

TECNICO SUPERIORE PER L'AUTOMAZIONE ED I SISTEMI MECCATRONICI

GETTING STARTED WITH THE WEB

GUIDA INTRODUTTIVA AL WEB

PROGETTARE UN SITO INTERATTIVO

Docente: GIOVANNI PELLEGRINI



2024

GETTING STARTED WITH THE WEB

è una serie concisa che ti introduce agli aspetti pratici dello sviluppo web. Imposterai gli strumenti necessari per creare una pagina Web semplice e pubblicare il tuo codice semplice.

LA STORIA DEL TUO PRIMO SITO WEB

Creare un sito web professionale richiede molto lavoro, quindi se sei alle prime armi con lo sviluppo web, ti invitiamo a iniziare in piccolo. Non costruirai subito un altro Facebook, ma non è difficile creare il tuo semplice sito web online. Quindi inizieremo da lì.

GUIDE

Lavorando attraverso i capitoli elencati di seguito, passerai dal nulla all'ottenere la tua prima pagina web online. Iniziamo il nostro viaggio !

INSTALLAZIONE DEL SOFTWARE DI BASE

Quando si tratta di strumenti per la creazione di un sito web, c'è molto da scegliere. Se hai appena iniziato, potresti essere confuso dalla serie di editor di codice, framework e strumenti di test disponibili. In questo capitolo, ti mostreremo passo dopo passo come installare il software di cui hai bisogno per iniziare uno sviluppo web di base.

COME SARÀ IL TUO SITO WEB ?

Prima di iniziare a scrivere il codice per il tuo sito web, dovresti prima pianificarlo. Quali informazioni stai mostrando ? Quali caratteri e colori stai usando ? Qui descriveremo un semplice metodo che puoi seguire per pianificare il contenuto e il design del tuo sito.

GESTIONE DEI FILE

Un sito Web è costituito da molti file: contenuto di testo, codice, fogli di stile, contenuto multimediale e così via. Quando crei un sito web, devi assemblare questi file in una struttura sensata e assicurarti che possano comunicare tra loro. Questo capitolo spiega come impostare una struttura di file sensata per il tuo sito web e di quali problemi dovresti essere a conoscenza.

NOZIONI DI BASE SULL'HTML

HyperText Markup Language (HTML) è il codice che utilizzi per strutturare i tuoi contenuti web e dargli significato e scopo. Ad esempio, il mio contenuto è un insieme di paragrafi o un elenco puntato ? Ho delle immagini inserite nella mia pagina ? Ho una tabella dati ? Senza sopraffarti, questo capitolo ti fornirà informazioni sufficienti per familiarizzare con l'HTML.

NOZIONI DI BASE SUI CSS

Cascading Style Sheets (CSS) è il codice che usi per definire lo stile del tuo sito web. Ad esempio, vuoi che il testo sia nero o rosso ? Dove deve essere disegnato il conte-

nuto sullo schermo ? Quali immagini e colori di sfondo dovrebbero essere utilizzati per decorare il tuo sito web ? In questo capitolo, ti guideremo attraverso ciò di cui hai bisogno per iniziare.

NOZIONI DI BASE SU JAVASCRIPT

JavaScript è il linguaggio di programmazione che utilizzi per aggiungere funzionalità interattive al tuo sito web. Alcuni esempi potrebbero essere giochi, cose che accadono quando si premono pulsanti o si immettono dati nei moduli, effetti di stile dinamici, animazioni e molto altro. In questo capitolo, ti daremo un'idea di ciò che è possibile fare con questa entusiasmante lingua e di come iniziare.

PUBBLICAZIONE DEL CODICE DI ESEMPIO

Una volta che hai finito di scrivere il codice e organizzare i file che compongono il tuo sito web, devi metterlo tutto online in modo che le persone possano trovarlo. Questo capitolo descrive come ottenere il codice di esempio semplice online con il minimo sforzo.

COME FUNZIONA IL WEB

Quando accedi al tuo sito web preferito, in background accadono molte cose complicate di cui potresti non essere a conoscenza. Qui descriveremo cosa succede quando visualizzi una pagina web sul tuo computer.

INSTALLAZIONE DEL SOFTWARE DI BASE

In Installazione di software di base, ti mostriamo quali strumenti ti servono per eseguire un semplice sviluppo web e come installarli correttamente.

Di quali strumenti ho effettivamente bisogno, in questo momento?

In questo capitolo, illustreremo il minimo indispensabile: un editor di testo e alcuni browser Web moderni.

INSTALLAZIONE DI UN EDITOR DI TESTO

Probabilmente hai già un editor di testo di base sul tuo computer. Per impostazione predefinita, Windows include Blocco note e macOS viene fornito con TextEdit. Le distribuzioni Linux variano; la versione Ubuntu 22.04 LTS viene fornita con GNOME Text Editor per impostazione predefinita.

Per lo sviluppo web, probabilmente puoi fare meglio di Blocco note o TextEdit. È consigliabile iniziare con Visual Studio Code, che è un editor gratuito, che offre anteprime in tempo reale e suggerimenti sul codice.

INSTALLAZIONE DI BROWSER WEB MODERNI

Un paio di browser Web desktop servono a testare il nostro codice.

- Per Linux: Firefox, Chrome, Opera, Brave.
- Per Windows: Firefox, Chrome, Opera, Microsoft Edge, Brave (Windows 10 viene fornito con Edge per impostazione predefinita; in caso contrario, è necessario installare un browser alternativo).
- Per macOS: Firefox, Chrome, Opera, Safari, Brave (macOS e iOS vengono forniti con Safari per impostazione predefinita).

QUALI STRUMENTI UTILIZZANO I PROFESSIONISTI ?

Quello che segue sembra un elenco spaventoso, ma fortunatamente puoi iniziare a sviluppare web senza sapere nulla della maggior parte di questi.

- **Un computer.**
Forse sembra ovvio per alcune persone, ma alcuni di voi stanno leggendo questo articolo sul proprio telefono o sul computer di una biblioteca. Per uno sviluppo web serio, è meglio investire in un computer desktop o laptop con Windows, macOS o Linux.
- **Un editor di testo, in cui scrivere codice.**
Può trattarsi di un editor di testo (ad esempio, Visual Studio Code, Notepad++, Sublime Text, GNU Emacs o VIM) o di un editor ibrido (ad esempio, Dreamweaver o WebStorm). Gli editor di documenti di Office non sono adatti a questo uso, in quanto si basano su elementi nascosti che interferiscono con i motori di rendering utilizzati dai browser Web.
- **Browser Web, per testare il codice.**
Attualmente, i browser più utilizzati sono Firefox, Chrome, Safari e Microsoft Edge. Dovresti anche testare le prestazioni del tuo sito sui dispositivi mobili e su

tutti i vecchi browser che il tuo pubblico di destinazione potrebbe ancora utilizzare (come IE 8-10). Lynx, un browser Web per terminali basato su testo, è ottimo per vedere come il tuo sito venga visto dagli utenti ipovedenti.

- **Un editor grafico.**

Applicazioni come GIMP, Figma, Paint.NET, Photoshop, Sketch o XD, per creare immagini o grafica per le tue pagine web.

- **Un sistema di controllo delle versioni.**

Per gestire i file sui server, collaborare a un progetto con un team, condividere codice e risorse ed evitare conflitti di modifica. In questo momento, Git è il sistema di controllo della versione più popolare insieme al servizio di hosting GitHub o GitLab.

- **Un programma FTP, utilizzato su vecchi account di web hosting per gestire i file sui server (Git sta sempre più sostituendo FTP per questo scopo).**

Sono disponibili un sacco di programmi (S)FTP, tra cui Cyberduck, Fetch e FileZilla.

- **Un sistema di automazione.**

Webpack, Grunt o Gulp eseguono automaticamente attività ripetitive, come la minimizzazione del codice e l'esecuzione di test.

- **Librerie, framework, ecc., per velocizzare la scrittura di funzionalità comuni.**

Una libreria tende ad essere un file JavaScript o CSS esistente che fornisce funzionalità pronte per l'uso nel codice. Un framework tende a portare avanti questa idea, offrendo un sistema completo con alcune sintassi personalizzate su cui scrivere un'app web.

Più strumenti per produrre più velocemente !

COME SARÀ IL TUO SITO WEB ?

Illustra il lavoro di pianificazione e progettazione che devi eseguire per il tuo sito Web prima di scrivere il codice, tra cui: “quali informazioni offre il mio sito Web?”, “quali tipi di carattere e colori voglio?” e “cosa fa il mio sito?”

PER PRIMA COSA: LA PIANIFICAZIONE

Prima di fare qualsiasi cosa, hai bisogno di alcune idee. Cosa dovrebbe fare effettivamente il tuo sito web? Un sito web può fare praticamente qualsiasi cosa, ma, per il tuo primo tentativo, dovresti mantenere le cose semplici. Inizieremo creando una semplice pagina web con un'intestazione, un'immagine e alcuni paragrafi.

Per iniziare, dovrai rispondere a queste domande:

- Di cosa tratta il tuo sito web? Ti piacciono i cani, New York o Pac-Man?
- Quali informazioni state presentando sull'argomento?

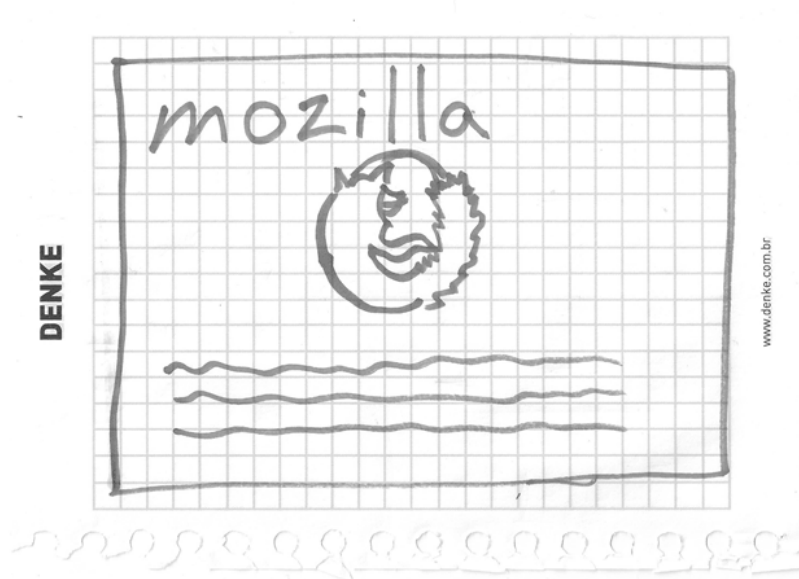
Scrivi un titolo e alcuni paragrafi e pensa a un'immagine che vorresti mostrare sulla tua pagina.

- Che aspetto ha il tuo sito web, in termini semplici e di alto livello? Qual è il colore di sfondo? Che tipo di carattere è appropriato: formale, cartonesco, audace, sottile?

Nota: I progetti complessi richiedono linee guida dettagliate che entrino in tutti gli aspetti relativi a colori, caratteri, spaziatura tra gli elementi di una pagina, stile di scrittura appropriato e così via. Questo è, ingergo tecnico, definito, guida alla progettazione o sistema di progettazione.

ABBOZZARE IL TUO PROGETTO

Quindi, prendi carta e penna e disegna approssimativamente come vuoi che appaia il tuo sito. Per la tua prima semplice pagina web, non c'è molto da abbozzare, ma dovresti prendere l'abitudine di farlo ora. Aiuta davvero: non devi essere Van Gogh!



Un disegno approssimativo e uno schizzo di un sito web su carta

Nota: Anche su siti web reali e complessi, i team di progettazione di solito iniziano con schizzi grezzi su carta e successivamente costruiscono mockup digitali utilizzando un editor grafico o tecnologie web.

Mockup

Nella produzione e nella progettazione, un modello, o mock-up, è un modello in scala o a grandezza naturale di un progetto o di un dispositivo, utilizzato per l'insegnamento, la dimostrazione, la valutazione del progetto, la promozione e altri scopi.

I team Web spesso includono sia un graphic designer che un designer dell'esperienza utente (UX). I grafici mettono insieme le immagini del sito web. Gli UX designer hanno un ruolo un po' più astratto nell'affrontare il modo in cui gli utenti sperimenteranno e interagiranno con il sito web.

SCELTA DEGLI ELEMENTI

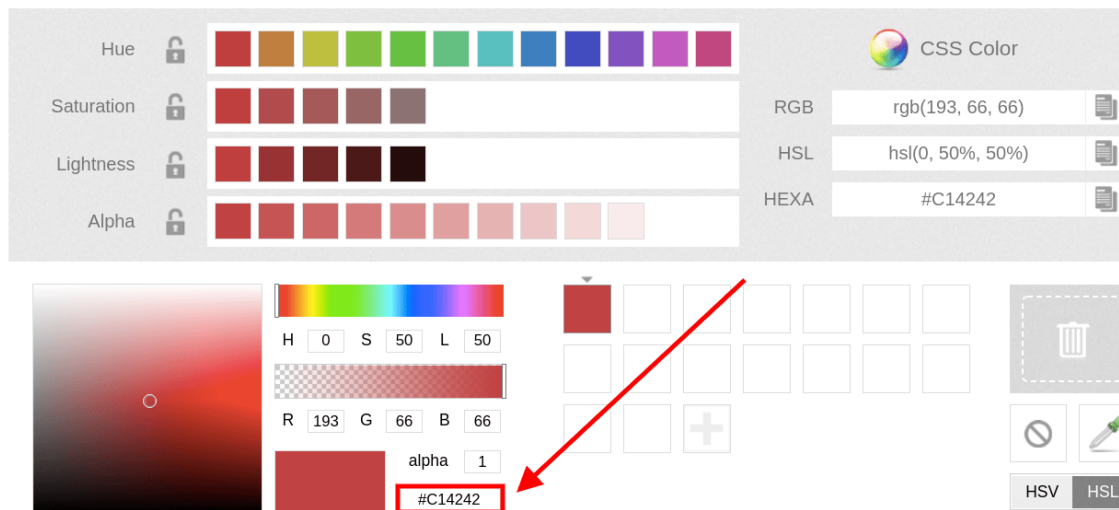
A questo punto, è bene iniziare a mettere insieme gli elementi che alla fine appariranno sulla tua pagina web.

- **Testo**

Dovresti avere, ben chiaro, il titolo, i paragrafi, le didascalie, le caselle di testo, i menu.

- **Colore del tema**

Per scegliere un colore, nell'editor grafico scelto, vai al **Selettore colore** e trova un colore che ti piaccia. Quando fai clic su un colore, vedrai uno strano codice di sei caratteri come il seguente: **#660066**. Si chiama **codice esadecimale** e rappresenta il tuo colore. Copia il codice in un posto sicuro per ora.

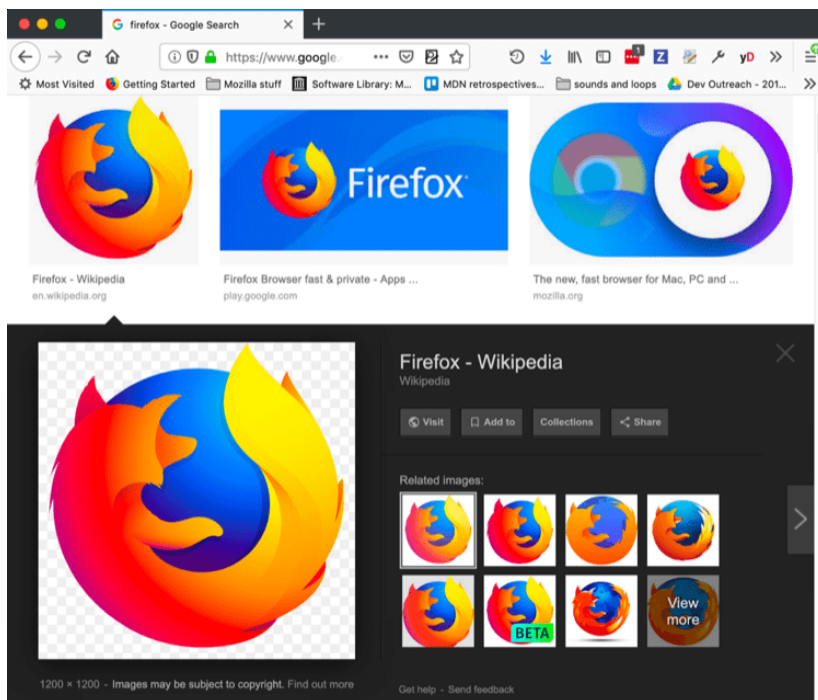


Color-Picker-Tool sul sito Web di MDN Docs con colori RGB, HSL e HEX

- **Immagini**

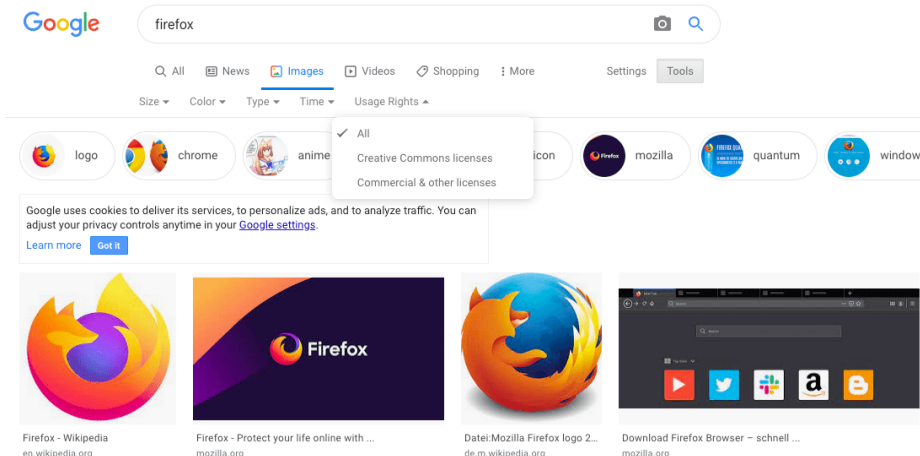
Per scegliere un'immagine, vai su [Google Immagini](https://www.google.com/search?q=Google+Immagini) e cerca un elemento grafico appropriato relativo ai contenuti del tuo sito.

- Quando trovi l'immagine che desideri, fai clic sull'immagine per ingrandirla.
- Fai clic con il pulsante destro del mouse sull'immagine (Ctrl + clic su Mac), scegli Salva immagine con nome... e scegli una posizione sicura per salvare l'immagine. In alternativa, copia l'indirizzo web dell'immagine dalla barra degli indirizzi del browser per un uso successivo.



Risultati di ricerca per un termine di ricerca su Google Immagini

Tieni presente che la maggior parte delle immagini sul Web, incluso Google Immagini, sono protette da copyright. Per ridurre la probabilità di violare il copyright, puoi utilizzare il filtro delle licenze di Google. Fare clic sul pulsante Strumenti, quindi sull'opzione Diritti di utilizzo risultante che appare di seguito. Dovresti scegliere l'opzione Licenze Creative Commons.



Risultati di ricerca filtrati per ottenere immagini di licenze Creative Commons su Google Immagini

- **Font**

Come per le immagini, molti font sono protetti da licenze, il che significa che non puoi usarli liberamente nel tuo sito. [Google Fonts](#) è un servizio web di proprietà di Google che fornisce l'accesso a molti font.

Una volta trovato un font, ci sono due modi principali per utilizzarlo:

- Aggiungi un riferimento nel tuo codice per caricare il font dai server di Google.
- Scarica il file del font sul tuo sistema, ospita tu stesso il font e usa la tua copia ospitata nel codice del tuo sito web.

Nota: La pubblicazione di font ospitati su Google Fonts potrebbe essere incompatibile con il GDPR, il regolamento sulla privacy dei dati dell'Unione Europea, in quanto il servizio di font espone l'indirizzo IP dell'utente. Se questo è un potenziale problema per te, scegli la seconda opzione.

In alternativa, è possibile utilizzare font Web sicuri come Arial, Times New Roman o Courier New.

GESTIONE DEI FILE

Un sito Web è costituito da molti file: contenuto di testo, codice, fogli di stile, contenuto multimediale e così via. Quando crei un sito web, devi assemblare questi file in una struttura ben organizzata sul tuo computer locale, assicurarti che possano comunicare tra loro e ottenere tutti i tuoi contenuti prima di caricarli su un server.

Questo capitolo analizza alcuni problemi di cui dovresti essere a conoscenza in modo da poter impostare una struttura di file ben modulata per il tuo sito web.

DOVE DOVREBBE RISIEDERE IL TUO SITO WEB SUL TUO COMPUTER ?

Quando si lavora su un sito Web localmente sul computer, è necessario conservare tutti i file correlati in un'unica cartella che rispecchi la struttura dei file del sito Web pubblicato sul server. Questa cartella può risiedere ovunque tu voglia, ma dovresti metterla da qualche parte dove possa trovarla facilmente, magari sul desktop, nella cartella Home o nella radice del tuo disco rigido.

- Scegli un posto dove archiviare i progetti del tuo sito web. All'interno della posizione scelta, crea una nuova cartella chiamata web-projects (o simile). È qui che risiederanno tutti i progetti del tuo sito web.
- All'interno di questa prima cartella, crea un'altra cartella in cui archiviare il tuo primo sito web. Chiamatela sito (o qualcosa di più fantasioso).

UNA PRECISAZIONE SULLA STRUTTURA E LA SPAZIATURA

Noterai che in questo capitolo ti chiediamo di nominare le cartelle e i file completamente in minuscolo senza spazi. Questo perché:

- Molti computer, in particolare i server Web, fanno distinzione tra maiuscole e minuscole. Quindi, ad esempio, se inserisci un file di immagine chiamato: sito/MyImage.jpg sul tuo sito Web in e poi in un file diverso provi a richiamare l'immagine come sito/myimage.jpg, potrebbe non funzionare.
- I browser, i server Web e i linguaggi di programmazione non gestiscono gli spazi in modo coerente. Ad esempio, se si utilizzano spazi nel nome del file, alcuni sistemi potrebbero considerare il nome del file come due nomi di file. Alcuni server sostituiranno gli spazi nei nomi dei file con "%20" (il codice carattere per gli spazi negli URL), con il risultato che tutti i collegamenti verranno interrotti. È meglio separare le parole con trattini my-file.html, piuttosto che con caratteri di sottolineatura my_file.html

In sostanza, è bene usare un trattino per i nomi dei tuoi file. Il motore di ricerca di Google considera un trattino come un separatore di parole, ma non considera un carattere di sottolineatura in questo modo. Per questi motivi, è meglio prendere l'abitudine di scrivere i nomi delle cartelle e dei file in minuscolo, senza spazi e con le parole separate da trattini, almeno fino a quando non si sa cosa si stia facendo. In questo modo incontrerai meno problemi lungo la strada.

CHE STRUTTURA DOVREBBE AVERE IL TUO SITO WEB ?

Successivamente, diamo un'occhiata a quale struttura dovrebbe avere il nostro sito di test. Le cose più comuni che avremo in qualsiasi progetto di sito Web che creiamo sono un file HTML di indice e cartelle per contenere immagini, file di stile e file di

script. Creiamoli ora:

1. **index.html**: questo file conterrà generalmente il contenuto della tua home page, ovvero il testo e le immagini che le persone vedono quando visitano per la prima volta il tuo sito. Utilizzando l'editor di testo, crea un nuovo file chiamato **index.html** e salvalo all'interno della cartella **sito**
2. **cartella "images"**: questa cartella conterrà tutte le immagini che utilizzi sul tuo sito. Crea una cartella denominata **images**, all'interno della tua cartella **sito**
3. **cartella "styles"**: questa cartella conterrà il codice CSS utilizzato per definire lo stile del contenuto (ad esempio, l'impostazione dei colori del testo e dello sfondo). Crea una cartella denominata **styles**, all'interno della tua cartella **sito**
4. **cartella "scripts"**: questa cartella conterrà tutto il codice JavaScript utilizzato per aggiungere funzionalità interattive al tuo sito (ad esempio, pulsanti che caricano i dati quando vengono cliccati). Crea una cartella denominata **scripts**, all'interno della tua cartella **sito**.

Nota: Sui computer Windows, potresti avere problemi a vedere i nomi dei file, perché Windows ha un'opzione chiamata Nascondi estensioni per i tipi di file noti attivata per impostazione predefinita. In genere, è possibile disattivare questa opzione accedendo a Esplora risorse, selezionando l'opzione Opzioni cartella..., deselezionando la casella di controllo Nascondi estensioni per tipi di file noti, quindi facendo clic su OK. Per informazioni più specifiche sulla versione di Windows in uso, è possibile eseguire una ricerca sul Web.

PERCORSI DEI FILE

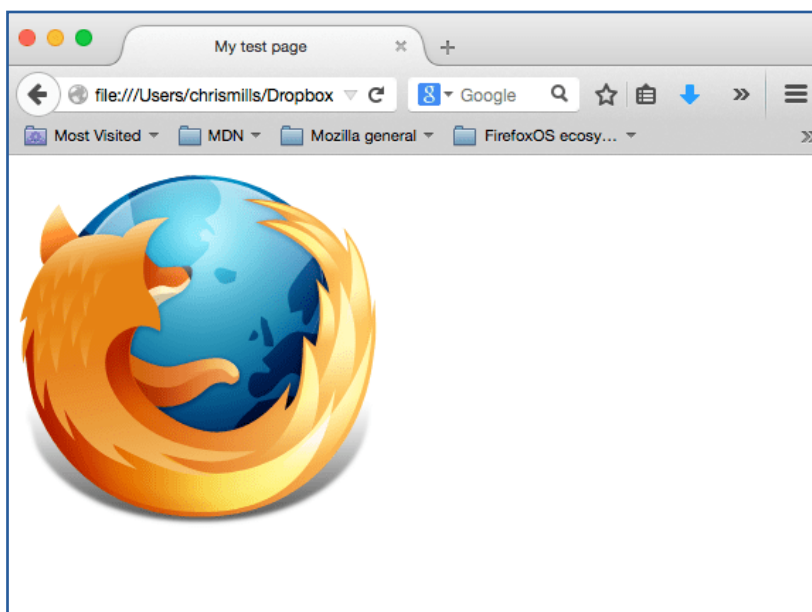
Per fare in modo che i file comunichino tra loro, è necessario fornire loro un percorso, in modo che un file sappia dove si trovi un altro. Per dimostrarlo, inseriremo un po' di HTML nel nostro file **index.html** e gli faremo visualizzare l'immagine scelta.

In alternativa, puoi scegliere un'immagine esistente a tua disposizione, sul tuo computer o dal Web, e utilizzarla nei seguenti passaggi:

- Copia l'immagine che hai scelto in precedenza nella cartella.images
- Apri il file e inserisci il codice seguente esattamente come illustrato. Non preoccuparti di cosa significhi tutto questo per ora: esamineremo le strutture in modo più dettagliato più avanti.

```
<!doctype html>
<html lang="en-US">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width" />
  <title>My test page 3</title>
</head>
<body>
  
</body>
</html>
```

- La riga `` è il codice HTML che inserisce un'immagine nella pagina. Dobbiamo dire all'HTML dove si trovi l'immagine. L'immagine si trova all'interno della directory **images**, che si trova nella **stessa directory** di **index.html**. Per scorrere la struttura del file della nostra immagine, il percorso del file di cui abbiamo bisogno è **images/your-image-filename**. Ad esempio, la nostra immagine si chiama **firefox.png**, quindi il percorso del file è **images/firefox.png**.
- Inserisci il percorso del file nel codice HTML `src=""` tra le virgolette doppie del codice.
- Modificare il contenuto dell'attributo **alt** in una descrizione dell'immagine che si sta includendo. In questo caso: `alt="Firefox logo: flaming fox wrapping the world"` (Logo di Firefox: volpe fiammeggiante che avvolge il mondo)
- Salva il file HTML, quindi caricalo nel browser web (fai doppio clic sul file). Dovresti vedere la tua nuova pagina web che mostra la tua immagine!



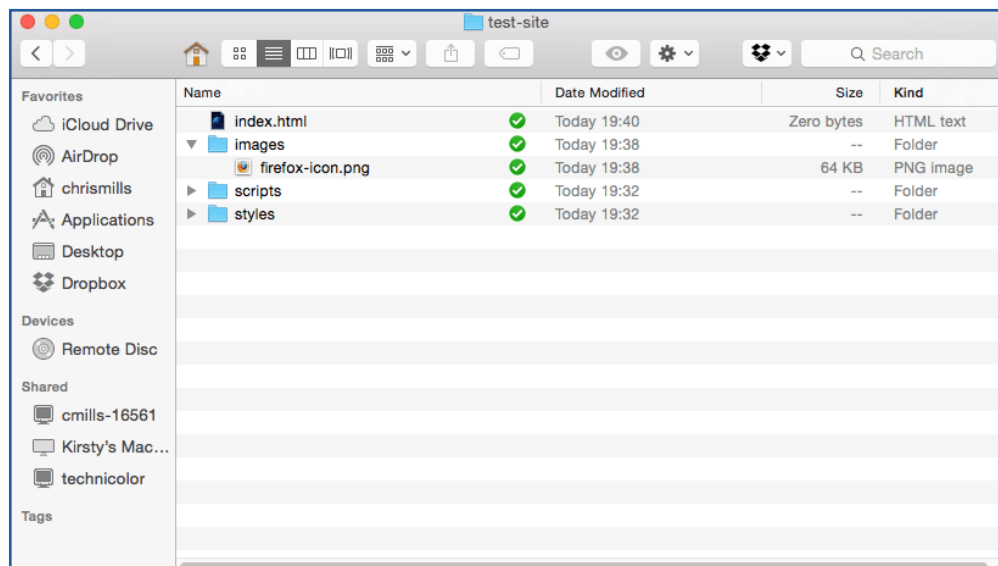
Alcune regole generali per i percorsi dei file:

- Per collegarsi a un file di destinazione **nella stessa directory del file HTML** che lo richiama, è sufficiente utilizzare il nome del file, ad esempio: **firefox.png**
- Per fare riferimento a un file in una sottodirectory, scrivere il nome della directory davanti al percorso, più una barra, ad esempio: **images/firefox.png**

COS'ALTRO SI DOVREBBE FARE ?

Per ora è tutto.

La struttura delle cartelle dovrebbe essere simile a questa:



NOZIONI DI BASE SULL'HTML

HTML (HyperText Markup Language) è il codice che viene utilizzato per strutturare una pagina web e il suo contenuto. Ad esempio, il contenuto potrebbe essere strutturato all'interno di una serie di paragrafi, un elenco puntato o utilizzando immagini e tabelle di dati. Come suggerisce il titolo, questo capitolo ti fornirà una conoscenza di base dell'HTML e delle sue funzioni.

CHE COS'È L'HTML ?

L'HTML è un linguaggio di markup che definisce la struttura dei contenuti. L'HTML è costituito da una serie di elementi, che vengono utilizzati per racchiudere, o avvolgere, diverse parti del contenuto per farlo apparire in un certo modo o agire in un certo modo. I **tag** di inclusione possono creare un collegamento ipertestuale a una parola o a un'immagine da qualche altra parte, possono mettere in corsivo le parole, possono ingrandire o rimpicciolire il carattere e così via. Si consideri, ad esempio, la seguente riga di contenuto:

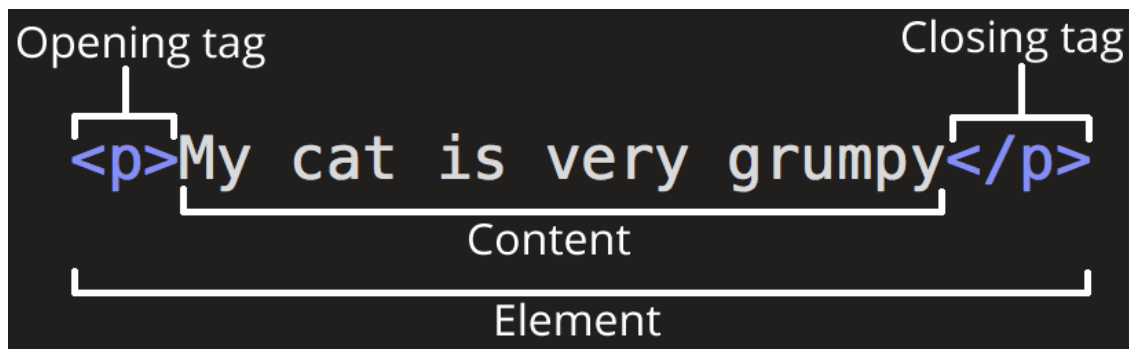
My cat is very grumpy

Se volessimo che la riga stesse in piedi da sola, potremmo specificare che si tratti di un paragrafo racchiudendolo nei tag di paragrafo:

`<p>My cat is very grumpy</p>`

ANATOMIA DI UN ELEMENTO HTML

Esploriamo un po' più a fondo questo elemento del paragrafo.



Le parti principali del nostro elemento sono le seguenti:

- Il tag di apertura: è costituito dal nome dell'elemento (in questo caso, p), racchiuso tra parentesi angolari di apertura e chiusura. Indica dove l'elemento inizi o inizi ad avere effetto, in questo caso dove inizia il paragrafo.
- Il tag di chiusura: è lo stesso del tag di apertura, tranne per il fatto che include una barra prima del nome dell'elemento. Indica dove finisca l'elemento, in questo caso dove finisce il paragrafo. La mancata aggiunta di un tag di chiusura è uno degli errori standard per principianti e può portare a risultati strani.

- Il contenuto: Questo è il contenuto dell'elemento, che in questo caso è solo testo.
- L'elemento: Il tag di apertura, il tag di chiusura e il contenuto insieme costituiscono l'elemento.

Gli elementi possono anche avere attributi simili ai seguenti:

Attribute

```
<p class="editor-note">My cat is very grumpy</p>
```

Gli attributi contengono informazioni aggiuntive sull'elemento che non si desidera visualizzare nel contenuto effettivo. Qui, **class** è il nome dell'attributo e **editor-note** il valore dell'attributo. L'attributo **class** consente di assegnare all'elemento un identificatore non univoco che può essere utilizzato per scegliere, come target (e qualsiasi altro elemento con lo stesso valore), informazioni sullo stile e altri elementi.

Gli attributi che impostano un valore hanno sempre:

- Nome dell'attributo seguito da un segno di uguale.
- Valore dell'attributo racchiuso tra virgolette di apertura e chiusura.

ELEMENTI DI ANNIDAMENTO

È possibile inserire elementi anche all'interno di altri elementi: questo si chiama **annidamento**. Se volessimo affermare che il nostro gatto è molto scontroso, potremmo racchiudere la parola "molto" in un elemento **<forte>**, il che significa che la parola è da enfatizzare fortemente:

```
<p>My cat is <strong>very</strong> grumpy.</p>
```



Tuttavia, è necessario assicurarsi che gli elementi siano nidificati correttamente. Nell'esempio precedente, abbiamo aperto prima l'elemento **<p>**, quindi l'elemento ****; Pertanto, dobbiamo chiudere prima l'elemento ****, quindi l'elemento **</p>**. Quanto segue **non è corretto**:

```
<p>My cat is <strong>very grumpy.</p></strong>
```



Gli elementi devono aprirsi e chiudersi correttamente in modo che siano chiaramente all'interno o all'esterno l'uno dell'altro. Se si sovrappongono come mostrato sopra, il tuo browser web cercherà di indovinare al meglio ciò che stavi cercando di dire, il che può portare a risultati inaspettati.

ELEMENTI VUOTI

Alcuni elementi non hanno contenuto e sono chiamati elementi void. Prendiamo l'elemento **** che abbiamo già nella nostra pagina HTML:

```

```


Contiene due attributi, ma non c'è alcun tag di chiusura `` e nessun contenuto interno. Ciò è dovuto al fatto che un elemento immagine non esegue il wrapping del contenuto per influenzarlo. Il suo scopo è quello di incorporare un'immagine nella pagina HTML nel punto in cui appare.

ANATOMIA DI UN DOCUMENTO HTML

Ora vedremo come i singoli elementi vengono combinati per formare un'intera pagina HTML. Esaminiamo nuovamente il codice che abbiamo inserito nel nostro esempio `index.html` (che abbiamo incontrato per la prima volta nel capitolo Gestione dei file).

```
<!doctype html>
<html lang="en-US">
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width" />
  <title>My test page</title>
</head>
<body>
  
</body>
</html>
```

Qui, abbiamo quanto segue:

- `<!DOCTYPE html>` — **doctype**. E' un preambolo d'obbligo. Nella notte dei tempi, quando l'HTML era giovane (intorno al 1991/92), i doctype erano pensati per fungere da collegamenti a un insieme di regole che la pagina HTML doveva seguire per essere considerata un buon HTML, il che poteva significare il controllo automatico degli errori e altre cose utili. Tuttavia, al giorno d'oggi, non fanno molto e sono fundamentalmente necessari solo per assicurarsi che il tuo documento si comporti correttamente. Questo è tutto ciò che devi sapere per ora.
- `<html></html>` — **l'elemento <html>**. Questo elemento racchiude tutto il contenuto dell'intera pagina ed è talvolta noto come elemento radice. Include anche l'attributo `lang`, impostando la lingua principale del documento.
- `<head></head>` — **l'elemento <testa>**. Questo elemento funge da contenitore per tutti gli elementi che desideri includere nella pagina HTML che non sono i contenuti che stai mostrando ai visualizzatori della tua pagina. Ciò include cose come parole chiave e una descrizione della pagina che desideri appaia nei risultati di ricerca, CSS per definire lo stile dei nostri contenuti, dichiarazioni di set di caratteri e altro ancora.
- `<meta charset="utf-8">` — Questo elemento imposta il set di caratteri che il documento deve utilizzare su UTF-8, che include la maggior parte dei caratteri della stragrande maggioranza delle lingue scritte. In sostanza, ora è in grado di gestire qualsiasi contenuto testuale che potresti inserire su di esso. Non c'è motivo per non impostarlo e può aiutare a evitare alcuni problemi in seguito.

`<meta name="viewport" content="width=device-width">` : questo elemento della finestra garantisce che la pagina venga visualizzata alla larghezza della finestra, impedendo ai browser per dispositivi mobili di eseguire il rendering delle pagine più larghe della finestra e quindi di ridurle.

`<title></title>` — l'elemento `<title>`. In questo modo viene impostato il titolo della pagina, che è il titolo che appare nella scheda del browser in cui viene caricata la pagina. Viene anche utilizzato per descrivere la pagina quando la aggiungi ai segnalibri/ o aggiungi ai preferiti.

`<body></body>` — l'elemento `<corpo>`. Questo include tutti i contenuti che vuoi mostrare agli utenti web quando visitano la tua pagina, che si tratti di testo, immagini, video, giochi, tracce audio riproducibili o qualsiasi altra cosa.

IMMAGINI

Rivolgiamo di nuovo la nostra attenzione all'elemento ``:

```

```

Come abbiamo detto prima, incorpora un'immagine nella nostra pagina nella posizione in cui appare. Lo fa tramite l'attributo `src` (source), che contiene il percorso del nostro file immagine.

Abbiamo anche incluso un attributo `alt` (alternativo). Nell'attributo `alt` è possibile specificare un testo descrittivo per gli utenti che non possono visualizzare l'immagine, probabilmente per i seguenti motivi:

- Sono ipovedenti. Gli utenti con disabilità visive significative utilizzano spesso strumenti chiamati screen reader per leggere loro il testo alternativo.
- Qualcosa è andato storto causando la mancata visualizzazione dell'immagine. Ad esempio, prova a modificare deliberatamente il percorso all'interno dell'attributo `src` per renderlo errato. Se salvi e ricarichi la pagina, dovresti vedere qualcosa di simile a questo al posto dell'immagine:

My test image

Le parole chiave per il testo alternativo sono "testo descrittivo". Il testo alternativo che scrivi dovrebbe fornire al lettore informazioni sufficienti per avere una buona idea di ciò che l'immagine trasmette. In questo esempio, il testo corrente di "My test image" ("La mia immagine di prova") non è affatto valido. Un'alternativa, decisamente migliore, per il nostro logo di Firefox sarebbe: "Il logo di Firefox: una volpe fiammeggiante che circonda la Terra".

Prova subito a trovare un testo alternativo migliore per la tua immagine.

CONTRASSEGNARE IL TESTO

Questa sezione tratterà alcuni elementi HTML essenziali che utilizzerai per contrassegnare il testo.

INTESTAZIONI

Gli elementi di intestazione consentono di specificare che alcune parti del contenuto

siano intestazioni o sottotitoli. Allo stesso modo in cui un libro ha il titolo principale, i titoli dei capitoli e i sottotitoli, anche un documento HTML può farlo. L'HTML contiene 6 livelli di intestazione, `<h1>` - `<h6>`, anche se in genere ne usi solo 3 o 4 al massimo:

```
<!-- 4 heading levels: -->
```

```
<h1>My main title</h1>
```

```
<h2>My top level heading</h2>
```

```
<h3>My subheading</h3>
```

```
<h4>My sub-subheading</h4>
```

```
<!-- 4 livelli di intestazione: -->
```

```
<h1>Il mio titolo principale</h1>
```

```
<h2>La mia intestazione di livello superiore</h2>
```

```
<h3>Il mio sottotitolo</h3>
```

```
<h4>Il mio sotto- sottotitolo</h4>
```

Nota: Tutto ciò che è in HTML compreso tra `<!--` e `-->` (come `<!-- 4 livelli di intestazione: -->`), è un commento HTML. Il browser ignora i commenti durante il rendering del codice. In altre parole, non sono visibili nella pagina, ma solo nel codice. I commenti HTML sono un modo per scrivere note utili sul codice o sulla logica.

Ora prova ad aggiungere un **titolo adatto** alla tua pagina HTML appena sopra l'elemento ``.

Nota: Vedrai che il tuo livello di intestazione 1 ha uno stile implicito. Non modificare gli elementi di intestazione ingrandendone il testo, perché vengono utilizzati per l'accessibilità e per altri motivi, ad esempio per la SEO. Cerca di creare una sequenza significativa di titoli sulle tue pagine, senza "saltare" i livelli.

PARAGRAFI

Come spiegato sopra, gli elementi `<p>` servono a contenere paragrafi di testo; li utilizzerai, frequentemente, quando eseguirai il markup del contenuto di testo normale:

```
<p>This is a single paragraph</p>
```

```
<p>Questo è un singolo paragrafo</p>
```

Aggiungi il tuo testo di esempio in uno o più paragrafi, posizionati direttamente sotto l'elemento ``.

LISTE

Gran parte del contenuto del web è costituito da elenchi e l'HTML ha elementi speciali per questi. Le liste di markup sono sempre composte da almeno 2 elementi. I tipi di elenco più comuni sono gli elenchi ordinati e non ordinati:

- Gli elenchi non ordinati sono utilizzati per gli elenchi in cui l'ordine degli elementi

non ha importanza, ad esempio una lista della spesa. Questi sono racchiusi in un elemento `` (unordered list)(lista non ordinata).

- Gli elenchi ordinati sono utilizzati per gli elenchi in cui l'ordine degli elementi è importante, ad esempio una ricetta. Questi sono racchiusi in un elemento `` (ordered list)(lista ordinata).

Ogni elemento all'interno delle liste viene inserito all'interno di un elemento `` (voce della lista).

Ad esempio, se volessimo trasformare la parte del frammento di paragrafo seguente in un elenco:

```
<p>
```

At Mozilla, we're a global community of technologists, thinkers, and builders working together...

```
</p>
```

```
<p>
```

In Mozilla siamo una comunità globale di tecnici, pensatori e costruttori che lavorano insieme...

```
</p>
```

Potremmo modificare il markup in questo modo:

```
<p>At Mozilla, we're a global community of</p>
```

```
<ul>
```

```
<li>technologists</li>
```

```
<li>thinkers</li>
```

```
<li>builders</li>
```

```
</ul>
```

```
<p>working together...</p>
```

```
<p>In Mozilla siamo una comunità globale di</p>
```

```
<ul>
```

```
<li>tecnici</li>
```

```
<li>pensatori</li>
```

```
<li>costruttori</li>
```

```
</ul>
```

```
<p>che lavorano insieme...</p>
```

Prova ad aggiungere un elenco ordinato o non ordinato alla pagina di esempio, salva il file html nella cartella sito e, facendo doppio click su di esso, verifica il risultato nel tuo browser.

LINKS

I links sono molto importanti: sono ciò che rende il web un web ! Per aggiungere un link, dobbiamo usare un elemento semplice - `<a>` - (“a” è la forma abbreviata di “anchor”). Per trasformare il testo all’interno del paragrafo in un link, procedi nel seguente modo:

- Scegli del testo. Abbiamo scelto il testo “Manifesto Mozilla”.
- Disponi il testo in un elemento `<a>`, come illustrato di seguito:

```
<a>Mozilla Manifesto</a>
```

- Assegna un attributo `href` all’elemento `<a>`, come illustrato di seguito:

```
<a href="">Mozilla manifesto</a>
```

Inserisci il valore di questo attributo, tra le virgolette, con l’indirizzo web di cui vuoi inserire il link:

```
<a href="https://www.mozilla.org/en-GB/about/manifesto/">
  Mozilla Manifesto
</a>
```

È possibile ottenere risultati imprevisti se si omette la parte `https://` o `http://`, denominata protocollo, all’inizio dell’indirizzo Web. Dopo aver creato un link, fai clic su di esso per assicurarti che ti stia indirizzando dove volessi.

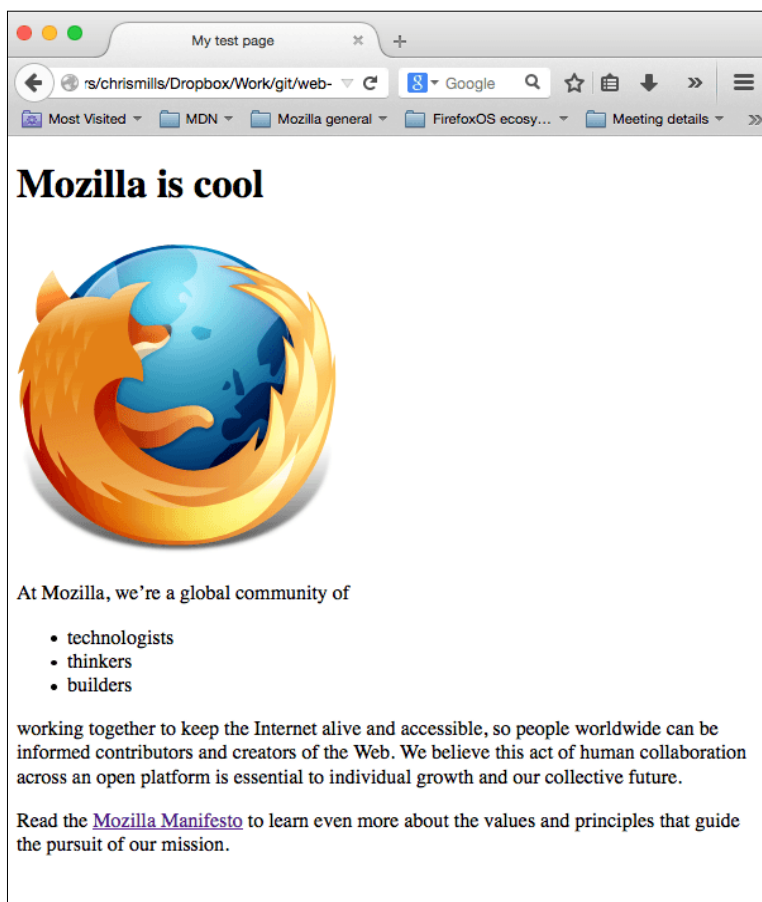
Nota: all’inizio potrebbe sembrare una scelta piuttosto oscura `href` per il nome di un attributo. Se hai difficoltà a ricordarlo, ricorda che `href` è l’acronimo di hypertext reference.

Aggiungi subito un link alla tua pagina, se non l’hai già fatto, salva il file html nella cartella sito e, facendo doppio click su di esso, verifica il risultato nel tuo browser.

CONCLUSIONE

Seguendo tutte le istruzioni in questo percorso, dovresti essere in grado di “costruire” pagine web simili a quelle qui riportate:

(versione inglese)



(Questa pagina puoi anche visualizzarla a questo [indirizzo](#)):

(Il codice html finale di questa versione inglese puoi visualizzarlo a pagina 23):

(versione italiana)



(Il codice html finale di questa versione italiana puoi visualizzarlo a pagina 24):

CODICE HTML FINALE PAGINA WEB NELLA VERSIONE INGLESE

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My test page</title>
  </head>
  <body>
    <h1>Mozilla is cool</h1>
    

    <p>At Mozilla, we're a global community of</p>

    <ul> <!-- changed to list in the tutorial -->
      <li>technologists</li>
      <li>thinkers</li>
      <li>builders</li>
    </ul>

    <p>working together to keep the Internet alive and accessible, so people worldwi-
de can be informed contributors and creators of the Web.
    We believe this act of human collaboration across an open platform is essential to
individual growth and our collective future.</p>

    <p>Read the <a href="https://www.mozilla.org/en-US/about/manifesto/">Mozilla
Manifesto</a>
    to learn even more about the values and principles that guide the pursuit of our
mission.</p>
  </body>
</html>

```

CODICE HTML FINALE PAGINA WEB NELLA VERSIONE ITALIANA

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>My test page</title>
  </head>
  <body>
    <h1>Mozilla è fantastico</h1>
    

    <p>In Mozilla siamo una comunità globale di</p>

    <ul> <!-- changed to list in the tutorial -->
      <li>tecnologi</li>
      <li>pensatori</li>
      <li>costruttori</li>
    </ul>

    <p>che lavorano insieme per mantenere Internet vivo e accessibile, in modo che le persone di tutto il mondo possano essere contributori e creatori informati del Web.
    Riteniamo che questo atto di collaborazione umana attraverso una piattaforma aperta sia essenziale per la crescita individuale e il nostro futuro collettivo.</p>

    <p>Leggi il <a href="https://www.mozilla.org/en-US/about/manifesto/">Manifesto Mozilla</a>
    per saperne di più sui valori e i principi che guidano il perseguimento della nostra missione.</p>
  </body>
</html>
```

NOZIONI DI BASE SUI CSS

CSS (Cascading Style Sheets) è il codice che modella i contenuti web. Le nozioni di base sui CSS illustrano ciò di cui hai bisogno per iniziare. Risponderemo a domande come: Come faccio a rendere il testo rosso? Come faccio a fare in modo che il contenuto venga visualizzato in una determinata posizione nel layout (pagina Web)? Come faccio a decorare la mia pagina web con immagini e colori di sfondo?

CHE COS'È IL CSS?

Come l'HTML, il CSS non è un linguaggio di programmazione. Non è nemmeno un linguaggio di markup. CSS è un linguaggio per fogli di stile. CSS è ciò che usi per stilizzare selettivamente gli elementi HTML. Ad esempio, questo CSS seleziona il testo in paragrafi, impostando il colore sul rosso:

```
p {  
  color: red;  
}
```

Proviamoci! Utilizzando un editor di testo, incolla le tre righe di CSS (sopra) in un nuovo file. Salva il file con il nome **style.css** in una directory denominata **styles**. Per far funzionare il codice, devi, comunque, applicare questo CSS (sopra) al tuo documento HTML. In caso contrario, lo stile non modificherà l'aspetto dell'HTML.

- Apri il file index.html (o il file, con il nome da te scelto e salvato, che contiene tutto il codice html fin qui elaborato). Incolla la seguente riga nell'intestazione (tra l'</head> e i tag):

```
<link href="styles/style.css" rel="stylesheet" />
```

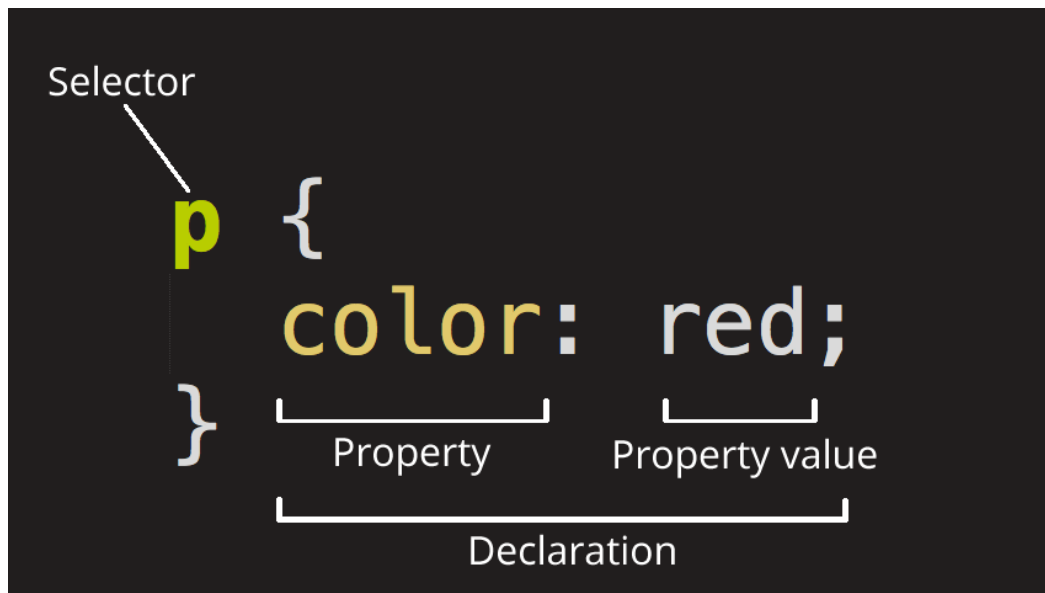
Salvalo e caricalo nel tuo browser. Se il testo del paragrafo è rosso, congratulazioni! Il tuo CSS funziona!

Dovresti vedere qualcosa di simile a questo



ANATOMIA DI UN SET DI REGOLE CSS

Analizziamo il codice CSS per il testo del paragrafo rosso per capire come funzioni:



L'intera struttura è chiamata **set di regole**. (Il termine set di regole è spesso indicato come **regola semplice**). Notare i nomi delle singole parti:

Selettore

Questo è il nome dell'elemento HTML all'inizio del set di regole. Definisce gli elementi a cui applicare lo stile (in questo esempio, gli elementi `<p>`). Per applicare uno stile a un elemento diverso, modificare il selettore.

Dichiarazione

Si tratta di una singola regola come **color: red;**. Specifica a quale proprietà dell'elemento si desidera applicare lo stile.

Proprietà

Questi sono i modi in cui è possibile applicare uno stile a un elemento HTML. In questo esempio, **color** è una proprietà degli elementi `<p>`). In CSS, è possibile scegliere le proprietà che si desidera modificare nella regola.

Valore della proprietà

A destra della proprietà **color**, dopo i due punti, c'è il valore della proprietà. In questo modo si sceglie uno dei tanti aspetti possibili per una determinata proprietà. Ad esempio, sono presenti molti valori oltre a **red**.

Si notino le altre parti importanti della sintassi:

A parte il selettore, ogni set di regole deve essere racchiuso tra parentesi graffe. (`{}`) All'interno di ogni dichiarazione, è necessario utilizzare i due punti (`:`) per separare la proprietà dal valore o dai valori.

All'interno di ogni set di regole, è necessario utilizzare un punto e virgola (;) per separare ogni dichiarazione da quella successiva.

Per modificare più valori di proprietà in un set di regole, scriverli separati da punti e virgola, in questo modo:

```
p {
  color: red;
  width: 500px;
  border: 1px solid black;
}
```

SELEZIONE DI PIÙ ELEMENTI

È inoltre possibile selezionare più elementi e applicare un unico set di regole a tutti gli elementi. Separa più selettori con virgole. Per esempio:

```
p,
li,
h1 {
  color: red;
}
```

DIVERSI TIPI DI SELETTORI

Esistono molti tipi diversi di selettori. Negli esempi precedenti vengono utilizzati i selettori di elementi, che selezionano tutti gli elementi di un determinato tipo. Ma possiamo anche fare selezioni più specifiche.

Per ulteriori informazioni, consulta la [guida ai selettori MDN](#).

CARATTERI E TESTO

Ora che abbiamo esplorato alcuni concetti fondamentali sui CSS, miglioriamo l'aspetto dell'esempio aggiungendo più regole e informazioni al file style.css

- Innanzitutto, trovato l'output di [Google Fonts](#) salvato in precedenza, aggiungi al file index.html (o al file come da te nominato) l'elemento `<link>` da qualche parte all'interno della intestazione (ovunque tra la `</head>` e i tag) così come segue:

```
<link
  href="https://fonts.googleapis.com/css?family=Open+Sans"
  rel="stylesheet" />
```

Questo codice collega la pagina a un foglio di stile che carica la famiglia di font Open Sans con la pagina Web.

- Quindi, riapri il file `style.css` dalla cartella styles, elimina quanto scritto in precedenza (a proposito del colore rosso dei paragrafi). È stato un buon test, ma preparati ad inserire nuovi elementi.
- Aggiungi le righe mostrate di seguito. La proprietà `font-family` si riferisce ai tipi di carattere che si desidera utilizzare per il testo. Questa regola `font-family` definisce

un carattere **font-family** di base globale e una dimensione **font-size** del carattere per l'intera pagina. Poiché `<html>` è l'elemento padre dell'intera pagina, tutti gli elementi al suo interno ereditano lo stesso **font-family**.

```
html {  
  font-size: 10px; /* px means "pixels": the base font size is now 10 pixels high */  
  font-family: "Open Sans", sans-serif; /* this should be the rest of the output you got  
from Google Fonts */  
}
```

Nota: Tutto ciò che è in CSS tra `/*` e `*/` è un commento CSS. Il browser ignora i commenti durante il rendering del codice. I commenti CSS sono un modo per scrivere note utili sul codice o sulla logica.

Ora impostiamo le dimensioni dei caratteri per gli elementi che avranno testo all'interno del corpo HTML (`<h1>`, `` e `<p>`). Centriamo anche l'intestazione. Infine, espandiamo il secondo set di regole (sotto) con le impostazioni per l'altezza della riga e la spaziatura delle lettere per rendere il contenuto del corpo più leggibile.

```
h1 {  
  font-size: 60px;  
  text-align: center;  
}  
  
p,  
li {  
  font-size: 16px;  
  line-height: 2;  
  letter-spacing: 1px;  
}
```

Regola i valori **px** come preferisci. Il tuo lavoro in corso dovrebbe essere simile a quanto indicato nella pagina seguente.



I BOXES IN CSS

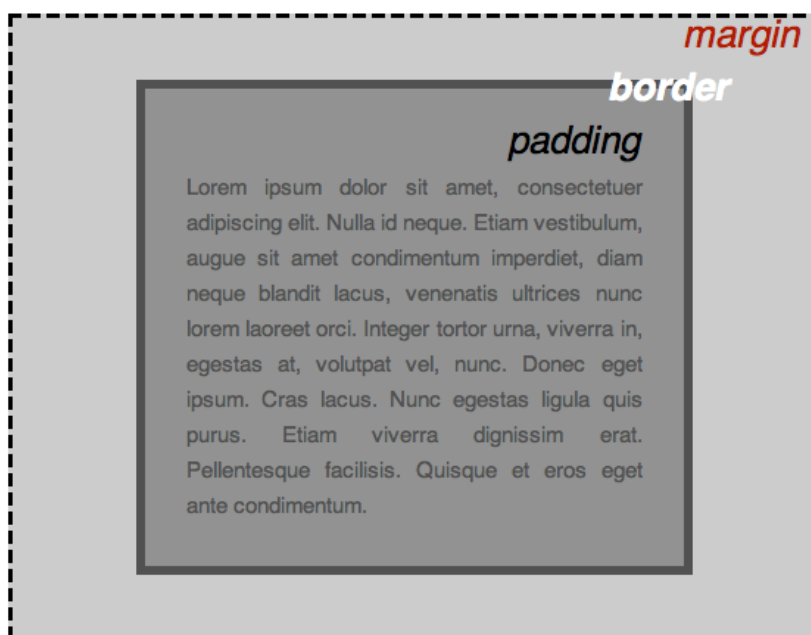
Elementi importanti in CSS sono i **boxes**. Fondamentale è l'impostazione delle dimensioni, del colore e della posizione. La maggior parte degli elementi HTML della pagina può essere considerata come un riquadro sovrapposto ad altri riquadri.



Foto da <https://www.geograph.org.uk/photo/3418115> Copyright © Jim Barton cc-by-sa/2.0

Il layout CSS si basa principalmente sul **modello a scatola**. Ogni casella che occupa spazio sulla tua pagina ha proprietà come:

- **padding**, lo spazio intorno al contenuto. Nell'esempio seguente, è lo spazio intorno al testo del paragrafo.
- **border**, la linea continua che si trova appena fuori dalla spaziatura interna.
- **margin**, lo spazio intorno all'esterno del bordo.



In questa sezione utilizziamo anche:

- **width**, (larghezza di un elemento).
- **background-color**, il colore dietro il contenuto e la spaziatura interna di un elemento.
- **color**, il colore del contenuto di un elemento (in genere testo).
- **text-shadow**, Imposta un'ombreggiatura sul testo all'interno di un elemento.
- **display**, imposta la modalità di visualizzazione di un elemento

Per continuare, aggiungiamo altri CSS. Continua ad aggiungere queste nuove regole nella parte inferiore del file **style.css** . Sperimenta, con la variazione dei valori, le modifiche relative.

MODIFICA DEL COLORE DELLA PAGINA

```
html {
  background-color: #00539f;
}
```

Questa regola imposta un colore di sfondo per l'intera pagina. Cambia il codice colore utilizzando il selettore colore (come indicato nel capitolo “Come apparirà il mio sito web ?”).

STILE DEL CORPO DELLA PAGINA

```
body {
  width: 600px;
  margin: 0 auto;
  background-color: #ff9500;
  padding: 0 20px 20px 20px;
  border: 5px solid black;
}
```

Sono disponibili diverse dichiarazioni per l'elemento **<body>**. Esaminiamoli riga per riga:

- **width: 600px**; Questo costringe il corpo ad avere sempre una larghezza di 600 pixel.
- **margin: 0 auto**; Quando si impostano due valori su una proprietà come **margin** o **padding** , il primo valore influisce sul lato superiore e inferiore dell'elemento (impostandolo su **0** in questo caso); Il secondo valore influisce sul lato sinistro e destro. (In questo caso, **auto** è un valore speciale che divide lo spazio orizzontale disponibile in modo uniforme tra sinistra e destra).
- **background-color: #FF9500**; Imposta il colore di sfondo dell'elemento. Questo progetto utilizza un arancione rossastro per il colore di sfondo del corpo, in contrasto con il blu scuro per l'elemento **<html>**. (Sentiti libero di sperimentare.)
- **padding: 0 20px 20px 20px**; In questo modo vengono impostati quattro valori

per la spaziatura interna. L'obiettivo è quello di mettere un po' di spazio intorno al contenuto. In questo esempio, non c'è spaziatura interna nella parte superiore del corpo e 20 pixel a destra, in basso e a sinistra. I valori sono impostati in quest'ordine: in alto, a destra, in basso, a sinistra. Come per **margin**, è possibile utilizzare uno, due, tre o quattro valori.

- **border: 5px solid black;** Imposta i valori per la larghezza, lo stile e il colore del bordo. In questo caso, si tratta di un bordo nero solido largo cinque pixel, su tutti i lati del corpo.

POSIZIONAMENTO E STILE DEL TITOLO DELLA PAGINA PRINCIPALE

```
h1 {  
  margin: 0;  
  padding: 20px 0;  
  color: #00539f;  
  text-shadow: 3px 3px 1px black;  
}
```

Potresti aver notato che c'è un'orribile fessura nella parte superiore del corpo. Ciò accade perché i browser applicano lo stile predefinito all'elemento h1 (tra gli altri). Potrebbe sembrare una cattiva idea, ma l'intento è quello di fornire una leggibilità di base per le pagine senza stile. Per eliminare lo spazio, sovrascriviamo lo stile predefinito del browser con l'impostazione **margin: 0**

Successivamente, impostiamo la spaziatura interna superiore e inferiore dell'intestazione su 20 pixel.

Quindi, impostiamo il testo dell'intestazione in modo che sia dello stesso colore del colore di sfondo HTML (per esempio:black).

Infine, applica un'ombreggiatura **text-shadow** al contenuto di testo dell'elemento. I suoi quattro valori sono:

1. Il **primo valore** in pixel imposta l'**offset orizzontale** dell'ombra dal testo: quanto si sposta attraverso.
2. Il **secondo valore** in pixel imposta l'**offset verticale** dell'ombra dal testo: quanto si sposta verso il basso.
3. Il **terzo valore** di pixel imposta il **raggio di sfocatura dell'ombra**. Un valore maggiore produce un'ombreggiatura dall'aspetto più sfocato.
4. Il **quarto valore** imposta il **colore di base dell'ombra**.

Prova a sperimentare con valori diversi per vedere come cambi l'aspetto.

CENTRATURA DELL'IMMAGINE

```
img {  
  display: block;  
  margin: 0 auto;  
}
```

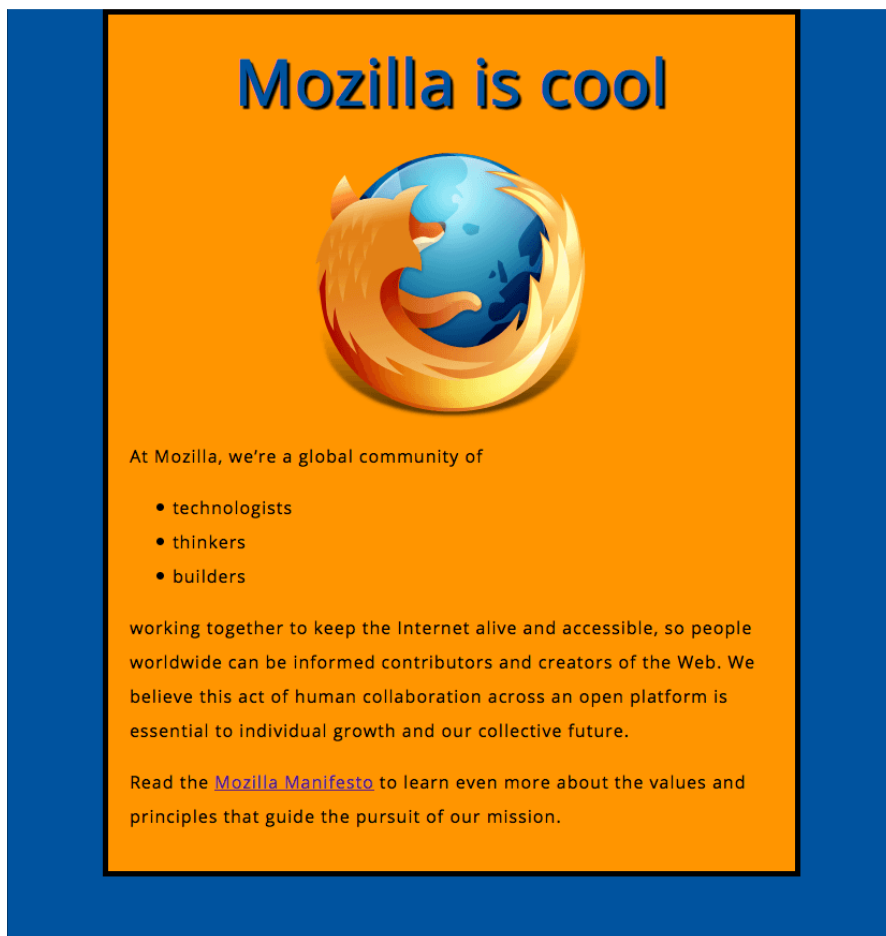
Successivamente, centriamo l'immagine per migliorarne l'aspetto. Potremmo usare di nuovo il trucco come abbiamo fatto per il corpo. Ma ci sono differenze che richiedono un'impostazione aggiuntiva per far funzionare il CSS.

Il `<body>` è un elemento a blocchi, il che significa che occupa spazio nella pagina. Il margine applicato a un elemento di blocco verrà rispettato dagli altri elementi della pagina. Al contrario, le immagini sono elementi in linea, affinché il trucco del margine automatico funzioni su questa immagine, dobbiamo dargli un comportamento a livello di blocco usando `.display: block;`

Nota: Le istruzioni precedenti presuppongono che tu stia utilizzando un'immagine più piccola della larghezza impostata sul corpo. (600 pixel) Se l'immagine è più grande, traboccherà dal corpo, riversandosi nel resto della pagina. Per risolvere questo problema, puoi: 1) ridurre la larghezza dell'immagine utilizzando un editor grafico o 2) utilizzare CSS per ridimensionare l'immagine impostando la proprietà `width` sull'elemento con un valore inferiore.

CONCLUSIONE

Se sono state seguite tutte le istruzioni, la pagina dovrebbe essere così strutturata:



NOZIONI DI BASE SU JAVASCRIPT

JavaScript è un linguaggio di programmazione che aggiunge interattività al tuo sito web. Questo accade nei giochi, nel comportamento delle risposte quando vengono premuti i pulsanti o con l'inserimento dei dati sui moduli; con uno stile dinamico; con animazione, ecc. Questo capitolo ti aiuta a iniziare con JavaScript e ti aiuta a comprendere meglio ciò che sia possibile fare.

CHE COS'È JAVASCRIPT?

JavaScript è un potente linguaggio di programmazione in grado di aggiungere interattività a un sito web. È stato inventato da Brendan Eich.

JavaScript è versatile e adatto ai principianti. Con più esperienza, sarai in grado di creare giochi, grafica animata 2D e 3D, app complete basate su database e molto altro ! JavaScript stesso è relativamente compatto, ma molto flessibile. Gli sviluppatori hanno scritto una varietà di strumenti basati sul linguaggio JavaScript di base, sbloccando una vasta quantità di funzionalità con il minimo sforzo. Questi includono:

- Interfacce di programmazione delle applicazioni (API) integrate nei browser Web, che forniscono funzionalità come la creazione dinamica di HTML e l'impostazione di stili CSS; la raccolta e la manipolazione di un flusso video dalla webcam di un utente o la generazione di grafica 3D e campioni audio.
- API di terze parti che consentono agli sviluppatori di incorporare funzionalità nei siti di altri fornitori di contenuti, come Twitter o Facebook.
- Framework e librerie di terze parti che è possibile applicare all'HTML per accelerare il lavoro di creazione di siti e applicazioni.

Il capitolo seguente introduce alcuni aspetti del linguaggio di base e offre l'opportunità di giocare anche con alcune funzionalità API del browser. Buon divertimento !

UN ESEMPIO DI "HELLO WORLD!"

JavaScript è una delle tecnologie web moderne più popolari ! Man mano che le tue competenze JavaScript crescono, i tuoi siti web entrano in una nuova dimensione di potenza e creatività.

Tuttavia, prendere confidenza con JavaScript è più impegnativo che prendere confidenza con HTML e CSS. Potrebbe essere necessario iniziare in piccolo e progredire gradualmente. Per iniziare, esaminiamo come aggiungere JavaScript alla tua pagina per creare un esempio **Hello world ! (Ciao mondo !)**.

- Nella cartella di prova, chiamata **sito**, crea una nuova cartella denominata **scripts**. All'interno della cartella **scripts**, crea un nuovo documento di testo denominato **main.js** e salvarlo.
- Nel file **index.html**, inserisci questo codice in una nuova riga, appena prima del tag di chiusura **</body>**:

```
<script src="scripts/main.js"></script>
```

- Questo fa lo stesso lavoro dell'elemento **<link>** per CSS. Applica il JavaScript alla pagina, in modo che possa avere un effetto sull'HTML (insieme al CSS e a qualsiasi

altra cosa sulla pagina).

- Aggiungi il codice seguente al file: **main.js**

```
const myHeading = document.querySelector("h1");
myHeading.textContent = "Hello world!";
```

Assicurati che i file HTML e JavaScript siano salvati. Quindi carica il file index.html nel tuo browser. Dovresti vedere qualcosa di simile a questo:



Nota: Il motivo per cui le istruzioni (sopra) posizionano l'elemento `<script>` vicino alla parte inferiore del file HTML è che il browser legge il codice nell'ordine in cui appare nel file.

Se il JavaScript viene caricato per primo e si suppone che influisca sull'HTML che non è ancora stato caricato, potrebbero esserci problemi. Posizionare JavaScript nella parte inferiore di una pagina HTML è un modo per adattarsi a questa dipendenza.

COS'È SUCCESSO?

Il testo dell'intestazione è stato modificato in Hello world! utilizzando JavaScript. A tale scopo, è stato utilizzata una **funzione** chiamata `querySelector()` per acquisire un riferimento all'intestazione e quindi memorizzarlo in una variabile denominata **myHeading**. Questo è simile a quello che abbiamo fatto usando i selettori CSS. Quando si desidera eseguire un'operazione su un elemento, è necessario prima selezionarlo.

Successivamente, il codice imposta il valore della proprietà `textContent` della variabile **myHeading** (che rappresenta il contenuto dell'intestazione) su **Hello world!**.

Nota: Entrambe le funzionalità utilizzate in questo esercizio fanno parte dell'API DOM (Document Object Model), che consente di modificare i documenti.

CORSO INTENSIVO DI NOZIONI DI BASE SULLA LINGUA

Per darti una migliore comprensione di come funzioni JavaScript, spieghiamo alcune delle caratteristiche principali del linguaggio. Vale la pena notare che queste caratteristiche sono comuni a tutti i linguaggi di programmazione. Se padroneggi questi fondamenti, hai un vantaggio anche nella programmazione in altri linguaggi!

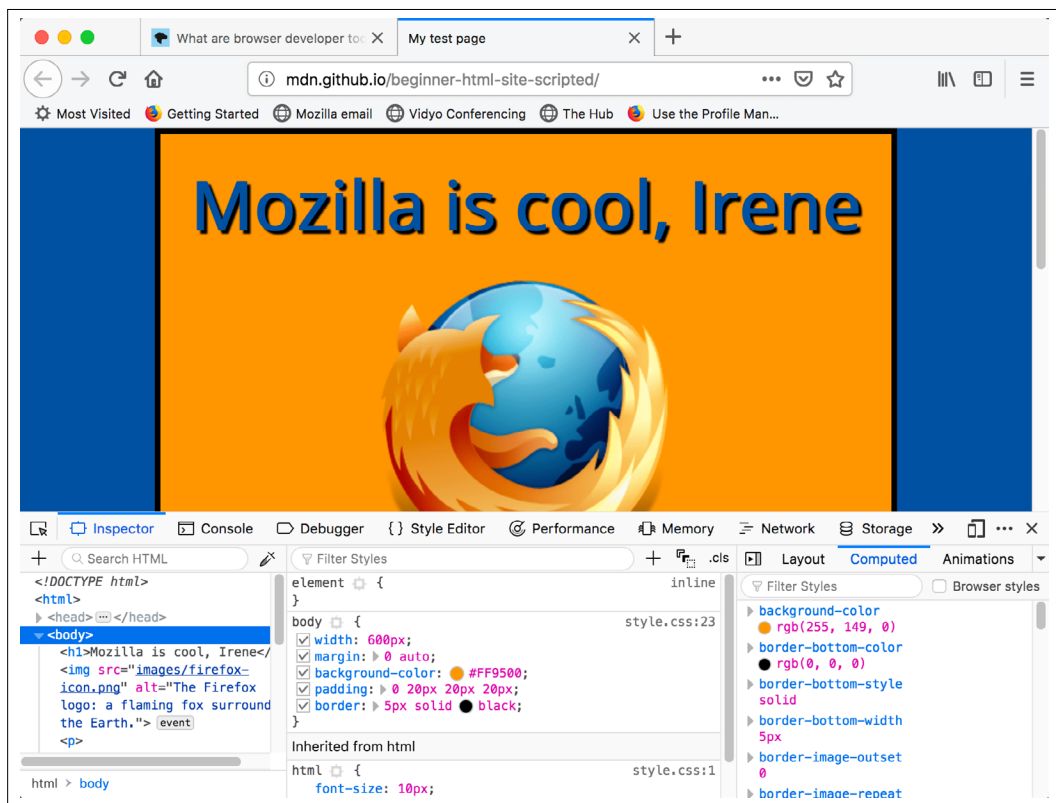
Nota: in questo capitolo, prova a inserire le righe di codice di esempio nella console JavaScript per vedere cosa accada.

COSA SONO GLI STRUMENTI DI BROWSER PER SVILUPPATORI?

Ogni browser web moderno include una potente suite di strumenti per sviluppatori. Questi strumenti fanno una serie di cose, dall'ispezione di HTML, CSS e JavaScript attualmente caricati alla visualizzazione di quali risorse la pagina ha richiesto e quanto tempo hanno impiegato per caricarsi.

COME APRIRE DEVTOOLS NEL BROWSER

Gli strumenti di sviluppo si trovano all'interno del browser in una sottofinestra che assomiglia più o meno alla seguente, a seconda del browser che si sta utilizzando:



Si aprono in tre modi:

Tastiera:

Windows: (Ctrl + Maiusc + I) o F12

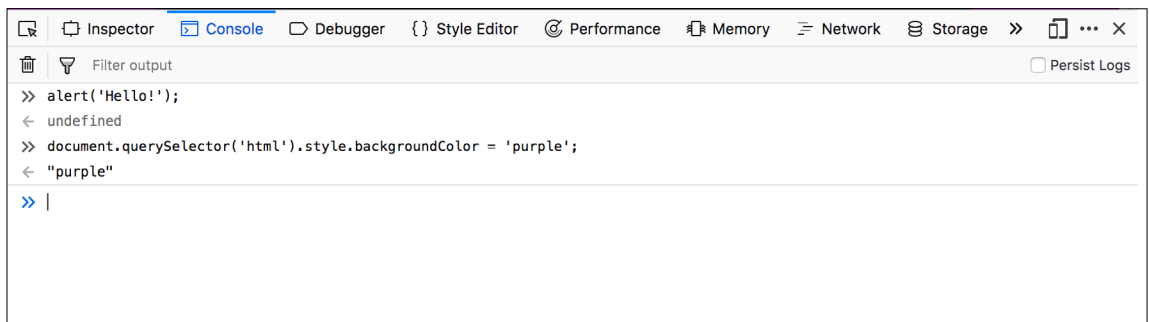
Barra dei menu:

Chrome: Altri strumenti → Strumenti per sviluppatori

Menu contestuale: Tieni premuto/fai clic con il **pulsante destro del mouse** su un punto in una pagina Web (Ctrl-clic su Mac) e scegli **Ispeziona (Esamina)** dal menu di scelta rapida visualizzato.

LA CONSOLE JAVASCRIPT

La console JavaScript è uno strumento incredibilmente utile per il debug di JavaScript che non funziona come previsto. Consente di eseguire righe di JavaScript sulla pagina attualmente caricata nel browser e segnala gli errori riscontrati durante il tentativo del browser di eseguire il codice. Per accedere alla console in qualsiasi browser: Se gli strumenti di sviluppo sono già aperti, fare clic o premere la scheda **Console**. Si aprirà una finestra come la seguente:



Per vedere cosa accade, prova a immettere i seguenti frammenti di codice nella console uno per uno (e quindi premi INVIO):

```
alert("hello!");
```

```
document.querySelector("html").style.backgroundColor = "purple";
```

```
const loginImage = document.createElement("img");
loginImage.setAttribute(
    "src",
    "https://raw.githubusercontent.com/mdn/learning-area/master/html/forms/image-type-example/login.png",
);
document.querySelector("h1").appendChild(loginImage);
```

VARIABILI

Le variabili sono contenitori in cui sono archiviati i valori. Si inizia dichiarando una variabile con la parola chiave **let**, seguita dal nome assegnato alla variabile:

```
let myVariable;
```

Un punto e virgola alla fine di una riga indica il punto in cui termina un'istruzione. È possibile assegnare qualsiasi nome a una variabile ma con alcune restrizioni. I **nomi delle variabili**, chiamati **identificatori**, sono, infatti, conformi a determinate regole.

Un identificatore JavaScript inizia in genere con una lettera, un carattere di sottolineatura (`_`) o un simbolo di dollaro (`$`). I caratteri successivi possono anche essere cifre (`0-9`). Poiché JavaScript fa distinzione tra maiuscole e minuscole, le lettere possono includere i caratteri dalla **A** alla **Z** (maiuscolo) e dalla **a** alla **z** (minuscolo).

Questo significa che **myVariable** non è la stessa cosa di **myvariable**.

Alcuni esempi di nomi legali sono: **Number_hits**, **temp99**, **\$credit**, e **_name**.

Dopo aver dichiarato una variabile, è possibile assegnarle un valore:

```
myVariable = "Bob";
```

Inoltre, è possibile eseguire entrambe le operazioni sulla stessa riga:

```
let myVariable = "Bob";
```

Per recuperare il valore, chiamare il nome della variabile:

```
myVariable;
```

Dopo aver assegnato un valore a una variabile, è possibile modificarlo in un secondo momento nel codice:

```
let myVariable = "Bob";
```

```
myVariable = "Steve";
```

Si noti che le variabili possono contenere valori con tipi di dati diversi:

| VARIABILE | SPIEGAZIONE | ESEMPIO |
|----------------|--|--|
| STRINGA | Si tratta di una sequenza di testo nota. Per significare che il valore è una stringa, racchiuderla tra virgolette singole o doppie. | <code>let myVariable = 'Bob';</code> <code>let myVariable = "Bob";</code> |
| NUMERO | I numeri non hanno virgolette intorno. | <code>let myVariable = 10;</code> |

| VARIABILE | SPIEGAZIONE | ESEMPIO |
|-----------|---|---|
| BOOLEANO | Si tratta di un valore Vero/Falso. Le parole true e false sono parole chiave speciali che non necessitano di virgolette. | <code>let myVariable = true;</code> |
| ARRAY | Si tratta di una struttura che consente di memorizzare più valori in un unico riferimento. | <code>let myVariable = [1,'Bob','Steve',10];</code> Fare riferimento a ciascuno dei membri dell'array in questo modo: <code>myVariable[0], myVariable[1]</code> |
| OGGETTO | Può trattarsi di qualsiasi cosa. Tutto in JavaScript è un oggetto e può essere memorizzato in una variabile. Tienilo a mente mentre impari. | Tutti gli esempi di cui sopra. |

Allora perché abbiamo bisogno di variabili ?

Le variabili sono necessarie per fare qualcosa di interessante nella programmazione. Se i valori non possono cambiare, non è possibile eseguire alcuna operazione dinamica, ad esempio personalizzare un messaggio di saluto o modificare un'immagine visualizzata in una raccolta di immagini.

COMMENTI

I commenti sono frammenti di testo che possono essere aggiunti insieme al codice. Il browser ignora il testo contrassegnato come commento. Puoi scrivere commenti in JavaScript proprio come puoi fare in CSS:

```
/*
```

```
Everything in between is a comment.
```

```
*/
```

Se il tuo commento non contiene interruzioni di riga, è possibile inserirlo dietro due barre:

```
// This is a comment
```

OPERATORI

Un **operatore** è un simbolo matematico che produce un risultato basato su due valori (o variabili). Nella tabella seguente puoi vedere alcuni degli operatori più semplici, insieme ad alcuni esempi da provare nella console JavaScript.

| OPERATORE | SPIEGAZIONE | SIMBOLO/I | ESEMPIO |
|---|---|-------------|--|
| ADDIZIONE | Somma due numeri insieme o combina due stringhe. | + | 6 + 9; 'Hello ' + 'world!'; |
| SOTTRAZIONE, MOLTIPLICAZIONE, DIVISIONE | Operazioni come nella matematica di base | - * / | 9 - 3; 8 * 2; 9 / 3; |
| ASSEGNAZIONE | Assegna un valore a una variabile | = | let myVariable = 'Bob'; |
| UGUAGLIANZA RIGOROSA | In questo modo viene eseguito un test per verificare se due valori sono uguali e dello stesso tipo di dati. Restituisce un risultato true / false (booleano). | === | let myVariable = 3; myVariable === 4; |
| NOT, NON-È-UGUALE | In questo modo viene restituito il valore logicamente opposto a quello precedente. Quando viene utilizzato accanto all'operatore di uguaglianza, l'operatore di negazione verifica se i due valori non sono uguali. | ! !== | Per "Not",(!) l'espressione di base è true , ma il confronto restituisce false perché lo neghiamo: let myVariable = 3; !(myVariable === 3); "Non-è-uguale"(!==) restituisce lo stesso risultato con diversa sintassi. let myVariable = 3; myVariable !== 3; |

Nota: La combinazione di tipi di dati può portare a risultati strani durante l'esecuzione dei calcoli. Fai attenzione a fare riferimento correttamente alle tue variabili e a ottenere i risultati che ti aspetti. Ad esempio, entra nella tua console ed esegui le seguenti operazioni: '35' + '25' 35 + 25.

Perché non ottieni il risultato che ti aspettavi? Perché le virgolette trasformano i numeri in stringhe, quindi hai finito per concatenare stringhe piuttosto che aggiungere numeri.

CONDIZIONALI

I condizionali sono strutture di codice utilizzate per verificare se un'espressione restituisca true o meno. Una forma molto comune di condizionali è l'istruzione **if...else**. Per esempio:

```
let iceCream = "chocolate";
if (iceCream === "chocolate") {
  alert("Yay, I love chocolate ice cream!");
} else {
  alert("Awwwww, but chocolate is my favorite...");
}
```

L'espressione **if (iceCream === "chocolate")** { è il test. In questo modo viene utilizzato l'operatore di uguaglianza rigorosa (**===**) per confrontare la variabile **iceCream** con la stringa **chocolate** per verificare se le due sono uguali. Se questo confronto restituisce **true**, viene eseguito il primo blocco di codice. Se il confronto non è vero, viene eseguito il secondo blocco di codice, dopo l'istruzione **else**.

FUNZIONI

Le funzioni sono un modo per creare pacchetti di funzionalità che si desidera riutilizzare. È possibile definire un corpo di codice come una funzione che viene eseguita quando si chiama il nome della funzione nel codice. Si tratta di una buona alternativa alla scrittura ripetuta dello stesso codice. Hai già visto alcuni usi delle funzioni. Per esempio:

```
alert("hello!");
```

Questa funzione **alert** è integrata nel browser.

Se vedi qualcosa che assomiglia al nome di una variabile, ma è seguito da parentesi, è probabile che si tratti di una funzione. Le funzioni spesso accettano argomenti: dati di cui hanno bisogno per svolgere il loro lavoro. Gli argomenti vanno all'interno delle parentesi **()**, separati da virgole se è presente più di un argomento.

Ad esempio, la funzione **alert()** fa apparire una finestra pop-up all'interno della finestra del browser, ma dobbiamo assegnarle una stringa come argomento per dire alla funzione quale messaggio visualizzare.

È inoltre possibile definire funzioni personalizzate.

Nell'esempio seguente, creiamo una semplice funzione che prende due numeri come argomenti e li moltiplica:

```
function multiply(num1, num2) {  
  let result = num1 * num2;  
  return result;  
}
```

Esegui la nella console provando con l'aggiunta di diversi argomenti. Per esempio:

```
multiply(4, 7);  
multiply(20, 20);  
multiply(0.5, 3);
```

Nota: L'istruzione **return** indica al browser di restituire la variabile **result** fuori dalla funzione in modo che sia disponibile per l'uso. Ciò è necessario perché le variabili definite all'interno delle funzioni sono disponibili solo all'interno di tali funzioni. Questa operazione è denominata **ambito della variabile**.

EVENTI

La vera interattività su un sito Web richiede gestori di eventi. Si tratta di strutture di codice che ascoltano l'attività nel browser ed eseguono il codice in risposta. L'esempio più ovvio è la gestione dell'evento **click**, che viene attivato dal browser quando si fa click su qualcosa con il mouse. Per dimostrarlo, apri una pagina web qualsiasi e inserisci quanto segue nella console Javascript (a cui accedere facendo click su F12), quindi fai clic sulla pagina Web corrente:

```
document.querySelector("html").addEventListener("click", function () {  
  alert("Questo è un esempio di gestione di un evento !!!");  
});
```

Esistono diversi modi per associare un gestore eventi a un elemento. Qui selezioniamo l'elemento **<html>**. Chiamiamo quindi la sua funzione **addEventListener()**, passando il nome dell'evento da ascoltare ("**click**") e una funzione da eseguire quando si verifica l'evento.

La funzione **addEventListener()** che abbiamo appena passato qui è chiamata **funzione anonima**, perché non ha un nome. Esiste un modo alternativo di scrivere funzioni anonime, che chiamiamo funzione freccia. Una funzione freccia utilizza **() =>** invece di **function ()**

```
document.querySelector("html").addEventListener("click", () => {  
  alert("Questo è un esempio di gestione di un evento !!!");  
});
```

POTENZIARE IL NOSTRO SITO WEB

Aggiungiamo alcune nuove funzionalità al nostro sito.

Prima di procedere, elimina il contenuto corrente del file `main.js`, ovvero la parte aggiunta in precedenza durante l'esempio "Hello world!", e salva il file vuoto. In caso contrario, il codice esistente si scontrerà con il nuovo codice che si sta per aggiungere.

AGGIUNTA DI UN CAMBIO IMMAGINE

In questa sezione imparerai come utilizzare le funzionalità di JavaScript e DOM API per alternare la visualizzazione di una delle due immagini. Questa modifica si verifica quando un utente fa clic sull'immagine visualizzata.

Scegli un'immagine che desideri inserire nel tuo sito. L'immagine dovrebbe avere, approssimativamente, le stesse dimensioni dell'immagine aggiunta in precedenza o il più vicino possibile.

Salva l'immagine nella tua cartella `images`

Rinominare l'immagine `firefox2.png`.

Aggiungi il seguente codice JavaScript al tuo file `main.js`

```
const myImage = document.querySelector("img");

myImage.onclick = () => {
  const mySrc = myImage.getAttribute("src");
  if (mySrc === "images/firefox-icon.png") {
    myImage.setAttribute("src", "images/firefox2.png");
  } else {
    myImage.setAttribute("src", "images/firefox-icon.png");
  }
};
```

Salva tutto nel file `index.html` e caricalo nel browser. Ora, quando fai clic sull'immagine, dovrebbe passare all'altra.

Questo è quello che è successo.

È stato archiviato un riferimento all'elemento `` in `.`. Successivamente, la proprietà del gestore eventi `myImage.onclick` è stata resa uguale a una funzione senza nome (una funzione "anonima") (`myImage.onclick = () => {}`).

Quindi, ogni volta che si fa clic su questo elemento:

1. Il codice recupera il valore dell'attributo `src` dell'immagine.
(`const mySrc = myImage.getAttribute("src");`)
2. Il codice utilizza un condizionale per verificare se il valore `src` è uguale al percorso dell'immagine originale (`if (mySrc === "images/firefox-icon.png") {`
 - In caso affermativo, il codice modifica il valore `src` nel percorso della seconda immagine, forzando il caricamento dell'altra immagine all'interno dell'elemento ``.
 - Se non lo è (significa che deve essere già cambiato), il valore `src` torna al percorso dell'immagine originale.

AGGIUNTA DI UN MESSAGGIO DI BENVENUTO PERSONALIZZATO

Successivamente, cambiamo il titolo della pagina in un messaggio di benvenuto personalizzato quando l'utente visita il sito per la prima volta. Questo messaggio di benvenuto persisterà. Se l'utente lascia il sito e ritorna in un secondo momento, salveremo il messaggio utilizzando l'**API Web Storage**. Includeremo anche un'opzione per cambiare l'utente e, quindi, il messaggio di benvenuto.

1. In **index.html**, aggiungi la seguente riga appena prima dell'elemento **<script>**:

```
<button>Change user</button>
```

2. In **main.js**, inserire il codice seguente nella parte inferiore del file, esattamente come è scritto. In questo modo vengono presi in considerazione i riferimenti al nuovo pulsante e all'intestazione, memorizzando ciascuna variabile interna.

```
let myButton = document.querySelector("button");  
let myHeading = document.querySelector("h1");
```

3. Aggiungi la seguente funzione per impostare il saluto personalizzato. Questo non farà ancora nulla, ma presto cambierà.

```
function setUsername() {  
  const myName = prompt("Please enter your name.");  
  localStorage.setItem("name", myName);  
  myHeading.textContent = `Mozilla is cool, ${myName}`;  
}
```

La funzione **setUsername()** **function setUsername() {** contiene una funzione **prompt()**, che visualizza una finestra di dialogo, simile a **alert()**. Questa funzione **prompt()**, più di **alert()**, non si limita a chiedere all'utente di immettere dati e a memorizzarli in una variabile dopo che l'utente ha fatto clic su OK. In questo caso, viene chiesto all'utente di immettere un nome **const myName = prompt("Please enter your name.");**. Successivamente, il codice chiama un'API, che ci consente di memorizzare i dati nel browser e recuperarli in un secondo momento. Usiamo la funzione di **localStorage.setItem()** **localStorage.setItem("name", myName);** per creare e memorizzare un elemento di dati chiamato **'name'**, impostando il suo valore sulla variabile **myName** che contiene la voce dell'utente per impostare il nome. Infine, impostiamo il **textContent** dell'intestazione su una stringa, più il nome appena memorizzato dell'utente **myHeading.textContent = `Mozilla is cool, \${myName}`;**.

4. Aggiungi il blocco di condizioni seguente. Potremmo chiamarlo **codice di inizializzazione**, in quanto struttura l'app quando viene caricata per la prima volta.

```
if (!localStorage.getItem("name")) {  
  setUsername();  
} else {
```



```
const storedName = localStorage.getItem("name");  
myHeading.textContent = `Mozilla is cool, ${storedName}`;  
}
```

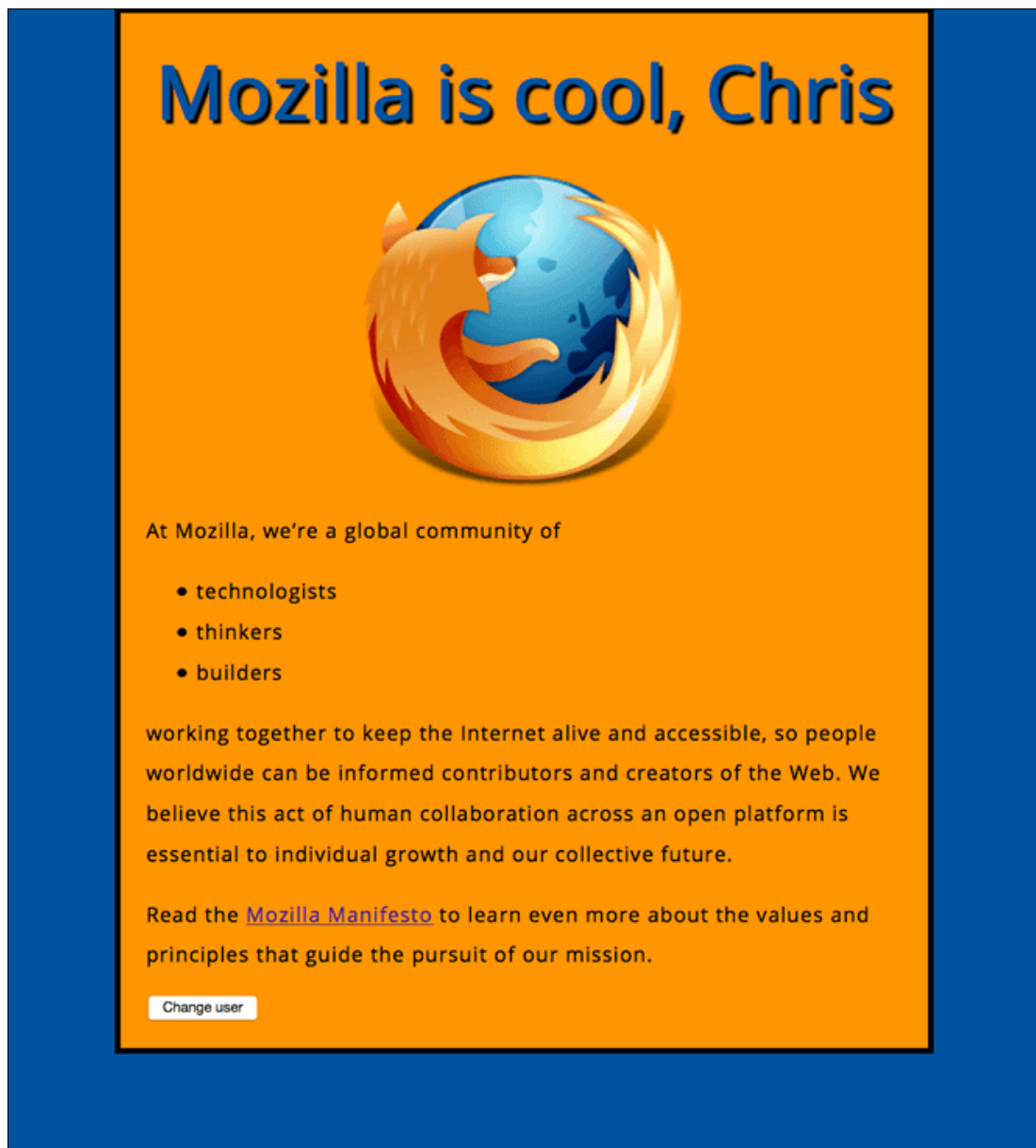
Questa prima riga di questo blocco utilizza l'operatore di negazione (logico NOT, rappresentato da !) `if (!localStorage.getItem("name"))` per verificare se i dati del nome esistono. In caso contrario, la funzione `setUserName()` viene eseguita per crearlo. Se esiste (cioè, l'utente ha impostato un nome utente durante una visita precedente), recuperiamo il nome memorizzato utilizzando `getItem()` `const storedName = localStorage.getItem("name");` e impostiamo il `textContent` (contenuto testuale) dell'intestazione su una stringa, più il nome dell'utente `myHeading.textContent = `Mozilla is cool, ${storedName}``, come abbiamo fatto all'interno di `setUserName()`.

5. Inserisci questo gestore eventi `onclick` (sotto) sul pulsante `myButton.onclick = () => {`. Quando si fa clic su di esso, `setUserName()` viene eseguito. Ciò consente all'utente di immettere un nome diverso premendo il pulsante.

```
myButton.onclick = () => {  
  setUserName();  
};
```

CONCLUSIONE

Se hai seguito tutte le istruzioni in questi capitoli, dovresti ritrovarti con una pagina simile all'immagine seguente.



PUBBLICARE IL TUO SITO WEB

Una volta che hai finito di scrivere il codice e organizzare i file che compongono il tuo sito web, devi metterlo tutto online in modo che le persone possano trovarlo. Questo capitolo spiega come mettere online il tuo semplice codice di esempio con poco sforzo.

QUALI SONO LE OPZIONI?

La pubblicazione di un sito web è un argomento complesso perché ci sono molti modi per farlo. Questo capitolo non tenta di documentare tutti i metodi possibili. Invece, spiega i vantaggi e gli svantaggi di tre approcci pratici per i principianti. Quindi tratta un metodo che può funzionare subito per molti lettori.

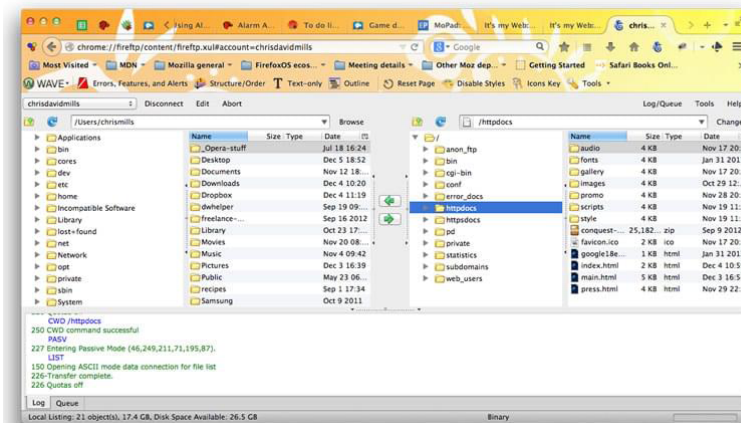
OTTENERE L'HOSTING E UN NOME DI DOMINIO

Per avere un maggiore controllo sui contenuti e sull'aspetto del sito web, la maggior parte delle persone sceglie di acquistare un web hosting e un nome di dominio:

- L'hosting Web è uno spazio file affittato sul server Web di una società di hosting. I file del sito Web vengono inseriti nel server Web. Il server web fornisce il contenuto del sito web ai visitatori del sito web.
- Un nome di dominio è l'indirizzo univoco in cui le persone trovano il tuo sito web, ad esempio: <https://www.mozilla.org> o <https://www.bbc.co.uk>. Puoi affittare il tuo nome di dominio per tutti gli anni che desideri da un **registrar di domini**. Un **registrar di domini** è un'azienda che gestisce e fa da intermediario nella vendita di nomi di dominio univoci su Internet¹. Per vendere domini, l'azienda deve essere accreditata dalla Internet Corporation for Assigned Names and Numbers (ICANN) o un dominio di primo livello con codice paese. Alcuni dei migliori registrar di domini del 2023 includono IONOS, DreamHost, Porkbun, Namecheap, Hostinger, NameSilo e Dynadot.

Molti siti web professionali vanno online in questo modo.

Inoltre, avrai bisogno di un programma FTP (File Transfer Protocol) per trasferire effettivamente i file del sito Web sul server. I programmi FTP variano notevolmente, ma in genere devi connetterti al tuo server web utilizzando i dettagli forniti dalla tua società di hosting (in genere nome utente, password, nome host). Quindi il programma mostra i file locali e i file del server Web in due finestre e fornisce un modo per trasferire i file avanti e indietro.



SUGGERIMENTI PER TROVARE HOSTING E DOMINI

- Per trovare società di hosting e registrar, basta cercare “web hosting” e “nomi di dominio”. Tutti i registrar avranno una funzione che ti permetterà di verificare se il nome di dominio che desideri sia disponibile.
- Il provider di servizi Internet di casa o dell'ufficio potrebbe fornire un hosting limitato per un piccolo sito Web. Il set di funzionalità disponibili sarà limitato, ma potrebbe essere perfetto per i tuoi primi esperimenti.
- Sono disponibili anche servizi gratuiti come **Neocities**, **Google Sites**, **Blogger** e **WordPress**. A volte si ottiene ciò per cui si paga, ma a volte queste risorse sono abbastanza buone per i primi esperimenti.
- Molte aziende forniscono hosting e domini.

UTILIZZO DI UNO STRUMENTO ONLINE COME GITHUB O GOOGLE APP ENGINE

Alcuni strumenti ti consentono di pubblicare il tuo sito web online:

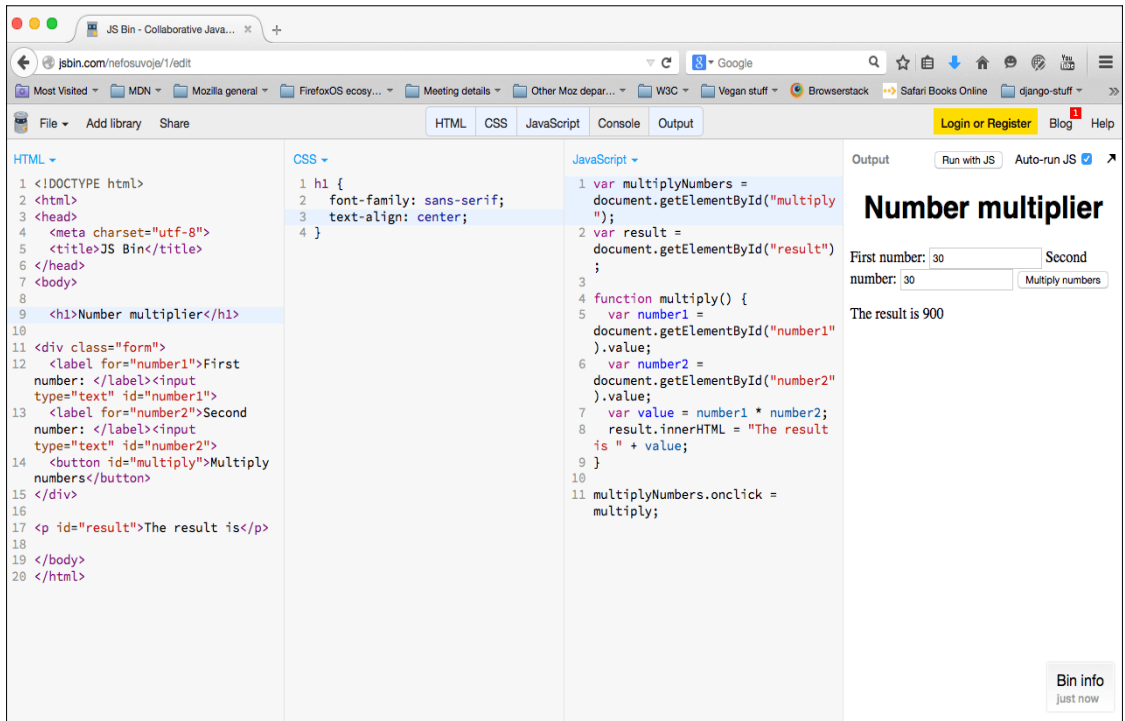
- GitHub è un sito di “social coding”. Consente di caricare repository di codice per l'archiviazione nel sistema di controllo della versione Git. È quindi possibile collaborare a progetti di codice e il sistema è open source per impostazione predefinita, il che significa che chiunque nel mondo può trovare il codice GitHub, usarlo, imparare da esso e migliorarlo. GitHub ha una funzionalità molto utile chiamata GitHub Pages, che ti consente di esporre il codice del sito Web in tempo reale sul Web.
- Google App Engine è una potente piattaforma che ti consente di creare ed eseguire applicazioni sull'infrastruttura di Google, sia che tu debba creare un'applicazione web a più livelli da zero o ospitare un sito web statico.

Queste opzioni sono generalmente gratuite, ma potresti superare il set di funzionalità limitato.

UTILIZZO DI UN IDE BASATO SUL WEB, AD ESEMPIO CODEPEN

Esistono numerose app Web che emulano un ambiente di sviluppo di siti Web, consentendo di immettere HTML, CSS e JavaScript e quindi visualizzare il risultato di tale codice come sito Web, il tutto in un'unica scheda del browser. In generale, questi strumenti sono relativamente facili, ottimi per l'apprendimento, buoni per condividere il codice (ad esempio, se si desidera condividere una tecnica o chiedere aiuto per il debug ai colleghi in un altro ufficio) e gratuiti (per le funzionalità di base). Ospitano la pagina di cui è stato eseguito il rendering in un indirizzo Web univoco. Tuttavia, le funzionalità sono limitate e queste app di solito non forniscono spazio di hosting per le risorse (come le immagini). Prova ad utilizzare alcuni degli strumenti seguenti per scoprire quale sia più indicato per te:

- **JSFiddle**
- **Glitch**
- **JS Bin**
- **CodePen**

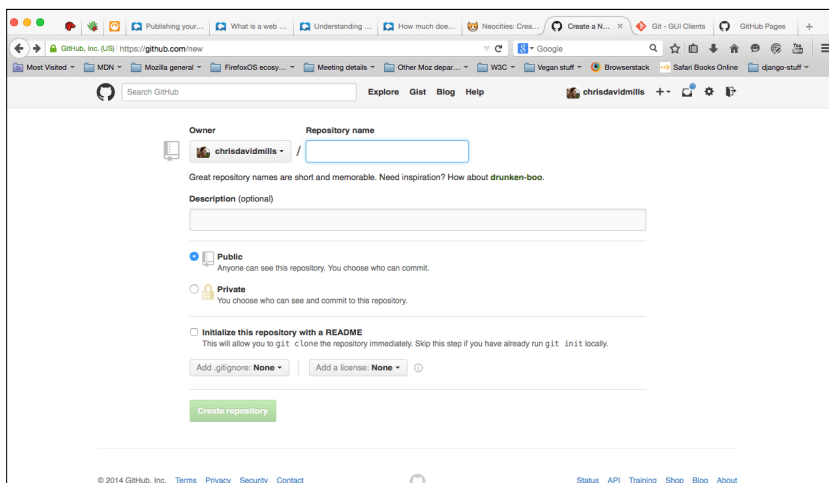


Esempio dell' IDE Js Bin

PUBBLICAZIONE TRAMITE GITHUB

Ora esaminiamo come pubblicare facilmente il tuo sito tramite **GitHub Pages**.

1. Prima di tutto, iscriviti a [GitHub](https://github.com/) (all'indirizzo <https://github.com/>) e verifica il tuo indirizzo e-mail.
2. Successivamente, è necessario creare un repository per archiviare i file.
3. In questa pagina, nella casella Nome repository, immettere **nomeutente.github.io**, dove nome utente è il tuo nome utente. Ad esempio, il nostro amico **Bob Smith** inseriva **bobsmith.github.io**. Seleziona la casella "Inizializza questo repository con un file README". Quindi fai clic su Crea repository.



4. Trascina e rilascia il contenuto della cartella del tuo sito web nel tuo repository. Quindi fai clic su **Commit changes**.

Nota: Assicurati che la tua cartella contenga un file `index.html`

5. Naviga nel tuo browser fino a `nomeutente.github.io` per vedere il tuo sito web online.

Ad esempio, per il nome utente `chrisdavidmills`, vai su `chrisdavidmills.github.io`.

Per il nome utente `gianni-pellegrini.github.io` vai su:

`https://gianni-pellegrini.github.io/giannipellegrini.github.io/`

Nota: Potrebbero essere necessari alcuni minuti prima che il tuo sito web venga pubblicato.

SITOGRAFIA DI RIFERIMENTO

- Maggiori informazioni su :

<https://developer.mozilla.org/>

- Gaming on line :

<https://breakout.enclavegames.com/lesson10.html>

Docente: GIOVANNI PELLEGRINI

Giorgio Pellegrini

