

HTML



CSS

PROGRAMMA



1

Introduzione HTML

2

Iniziare un progetto HTML

3

I tag base e gli attributi



1

Introduzione CSS

2

I selettori e la sintassi

3

Modificare gli elementi di una pagina
usando CSS

1 INTRODUZIONE HTML

HTML, acronimo di "Hypertext Markup Language", è il linguaggio fondamentale per la creazione di pagine web. Non è un linguaggio di programmazione, ma un linguaggio di markup, che utilizza i tag per definire gli elementi di un documento.

Un file HTML è, in origine, un semplice documento di testo non formattato. All'interno del documento, si inserisce non solo il testo che si vuole far apparire nella pagina web, ma anche alcune istruzioni specifiche (TAG o marcatori) del linguaggio HTML. Questi tag, contenuti all'interno dei simboli "<" e ">", forniscono all'HTML le indicazioni per la formattazione del documento, oltre che per l'inserimento di immagini e altri elementi multimediali.

Per scrivere codice HTML, si può utilizzare qualsiasi editor HTML testuale, o anche solo il Blocco Note di Windows. I siti web, prima della pubblicazione, vengono creati in locale, cioè sul tuo hard disk.

In sostanza, l'HTML è il linguaggio che stabilisce cosa un browser web deve mostrare a un visitatore quando arriva sul tuo sito. È il linguaggio che assicura che il testo e le immagini appaiano nel modo in cui desideri.

PROGRAMMA



Introduzione HTML



Iniziare un progetto HTML



I tag base e gli attributi



Introduzione CSS



I selettori e la sintassi



Modificare gli elementi di una pagina
usando CSS



INIZIARE UN PROGETTO HTML

Per iniziare un progetto html sarà sufficiente creare un file che abbia come estensione .html.

Per convenzione, il primo file html che andrà creato sarà sempre index.html.

Possiamo iniziare a fare tutto ciò seguendo i seguenti passaggi:

- Creare una cartella e nominarla con il nome del progetto che stiamo per andare a creare (nel nostro caso “Primo progetto html”;
- Apriamo visual studio code e apriamo la cartella appena creata.
- All’ interno di questa cartella creiamo un file nominato “index.html” e apriamo il file

La prima cosa che vedremo del nostro file html sarà il vuoto cosmico... tutto normale, dobbiamo ancora iniziare a scriverci quello che vogliamo far vedere, ma la struttura della nostra cartella, dovà avere un aspetto come questo:

✓ **PRIMO PROGETTO HTML**

✎ index.html

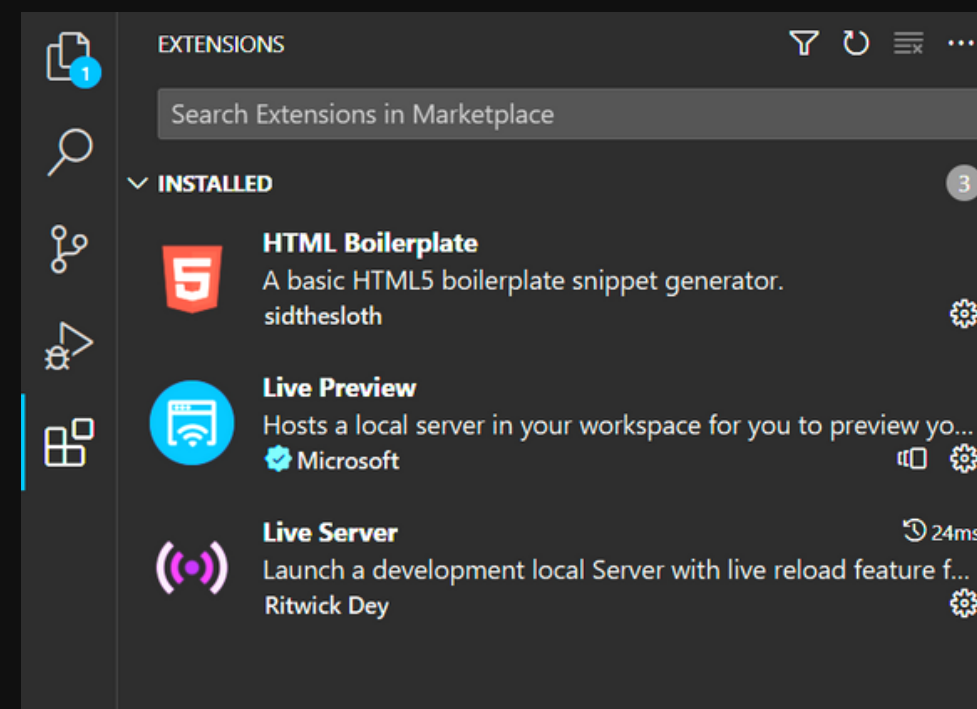


INIZIARE UN PROGETTO HTML

Abbiamo parlato della possibilità di creare HTML con qualsiasi tipo di editor di testo, giusto? Tuttavia, esistono degli strumenti chiamati IDE (Integrated Development Environment) che rendono questa creazione molto più efficiente e ottimizzata.

VSC (Visual Studio Code) è uno di quegli IDE gratuiti che spopola tra gli sviluppatori. Non solo per la sua interfaccia intuitiva, ma anche per l'ampia disponibilità di estensioni gratuite che lo rendono completo e versatile per svariate tipologie di linguaggi.

Quindi, prima di metterci all'opera, consiglio vivamente di aggiungere alcune estensioni essenziali nella sezione “extensions” sul lato sinistro del vostro IDE:

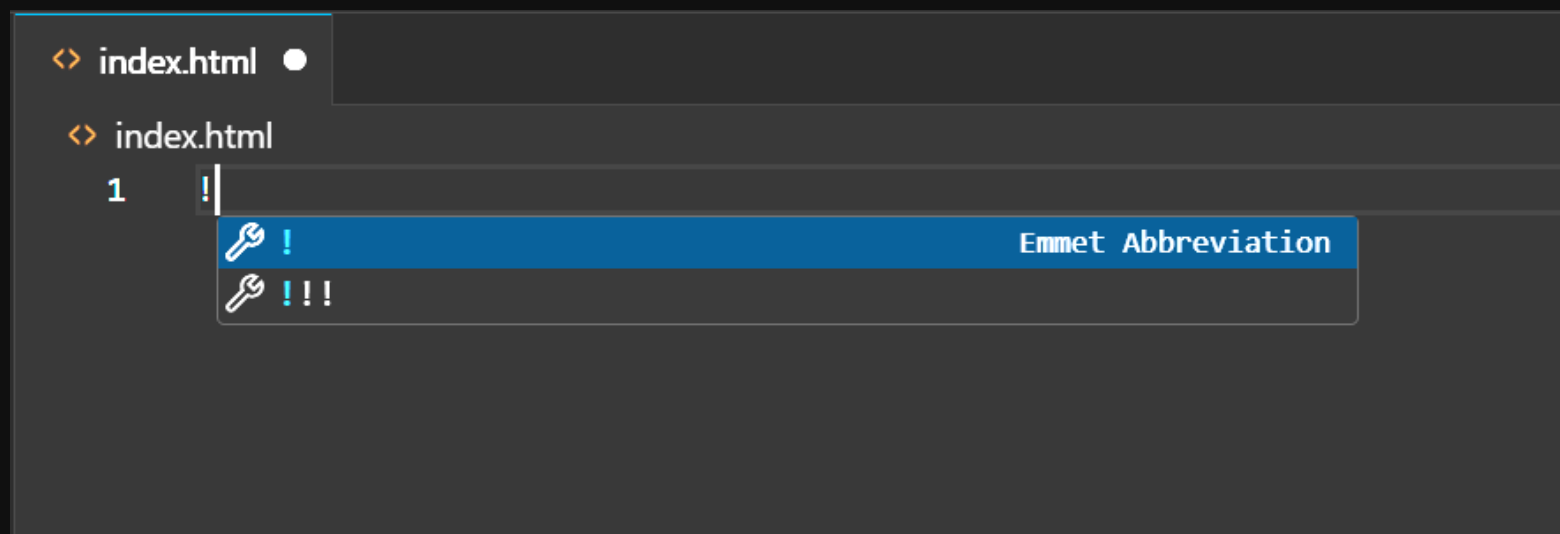


2

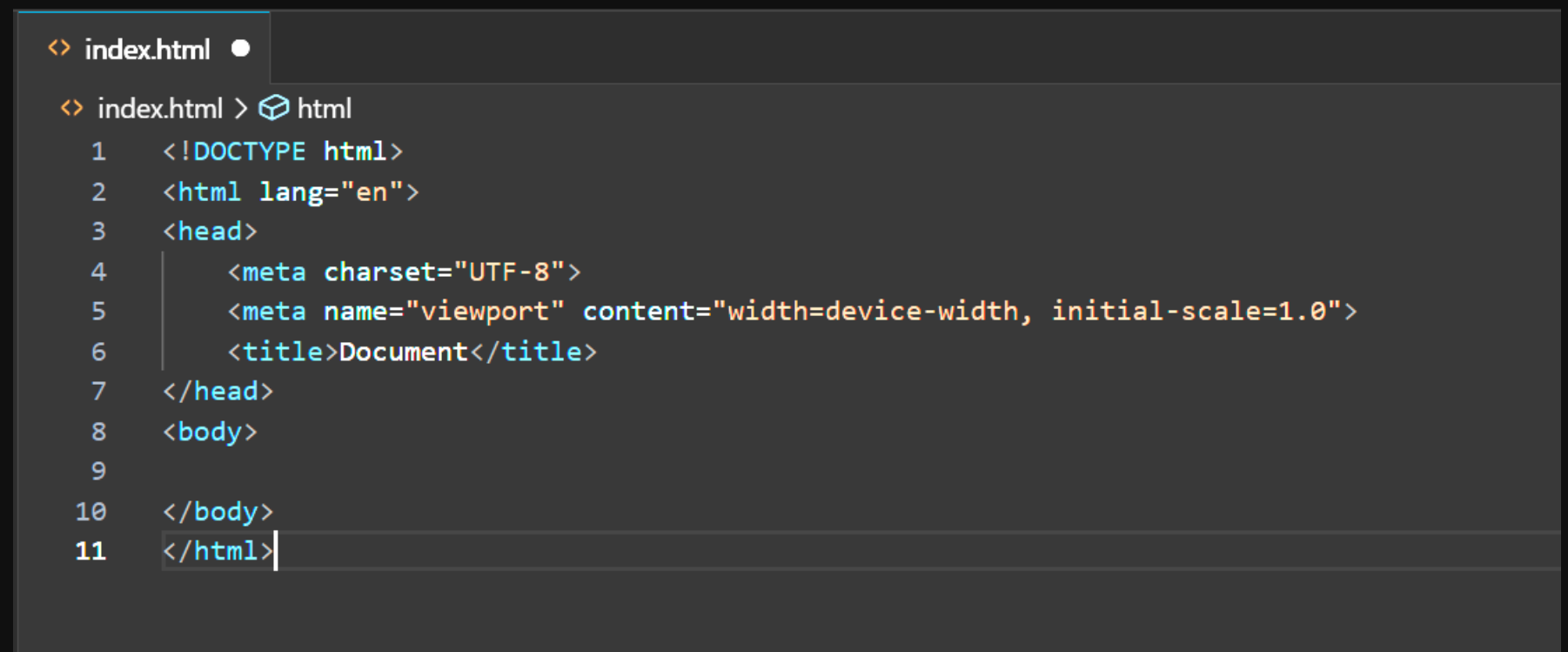
INIZIARE UN PROGETTO HTML

Iniziamo a riempire quel vuoto cosmico di cui abbiamo parlato prima!

Apriamo il nostro file index.html e grazie ad una shortcut di VSC, ci basterà premere “!” per visualizzare un suggerimento e selezionare il primo risultato che ci viene proposto.



A screenshot of the Visual Studio Code editor interface. The top tab bar shows 'index.html' with a code icon and a close button. Below it, the file explorer shows 'index.html'. The main editor area shows line 1 with the character '!' typed. A dropdown menu is open, displaying two options: '!' and '!!!', each preceded by a key icon. The text 'Emmet Abbreviation' is visible on the right side of the dropdown.



A screenshot of the Visual Studio Code editor interface showing the completed HTML boilerplate code. The top tab bar shows 'index.html' with a code icon and a close button. Below it, the file explorer shows 'index.html'. The main editor area shows the following code:

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Document</title>
7 </head>
8 <body>
9
10 </body>
11 </html>
```

2

INIZIARE UN PROGETTO HTML

```
<> index.html •
<> index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Document</title>
7  </head>
8  <body>
9
10 </body>
11 </html>
```

Tanto per cominciare, quelli che vedete qui, sono tag html che come abbiamo visto prima, sono avvolti dalle parentesi angolari "<>". Vediamo ora che cosa vogliono dire nello specifico:

- `<!DOCTYPE html>`: è una dichiarazione introduttiva in un documento HTML che specifica la versione di HTML utilizzata nella pagina. In pratica, indica al browser quale standard HTML seguire durante la visualizzazione del contenuto della pagina web.
- `<html>`: è il punto di partenza di ogni documento HTML. Racchiude l'intero contenuto della pagina e fornisce il contesto per gli altri elementi HTML.
- `<head>`: Contiene metadati e informazioni che non vengono visualizzate direttamente sulla pagina, ma forniscono indicazioni importanti al browser e ai motori di ricerca.
- `<body>`: è come il cuore pulsante della tua pagina HTML. È all'interno di questo tag che inseriamo tutto il contenuto visibile della pagina web.



INIZIARE UN PROGETTO HTML

I tag che abbiamo esaminato fino a ora sono fondamentali per comporre la nostra prima pagina.

Se ora apriremo il nostro file nel browser, ovviamente non vedremo nulla, poiché non abbiamo ancora inserito nulla nel body. Prima di iniziare a aggiungere e visualizzare elementi nella nostra pagina, è essenziale fare una breve panoramica sui principali tag HTML e il loro scopo.

PROGRAMMA



Introduzione HTML



Iniziare un progetto HTML



I tag base e gli attributi



Introduzione CSS



I selettori e la sintassi



Modificare gli elementi di una pagina
usando CSS



I TAG DI BASE E I LORO ATTRIBUTI

I tag di base in HTML sono elementi fondamentali utilizzati per strutturare e definire il contenuto di una pagina web. Ogni tag ha un ruolo specifico e può contenere attributi che forniscono ulteriori informazioni su come il tag dovrebbe comportarsi o essere visualizzato. Questi tag, vanno inseriti all'interno dei tag `<body>` che, come abbia detto prima, è il luogo in cui possiamo scrivere quello che vogliamo visualizzare all'interno della nostra pagina web.

TAG CONTENITORI

- `<div>`
- ``
- `<section>`
- `<article>`
- `<header>`
- `<footer>`
- `<nav>`
- `<aside>`
- `<main>`

TAG DI TESTO

- `<h1>`, `<h2>`, ..., `<h6>`
- `<p>`
- `<a>`
- ``
- ``
- `<abbr>`
- `<blockquote>`
- `<q>`
- `<cite>`
- `
`
- `<hr>`

TAG MEDIA

- ``
- `<audio>`
- `<video>`
- `<figure>`
- `<figcaption>`
- `<iframe>`



I TAG CONTENITORE

`<div>` (Divisione):

- Il tag `<div>` è un contenitore generico che viene utilizzato per raggruppare e strutturare altri elementi HTML.
- Non ha significato semantico proprio, ma può essere stilizzato con CSS e utilizzato per organizzare il layout della pagina.

``:

- Il tag `` è simile a `<div>`, ma è utilizzato per raggruppare elementi inline e applicare stili o script specifici a parti del testo senza influire sulla struttura del documento.

`<section>`:

- `<section>` è utilizzato per definire sezioni di contenuto tematicamente correlato in un documento HTML.
- Può contenere titoli, paragrafi, immagini e altri elementi per organizzare in modo logico il contenuto della pagina.

`<article>`:

- `<article>` è utilizzato per rappresentare un contenuto indipendente e distribuibile, come un articolo, un post di blog o un widget che può esistere autonomamente.

`<header>`:

- `<header>` è utilizzato per definire l'intestazione di una sezione o di tutto il documento.
- Contiene elementi come titoli (`<h1>` - `<h6>`), loghi, navigazione, o altri elementi di introduzione.

`<footer>`:

- `<footer>` definisce il piè di pagina di una sezione o di tutto il documento.
- Contiene solitamente informazioni di chiusura come copyright, link alle pagine legali, ecc.
-

`<nav>` (Navigazione):

- `<nav>` è utilizzato per definire una sezione di navigazione.
- Contiene spesso collegamenti a diverse parti del sito o a pagine correlate.

`<aside>`:

- `<aside>` rappresenta contenuto che è separato dal contenuto principale, come una barra laterale o un riquadro di contenuto affiancato.

`<main>`:

`<main>` è utilizzato per contenere il contenuto principale della pagina. È escluso da altre aree come intestazioni, piè di pagina o barre laterali.

Deve essere utilizzato una sola volta in un documento HTML.



I TAG DI TESTO

`<h1>`, `<h2>`, ..., `<h6>` (Intestazioni):

- `<h1>` rappresenta il titolo principale, mentre `<h6>` è il titolo meno importante.
- Vengono utilizzati per strutturare la gerarchia del testo e comunicare l'importanza relativa dei diversi titoli.

`<p>` (Paragrafo):

- Il tag `<p>` viene utilizzato per definire un paragrafo di testo.
- Ogni `<p>` inizia e termina con un nuovo spazio verticale, contribuendo a separare visivamente i diversi blocchi di testo.

`<a>` (Collegamento ipertestuale):

- Il tag `<a>` viene utilizzato per creare link ipertestuali verso altre pagine web, documenti, immagini, o risorse online.
- L'attributo principale è `href`, che indica il percorso della risorsa collegata.

`` (Testo in grassetto):

- `` è utilizzato per enfatizzare il testo in modo più forte, solitamente visualizzandolo in grassetto.

`` (Testo in corsivo):

- `` è utilizzato per enfatizzare il testo in modo leggero, solitamente visualizzandolo in corsivo.

`<abbr>` (Abbreviazione):

- `<abbr>` viene utilizzato per definire abbreviazioni o acronimi e può includere l'attributo `title` per fornire una spiegazione completa quando il cursore si posiziona sopra di esso.

`<blockquote>` (Blocco di citazione):

- `<blockquote>` viene utilizzato per citare un blocco di testo proveniente da un'altra fonte.
- Spesso accompagnato da `<cite>` per indicare la fonte della citazione.

`<q>` (Citazione):

- `<q>` è utilizzato per citare brevi frammenti di testo.
- Può essere utilizzato all'interno di un paragrafo.

`<cite>` (Citazione di una fonte):

- `<cite>` è utilizzato per indicare la fonte di una citazione.
- Spesso utilizzato all'interno di `<blockquote>` o `<q>`.

`
` (Interruzione di linea):

- `
` viene utilizzato per inserire un'interruzione di linea, spostando il testo successivo su una nuova riga.

`<hr>` (Righello orizzontale):

- `<hr>` viene utilizzato per creare una linea orizzontale che separa sezioni di contenuto.
- Può essere utilizzato per evidenziare divisioni logiche all'interno di una pagina.



I TAG DI MEDIA

`` (Immagine):

- Il tag `` viene utilizzato per incorporare un'immagine nella pagina web.
- L'attributo principale è `src`, che indica il percorso dell'immagine.
- L'attributo `alt` fornisce un testo alternativo che viene visualizzato se l'immagine non può essere caricata o per accessibilità.

`<audio>` (Audio):

- Il tag `<audio>` viene utilizzato per incorporare file audio nella pagina.
- Può contenere vari tag figlio come `<source>` per specificare diversi formati audio e fornire una migliore compatibilità del browser.

`<video>` (Video):

- `<video>` viene utilizzato per incorporare file video nella pagina.
- Come `<audio>`, può contenere tag figlio come `<source>` per specificare diversi formati video.

`<figure>`:

- Il tag `<figure>` è un contenitore che rappresenta qualsiasi contenuto indipendente, spesso utilizzato per immagini, grafici o tabelle.
- Fornisce un contesto semantico migliorato per il contenuto all'interno.

`<figcaption>`:

- `<figcaption>` è utilizzato all'interno di `<figure>` e fornisce una didascalia o una descrizione per l'elemento contenuto in `<figure>`.
- Aiuta a spiegare o fornire contesto aggiuntivo all'elemento in `<figure>`.

`<iframe>` (Inline Frame):

- `<iframe>` viene utilizzato per incorporare un'altra pagina web all'interno della pagina corrente.
- Può essere utilizzato per includere mappe, video di terze parti o altri contenuti esterni.

`<canvas>` (Area Disegnabile):

- `<canvas>` fornisce un'area disegnabile tramite JavaScript. Può essere utilizzato per creare grafici, animazioni o altre rappresentazioni visive dinamiche.
- È spesso utilizzato insieme a JavaScript per manipolare il contenuto visuale sulla pagina.



GLI ATTRIBUTI

Gli attributi in HTML forniscono informazioni aggiuntive su un elemento e vengono specificati all'interno del tag di apertura. Essi influenzano il comportamento o l'aspetto dell'elemento a cui sono associati. Ecco una spiegazione dettagliata e un elenco di attributi comuni:

class e id:

- Descrizione: Sono utilizzati per applicare stili CSS o identificare elementi tramite JavaScript.
- Dove vengono usati: Su qualsiasi elemento HTML.

```
html

<div class="classe-css" id="identificatore-elemento">Contenuto</div>
```

src (Source):

- Descrizione: Indica la fonte del contenuto, come il percorso di un'immagine o di un file multimediale.
- Dove viene usato: Sui tag , <audio>, <video>, <script>, <iframe>, ecc.

```
html


```

alt (Alternative Text):

- Descrizione: Fornisce testo alternativo per elementi multimediali e contribuisce all'accessibilità.
- Dove viene usato: Principalmente su , ma può essere usato anche su <area> e <input>.

```
html


```

href (Hypertext Reference):

- Descrizione: Specifica il destino di un link ipertestuale.
- Dove viene usato: Sui tag <a>, <link>, <area>, ecc.

```
html

<a href="https://www.esempio.com">Link a Esempio</a>
```

PROGRAMMA



Introduzione HTML



Iniziare un progetto HTML



I tag base e gli attributi



Introduzione CSS



I selettori e la sintassi



Modificare gli elementi di una pagina
usando CSS

1 INTRODUZIONE CSS

CSS, o Cascading Style Sheets, è un linguaggio di stile utilizzato per definire l'aspetto e la presentazione di documenti HTML nelle pagine web. Mentre HTML fornisce la struttura di base del contenuto di una pagina, CSS aggiunge il livello di design e formattazione, consentendo di creare layout accattivanti e interfacce utente gradevoli.

Immagina HTML come la struttura di una casa, con le pareti e le porte. Ora, immagina CSS come l'architetto che aggiunge il tocco magico: il colore delle pareti, il tipo di pavimento, e magari qualche quadro cool appeso al muro.

Perché usiamo CSS? CSS ci consente di rendere le cose belle e organizzate. Immagina di vestire il tuo sito con i vestiti più trendy del mondo del web!

E non dimentichiamoci della flessibilità! CSS è come il guru dello yoga per il tuo sito web. Adatta il layout e la formattazione a qualsiasi dimensione di schermo o dispositivo, rendendo il tuo sito web accessibile ovunque.

In sintesi, CSS è il tocco magico che trasforma un normale documento HTML in un'opera d'arte digitale.

PROGRAMMA



Introduzione HTML



Iniziare un progetto HTML



I tag base e gli attributi



Introduzione CSS



I selettori e la sintassi



Modificare gli elementi di una pagina
usando CSS

2 LA SINTASSI CSS

La sintassi css è molto semplice ed è fatta in questo modo:

```
selettore {  
  proprietà: valore  
}
```

Come potete vedere dall'immagine, la sintassi delle nostre regole è composta da:

- Selettore: l'elemento html a cui vogliamo cambiare lo stile;
- Due parentesi graffe che conterranno le regole che vogliamo applicare a quel selettore;
- Una proprietà o una sequenza di proprietà che non sono altro che le regole che vogliamo applicare;
- Due punti seguiti dal valore che vogliamo dare ad una specifica regola;

2 LA SINTASSI CSS

Ogni regola CSS avrà effetto solo al selettore inserito. Non sarà necessario mettere più selettori uguali con regole differenti per modificare un selettore, ma possiamo mettere un elenco di regole sul medesimo selettore per modificarlo su più aspetti come nell' esempio sotto. Qui andiamo a modificare il colore del testo, il colore di sfondo e la grandezza del testo di un paragrafo <p>:

```
p {  
    color: blue;  
    background-color: yellow;  
    font-size: 20px;  
}
```

- p: è il nostro selettore
- color, background-color, font-size: sono le proprietà che vogliamo applicare
- blue, yellow, 20px: sono i valori dati ad ogni singola proprietà

color per il testo, background-color per lo sfondo, font-size per la dimensione del testo.

2

I SELETTORI

Ecco un esempio di html con classi e ID:

```
<> index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9
10 <body>
11     <section id="content-wrapper">
12         <div class="title">
13             <h1>
14                 Il mio titolo
15             </h1>
16         </div>
17         <div class="paragraph">
18             <p>
19                 Il mio paragrafo
20             </p>
21         </div>
22     </section>
23 </body>
24
25 </html>
```

Quali sono i selettori che possiamo utilizzare? sicuramente i tag, ma qui abbiamo anche altri selettori che possono essere usati:

- gli ID
- le classi

Gli ID sono degli identificatori univoci, quindi se rispettiamo gli standard, saremo certi che “content-wrapper” sia uno soltanto in tutta la pagina e solo per specificare un elemento speciale mentre le classi, indicano gruppi di elementi che condividono caratteristiche simili.

Vediamo nelle prossime slide come usare i selettori e parleremo anche di priorità di regole.

2 I SELETTORI

```
<> index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9
10 <body>
11     <section id="content-wrapper">
12         <div class="title">
13             <h1>
14                 Il mio titolo
15             </h1>
16         </div>
17         <div class="paragraph">
18             <p>
19                 Il mio paragrafo
20             </p>
21         </div>
22     </section>
23 </body>
24
25 </html>
```

come abbiamo detto possiamo scegliere tra 3 tipi di selettori:

tag:

```
tag {
    attributo: valore;
}
```

ID:

```
#id {
    attributo: valore;
}
```

Class:

```
.class {
    attributo: valore;
}
```

2

I SELETTORI

```
<> index.html > html
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5      <meta charset="UTF-8">
6      <meta name="viewport" content="width=device-width, initial-scale=1.0">
7      <title>Document</title>
8  </head>
9
10 <body>
11     <section id="content-wrapper">
12         <div class="title">
13             <h1>
14                 Il mio titolo
15             </h1>
16         </div>
17         <div class="paragraph">
18             <p>
19                 Il mio paragrafo
20             </p>
21         </div>
22     </section>
23 </body>
24
25 </html>
```

Possiamo subito notare che per usare il selettore TAG ci basterà scrivere il nome del tag senza aggiungere altro.

Se vogliamo invece scegliere la CLASS dovremo iniziare a scrivere il suo nome con il punto “.”, nel nostro esempio: “.title” sarà il selettore per tutti gli elementi html che hanno come attributo ‘class=“title””

Se vogliamo invece usare l’ ID dovremmo iniziare a scrivere il suo nome con il cancelletto iniziale “#”, nel nostro esempio: “#content-wrapper” sarà il selettore per quell’elemento che ha come attributo ‘id=“content-wrapper”” ed essendo un ID saremo certi che sarà l’unico in tutta la pagina.



I SELETTORI

Bene. Abbiamo capito come selezionare un elemento html, come aggiungergli una proprietà e come dargli un valore. Ma quindi cosa scegliamo?

Facciamo un passo indietro:

Quando creiamo la nostra pagina html, dobbiamo innanzitutto progettarela. Progettarla vuol dire anche stabilire quali sono gli elementi comuni e quali quelli speciali.

Gli elementi comuni, sono quegli elementi che si ripetono all'interno di tutta la nostra web-app. Ad esempio, se la nostra web app ha un titolo e un sottotitolo su ogni pagina e abbiamo deciso o ci hanno detto che il titolo deve essere rosso e il sottotitolo blu in ogni pagina, dobbiamo scegliere class come attributo per quegli elementi perché saranno identici ovunque verranno utilizzati.

Gli elementi speciali invece sono quegli elementi che hanno una particolarità differente rispetto agli altri. Riprendendo l'esempio di prima, se i nostri titoli e sottotitoli devono essere tutti uguali, useremo la classe, ma, se uno di questi titoli ha la particolarità di avere ad esempio un colore diverso dagli altri e sarà l'unico in tutta l'app ad essere diverso, avrà senso usare l'ID.

Sappiate inoltre che all'interno di un tag html, possiamo avere sia classi che Id, inoltre, possiamo anche scegliere di dare più classi allo stesso elemento:

```
'<div id="elementoSpeciale" class="classeUno classeDue classeTre"> contenuto..... </div>'
```

Sulla base di queste informazioni, scegliete bene quali attributi dare agli elementi html che scrivete, perché farà la differenza sulla qualità del vostro css oltre che a quella del vostro html.

PROGRAMMA



Introduzione HTML



Iniziare un progetto HTML



I tag base e gli attributi



Introduzione CSS



I selettori e la sintassi



Modificare gli elementi di una pagina usando CSS



ESEMPI SELETTORI

```
<style>
  /*
    SUPPONIAMO DI AVERE 4 PARAGRAFI E VOLER DARE AD OGNUNO DI LORO
    DELLE PARATTERISTICHE CONDIVISE E ALTRE UNICHE

    VOGLIAMO CHE TUTTI I TESTI SIANO ROSSI
    CHE SOLO UNO ABBIA LO SFONDO
    CHE SOLO UNO ABBIA UN FONT DI 20PX
    CHE SOLO UNO ABBIA IL COLORE DEL TESTO FUCSIA:
    CHE SOLO UNO ABBIA TUTTI QUESTI ASPETTI AD ECCEZIONE DEL FUCSIA
  */
  p {
    color: red;
  }

  /* AVRA' EFFETTO SOLO AGLI ELEMENTI CON QUESTA CLASSE */
  .backgroundYellow {
    background-color: yellow;
  }

  /* AVRA' EFFETTO SOLO AGLI ELEMENTI CON QUESTA CLASSE */
  .textFont-20 {
    font-size: 20px;
  }

  /* AVRA' EFFETTO SOLO AGLI ELEMENTI CON QUESTA ID */
  #specialColor {
    color: fuchsia;
  }
</style>

<body>
  <p>COLORE ROSSO</p>
  <p class="backgroundYellow">COLORE ROSSO E SFONDO GIALLO</p>
  <p class="textFont-20">COLORE ROSSO E FONT 20PX</p>
  <p id="specialColor">COLORE FUCSIA</p>
  <p class="backgroundYellow textFont-20">
    COLORE ROSSO SFONDO GIALLO FONT 20PX
  </p>
</body>

</html>
```

COLORE ROSSO

COLORE ROSSO E SFONDO GIALLO

COLORE ROSSO E FONT 20PX

COLORE FUCSIA

COLORE ROSSO SFONDO GIALLO FONT 20PX



GERARCHIA DEI SELETTORI

Selettore di Tipo (Tag):

- Gli stili definiti per un selettore di tipo (es. `div { color: red; }`) si applicano a tutti gli elementi di quel tipo nella pagina.

Selettore di Classe:

- Gli stili definiti per un selettore di classe (es. `.mia-classe { font-size: 16px; }`) si applicano a tutti gli elementi che hanno quella classe.

Selettore di ID:

- Gli stili definiti per un selettore di ID (es. `#mio-id { background-color: #eee; }`) si applicano all'elemento con quell'ID specifico.

Stile Inline:

- Gli stili inline (direttamente nell'elemento HTML) hanno la massima priorità e sovrascrivono gli stili definiti altrove.
- Esempio: `<div style="color: green;">Testo verde</div>`.

Classi Concatenate e ID e Classi Concatenate:

- Quando più classi sono concatenate in un selettore (es. `.classe1.classe2`), lo stile si applica agli elementi che hanno entrambe le classi.
- Lo stesso vale per la concatenazione di ID e classi (es. `#mio-id.mia-classe`).

Classi Concatenate e ID e Classi Concatenate:

- Quando più classi sono concatenate in un selettore (es. `.classe1.classe2`), lo stile si applica agli elementi che hanno entrambe le classi.
- Lo stesso vale per la concatenazione di ID e classi (es. `#mio-id.mia-classe`).

Priorità nella Sovrascrittura:

- In generale, stili più specifici (ad esempio quelli definiti con ID) hanno la priorità su quelli meno specifici (ad esempio quelli definiti con tag o classe).
- La specificità è valutata in base al numero di ID, classi e elementi in un selettore.

Ordine di Applicazione:

- Gli stili vengono applicati nell'ordine in cui sono definiti nel CSS.
- Gli stili definiti più tardi nel file CSS sovrascrivono quelli definiti precedentemente.

PROGRAMMA



Introduzione HTML



Iniziare un progetto HTML



I tag base e gli attributi



Introduzione CSS



I selettori e la sintassi



Modificare gli elementi di una pagina
usando CSS