



**UNIMORE**

UNIVERSITÀ DEGLI STUDI DI  
MODENA E REGGIO EMILIA

Dipartimento di Scienze e Metodi  
dell'Ingegneria

# Tecnologie Web e Internet of Things

Stefania Monica  
[stefania.monica@unimore.it](mailto:stefania.monica@unimore.it)



## Il Linguaggio HTML

**HTML** è l'acronimo di *HyperText Markup Language*

- È il linguaggio utilizzato per descrivere le pagine che costituiscono i nodi dell'ipertesto
- È un linguaggio di codifica del testo del tipo a **marcatori** (markup)
- Un **linguaggio di codifica del testo** è un formalismo con il quale è possibile rappresentare un documento su supporto digitale in modo che sia trattabile dall'elaboratore in quanto testo

## Sistemi di Codifica dei Caratteri

- I formalismi più elementari per la codifica informatica del testo sono i *sistemi di codifica dei caratteri*
- In generale, ogni documento elettronico è costituito da una stringa di caratteri
- Come qualsiasi altro tipo di dato, anche i caratteri vengono rappresentati all'interno di un elaboratore mediante una codifica binaria
- Per codificare i caratteri si stabilisce una corrispondenza biunivoca tra gli elementi di una collezione ordinata di caratteri e un insieme di codici numerici
- Si ottiene così un *coded character set* che di solito si rappresenta in forma di tabella (code page o code table)

## Linguaggi a Marcatori

- La codifica dei caratteri non esaurisce i problemi di rappresentazione di un testo
- Un testo è un oggetto complesso caratterizzato da molteplici livelli strutturali che non si limitano alla sequenza di simboli del sistema di scrittura
- Si parla propriamente di *linguaggio di codifica testuale* solo in riferimento ai *linguaggi che consentono la rappresentazione o il controllo di uno o più livelli strutturali di un documento testuale*
  - Tali linguaggi vengono correntemente denominati linguaggi a marcatori (*markup languages*)

## Linguaggi a Marcatori

Un linguaggio di markup è composto da:

- un insieme di istruzioni dette **tag** o **markup** (marcatori) che rappresentano le caratteristiche del documento testuale
  - una **grammatica** che regola l'uso del markup
  - una **semantica** che definisce il dominio di applicazione e la funzione del markup
- 
- I marcatori vengono inseriti direttamente all'interno del testo cui viene applicato
  - NOTA: Ogni tag è a sua volta costituito da una sequenza di caratteri, preceduta da caratteri speciali che la delimitano e permettono all'elaboratore di distinguere il testo dai marcatori

## Classificazione

Tradizionalmente i linguaggi di markup sono stati divisi in due tipologie:

- linguaggi *procedurali* o imperativi
- linguaggi *dichiarativi* o descrittivi

Questa classificazione risale a Charles Goldfarb, il padre di **SGML**

- Nei linguaggi *procedurali* il markup specifica quali operazioni un dato programma deve compiere su un documento elettronico per ottenere una determinata presentazione
  - Tex, LaTeX, ...
- Nei linguaggi *dichiarativi* il markup descrive la struttura di un documento testuale, identificandone i componenti
  - SGML, HTML, XML, ...

## Linguaggi Dichiarativi

Viene descritta la struttura editoriale, costituita da componenti (*content object*) organizzati in modo gerarchico

- Frontespizio, introduzione, corpo, appendice...
- Capitoli, sottocapitoli, atti, scene, canti...
- Titoli, epigrafi, abstract...
- Paragrafi, versi, battute, entrate di dizionario...
- Enfasi, citazioni...



## SGML

- E' l'acronimo di **Standard Generalized Markup Language**
- È uno standard ISO (8879) pubblicato nel 1986
- È un meccanismo flessibile e portabile per rappresentare documenti elettronici, proposto da Charles Goldfarb
- Un documento SGML comprende oggetti di varie classi chiamati elementi
  - Capitoli, titoli, riferimenti, oggetti grafici, etc.
- SGML identifica gli estremi degli elementi tramite **tag iniziali** e **tag finali**
- NOTA: Non contiene sequenze di istruzioni di formattazione
- Gli elementi sono organizzati in una **gerarchia**
  - Un capitolo contiene un titolo ed una o più sezioni che a loro volta contengono altri elementi, ecc.

- HTML è un'applicazione SGML, ovvero un linguaggio per la rappresentazione di un tipo di documento SGML
- Ideato da Tim Berners-Lee a inizio anni novanta
- Tramite HTML è possibile realizzare documenti con una struttura semplice contenenti:
  - testo
  - immagini e contenuti multimediali
  - oggetti interattivi
  - connessioni ipertestuali ad altri documenti
  - ...
- Oltre a descrivere il contenuto, HTML associa anche significati *grafici* agli elementi che definisce
  - Istruzioni più o meno precise su come rendere graficamente gli elementi che definisce
- Questa commistione crea diversi problemi...

## Tag

- I tag HTML sono usati per definire il markup di elementi HTML
- Sono preceduti e seguiti rispettivamente da `<` e `>` (parentesi angolari)
- Sono normalmente accoppiati
  - un esempio è dato da: `<p>` e `</p>`, detti rispettivamente **start tag** ed **end tag**
- Il testo tra start tag ed end tag è detto **contenuto** dell'elemento
- Un documento HTML contiene quindi elementi composti da testo semplice delimitato da tag

## Grammatica Poco Rigorosa

- HTML rispetta in maniera poco rigorosa le specifiche SGML
  - Ammette elementi senza chiusura come `<br>`
  - I tag non sono case sensitive
  - L'apertura e chiusura di tag annidati può essere “incrociata”, ad esempio:  
`<b><i>Testo corsivo grassetto</b></i>`
- Esistono però delle buone pratiche che è bene rispettare e che diventano un obbligo in una versione più rigorosa del linguaggio chiamata **XHTML**
  - Chiudere sempre anche i tag singoli:
  - `<br></br>` o in forma sintetica `<br/>`
  - Tag in minuscolo
  - Apertura e chiusura senza incroci, ad esempio: `<b><i>Testo corsivo grassetto</i></b>`

## Caratteri Speciali

HTML definisce un certo numero di entità (entity) per rappresentare i caratteri speciali senza incorrere in problemi di codifica:

- Caratteri riservati a HTML (<, >, &, “, ecc.)
- Caratteri non presenti nell'ASCII a 7 bit

### Esempi:

- &amp;      &
- &lt;      <
- &agrave;    à
- &Aacute;   Á
- ...

## Attributi

- Un elemento può essere dettagliato mediante **attributi**
- Gli attributi sono **coppie** “*nome = valore*” contenute nello *start tag* con una sintassi di questo tipo

`<tag attrib1='valore1' attrib2='valore2'>`

- I valori sono racchiusi da apici singoli o doppi

`<input type='submit' value="NAME">Ok</input>`

Gli apici possono essere omessi se il valore non contiene spazi

# Tecnologie Web e Internet of Things

- Lo **standard MIME** (Multipurpose Internet Mail Extensions) è nato originariamente per poter allegare file di dati (audio, video, immagini, ...) ai messaggi di posta elettronica
- Oggi, noto anche come *Internet Media Type*, **rappresenta il tipo di contenuto di un messaggio** (ad esempio: HTTP request)
- Classifica i tipi di contenuto sulla base di una logica a due livelli ed è largamente utilizzata in ambito di HTML e delle tecnologie Web in generale
- Un tipo MIME è espresso con questa sintassi:

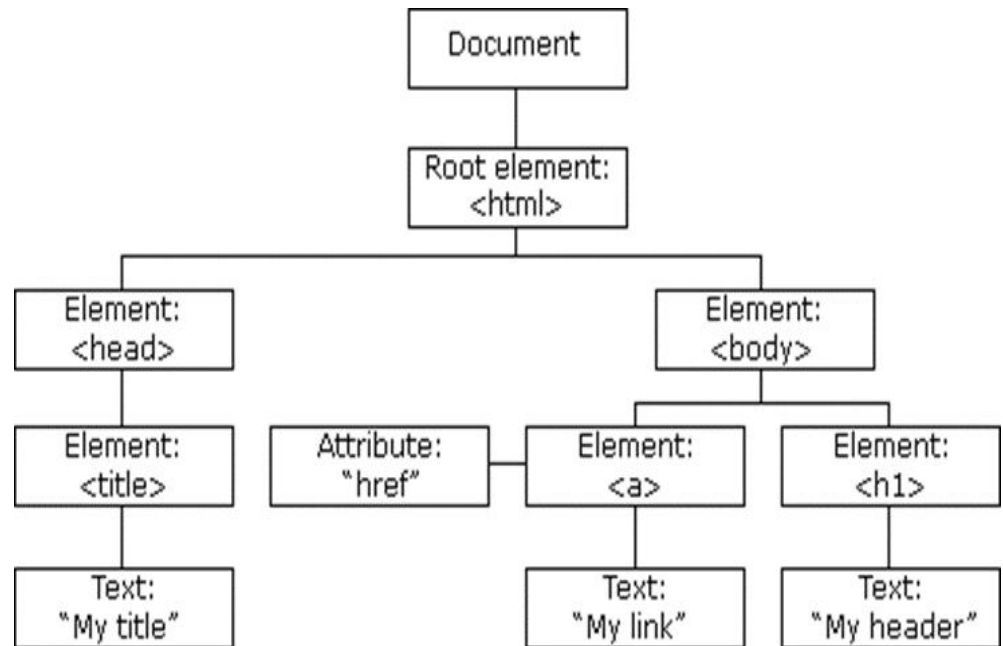
**tipo/sottotipo**

Esempi:

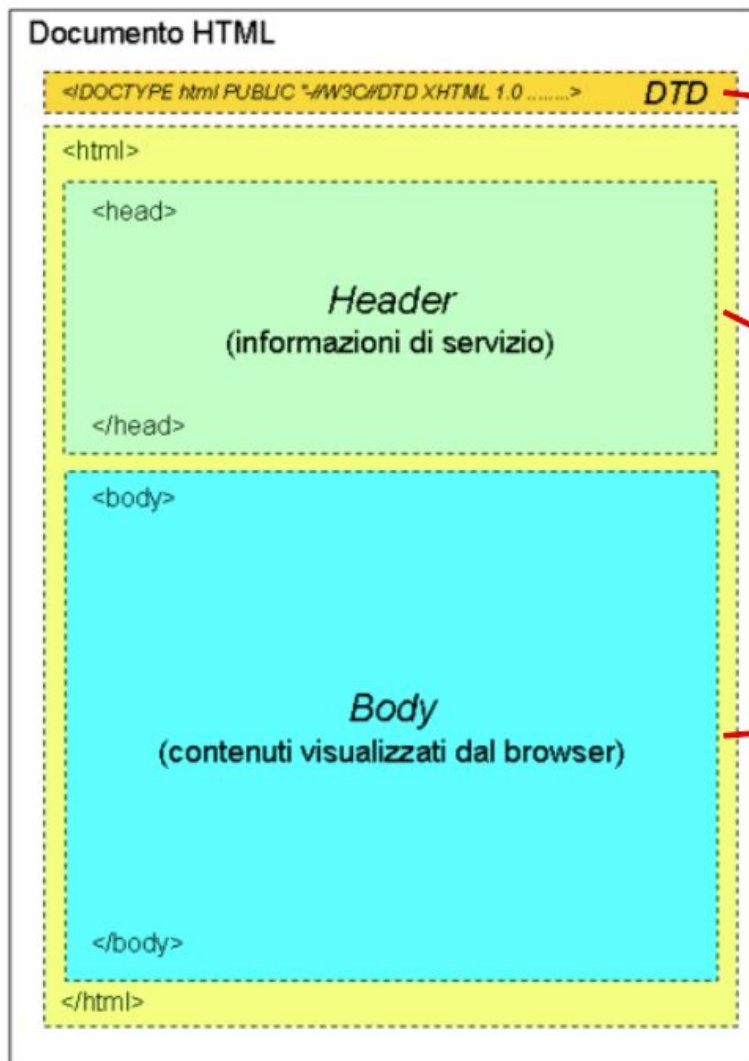
- text/plain: testo semplice
- text/html: testo HTML

## Struttura ad Albero

- Una pagina HTML può essere rappresentata come una struttura ad albero
- Questa struttura logica prende il nome di **DOM**: Document Object Model
- Quando un browser carica una pagina HTML la scompone e costruisce la struttura ad albero del DOM







```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```
<html>
```

```
<head>
  <title>Hello
document</title>
</head>
```

```
<body>
  Hello World!
</body>
```

```
</html>
```

## DTD

- Il primo elemento di un documento HTML è la definizione del tipo di documento (**Document Type Definition** o **DTD**):
- Serve al browser per identificare le regole di interpretazione e visualizzazione da applicare al documento

### Esempi:

- `<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" http://www.w3.org/TR/html4/loose.dtd>`
- `<!DOCTYPE HTML>`

## Header

- È identificato dal tag `<head>`
- Contiene elementi non visualizzati dal browser (informazioni di servizio)
  - `<title>` titolo della pagina (viene mostrato nella testata della finestra principale del browser)
  - `<meta>` metadati informazioni utili ad applicazioni esterne (es. motori di ricerca) o al browser (es. lingua, codifica dei caratteri utile per la visualizzazione di alfabeti non latini)
  - `<base>` definisce come vengono gestiti i riferimenti relativi nei link
  - `<link>` collegamenti verso file esterni: CSS, script, icone visualizzabili nella barra degli indirizzi del browser
  - `<script>` codice eseguibile utilizzato dal documento
  - `<style>` informazioni di stile (CSS locali)

## Header

- Gli elementi di tipo `<meta>` sono caratterizzati da una serie di attributi
- Esistono due tipi di elementi meta, distinguibili dal primo attributo: `http-equiv` o `name`
  - Gli elementi di tipo `http-equiv` danno informazioni al browser su come gestire la pagina
  - Hanno una struttura di questo tipo:

`<meta http-equiv=nome content=valore>`

- Gli elementi di tipo `name` forniscono informazioni utili, ma non critiche
- Hanno una struttura di questo tipo:

`<meta name=nome content=valore>`

## Elementi <meta> di tipo http-equiv

- refresh: indica che la pagina deve essere ricaricata dopo un numero di secondi definito dall'attributo content

`<meta http-equiv=refresh content=45>`

- expires: stabilisce una data scadenza (fine validità) per il documento

`<meta http-equiv=expires content="Tue, 22 Aug 2023 14:25:27 GMT">`

- content type: definisce il tipo di dati contenuto nella pagina (di solito il tipo MIME text/html):

`<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">`

## Elementi <meta> di tipo name

- author: autore della pagina  
`<meta name=author content='John Smith'>`
- description: descrizione della pagina  
`<meta name=description content="Home page UNIMORE">`
- copyright: indica che la pagina è protetta da un diritto d'autore  
`<meta name=copyright content="Copyright 2009, John Smith">`
- keywords: lista di parole chiave separate da virgole, usate dai motori di ricerca  
`<meta name=keywords lang="en" content="computer documentation, computers, computer help">`
- date: data di creazione del documento  
`<meta name="date" content="2008-05-07T09:10:56+00:00">`

## body

- Il tag `<body>` delimita il corpo del documento
- Contiene la parte che viene mostrata dal browser
- Ammette diversi attributi tra cui:
  - `background = uri`  
Definisce l'URI di una immagine da usare come sfondo per la pagina
  - `text = color`  
Definisce il colore del testo
  - `bgcolor = color`  
In alternativa a `background` definisce il colore di sfondo della pagina
  - `lang = linguaggio`  
Definisce la lingua utilizzata nella pagina, ad esempio `language="it"`

## Tipi di elementi del body

- **Intestazioni**: titoli organizzati in gerarchia
- **Strutture** di testo: paragrafi, testo indentato, ...
- **Aspetto** del testo: grassetto, corsivo, ...
- **Elenchi** e **liste**: numerati, puntati
- **Tabelle**
- **Form** (moduli elettronici): campi di inserimento, checkbox e radio button, menu a tendina, bottoni, ...
- **Collegamenti** ipertestuali e ancore
- **Immagini** e contenuti multimediali (audio, video, ...)
- **Contenuti interattivi**: script, applicazioni esterne



## Block-level, Inline e Liste

Dal punto di vista del layout della pagina gli elementi HTML si dividono in 3 grandi categorie:

- Elementi **block-level**: costituiscono un blocco attorno a sé, e di conseguenza «vanno a capo» (paragrafi, tabelle, form...)
- Elementi **inline**: non vanno a capo e possono essere integrati nel testo (link, immagini,...)
- **Liste**: numerate, puntate

Regole di composizione:

- Un elemento block-level può contenere altri elementi dello stesso tipo o di tipo inline
- Un elemento inline può contenere solo altri elementi inline

## Titoli

- I tag `<h1>`, `<h2>`, ... `<h6>` servono per definire dei titoli di importanza decrescente (`<h1>` è il più importante, cioè il più grande)
- La "h" sta per "heading", cioè titolo, e sono previste 6 grandezze
- I titoli appaiono in grassetto e lasciano una riga vuota prima e dopo di sé (block-element)
- Ammettono attributi di allineamento:

`<h1 align = left | center | right | justify>`

## Paragrafi

- Il paragrafo è l'unità di base entro cui suddividere un testo: è un block-element
- Il tag `<p>` lascia una riga vuota prima della sua apertura e dopo la sua chiusura
- Se si vuole andare a capo all'interno di un paragrafo si usa l'elemento `<br/>`
- Se al posto di `<p>` si usa il tag `<div>` il blocco di testo va a capo, ma non lascia spazi prima e dopo la sua apertura
- È il block-element per eccellenza

## span

- `<span>` è un contenitore generico che può essere annidato (ad esempio all'interno dei `<div>`)
- È un elemento *inline*, e quindi non va a capo ma continua sulla stessa linea del tag che lo include
- È un elemento molto utilizzato soprattutto insieme ai *fogli di stile* per dare un aspetto particolare ad un pezzo di testo in un blocco
- NOTA: Se non viene associato ad uno stile è invisibile

## Contenitori di testo: riepilogo

Ricapitolando: `<p>`, `<div>` e `<span>` sono tre diversi tipi di contenitori di testo

Si comportano in modo diverso:

- `<p>` è un elemento di blocco e lascia spazio prima e dopo la propria chiusura
- `<div>` è un elemento di blocco, non lascia spazio prima e dopo la propria chiusura, ma va a capo
- `<span>` è un elemento inline e quindi non va a capo

## Gli Stili del Testo

- Nella terminologia tipografica lo *stile di un testo* indica le possibili varianti di forma di un carattere
- HTML consente di definire lo stile di un frammento di testo, combinando fra loro anche più stili
- I tag che svolgono questa funzione vengono normalmente suddivisi in **fisici** e **logici**:
  - Tag **fisici**: definiscono lo stile del carattere in termini grafici  
`<tt>...</tt>`  
`<i>...</i>`  
`<b>...</b>`
  - Tag **logici**: forniscono informazioni sul ruolo svolto dal contenuto, e in base a questo adottano uno stile grafico  
`<address>`  
`<blockquote>`  
`<cite>`

## Liste

- Il tag `<ul>` (unordered list) permette di definire liste non ordinate (elenchi puntati)
  - L'attributo `type` definisce la forma dei punti: `disc`, `circle`, `square`
- Il tag `<ol>` (ordered list) permette di definire liste ordinate (elenchi numerati)
  - L'attributo `type` definisce il tipo di numerazione: `1` (1,2,...), `a` (a,b,...), `A` (A,B,...), `i` (i,ii,...) e `I` (I,II,...)
- Gli elementi di una lista vengono definiti mediante il tag `<li>` (list item)

## Tabelle

Il tag `<table>` racchiude la tabella

Attributi:

- `align = "{left|center|right}"`  
allineamento della tabella rispetto alla pagina;
- `width="n|n%"`  
larghezza della tabella (anche in percentuale rispetto alla pagina);
- `bgcolor="#xxxxxx"`  
colore di sfondo della tabella;



## Righe

`<tr>` è il tag che racchiude ciascuna riga della tabella

Attributi:

- `align = "{left|center|right|justify}"`  
allineamento del contenuto delle celle della riga;
- `valign = "{top|middle|bottom|baseline}"`  
allineamento verticale del contenuto delle celle della riga;
- `bgcolor="#xxxxxx"`  
colore di sfondo della riga

## Testate e Celle

<th> e <td> sono i tag che racchiudono le celle

<th> serve per le celle della testata

<td> serve per le celle del contenuto

Attributi:

- Gli stessi di <tr>
- width, height = {length|length%}

specifica le dimensioni (larghezza e altezza) della cella, dimensione assoluta (pixels) o valore percentuale;

- rowspan, colspan = n

indica su quante righe/colonne della tabella si estende la cella

## Immagini e Link

- Per inserire un'immagine si usa il tag `<img>` nel seguente modo:

```

```

- I Link in HTML sono collegamenti ipertestuali realizzati attraverso il tag `<a>`

```
<a href="URI">TESTO</a>
```

## Form

- Un **form** (modulo) è una sezione di documento HTML che contiene elementi di controllo che l'utente può utilizzare per inserire dati o in generale per interagire
- I dati inseriti possono essere poi inoltrati al server
- Gli elementi di controllo sono caratterizzati da un valore iniziale e da un valore corrente
- Gli elementi di controllo possono essere:
  - Bottoni di azione
  - Checkbox (caselle di spunta)
  - Radio Button (bottoni mutuamente esclusivi)
  - Liste di selezione (lista di opzioni)
  - Caselle di inserimento di testo
  - Oggetti nascosti (elementi valorizzati ma invisibili)

## input

- La maggior parte dei controlli viene definita mediante il tag
- `<input>`
- L'attributo `type` stabilisce il tipo di controllo
  - `text`: casella di testo monoriga
  - `password`: come `text` ma il testo non è leggibile (\*\*\*\*)
  - `file`: controllo che consente di caricare un file
  - `checkbox`: casella di spunta
  - `radio`: radio button
  - `submit`: bottone per trasmettere il contenuto del form
  - `image`: bottone di submit sotto forma di immagine
  - `reset`: bottone che riporta tutti i campi al valore iniziale
  - `button`: bottone di azione
  - `hidden`: campo nascosto
- Tutti gli input possono essere disabilitati utilizzando l'attributo `disabled` nella forma `disabled = "disabled"`

## input text

È un campo per l'inserimento di testo su una sola riga

Attributi:

- `name = text`  
nome del controllo
- `value = text`  
eventuale valore iniziale
- `size = n`  
lunghezza del campo (numero di caratteri)
- `maxlength = n`  
massima lunghezza del testo (numero di caratteri)

## **input checkbox**

Un input con tipo “checkbox” definisce una casella di spunta

Attributi:

- **name = text**  
nome del controllo
- **value = text**  
valore restituito se la casella viene spuntata
- **checked = “checked”**

## Bottoni

Per i bottoni si utilizzano tre valori dell'attributo type

- submit per il bottone che provoca la spedizione del form al server
- reset per il bottone che riporta il contenuto dei campi al valore originale
- button per un generico bottone di azione

L'etichetta che compare nel bottone viene definita usando l'attributo value

- In HTML 4 è stato introdotto il tag `<button>`



## Tendine

- Il tag `<select>` permette di costruire liste di opzioni
- Per definire le singole opzioni si usa il tag `<option>` ricorrendo all'attributo `value` si può attribuire il valore
- Con l'attributo `selected` si può indicare una scelta predefinita: `selected="selected"`
- L'aspetto di default è quello di un combo box (tendina a discesa)
- Se si utilizza l'attributo `multiple` (`multiple="multiple"`) non abbiamo più un combo ma una lista sempre aperta
  - Si può operare una scelta multipla tenendo premuto il tasto [Ctrl] durante la selezione
  - L'attributo `size` determina il numero di righe mostrate
- Con il tag `<optgroup>` è possibile organizzare la lista (sia a scelta singola che multipla) in gruppi
  - Molto utile per liste lunghe



## I Fogli di Stile

## Fogli di Stile

- E' possibile modificare lo stile, utilizzando un foglio di stile, con estensione .css
- Il collegamento al file CSS viene scritto nell'header della pagina html usando:

```
<link rel="stylesheet" href="style.css" type="text/css" media="all" />
```

- La sintassi è la seguente

Selettore di stile {

Attributo: Valore;

Attributo: Valore;

...

Attributo: Valore;

}

## Attributi e valori

### color

Stabilisce il colore del testo

### background-color

Stabilisce il colore di sfondo del testo

E' possibile inserire i colori in 3 modi:

- Un nome di colore valido (e.g., "red", "blue", "cyan", . . . )
- Un valore RGB value (e.g., "rgb(0, 84, 147)")
- Un valore esadecimale (e.g., #005493;)

## Attributi e valori

### font-family

serif

sans-serif

monospace

### font-style

normal

italic

### font-weight

normal

bold

## Attributi e valori

### font-size

Stabilisce la dimensione dei caratteri (e.g., 16px;).

### text-align

text-align:center;

text-align:left;

text-align:right;

text-align:justify;

## Attributi e valori

### margin/padding/border

Usato per impostare la dimensione dello spazio vuoto attorno a un elemento

auto: valori calcolati dal browser

larghezza: specifica la larghezza dei margini in px, em

Si possono specificare i valori per ogni lato, ad esempio

margin-top, margin-right, margin-bottom, margin-left

Esempi:

margin: 25px 50px 75px 100px

margin: 25px 50px 75px

margin: 25px 50px

margin: 25px

## Attributi e valori

border-style

dotted;

dashed;

solid;

double;

dotted dashed solid double;

border-width

border-color



## Attributi e valori

### height/width

usati per impostare l'altezza di un elemento

auto: valori calcolati dal browser

larghezza: specifica l'altezza in px, e

## Id e Classi

E' possibile definire **id** e/o **classi**

E' possibile associare uno stile a un elemento con un certo id o certa classe

**id**

id univoco

id definito con:

```
#nome1 {
```

```
...
```

```
}
```

id definito con **id="nome1"** nel file .html

## Id e Classi

### classi

classe non univoca

classe definita con:

```
.nome1 {  
    ...  
}
```

classe definita con `class="nome1"` nel file .html

# Tecnologie Web e Internet of Things

- L'attributo **display** può essere usato con valore flex per rendere flessibile un elemento
- L'attributo **flex-direction** specifica la direzione in cui rendere l'elemento flessibile
- L'attributo **overflow** può essere usato per controllare quello che succede quando un contenuto è troppo grande per una determinata area. I possibili valori sono:
  - visible (default).
  - hidden
  - scroll
  - overlay
  - auto (come scroll ma aggiunge la barra solo se necessario)
- L'attributo **box-sizing** è usato per includere padding e bordo nella larghezza/altezza totale di un elemento. Il valore da utilizzare è
  - border-box

# Tecnologie Web e Internet of Things

Maggiori informazioni su HTML e CSS:

<https://www.w3schools.com/HTML>

<https://www.w3schools.com/CSS>