

HTML

- Standard gestiti dal W3C <https://www.w3.org/standards/webdesign/>
 - HTML (struttura), CSS (stile), JavaScript (interattività)
 - MDN Mozilla Developer Network: <https://developer.mozilla.org/it/docs/Web>
- HTML è un **Markup Language**
 - Scrittura via text editor (vim, Nodepad++, ...)
 - Rendering (e debugging) via browser
- Progetto di riferimento
 - <https://github.com/egalli64/nesp> (*modulo 1*) – vedi *README.md*
 - **Node.js** <https://nodejs.org/en/download/> (LTS) Piattaforma per server app in JavaScript
 - **VS Code** <https://code.visualstudio.com/>

HTML: HyperText Markup Language

- Tim Berners-Lee (TimBL) @CERN ~1990 – Premio Turing 2016
 - World Wide Web Consortium (W3C) 1994 – HTML5 2014
- Descrive come rappresentare pagine web statiche
- La pagina viene acceduta via **URL** (*Uniform Resource Locator*)
 - Protocollo, host name (, app name), path (, query parameters)
 - `http://www.example.com/myApp/hello.html?name=Tom`
 - **HTTP**: Hypertext Transfer Protocol, **HTTPS** è la versione sicura (criptata)
 - Host (o “domain”) name tradotto in **indirizzo IP** da DNS (Domain Name System)
 - URI (*Uniform Resource Identifier*) non specifica il protocollo di accesso
- Un browser ne fa il **rendering** (*elevata permissività sintattica*)
 - Cerca di visualizzare anche pagine HTML scorrette
- Struttura ad albero, ogni nodo è un elemento **DOM**
 - Document Object Model

```
<!DOCTYPE html>
<!-- my hello page -->
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>Hello</title>
</head>

<body>
  <p>Hello world!</p>
</body>

</html>
```

hello.html

Web Developer Tools

- Firefox (Web Developer, Toggle Tools) / Chrome (More Tools, Developer Tools)
 - Scorciatoia comune per l'attivazione: ctrl+shift+i
- Settings (F1)
 - FF: Advanced settings, **Disable HTTP Cache** (when toolbox is open)
 - CH: Network, **Disable Cache** (while DevTools is open)
- Tab Inspector (FF) / Elements (CH)
 - Accesso alla struttura della pagina HTML
- Tab Console
 - Visualizzazione log del programmatore (JavaScript) e di sistema
 - ESC: mostra/nasconde la console nel tab corrente
- Tab Debugger (FF) / Sources (CH)
 - Accesso al codice

Elemento

- Componente di base di un documento HTML
- Normalmente delimitato da **open** – **close tag**
 - In alternativa, **elementi vuoti** non hanno il close tag
 - I tag name sono case-insensitive
- Può **contenere** testo e altri elementi
- Può avere **attributi** nella forma **nome="valore"**
 - L'ordine degli attributi di un elemento non è significativo
 - Gli attributi booleani sono nella forma *nome* o *nome="nome"*
- "!" indica che è un *non*-elemento
 - **DOCTYPE** tipo di documento. Indica al browser la versione del codice (qui: HTML5). Prima riga del file
 - Commenti HTML: <!-- ... -->

```
<!DOCTYPE html>
<!-- my hello page -->
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>Hello</title>
</head>

<body>
  <p>Hello world!</p>
</body>

</html>
```

hello.html

head vs body

- **html**

- Radice dell'albero DOM
- Contiene l'intero codice HTML della pagina
- l'attributo **lang** specifica il linguaggio del contenuto

- **head**

- Informazioni *sulla* pagina

- **body**

- Informazioni *nella* pagina
- Contenuto che vogliamo mostrare all'utente

```
<!DOCTYPE html>
<!-- my hello page -->
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>Hello</title>
</head>

<body>
  <p>Hello world!</p>
</body>

</html>
```

hello.html

head

- Gli elementi in head hanno lo scopo di descrivere la pagina corrente
 - **title**: il titolo della pagina, solitamente mostrato dal browser nella barra del titolo
 - Comunemente usato dal browser e motori di ricerca come riferimento al documento
 - **meta**: informazioni aggiuntive, come l'encoding usato e indicazioni per i motori di ricerca

```
<meta charset="utf-8">  
<meta name="description" content="Writing HTML code">  
<meta name="keywords" content="html, head, title, meta">
```
 - Per altri tipi di metadati vedi
 - The Open Graph protocol <https://ogp.me/>

Elementi di blocco vs inline

- Le due principali proprietà display degli elementi
 - **Blocco**, di solito rappresenta un elemento strutturale della pagina
 - Inizia a capo a sinistra e arriva in fondo a destra all'interno del contenitore in cui si trova
 - L'elemento che segue sarà su una nuova linea
 - Un blocco non dovrebbe essere contenuto da un inline
 - Un blocco generico è reso da un elemento **div**
 - **Inline**, spesso usato per stilare o gestire via JS solo una parte di un blocco
 - Contenuti in un blocco, occupano solo lo spazio necessario, a partire dal vicino di sinistra
 - Non implicano un andata a capo alla loro fine
 - Spesso associati a un paragrafo (elemento "p")
 - Un blocco generico è reso da un elemento **span**



Struttura della pagina

- Elementi semantici per le principali sezioni di una pagina web standard
 - **header** intestazione del sito web, di solito ripetuta nelle pagine
 - **nav** navigation bar, link alle sezioni principali del sito
 - **main** il contenuto principale della pagina
 - **article**, **section**, **div** aiutano a organizzare main
 - **aside** informazioni aggiuntive, link a risorse associate
 - **footer**: informazioni che devono essere presenti ma di importanza secondaria
- Semplicità di lettura e comprensione nello sviluppo
- Categorizzazione del contenuto da parte dei motori di ricerca
 - vedi SEO: Search Engine Optimization

Testo

- h1 ... h6
 - Titoli (heading) di parti del testo, ci si aspetta un solo H1 per pagina
- p
 - Paragrafo, unità di base per la suddivisione del testo
- b, i, u, sup, sub, ... (*Obsoleti se indicano solo uno stile – andrebbe usato CSS*)
 - Formattazione del testo, (bold → grassetto), <i>(italic → corsivo)</i>, sottolineato, esponente, pedice, ...
- strong, em
 - Enfasi, importante, equivalenti a e <i> ma veicolano più facilmente il loro senso (per cui sono detti “semantici”)
- br (*break, forza l'andata a capo*)
 - HTML ignora molteplicità di spazi, tab, andate a capo, etc. Ogni gruppo è interpretato come un singolo spazio bianco
- hr (*horizontal ruler*)
 - Separa visivamente blocchi nella pagina

id vs class

- L'attributo **id** identifica univocamente un elemento all'interno di una pagina
 - Un elemento può avere un solo id
- L'attributo **class** identifica un gruppo di elementi in un pagina
 - Un elemento può avere più classi, specificate nell'attributo class, separate da uno spazio
- I nomi di id e class sono case-sensitive, non possono contenere spazi
- Sono attributi “globali”, ogni elemento può averli
 - https://www.w3.org/wiki/HTML/Attributes/_Global
- L'uso di class e id è fondamentale per l'interazione tra
 - HTML – CSS – JavaScript

Atomic Design

- Atomo
 - Elementi di base
- Molecola
 - Più atomi che concorrono ad uno scopo comune
- Organismo
 - Componente complesso di molecole cooperanti
- Template
 - Sorta di pagina astratta, completa ma generica
- Pagina
 - Risultato concreto