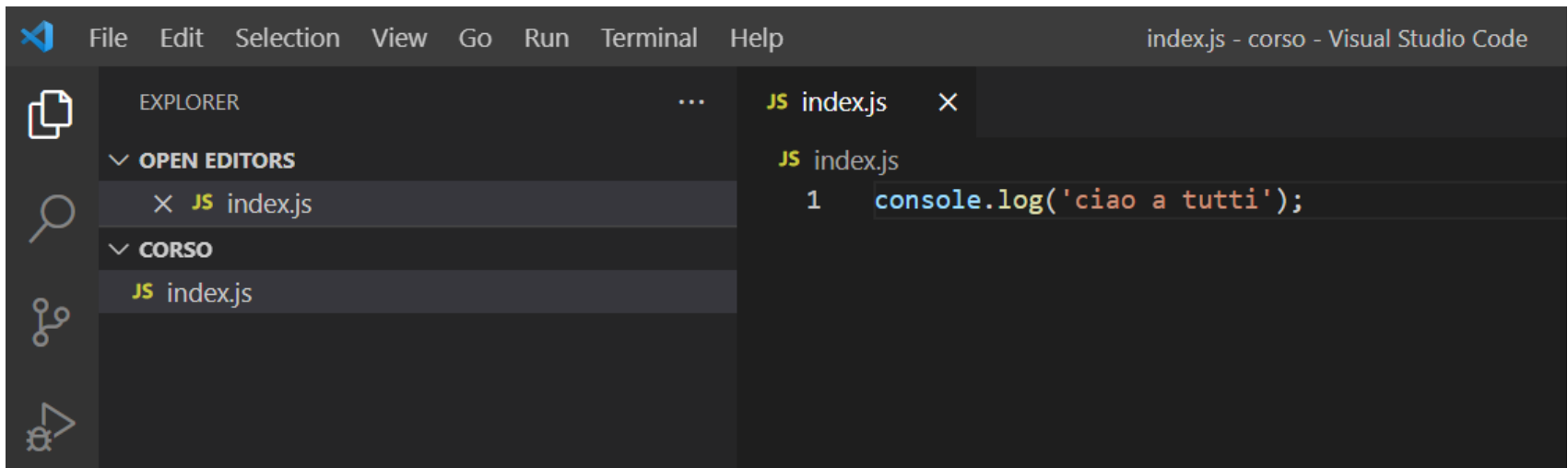
Node.js va utilizzato da riga di comando. Verifichiamo che l'installazione è andata a buon fine aprendo il terminale (dos) e digitare il comando `node -v`

A screenshot of a Windows Command Prompt window. The title bar at the top reads "Amministratore: Prompt dei comandi". The window content shows the following text: "Microsoft Windows [Versione 10.0.19042.985]" followed by "(c) Microsoft Corporation. Tutti i diritti sono riservati." Below this, the command "C:\WINDOWS\system32>node -v" has been entered, and the output "v14.17.0" is displayed. The prompt "C:\WINDOWS\system32>" is shown again at the bottom, indicating the command has completed.

```
Microsoft Windows [Versione 10.0.19042.985]
(c) Microsoft Corporation. Tutti i diritti sono riservati.

C:\WINDOWS\system32>node -v
v14.17.0

C:\WINDOWS\system32>
```



```
C:\Users\Rick\Desktop>cd corso
```

```
C:\Users\Rick\Desktop\corso>node index.js  
ciao a tutti
```

```
C:\Users\Rick\Desktop\corso>
```