**HTML PARTE 2**

**Cos’è la parte head dell’HTML**

Rivediamo lo schema di un documento HTML

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>My test page</title>

</head>

<body>

<p>This is my page</p>

</body>

</html>

La parte head dell’HTML è il contenuto presente all’interno dei tag **<head>** e **</head>**

Tale contenuto, a differenza di quello presente nel body, non viene mostrato nella pagina.

Il compito dell’head è di contenere dei metadati per il documento.

Nell’esempio precedente:

<head>

<meta charset="utf-8">

<title>My test page</title>

</head>

In pagine più grandi, comunque, l’header può contenere molto contenuto. Tramite Developer Tools del browser si può vedere il contenuto di un head di un sito a piacere, nel tab elements.

Vediamo come utilizzare i principali elementi dell’head

Aggiungere un titolo

Abbiamo visto l’elemento <title>. E’ quello che ci permette di aggiungere un titolo alla pagina.

Da non confondere con <h1> che è un elemento del body che è un’intestazione della pagina.

Vediamo un esempio:

Copiamo il seguente codice HTML su un file di testo

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>&lt;title&gt; element</title>

</head>

<body>

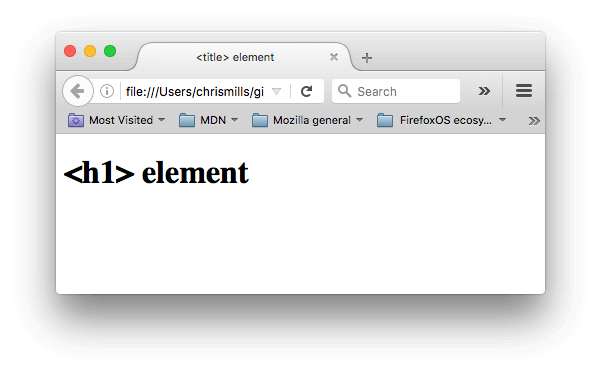
<h1>&lt;h1&gt; element</h1>

</body>

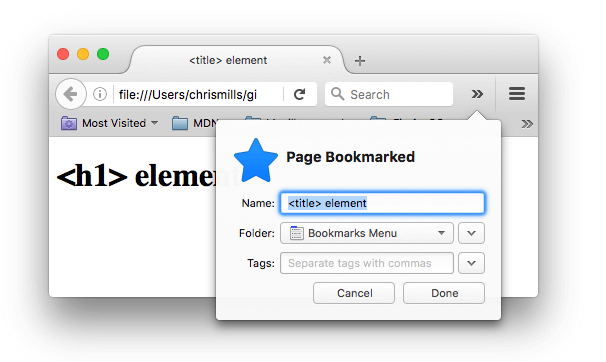
</html>

Lo salviamo in un luogo che preferiamo e lo apriamo con il browser.

Quello che otteniamo è qualcosa di simile a questo:



Il contenuto dell’elemento <title> può essere utilizzato anche in altri modi. Se si aggiunge ai preferiti la pagina il titolo verrà utilizzato come suggerimento come nome del sito preferito.



I contenuti del titolo vengono utilizzati nei risultati di ricerca, come vedremo dopo.

Metadati: l’elemento <meta>

I metadati sono dati che descrivono dati, e l’HTML ha un modo “ufficiale” per aggungere metadati a un documento: l’elemento **<meta>.**

Ci sono molti tipi di elementi <meta>che possono essere inclusi nell’head , ne vediamo solo alcuni, i più comuni e noti.

Specificare la codifica dei caratteri

Nel nostro esempio di partenza abbiamo visto

<meta charset="utf-8">

Tale elemento specifica la codifica dei caratteri nella pagina, il set di caratteri che il documento può utilizzare. Utf-8 è il set di caratteri universale che include praticamente qualsiasi carattere del linguaggio degli uomini. Questo significa che la nostra pagina web può gestire qualsiasi linguaggio; pertanto è una buona idea inserire questo elemento nelle nostre pagine web.

Per esempio si può gestire l’inglese e il giapponese senza problemi.



Se avessimo utilizzato una codifica diversa, come ad esempio ISO-8859-1, (l’insieme dei caratteri dell’alfabeto latino) la pagina sarebbe apparsa in maniera scorretta:



**Nota**: Alcuni browsers (es: Chrome) aggiustano automaticamente le codifiche errate, quindi a seconda del browser utilizzato il problema di codifica potrebbe non vedersi. E’ comunque consigliato utilizzare la codifica utf-8 sulle tue pagine, per evitare problemi con altri browsers.

Come esempio aggiungiamo al documento html salvato in precedenza il seguente codice:

<p>Japanese example: ご飯が熱い。</p>

Proviamo a utilizzare la codifica utf-8 e quella ISO-8859-1 per vedere cosa succede.

Aggiungere un autore e una descrizione

Molti tag <meta> includono gli attributi **name** e **content**

name – specifica il tipo di metadato, quale informazione contiene

content – specifica l’attuale contenuto meta

Tali elementi meta che sono utili da includere nelle tue pagine definiscono l’autore della pagina , e offrono una breve descrizione della pagina.

Vediamo un esempio:

<meta name="author" content="Chris Mills">

<meta name="description" content="The MDN Web Docs Learning Area aims to provide

complete beginners to the Web with all they need to know to get

started with developing web sites and applications.">

Specificare l’autore è utile in pratica solo per capire chi ha costruito la pagina.

Specificare una descrizione che includa parole chiave relative al contenuto della pagina è potenzialmente utile per far apparire la pagina più in alto nei risultati di una ricerca tramite motore di ricerca.

(Tali attività sono note come Search Engine Optimization – SEO)

Altri tipi di metadati

Navigando sul web si possono trovare molti tipi di metadati . Molte caratteristiche sono creazioni proprietarie , pensate per dare ad alcuni siti (come i social network) pezzi di informazione da utilizzare .

Ad esempio Open Graph Data è un protocollo di metadati inventato da Facebook offrire un set più ricco di metadati per i siti.

Nel codice sorgente del sito MDN Web Docs si trova:

<meta property="og:image" content="https://developer.cdn.mozilla.net/static/img/opengraph-logo.dc4e08e2f6af.png">

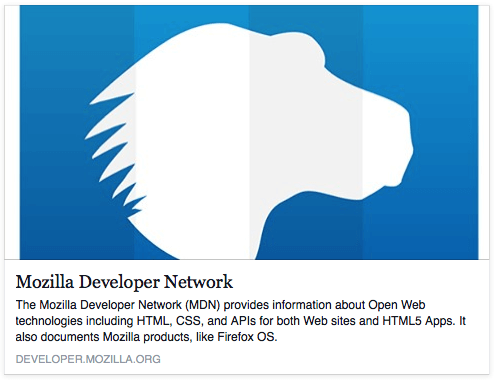
<meta property="og:description" content="The Mozilla Developer Network (MDN) provides

information about Open Web technologies including HTML, CSS, and APIs for both Web sites

and HTML5 Apps. It also documents Mozilla products, like Firefox OS.">

<meta property="og:title" content="Mozilla Developer Network">

Un effetto dell’utilizzo di OGD è che quando si cerca MDN Web Docs su facebook, il link appare con un’immagine



Twitter ha un insieme di metadati simile, si usa per esempio

<meta name="twitter:title" content="Mozilla Developer Network">

Aggiungere un’immagine personalizzata al sito

Per arricchire ulteriormente il progetto del tuo sito, puoi aggiungere dei riferimenti a icone personalizzate all’interno dei metadati, e tali icone verranno mostrate in certe situazioni.

La più usata di queste è la arci-nota **favicon** (abbreviazione di "favorites icon") . Ci riferisce al termine favorita per indicare il suo uso nei siti preferiti (ovvero favorites)

La Favicon è la prima della stirpe, è stata presente per anni. La prima di questo tipo era un’icona di 16pixel quadrati .

Per creare una favicon:

1. Si cerca o si costruisce un’immagine e la si salva allo stesso livello del file index.html nel formato .ico (si possono utilizzare anche .gif o .png ma non è garantito che funzionino su tutti i browser, specialmente quelli più datati di IE)
2. Si aggiunge la seguente riga nell’head:

<link rel="shortcut icon" href="favicon.ico" type="image/x-icon">

Dove favicon.ico è il nome prescelto per la nostra favicon

Esistono molti tipi di icona da considerare al giorno d’oggi. Vediamo un esempio.

<!-- third-generation iPad with high-resolution Retina display: -->

<link rel="apple-touch-icon-precomposed" sizes="144x144" href="https://developer.cdn.mozilla.net/static/img/favicon144.a6e4162070f4.png">

<!-- iPhone with high-resolution Retina display: -->

<link rel="apple-touch-icon-precomposed" sizes="114x114" href="https://developer.cdn.mozilla.net/static/img/favicon114.0e9fabd44f85.png">

<!-- first- and second-generation iPad: -->

<link rel="apple-touch-icon-precomposed" sizes="72x72" href="https://developer.cdn.mozilla.net/static/img/favicon72.8ff9d87c82a0.png">

<!-- non-Retina iPhone, iPod Touch, and Android 2.1+ devices: -->

<link rel="apple-touch-icon-precomposed" href="https://developer.cdn.mozilla.net/static/img/favicon57.a2490b9a2d76.png">

<!-- basic favicon -->

<link rel="shortcut icon" href="https://developer.cdn.mozilla.net/static/img/favicon32.e02854fdcf73.png">

I commenti spiegano l’utilizzo delle varie icone. Si usano icone specifiche per risoluzioni schermo differenti.

Al momento però non preoccupiamoci troppo di questo aspetto. Si tratta di questioni avanzate di html. L’importante è sapere che queste cose esistano, al momento.

Applicare CSS e JavaScript all’HTML

Tutti i siti moderni impiegano CSS per rendere più fico l’HTML e JavaScript per offrire interattività al sito (player video, giochi, mappe e ancora altro).

Questi elementi si applicano all’html tramite il tag **<link>** e **<script>** rispettivamente.

* L’elemento <link> va inserito sempre all’interno del tag <head>. Prende due attributi:

1. **rel=”stylesheet”** che indica che il documento è un foglio di stile
2. **href** che indica il percorso del file css all’interno del proprio server web

<link rel="stylesheet" href="my-css-file.css">

* L’elemento <script> non deve necessariamente andare nell’head. Infatti a volte è meglio metterlo alla fine del file, prima della chiusura del </body> del documento, per assicurarsi che il documento sia completamente caricato , prima di assegnargli dei comportamenti interattivi.

(Se il javascript prova ad accedere a degli elementi che non sono stati ancora caricati lancia un errore)

La sintassi è la seguente:

<script src="my-js-file.js"></script>

L’elemento script potrebbe apparire come un element vuoto, ma non è così e pertanto richiede obbligatoriamente il tag di fine elemento ( </script> )

Invece di puntare a un file esterno si può anche scegliere di scrivere il codice javascript all’interno dei tag <script></script>

Esercizio:

Copiare i seguenti tre file meta-example.html, script.js e style.css in tre file e assegnare l’opportuno percorso. Verificare che il sito funzioni, dopo aver aggiunto <link> e <script> al file html

**meta-example.html**

<!DOCTYPE html>

<html lang="en-US">

<head>

<meta charset="utf-8">

<title>Meta examples</title>

<meta name="author" content="Chris Mills">

<meta name="description" content="This is an example page to demonstrate usage of metadata on web pages.">

<meta property="og:image" content="https://developer.cdn.mozilla.net/static/img/opengraph-logo.dc4e08e2f6af.png">

<meta property="og:description" content="This is an example page to demonstrate usage of metadata on web pages.">

<meta property="og:title" content="Metadata; The HTML &lt;head&gt;, on MDN">

<link rel="Shortcut Icon" href="favicon.ico" type="image/x-icon">

</head>

<body>

<h1>Meta examples</h1>

<p>Japanese example: ご飯が熱い。</p>

</body>

</html>

**script.js**

var list = document.createElement('ul');

var info = document.createElement('p');

var html = document.querySelector('html');

info.textContent = 'Below is a dynamic list. Click anywhere outside the list to add a new list item. Click an existing list item to change its text to something else.';

document.body.appendChild(info);

document.body.appendChild(list);

html.onclick = function() {

var listItem = document.createElement('li');

var listContent = prompt('What content do you want the list item to have?');

listItem.textContent = listContent;

list.appendChild(listItem);

listItem.onclick = function(e) {

e.stopPropagation();

var listContent = prompt('Enter new content for your list item');

this.textContent = listContent;

}

}

**style.css**

html {

background-color: green;

font-size: 20px;

}

ul {

background: red;

padding: 10px;

border: 1px solid black;

}

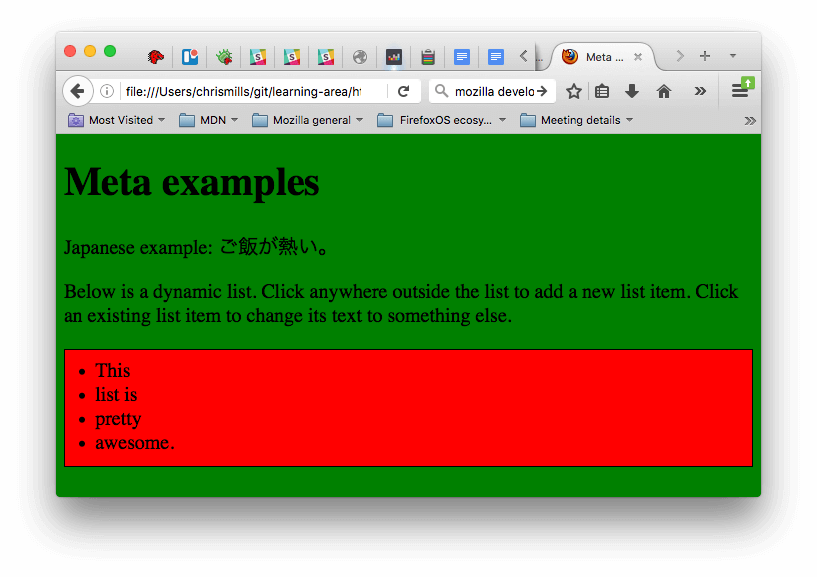
li {

margin-left: 20px;

}

Aprire i file sia nel broswer che nell’editor

Se tutto è fatto bene deve apparire la seguente schermata

****

Cliccando su un elemento esistente è possibile cambiarne il nome.

Cliccando altrove si può aggiungere un elemento.

Il CSS crea uno sfondo verde , il testo è più grande, la lista ha sfondo rosso e bordo nero.

Impostare il linguaggio principale del sito

Ultima cosa che vale la pena far notare è la possibilità di impostare il linguaggio principale del sito web. Questo può essere fatto aggiungendo l’attributo **lang** al tag html di apertura, nel seguente modo:

<html lang="en-US">

Questa aggiunta è utile in diversi modi.

Il documento HTML sarà indicizzato in maniera migliore dai motori di ricerca se si imposta il suo linguaggio. (§appare per esempio correttamente in risultati specifici per lingua), ed è utile anche con persone con problemi di vista che utilizzano lettori automatici (ad esempio la parola “six” esiste sia in inglese che in francese, ma viene pronunciata diversamente nelle due lingue).

Si possono anche specificare dei linguaggi in sottosezioni delle nostre pagine web. Nel nostro esempio, il sito è in inglese ma una frase è in giapponese.

<p>Japanese example: <span lang="ja">ご飯が熱い。</span>.</p>

I codici sono definiti nello standard ISO 639-1 (<https://en.wikipedia.org/wiki/ISO_639-1> )