**HTML PARTE 4**

**Hyperlinks**

Gli hyperlink (link ipertestuali) è la più grande innovazione offerta dal Web. Esistono dalla nascita del Web e lo rendono ciò che è , ovvero il Web stesso.

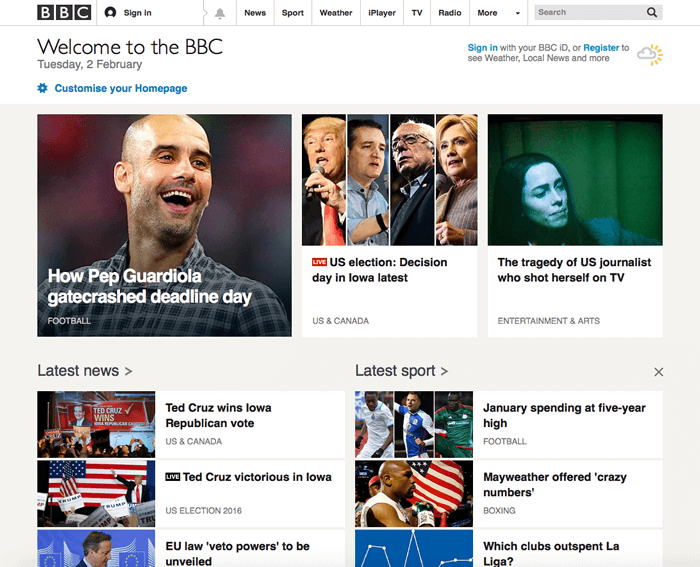
Ci permettono di collegare i nostri documenti con altri documenti (o altre risorse) che vogliamo.

Possiamo fare link a parti specifiche del documento e possiamo rendere le app disponibile a un semplice indirizzo web (in contrasto con le app native che vanno installate e richiedono più operazioni per gestirle)

Così come qualsiasi contenuto web può essere convertito in un link , allo stesso modo quando viene cliccato ( in altre parole “attivato”) permette al browser web di andare in un altro indirizzo web (URL)

**Nota**: Un URL può puntare a file HTML, file di testo, immagini, documenti di testo, video e audio e qualsiasi cosa che possa vivere sul Web. Se il browser Web non sa come mostrare o gestire il file, ti chiede se lo vuoi aprire (in questo caso il compito di aprire o gestire il file è passato alla app disponibile sul dispositivo) oppure se vuoi salvare il file (così da trattare il file in un secondo momento)

La home page della BBC , per esempio, contiene un gran numero di link che puntano non solo a molti articoli di notizie, ma anche a diverse aree del sito (funzionalità di navigazione), pagina di login e di registrazione e molti altri ancora.



**Anatomia di un link**

Un link di base si crea inserendo il testo (o altro contenuto, tipo immagini) che vuoi far diventare un link dentro un elemento **<a>**,e dando a <a> un attributo **href** (noto come **Hypertext Reference** o **target**) che contiene la destinazione del link

<p>I'm creating a link to

<a href="https://www.mozilla.org/en-US/">the Mozilla homepage</a>.

</p>

Il risultato è

I'm creating a link to [the Mozilla homepage](https://www.mozilla.org/en-US/).

**Aggiungere informazioni di supporto con l’attributo title**

Un altro attributo da aggiungere ad <a> è l’attributo title, che è pensato per contenere delle informazioni utili sul link, come il tipo di informazione del link oppure cose su cui porre l’attenzione. Vediamo un esempio:

p>I'm creating a link to

<a href="https://www.mozilla.org/en-US/"

title="The best place to find more information about Mozilla's

mission and how to contribute">the Mozilla homepage</a>.

</p>

Il risultato è:

I'm creating a link to [the Mozilla homepage](https://www.mozilla.org/en-US/).

**Nota**: Un titolo di un link viene rivelato solo al passaggio del mouse (mouse hover), il ché significa che le persone che si basano sulla tastiera per navigare sul Web avranno difficoltà ad accedere all’informazione del titolo. Se l’informazione del titole è di vitale importanza per l’usabilità della pagina, meglio metterla nel contenuto, piuttosto che come titolo di un link.

Esercizio: Crea un documento HTML usando l’editor di testo che usi di solito

* Dentro il body HTML prova ad aggiungere uno o più paragrafi o altri tipi di contenuto che conosci.
* Fai diventare qualche contenuto del link
* Includi gli attributi title

**Link a livello di blocco**

Come detto prima, si può trasformare in link qualsiasi contenuto, anche elementi a livello di blocco. Se si volesse far diventare link un’immagine, basterebbe racchiudere l’immagine nell’elemento <a>

<a href="https://www.mozilla.org/en-US/">

<img src="mozilla-image.png" alt="mozilla logo that links to the mozilla homepage">

</a>

Vedremo le immagini in altre sezioni.

Veloce introduzione agli URL e ai paths (percorsi)

Per capire a pieno il target dei link bisogna comprendere cosa siano gli URL e i percorsi.

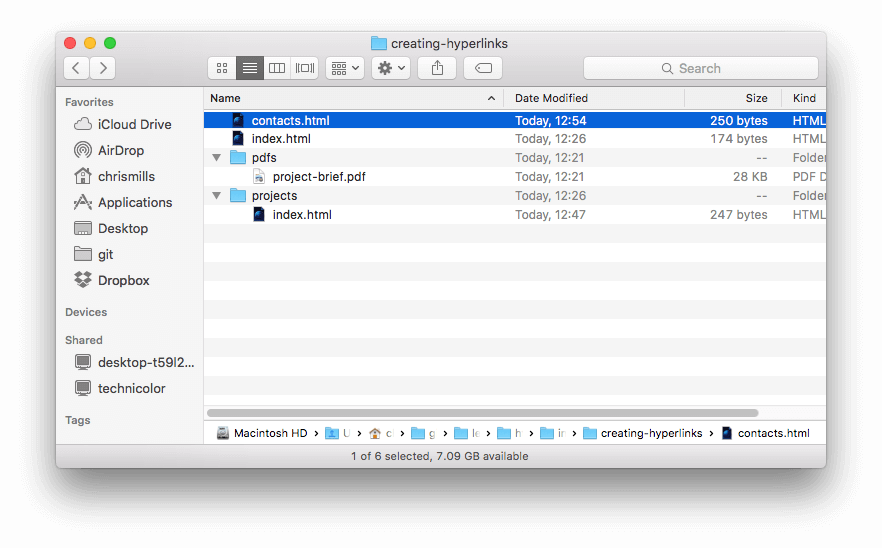
Un URL, Uniform Resource Locator, è una stringa di testo che definisce dove qualche risorsa è presente sul Web.

La pagina di Mozilla in inglese è presente all’URL

<https://www.mozilla.org/en-US/>

Gli URL usano i paths per cercare i file. I Paths specificano dove nel filesystem sono presenti i file di tuo interesse.

Vediamo un esempio di struttura di directory:



La radice (root) della directory si chiama **creating-hyperlinks**

Quando si lavora a un sito in locale si scegli una directory dentro la quale viene sviluppato l’intero sito.

Dentro la root abbiamo il file **index.html** e **contacts.html**

In un sito web reale index.html è la nostra homepage e la cosiddetta landing page (la pagina di atterraggio, ovvero una pagina web che funziona da punto d’ingresso di un sito web o una particolare sezione del sito web )

Ci sono anche delle directory nel nostro sito **pdfs** e **projects**, entrambe con un solo file all’interno, rispettivamente project-brief.pdf e index.html.

Nota che due file con lo stesso nome (nel nostro caso index.html) possono tranquillamente coesistere in un sito, a patto che risiedano in due directory differenti. Nel nostro esempio il secondo index.html potrebbe essere la landing page di una sezione del sito relativa a informazioni legate a progetti.

**Stessa directory:** se volessi includere un hyperlink all’interno di index.html (quella top-level) che punti a contacts.html dovresti solo specificare il nome del file di destinazione , dato che è nella stessa directory di index.html

<p>Want to contact a specific staff member?

Find details on our <a href="contacts.html">contacts page</a>.</p>

Muoversi nelle sottodirectory: se volessi inserire un link a projects/index.html dovresti andare sotto di un livello nella directory projects e poi indicare il file index.html. Questo risultato si ottiene specificando il nome della directory seguito da slash seguito dal nome del file , ovvero **projects/index.html**

<p>Visit my <a href="projects/index.html">project homepage</a>.</p>

Tornare sopra nella parent directory: se volessi inserire un link nella pagina projects/index.html che punti a pdfs/project-brief.pdf dovresti risalire di un livello dalla directory projects nella directory root , scendere di un livello nella directory pdfs per arrivare finalmente al file pdf.

Pertanto l’URL è:

…/pdfs/project-brief.pdf

<p>A link to my <a href="../pdfs/project-brief.pdf">project brief</a>.</p>

**Nota**: Puoi combinare istanze multiple di questa caratteristica in URL complessi, se necessario. es. ../../../complex/path/to/my/file.html.

**Frammenti di documento**

E’ possibile eseguire il link a una parte specifica del documento HTML (noto come frammento di documento, document fragment), oppure all’inizio del documento

Per fare questo bisogna per prima cosa assegnare un attributo id all’elemente che vuoi venga raggiunto.

Di solito si associa l’id a un heading, quindi si scrive:

<h2 id="Mailing\_address">Mailing address</h2>

Poi per raggiungere questa intestazione, bisogna includerla nell’url preceduta dal carattere cancelletto #

<p>Want to write us a letter? Use our <a href="contacts.html#Mailing\_address">mailing address</a>.</p>

Si può anche utilizzare il riferimento al frammento del documento da solo

<p>The <a href="#Mailing\_address">company mailing address</a> can be found at the bottom of this page.</p>

**URL assoluti vs URL relative**

URL assoluto: punta a un luogo definito dalla sua posizione assoluta sul web, incluso protocollo e nome dominio.

Per esempio, se un file index.html è caricata in una directory chiamata projects che risiede nella radice di un web server e il dominio del sito è <http://www.example.com> , la pagina sarebbe allora disponibile all’URL assoluto

<http://www.example.com/projects/index.html>

(o anche solo a <http://www.example.com/projects> se il server web è configurato in modo tale da scegliere come landing page la pagina index.html)

Un URL assoluto punterà sempre allo stesso luogo , indipendentemente da dove viene utilizzato

URL relativo: punta a un luogo che è relativo al file da cui stai puntando, similmente a come abbiamo visto nelle precedenti sezioni. Per esempio, se si vuole dal nostro file di esempio

<http://www.example.com/projects/index.html> un file pdf project-brief.pdf che è nella stessa directory, ovvero in projects, bisogna utilizzare solo il nome del file pdf senza altre informazioni.

Se il pdf è presente in una cartella pdfs che sta sotto projects, basta scrivere pdfs/project-brief.pdf

L’equivalente percorso assoluto sarebbe

<http://www.example.com/projects/pdfs/project-brief.pdf>

Un URL relativo punta a diversi luoghi a seconda del percorso dal quale è referenziato.

Per esempio se spostiamo index.html dalla cartella projects e lo mettiamo nella cartella root, il percorso relativo pdfs/project-brief.pdf non punta più a

<http://www.example.com/projects/pdfs/project-brief.pdf>

ma a

<http://www.example.com/pdfs/project-brief.pdf>

Ovviamente la risorsa pdf non si sposta automaticamente pertanto il link potrebbe risultare rotto (broken) , ovvero non si raggiungerebbe il file (a meno che non venga spostato a mano).

**Best practices per i link**

* **Utilizzare parole chiare per i link**

I link devono essere accessibili a tutti i lettori. Ad esempio

1. Gli utenti dei lettori di schermo amano saltare da link a link sulla pagina e leggono il testo del link incuranti del contesto
2. I motori di ricerca utilizzano il testo dei link per l’indicizzazione. Pertanto è una buona idea utilizzare parole chiave nel testo del link per descrivere a cosa ci si sta collegando.
3. I lettori comuni saltano grossi pezzi di testo durante la lettura del sito e si soffermano sui link, pertanto è utile che descrivano bene dove portano.

Vediamo un esempio e un contro-esempio

Esempio di buon link

<p><a href="https://firefox.com/">

Download Firefox

</a></p>

Esempio di link mal realizzato

<p><a href="https://firefox.com/">

Click here

</a>

to download Firefox</p>

Altri consigli:

1. Non ripetere il link come descrizione testuale del link.
2. Non scrivere la parola link o links nelle descrizioni testuali del link
3. Usa una descrizione più breve possibile
4. Minimizza i casi in cui molte copie dello stesso testo portano a pagine differenti (Esempio “clicca qui” porta a confusione certa)

* **Utilizzare link relativi quando è possibile**

Per questioni di efficienza. L’uso dei link assoluti è prolisso perché il browser scansiona tutto l’indirizzo, cerca il server nel DNS anche se già sei su quel server, e solo dopo cerca il file.

* Link a risorse NON-HTML, segnala chiaramente

Quando si crea un link a una risorsa da scaricare (un pdf o un documento word, per esempio) o in streaming (video o audio) o che ha effetti potenzialmente non attesi (si apre una finestra di popup o si carica un file flash) bisogna specificarlo chiaramente per non generare confusione o problemi.

E’ spiacevole infatti

* Se navighi con una connessione che non è a banda larga, clicchi su un link e parte un download di un grosso file
* Se non hai installato il player flash , clicchi su un link e si apre una pagina che dice di scaricare flash

Vediamo degli esempi di link che descrivono bene queste situazioni

<p><a href="http://www.example.com/large-report.pdf">

Download the sales report (PDF, 10MB)

</a></p>

<p><a href="http://www.example.com/video-stream/" target="\_blank">

Watch the video (stream opens in separate tab, HD quality)

</a></p>

<p><a href="http://www.example.com/car-game">

Play the car game (requires Flash)

</a></p>

* **Utilizza l’attributo download quando si crea un link a un file da scaricare**

Quando si crea un link a una risorsa da scaricare si può utilizzare l’attributo download per specificare un nome file di default.

<a href="https://download.mozilla.org/?product=firefox-latest-ssl&os=win64&lang=en-US"

download="firefox-latest-64bit-installer.exe">

Download Latest Firefox for Windows (64-bit) (English, US)

</a>

**Esercizio**

**Creazione di un menu di navigazione**

Colleghiamo alcune pagine insieme in un menu di navigazione per creare un sito multi-pagina.

Questo è il modo comune in cui si crea un sito web. La stessa struttura di pagina è utilizzata su ogni pagina, incluso lo stesso menu di navigazione, pertanto quando clicchi su un link si ha l’impressione di rimanere sulla stessa pagina e che vari solo il contenuto.

Creiamo una nuova cartella e mettiamo dentro i seguenti quattro file:

**index.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Homepage</title>

</head>

<body>

<!-- put navigation menu just below here -->

<h1>Homepage</h1>

<p>Welcome to my exciting homepage</p>

</body>

</html>

**projects.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Projects</title>

</head>

<body>

<!-- put navigation menu just below here -->

<h1>My projects</h1>

<p>Welcome to my project page, showing what exciting things I am currently doing.</p>

</body>

</html>

**pictures.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Pictures</title>

</head>

<body>

<!-- put navigation menu just below here -->

<h1>Pictures</h1>

<p>My pictures will go here, when I start taking some.</p>

</body>

</html>

**social.html**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Social</title>

</head>

<body>

<!-- put navigation menu just below here -->

<h1>Social media</h1>

<p>Welcome to my social media page. I am currently antisocial, but will start putting my social media widgets on here when the time is right.</p>

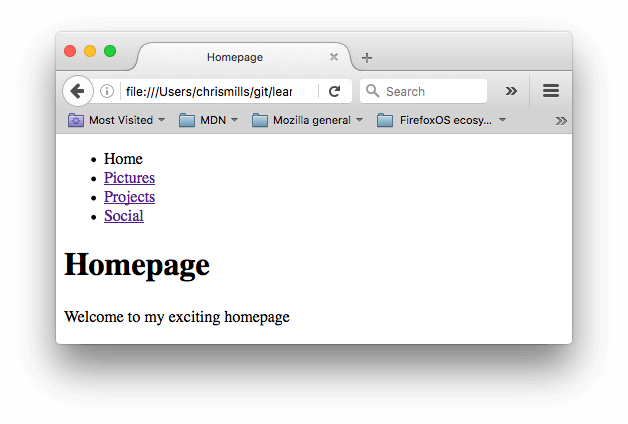
</body>

</html>

Dovresti:

1. Aggiungere una lista ordinate nel luogo indicato su una pagina, contenente i nomi delle pagine da linkar. Un menu di navigazione di solito è una lista di link, quindi semanticamente va bene realizzarla in questo modo.
2. Fai diventare ogni voce della lista un link a quella pagina..
3. Copia il menu di navigazione sulle altre tre pagine.
4. In ogni pagina rimuovi solo il link alla pagina corrente.

L’esempio finito deve apparire più o meno nel seguente modo:



E-mail link

E’ possibile creare link o bottoni che, se cliccati, aprono un nuovo messaggio di posta elettronica, invece di aprire una pagina html o scaricare risorse. Si ottiene questo comportamento con l’elemento **<a>** e lo schema di URL **mailto:**

Nel suo formato base e più utilizzato , un link mailto: indica semplicemente l’indirizzo email a cui mandare la mail.

Per esempio:

<a href="mailto:nowhere@mozilla.org">Send email to nowhere</a>

(Prova e vedi come appare)

L’indirizzo email è opzionale. Se non lo si mette e si utilizza solo lo schema mailto: si apre un nuovo messaggio di posta vuoto. Questo di solito è un link “SHARE” che gli utenti utilizzano per mandare mail a chi vogliono con comodità, rimandendo sul sito.

**Specificare i dettagli**

Oltre all’indirizzo email si possono indicare altri dettagli. I più comuni sono *subject*, *cc*, e *body* (permette di inserire un breve messaggio di default nella mail).

Un esempio:

<a href="mailto:nowhere@mozilla.org?cc=name2@rapidtables.com&bcc=name3@rapidtables.com&subject=The%20subject%20of%20the%20email&body=The%20body%20of%20the%20email">

Send mail with cc, bcc, subject and body

</a>

**Nota:** I valori di ogni campo devono essere URL-encoded, cioè senza caratteri non stampati (caratteri invisibili come tabs, return o interruzioni di riga) e spazi (da sostituire con %20). Si noti anche l’utilizzo del punto interrogativo (?) per separare l’URL principale dai valori dei campi, e il carattere & per separare ogni campo nell’URL  mailto: . Questa è una notazione standard delle query delle URL. Vedremo il metodo GET nell’invio dei form successivamente.