

# Introduzione a HTML 5 – Parte 1

## Le basi

# Il World Wide Web (WWW)

- Il WWW nasce al CERN di Ginevra nel 1991 da un'idea di Tim Berners Lee
- L'idea originale era di creare un archivio di informazioni collegate tra loro (ipertesti) distribuito su tutto il pianeta
  - In altre parole, ogni persona poteva creare e rendere pubblico il proprio sito web, con collegamenti ad altre pagine e ad altri siti web che gli utenti potevano seguire se interessati
  - Più o meno simile a Wikipedia, ma non centralizzato su un solo sito

# L'architettura del web

- Un **server** (host) ospita tutti i file del sito web (pagine HTML, immagini, stile, ecc.)
- I **client** si collegano al server tramite un **browser**, il quale comunica con il server tramite il protocollo HTTP (o HTTPS) per richiedere i file del sito man mano che servono
- Il browser è un'applicazione che permette di visualizzare pagine web, ad esempio Firefox, Google Chrome, Microsoft Edge, ecc.

# Il primo sito web della storia

Sul sito del CERN è ancora possibile visualizzare la prima pagina web della storia

<http://info.cern.ch/hypertext/WWW/TheProject.html>

## World Wide Web

The WorldWideWeb (W3) is a wide-area [hypermedia](#) information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an [executive summary](#) of the project, [Mailing lists](#) , [Policy](#) , November's [W3 news](#) , [Frequently Asked Questions](#) .

### [What's out there?](#)

Pointers to the world's online information, [subjects](#) , [W3 servers](#), etc.

### [Help](#)

on the browser you are using

### [Software Products](#)

A list of W3 project components and their current state. (e.g. [Line Mode](#) ,X11 [Viola](#) , [NeXTStep](#) , [Servers](#) , [Tools](#) , [Mail robot](#) , [Library](#) )

### [Technical](#)

Details of protocols, formats, program internals etc

### [Bibliography](#)

Paper documentation on W3 and references.

### [People](#)

A list of some people involved in the project.

### [History](#)

A summary of the history of the project.

### [How can I help ?](#)

If you would like to support the web..

### [Getting code](#)

Getting the code by [anonymous FTP](#) , etc.

# HTML

- HTML (HyperText Markup Language) è il linguaggio utilizzato per creare pagine web
- Non è un vero e proprio linguaggio di programmazione, ma è un **linguaggio di markup**, ossia permette di scrivere del testo aggiungendo alcune informazioni ad alcune parti di testo (ad esempio i link)
- Dal 1991 ad oggi, il linguaggio HTML si è evoluto tantissimo e permette di creare pagine estremamente complesse con immagini, tabelle, elenchi puntati, barre, audio/video, ecc.

# HTML 5

- La versione attuale (e probabilmente anche l'ultima) di HTML è HTML 5, definita nel 2008 e in costante evoluzione da allora
- Un documento HTML non è altro che un file di testo (UTF-8) che contiene testo e alcune parole chiave dette **tag** che marciano alcune parti di testo come „speciali“
- Lo standard HTML 5 prevede che i file HTML vengano usati solo per definire la **struttura e il contenuto della pagina**
- La **presentazione** grafica (stile) viene realizzata con il linguaggio CSS che vedremo in seguito

# Siti importanti

- <https://w3schools.com>: un sito pieno di tutorial, documentazione, esempi ed esercizi. È il miglior posto per imparare HTML, CSS e tutte le altre tecnologie web
- <https://validator.w3.org>: validatore di pagine HTML sviluppato dal W3C, vi dice se il vostro file HTML contiene errori e come correggerli (è un po' pignolo)
- <https://www.w3.org>: sito del W3C (World Wide Web Consortium), l'associazione che definisce gli standard del web

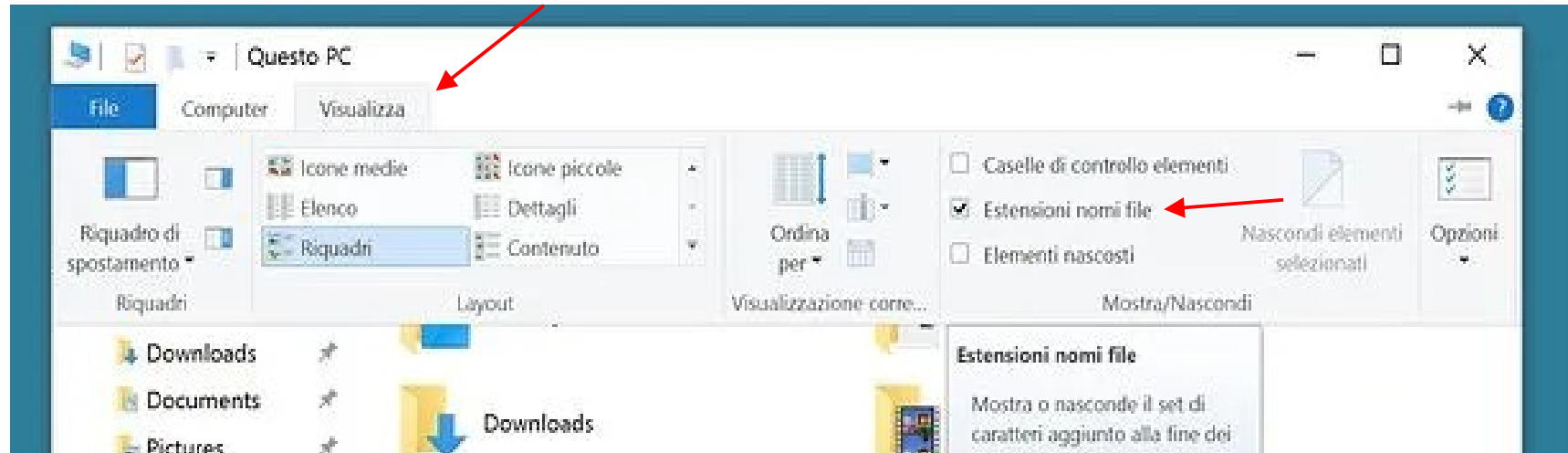
# Come creare una pagina HTML

- Per iniziare a fare sviluppo web, abbiamo bisogno solo di un **browser** e di un **editor di testo**, ad esempio
  - Notepad++ (solo per Windows)
  - Kate
  - VS Code (o meglio ancora, [VSCodium](#))
- **Attenzione: un word processor NON è un editor di testo!**  
Applicazioni come LibreOffice Writer e Microsoft Word NON vanno bene per fare sviluppo web (o programmazione in generale)



# Preparazione

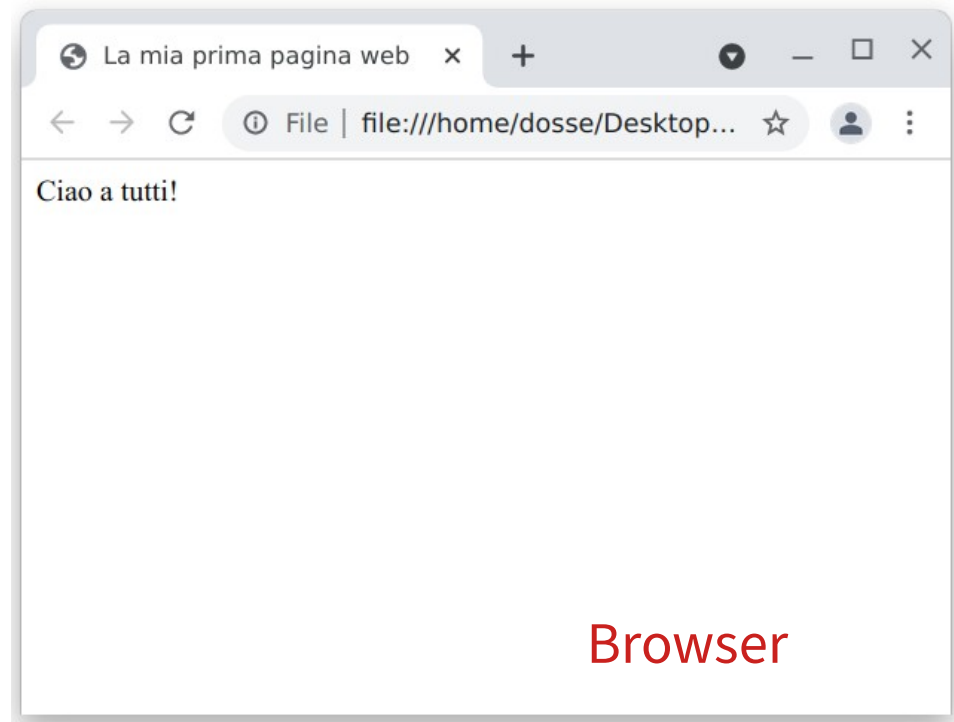
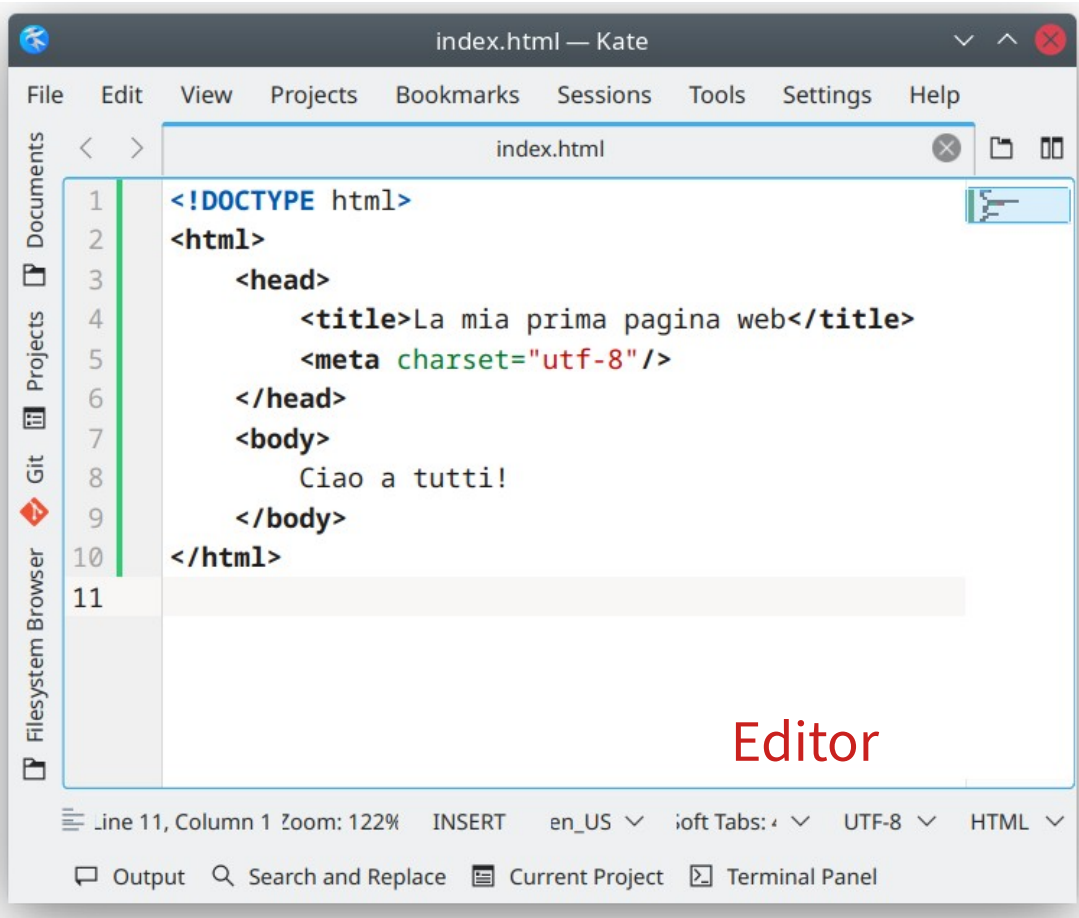
- Un sito web è formato da molti file, non solo uno, per cui la prima cosa da fare è creare una cartella per il sito in cui andremo a mettere tutti i file
- **Su Windows, è necessario rendere visibili le estensioni dei file** entrando nella cartella, cliccando su Visualizza e mettendo la spunta su Estensioni nomi file



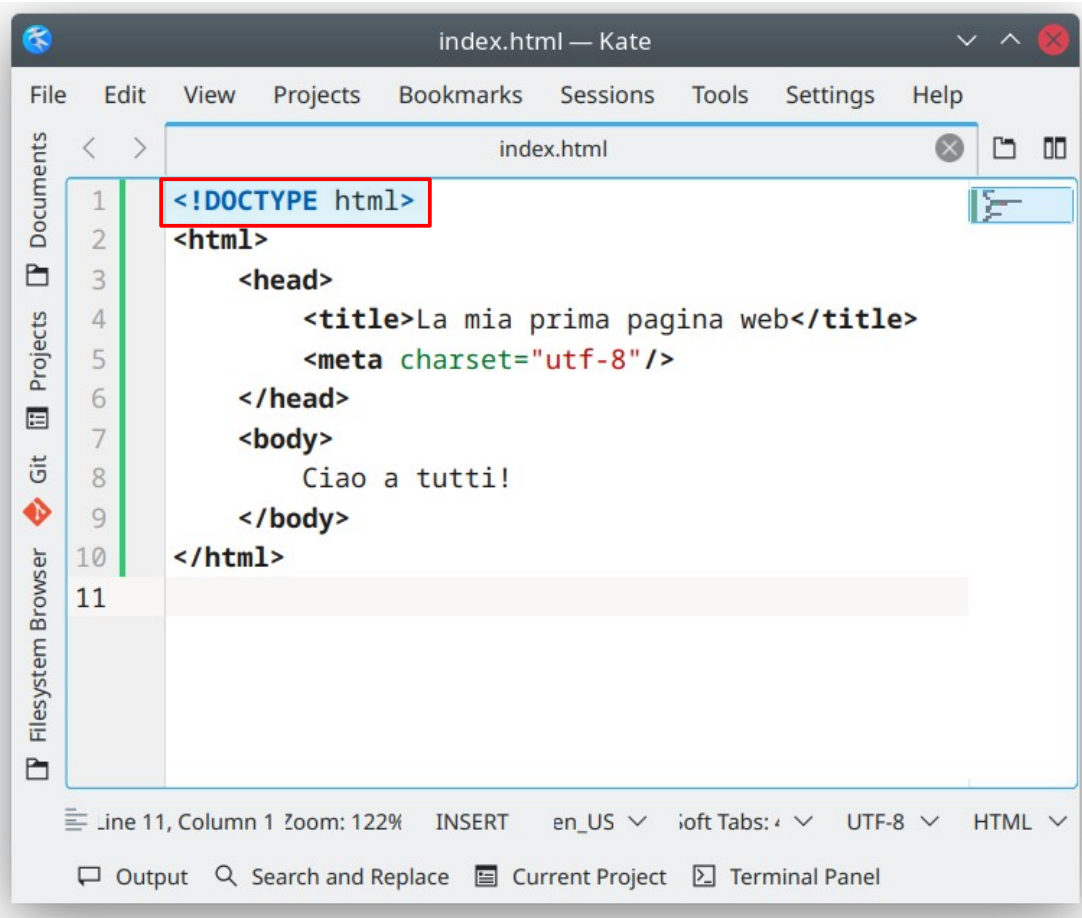
# Preparazione

- Nella cartella creiamo un file di testo con nome **index.html**, questa sarà la pagina iniziale del nostro sito
- Se hai fatto giusto, l'icona del file di testo dovrebbe sparire e **dovresti vedere l'icona del tuo browser** e facendo doppio click sul file dovrebbe aprirsi il browser con una pagina vuota
- Adesso possiamo **aprire il file nell'editor di testo** e iniziare a scrivere un po' di codice

# La mia prima pagina web



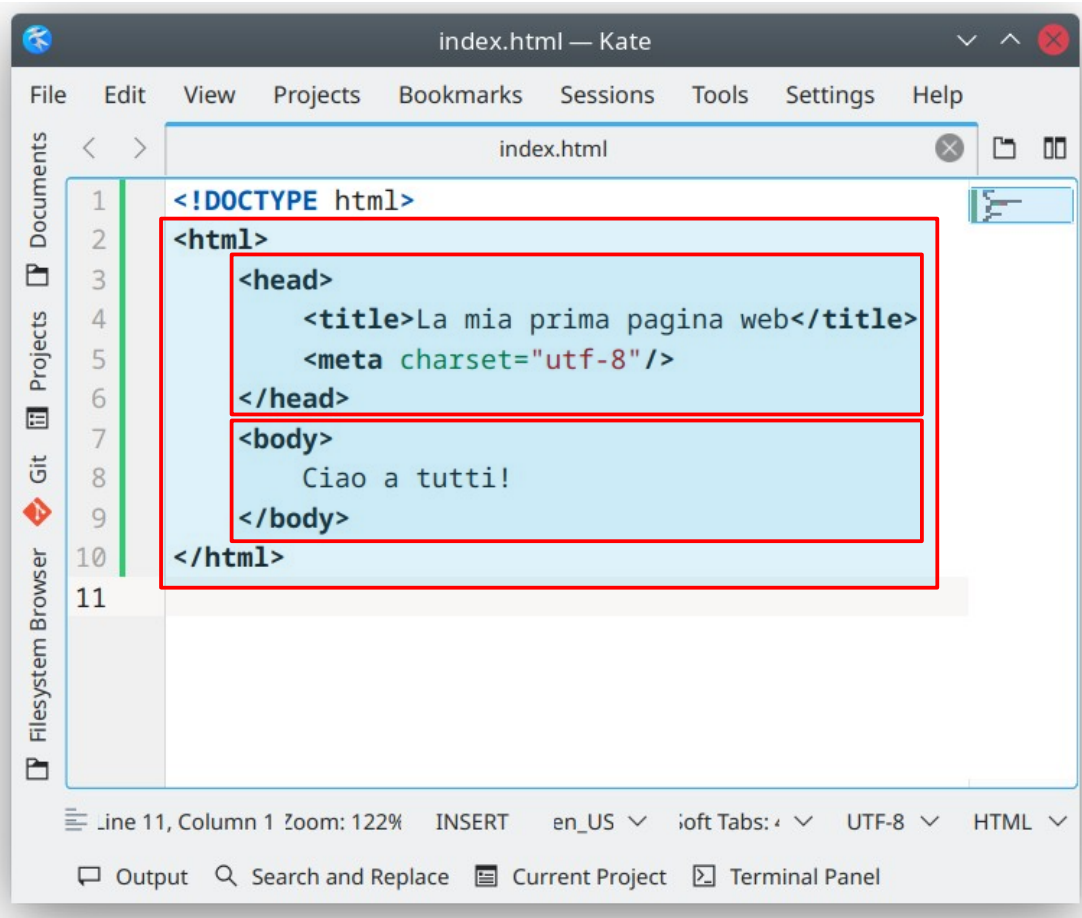
# La mia prima pagina web



```
index.html — Kate
File Edit View Projects Bookmarks Sessions Tools Settings Help
index.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>La mia prima pagina web</title>
5     <meta charset="utf-8"/>
6   </head>
7   <body>
8     Ciao a tutti!
9   </body>
10 </html>
11
```

- **Il documento inizia con**  
`<!DOCTYPE html>`
- Questo identifica il file come un documento HTML 5
- Se lo dimentichi, il browser penserà che è una versione precedente di HTML e potresti vedere comportamenti strani

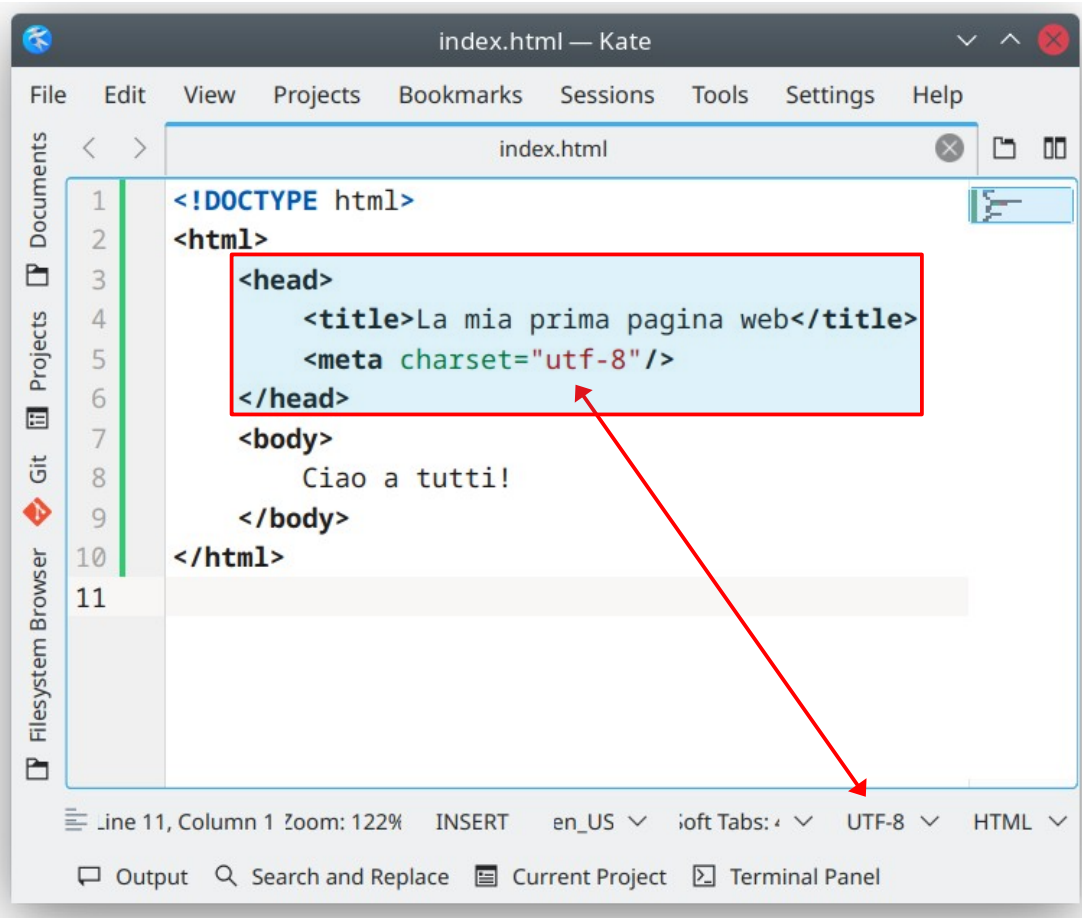
# La mia prima pagina web



```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>La mia prima pagina web</title>
5     <meta charset="utf-8"/>
6   </head>
7   <body>
8     Ciao a tutti!
9   </body>
10 </html>
11
```

- L'intero documento è inserito tra il tag di apertura `<html>` e il tag di chiusura `</html>`
- All'interno sono presenti due parti obbligatorie:
  - `head`: la parte NON visibile della pagina, contenente i metadati
  - `body`: la parte visibile della pagina, contiene tutto il resto

# La mia prima pagina web

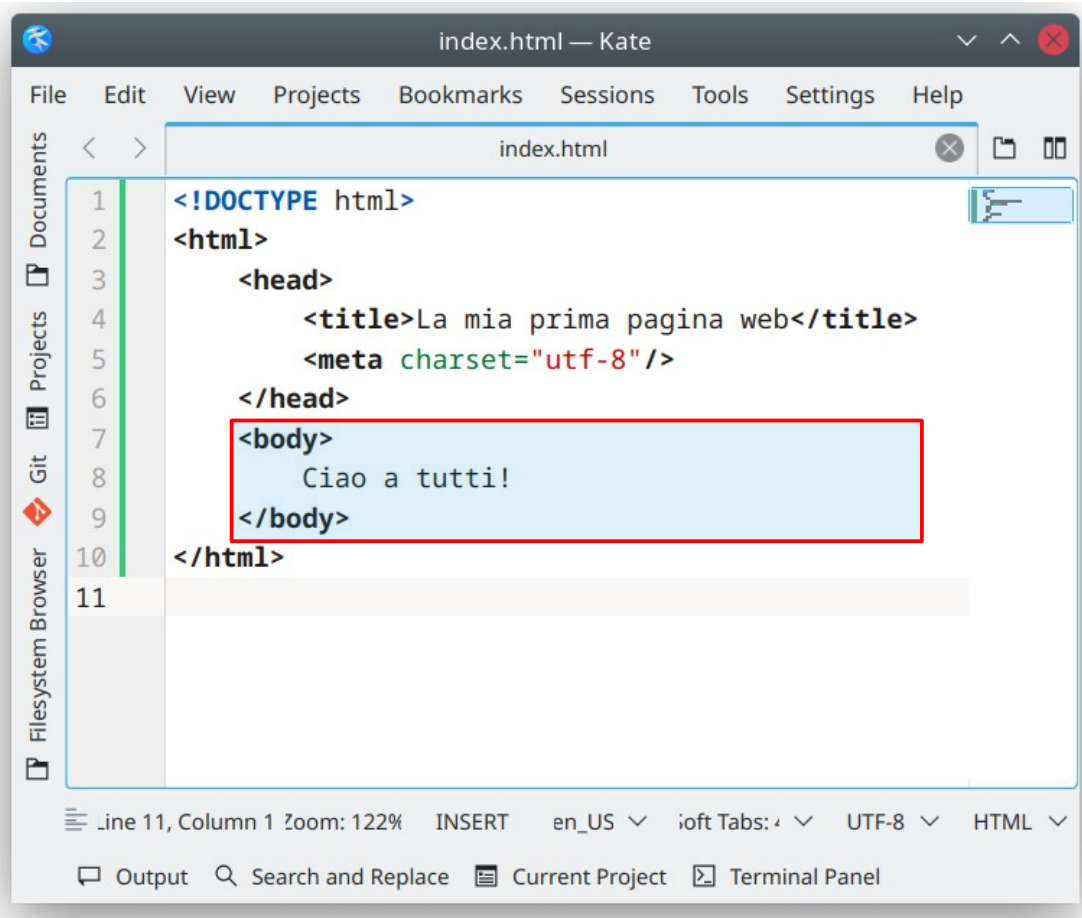


```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>La mia prima pagina web</title>
5     <meta charset="utf-8"/>
6   </head>
7   <body>
8     Ciao a tutti!
9   </body>
10 </html>
11
```

Line 11, Column 1 Zoom: 122% INSERT en\_US 4 Tabs: 4 UTF-8 HTML

- All'interno dell'elemento **head** è necessario specificare:
  - **title**: il titolo della pagina (viene visualizzato in alto, nel nome della scheda)
  - La codifica dei caratteri utilizzata (lo standard prevede UTF-8). Se non la specifichi o se ne usi una diversa, ci saranno problemi a visualizzare alcuni caratteri come lettere accentate, emoji, caratteri stranieri, ecc.

# La mia prima pagina web

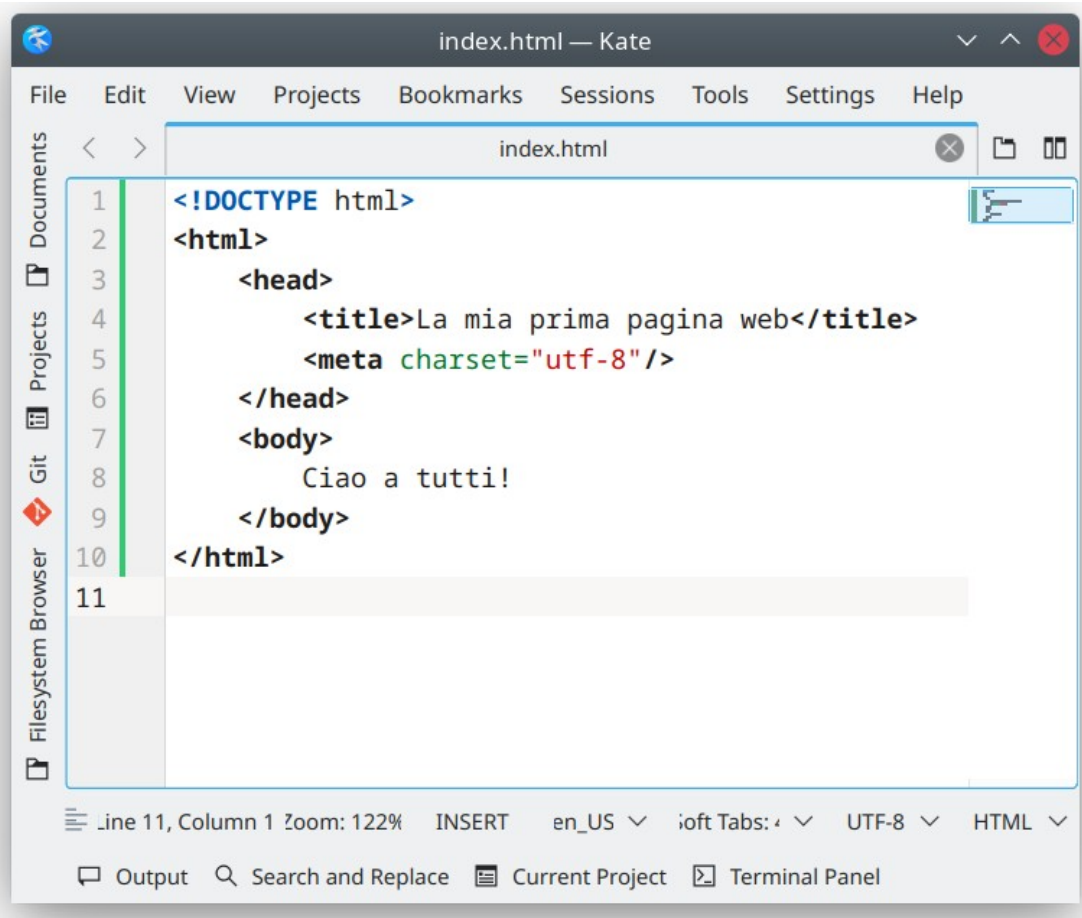


```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>La mia prima pagina web</title>
5     <meta charset="utf-8"/>
6   </head>
7   <body>
8     Ciao a tutti!
9   </body>
10 </html>
11
```

- All'interno dell'elemento **body** si mette tutta la parte visibile della pagina
- In questo esempio è semplicemente testo, ma generalmente è più complesso



# La mia prima pagina web



```
index.html — Kate
File Edit View Projects Bookmarks Sessions Tools Settings Help
index.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>La mia prima pagina web</title>
5     <meta charset="utf-8"/>
6   </head>
7   <body>
8     Ciao a tutti!
9   </body>
10 </html>
11
Line 11, Column 1 Zoom: 122% INSERT en_US 4oft Tabs: 4 UTF-8 HTML
Output Search and Replace Current Project Terminal Panel
```

- **Notate l'indentazione**
- Quando inserisco un tag all'interno di un altro, vado a capo e poi a destra premendo il tasto Tab sulla tastiera
- L'indentazione non è obbligatoria, ma **rende leggibile il codice e vi facilita la vita**



# Attenzione!

- I browser moderni sono delle applicazioni molto intelligenti e adattabili poichè devono poter visualizzare pagine scritte molto tempo fa e/o con errori nel codice
- **Se fate errori nel codice HTML, potreste non accorgervene nemmeno perchè il browser li corregge automaticamente**
- Non viene dato nessun avviso della presenza di questi errori
- **Usate il [validatore](#) per controllare se ci sono errori**

# Elementi basilari di HTML

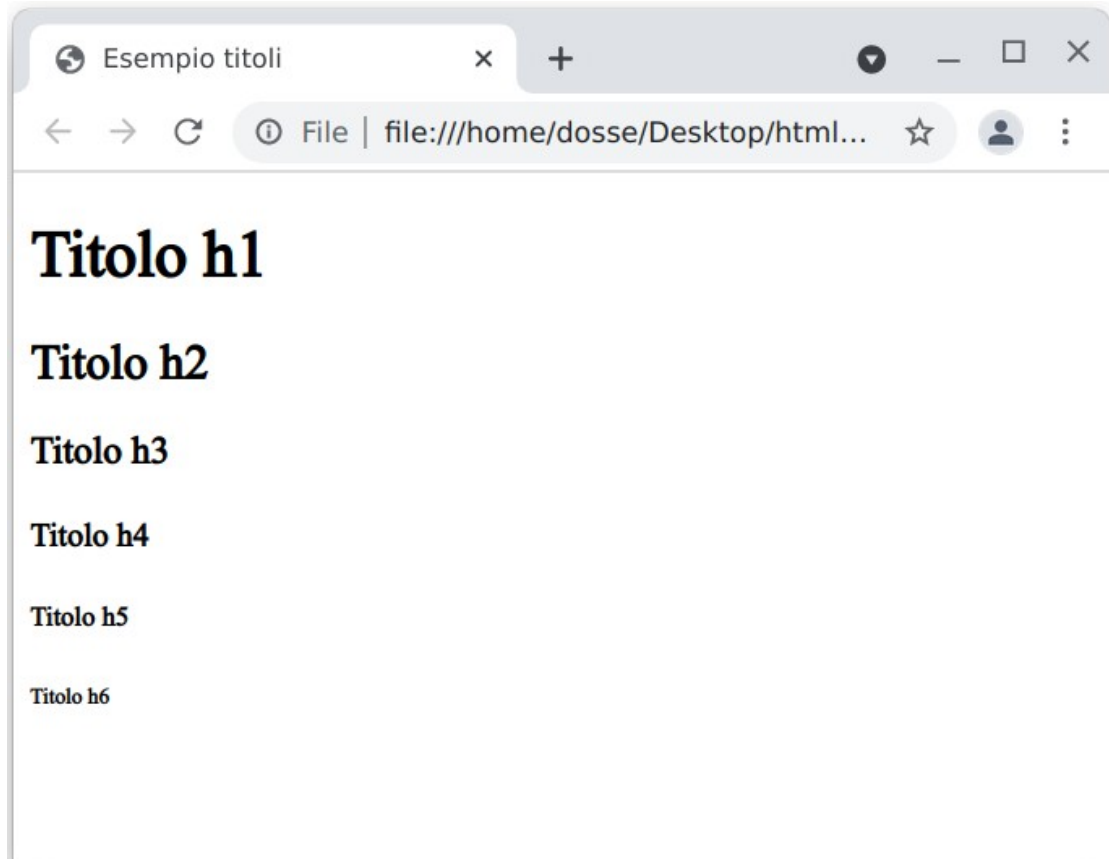
- Titoli
- Paragrafi
- Formattazione del testo
- Link
- Immagini
- Entità
- Elenchi puntati e numerati
- Tabelle
- Audio e Video

# Titoli

- I tag `h1`, `h2`, `h3`, `h4`, `h5` e `h6` definiscono 6 livelli di titoli
- Visivamente, `h1` è il titolo più grande e `h6` è quello più piccolo
- **I titoli vanno usati correttamente per definire una „scaletta“ del documento.** Ad esempio, `h1` può essere il titolo principale del documento, `h2` può essere usato per i capitoli, `h3` per le sezioni all'interno di un capitolo, ecc.

# Titoli

```
<!DOCTYPE html>
<html>
  <head>
    <title>Esempio titoli</title>
    <meta charset="utf-8"/>
  </head>
  <body>
    <h1>Titolo h1</h1>
    <h2>Titolo h2</h2>
    <h3>Titolo h3</h3>
    <h4>Titolo h4</h4>
    <h5>Titolo h5</h5>
    <h6>Titolo h6</h6>
  </body>
</html>
```

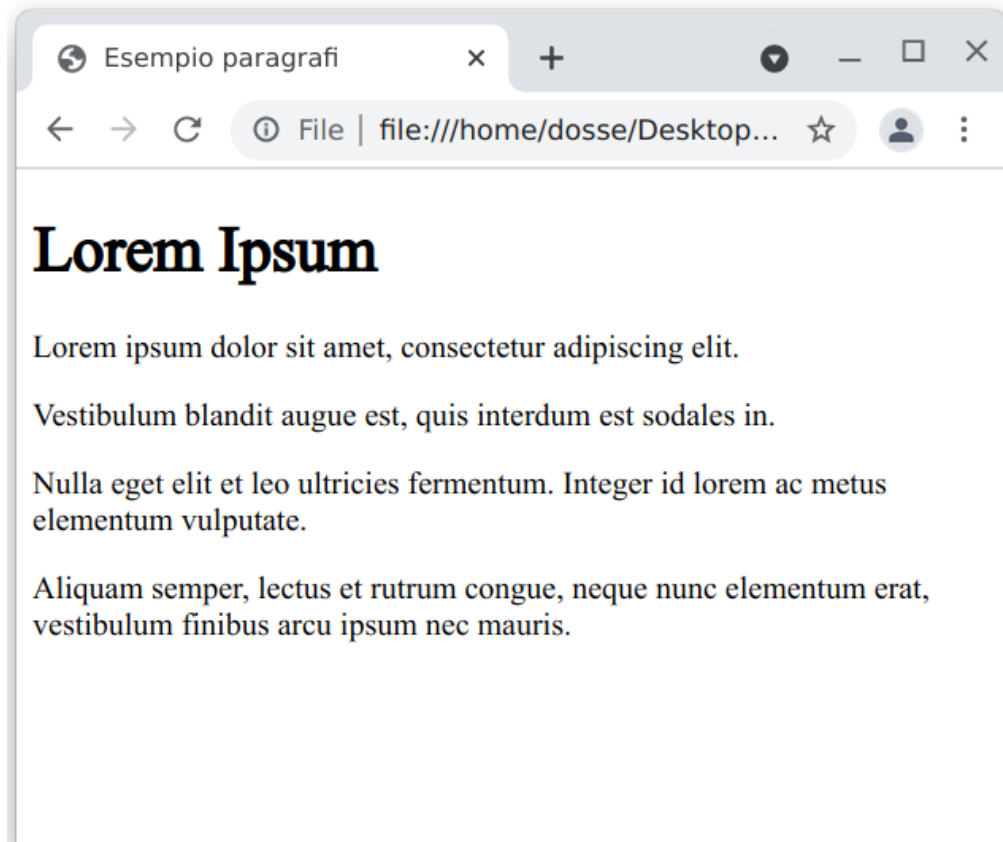


# Paragrafi

- Il tag `<p>` permette di definire dei paragrafi
- Un paragrafo è un blocco di testo con dello spazio sopra e sotto per separare i paragrafi
- **All'interno di un paragrafo si possono inserire solo testo, link e immagini.** L'inserimento di altri elementi è un errore che il browser corregge automaticamente spezzando il paragrafo in due, uno prima e uno dopo l'elemento non consentito

# Paragrafi

```
<!DOCTYPE html>
<html>
  <head>
    <title>Esempio paragrafi</title>
    <meta charset="utf-8"/>
  </head>
  <body>
    <h1>Lorem Ipsum</h1>
    <p>
      Lorem ipsum dolor sit amet, consectetur adipiscing elit.
    </p>
    <p>
      Vestibulum blandit augue est, quis interdum est sodales in.
    </p>
    <p>
      Nulla eget elit et leo ultricies fermentum. Integer id lorem ac metus
      ac metus elementum vulputate.
    </p>
    <p>
      Aliquam semper, lectus et rutrum congue, neque nunc elementum
      erat, vestibulum finibus arcu ipsum nec mauris.
    </p>
  </body>
</html>
```



# Paragrafi

- Il tag `<div>` funziona in modo simile a `<p>` ma è un contenitore più generico, senza uno stile predefinito, **può contenere qualsiasi cosa, inclusi altri div**
- Gli elementi `div` ci saranno **molto utili quando vedremo CSS** poichè ci permettono di creare delle „sezioni“ nella pagina a cui dare uno stile

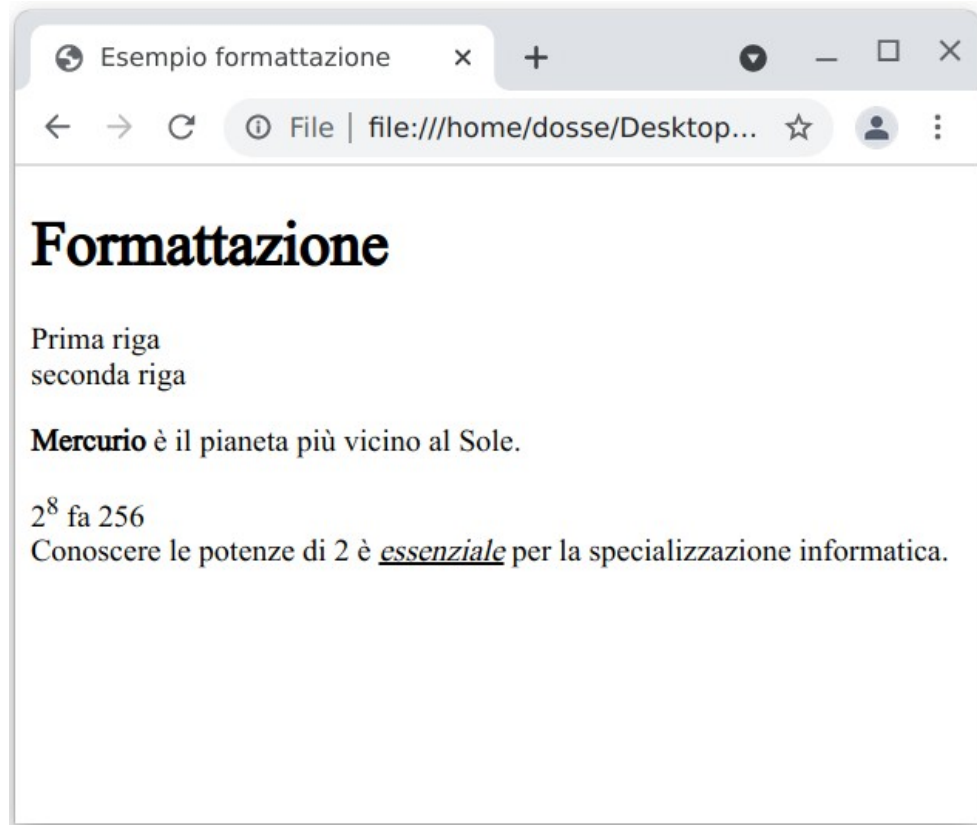
# Formattazione del testo

- I seguenti tag possono essere usati per formattare il testo:
  - `<br/>`: va a capo (in HTML, premere Invio non manda a capo il testo)  
Nota: questo tag non ha contenuto e non ha bisogno di essere chiuso
  - `<hr/>`: separatore orizzontale (si usa come il `<br/>` ma mette una riga orizzontale)
  - `<b>`: grassetto
  - `<i>`: corsivo
  - `<u>`: sottolineato
  - `<code>`: testo scritto così, utile per codici, sequenze di numeri, ecc.
  - `<del>`: barrato
  - `<sup>`: apice
  - `<sub>`: pedice
- Formattazione più avanzata, come centraggio, dimensioni, font, spaziature, ecc. può essere fatta con CSS



# Formattazione del testo

```
<!DOCTYPE html>
<html>
  <head>
    <title>Esempio formattazione</title>
    <meta charset="utf-8"/>
  </head>
  <body>
    <h1>Formattazione</h1>
    <p>
      Prima riga<br/>
      seconda riga
    </p>
    <p>
      <b>Mercurio</b> è il pianeta più vicino al Sole.
    </p>
    <p>
      2<sup>8</sup> fa 256<br/>
      Conoscere le potenze di 2 è <i><u>essenziale</u></i> per
      la specializzazione informatica.
    </p>
  </body>
</html>
```



# Link

- **Un link è un collegamento dalla pagina corrente a un'altra pagina o un altro sito**
- Per inserire un link si usa  
`<a href="indirizzo">Testo</a>`
- L'indirizzo può essere il percorso di un **altro file HTML**, ad esempio `pianeti/mercurio.html`
- Oppure può essere **un URL verso un altro sito**, ad esempio `https://google.com`

# Link

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Esempio link</title>
```

```
    <meta charset="utf-8"/>
```

```
  </head>
```

```
  <body>
```

```
    <h1>Il World Wide Web</h1>
```

```
    <p>
```

```
      Il World Wide Web è nato come una vasta area di <a
```

```
      href="ipertesti.html">ipertesti</a>. Nel tempo, si è evoluto  
      e oggi abbiamo siti web dinamici e molto interattivi, come
```

```
      <a href="https://google.com" target="_blank">Google</a>.
```

```
    </p>
```

```
  </body>
```

```
</html>
```

Esempio link



File | file:///home/dosse/Desktop...



## Il World Wide Web

Il World Wide Web è nato come una vasta area di [ipertesti](#). Nel tempo, si è evoluto e oggi abbiamo siti web dinamici e molto interattivi, come [Google](https://google.com).

# Link

- Aggiungendo l'attributo `target="_blank"` a un link, possiamo chiedere al browser di aprire il link in una nuova scheda, anzichè nella scheda corrente
- Ad esempio:  
`<a href="https://google.com" target="_blank">Cliccami</a>`  
Apri Google in una nuova scheda
- È una piccola cortesia per l'utente, apprezzata soprattutto se la destinazione è un altro sito

# Link

- Nell'href si può inserire qualsiasi protocollo supportato da una qualche applicazione installata sul dispositivo
  - `mailto:info@dittaprandini.it` apre il client email per mandare una mail all'indirizzo indicato
  - `tel:+391234567890` fa partire una telefonata verso il numero indicato (deve esserci il prefisso internazionale)
  - `tg://resolve?domain=canale` apre il canale specificato nell'app di Telegram
  - ...

# Immagini

- Con il tag `<img>` si possono inserire delle immagini nei documenti
- L'attributo obbligatorio `src` punta al file con l'immagine
  - Può essere un percorso verso un'immagine nello stesso sito, ad esempio `immagini/gatto.jpg`
  - Oppure può essere un URL verso un'immagine su un altro sito (se quel sito lo consente), ad esempio `https://fdossena.com/photo_home.jpg`
- L'attributo obbligatorio `alt` (anche se spesso dimenticato) specifica una breve descrizione testuale dell'immagine che viene usata dai motori di ricerca, dai lettori di schermo per i non vedenti e se l'immagine non può essere caricata

# Immagini

```
<!DOCTYPE html>
<html>
  <head>
    <title>Esempio immagini</title>
    <meta charset="utf-8"/>
  </head>
  <body>
    <h1>Esempio immagini</h1>
    <p>
      <b>Mercurio</b> è il pianeta più vicino al Sole.
    </p>
    
    <p>
      La temperatura della superficie varia tra i -173°C e i 427°C
      tra il giorno e la notte, la più alta escursione termica tra
      i pianeti del sistema solare.
    </p>
  </body>
</html>
```



# Immagini

- **Non avendo nessuna informazione di stile, l'immagine resta alle sue dimensioni originali, per cui potrebbe essere troppo grande**
- Quando faremo CSS vedremo meglio come dare uno stile alle immagini, per ora vediamo almeno come rimpicciolirla
- L'attributo `style` permette di specificare alcune informazioni di stile per uno specifico elemento. Nell'esempio sotto è stata impostata una larghezza fissa di 10 caratteri e l'altezza calcolata automaticamente

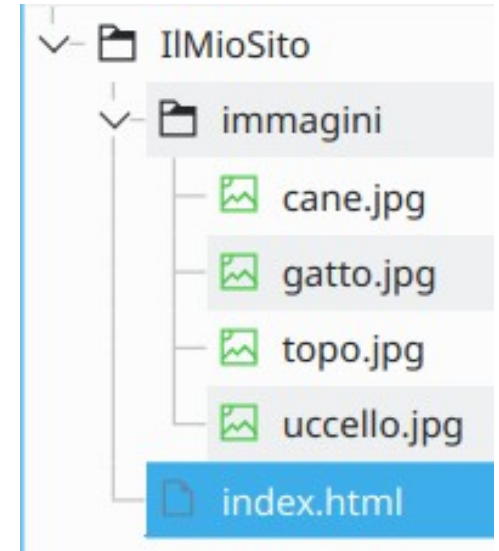
```

```



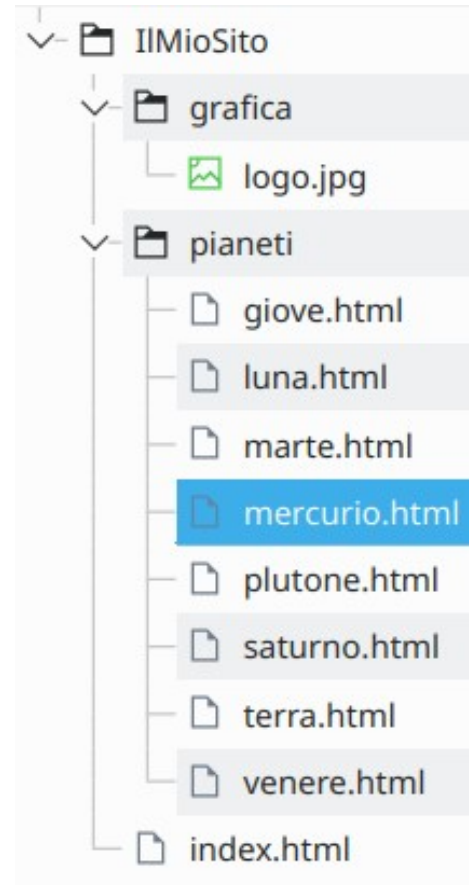
# Attenzione ai percorsi

- Quando fa riferimento al file di un'immagine (o un link, o qualsiasi altra cosa), questo è **relativo a dove si trova il file HTML**
- Nell'esempio a destra, la cartella `IlMioSito` contiene `index.html` e una cartella `immagini` con dentro alcuni file. Questo è lo scenario più comune
- Per fare riferimento a `gatto.jpg` dall'interno di `index.html`, bisogna utilizzare il percorso relativo: `immagini/gatto.jpg`



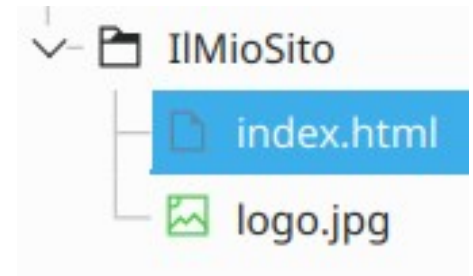
# Attenzione ai percorsi

- In questo esempio, la cartella `ILMioSito` contiene `index.html` e due cartella: `grafica` e `pianeti`
- All'interno di `pianeti` c'è un file HTML per ogni pianeta
- Se da `mercurio.html` voglio inserire `logo.jpg`, con il percorso relativo devo spostarmi al livello superiore scrivendo `..`, entrare nella cartella `grafica`, e prenderlo:  
`../grafica/logo.jpg`



# Attenzione ai percorsi

- In quest'ultimo esempio, la cartella `IlMioSito` contiene `index.html` e il file `logo.jpg`
- Essendo nella stessa cartella, per prendere il logo basta scrivere il nome del file:  
`logo.jpg`



# Attenzione ai percorsi

- Quando scrivete i percorsi dovete stare attenti a
  - Maiuscole e minuscole
  - Utilizzare il simbolo slash / e non il backslash \
  - **Non utilizzare percorsi assoluti** tipo  
C:\Users\Luca\Desktop\IlMioSito\immagine.jpg  
/home/luca/Desktop/IlMioSito/immagine.jpg
- **Questi errori faranno sì che il sito funzioni solo sul vostro computer**, nel momento in cui lo caricate su Internet i percorsi non funzioneranno più

# Entità HTML

- Le entità HTML consentono di inserire dei caratteri che normalmente non sarebbero consentiti
- Ad esempio, il simbolo < indica l'inizio di un tag, per cui non può essere usato nel testo, bisogna usare l'entità
- Entità principali:

Entità	Significato
&lt;	<
&gt;	>
&amp;	&
&nbsp;	Spazio piccolo
&nbsp;	Spazio grande

# Entità HTML

```
<!DOCTYPE html>
<html>
  <head>
    <title>Esempio entità</title>
    <meta charset="utf-8"/>
  </head>
  <body>
    <h1>Entità &amp; Entità</h1>
    <p>
      Grazie alle entità posso scrivere &lt;h1&gt;
    </p>
    <p>
      &emsp;&emsp;Questo testo è leggermente indentato
      usando due spazi grandi. Testo Testo Testo Testo
      Testo Testo Testo Testo Testo Testo Testo Testo
      Testo Testo Testo Testo Testo Testo Testo Testo
      Testo Testo Testo Testo Testo Testo Testo Testo
      Testo Testo Testo Testo Testo Testo Testo Testo
      Testo Testo Testo Testo
    </p>
  </body>
</html>
```



# To be continued...

- Scarica il secondo pacco di slide per continuare