



# ~~JavaScript~~ HTML5/CSS3

## Introduzione a HTML

Giovanni Iovino: [giovanni.iovino@intecs.it](mailto:giovanni.iovino@intecs.it)



## Contenuto:

Introduzione a HTML e CSS per la creazione di interfacce grafiche.



# HTML

## breve storia

---

- Creato agli inizi degli anni 90 al CERN assieme al protocollo HTTP usato per lo scambio di documenti scritti in HTML
- Rilasciato pubblicamente nel '93 con la collaborazione dell'IETF
- A seguito del suo successo nasce il W3C (World Wide Web Consortium) che da allora si occupa di portare avanti la definizione delle nuove versioni di HTML fino alla 4
- Il W3C si occupa di molti standard relativi al WWW:
  - HTML, CSS, PNG, HTTP, DOM, XML, WebRTC, URI, URL
  - ... oltre 50 in totale
- Nel 2004 Apple, Google, Microsoft, Mozilla ed altri formano il WHATWG (Web Hypertext Application Technology Working Group) perchè ritengono che gli sviluppi di HTML procedano a rilento.
- Nel 2008 il WHATWG rilascia il primo draft di HTML5 che viene poi adottato dal W3C (2014)
- Con HTML5 ci riferiamo genericamente all'ultima versione dello standard, tuttavia non esistono rilasci stabili, HTML è un **living standard**, con update continui che vengono adottati dai browser nel tempo
- Dal 2019 HTML e DOM sono sotto il controllo del solo WHATWG

HTML living standard: <https://html.spec.whatwg.org/multipage/>



# HTML

what is and what is not

- HTML (Hyper Text Markup Language) è un linguaggio di markup per ipertesti, ovvero un set di regole che descrivono la struttura e la presentazione di un testo e relazioni con altri documenti (ipertesto)
- **HTML NON è un linguaggio di programmazione**
- HTML è la base del WWW



## Sintassi HTML - elementi e tag

— — —

- La sintassi di HTML utilizza i tag per descrivere una serie di elementi che costituiscono la struttura del documento
- Un elemento è costituito da:
  - un tag di apertura
  - seguito opzionalmente da un contenuto
  - da un tag di chiusura

```
<tagname>Content ...</tagname>
```

```
<tagname>  
    <othertagname>  
        Content ...  
    </othertagname>  
</tagname>
```



# Sintassi HTML - elementi e tag

— — —

- Alcuni elementi che normalmente prevedono un contenuto, possono essere scritti con un tag “*auto-chiudente*” se non hanno un contenuto
- I nomi dei tag non sono case-sensitive, ma è buona norma usare solo valori con lettere minuscole
- Gli elementi possono avere degli attributi, che offrono informazioni aggiuntive sull’elemento stesso.

```
<tagName>Content ...</tagName>

<tagName>
  <othertagName>Content ...</othertagName>
</tagName>

<tagName />

<tagName attr1="value1" attr2="value2">Content
...</tagName>

<tagName attr1="value1" attr2="value2" />
```



# Sintassi HTML - elementi e tag

— — —

- Gli attributi vanno inseriti all'interno del tag iniziale di un elemento.
- Alcuni attributi sono globali, possono essere applicati a qualsiasi elemento, mentre altri sono specifici per alcuni elementi.
- Alcuni attributi possono essere usati per agganciare degli eventi!

Lista degli elementi HTML:

<https://www.w3schools.com/tags/default.asp>

Lista degli attributi degli elementi html:

[https://www.w3schools.com/tags/ref\\_attributes.asp](https://www.w3schools.com/tags/ref_attributes.asp)

```
<tagname>Content ...</tagname>

<tagname>
    <othertagname>Content ...</othertagname>
</tagname>

<autoclosingtag />

<tagname attr1="value1" attr2="value2">Content
...</tagname>

<autoclosingtag attr1="value1" attr2="value2" />
```



# Struttura di una pagina HTML

---

La struttura di una pagina HTML è composta da tre sezioni principali:

- **Document type:**
  - Fino alla versione 4 di HTML era necessario indicare la versione specifica affinché i browser riuscissero a visualizzare il documento senza errori
  - Per HTML5 utilizziamo sempre `<!DOCTYPE html>`
- **Tag `<html>`:** Utilizzato per delimitare la struttura del file HTML. L'intero contenuto del documento è racchiuso nel tag.
- **Tag `<head>`:** Deve essere il primo elemento all'interno del tag `html`.
- **Tag `<body>`:** Deve essere il secondo elemento all'interno del tag `html`.

```
<!DOCTYPE html>
<html>
    <head></head>
    <body></body>
</html>
```





# <head>

— — —

Contiene i metadati del documento (es. titolo, parole chiave per indicizzazione, script da eseguire etc.). Seguono alcuni elementi che possono essere inseriti nel head.

- **Title:** Il titolo del documento. Obbligatorio.
- **Links:** Questi elementi permettono di collegare risorse esterne da utilizzare per la corretta visualizzazione del documento HTML (es. fogli di stile CSS, script JavaScript, fonts etc.). Utilizza l'attributo **rel** per specificare la relazione della risorsa esterna con il documento, e gli attributi **href** o **imgsrcset** per indicare l'URL della risorsa esterna.
- **Base:** Elemento che può essere utilizzato come indirizzo base da utilizzare in tutti i link del documento che hanno un url relativo e non assoluto
- **Script:** utilizzato per includere script esterni o scrivere il codice direttamente nel documento html. Il tag <script> può essere utilizzato anche nel body. Il codice viene eseguito in base a dove è inserito l'elemento!

```
<!DOCTYPE html>
<html>
  <head>
    <title>Titolo della pagina</title>
    <link rel="stylesheet" href="styles.css">
    <link rel="icon" href="images/my-icon.png">
    <script type="module" src="modulo.js"></script>
    <script src="script.js"></script>
    <script defer src="script_2.js"></script>
    <script async src="script_3.js"></script>
    <script type="text/javascript">
      var a = 5;
      a++;
    </script>
    <base href="https://www.example.com/"
target="_blank">
  </head>
  <body></body>
</html>
```



# <body>

— — —

Rappresenta il corpo del documento. Può esserci un solo body in un documento HTML. Alcuni esempi di elementi che si possono utilizzare nel body:

- **<h1> ... <h6>**: intestazioni
- **<p>**: paragrafi
- **<b>**: testo in grassetto
- **<br>**: break-line
- **<em>**: testo enfaticizzato
- **<i>**: testo italico
- **<img>**: immagini
- **<a>**: link

```
<!DOCTYPE html>
<html>
  <head><title>Test1</title></head>
  <body>
    <h1>Titolo 1</h1>
    <h2>Titolo 2</h2>
    <h3>Titolo 3</h3>
    <h4>Titolo 4</h4>
    <h5>Titolo 5</h5>
    <h6>Titolo 6</h6>
    <a href="http://www.google.com" target="_blank">Visita google se non
sai le cose</a>
    <a href="2.html">Questo è un link relativo</a>
    <a href="#id1">Questo è un link interno</a>
    <p> Lorem <b>ipsum</b> dolor sit amet,<br> consectetur adipiscing
elit.<em> Integer nec sem lacinia</em>, bibendum est sit amet,<i> ullamcorper
nunc</i>. Suspendisse ipsum felis, pellentesque vel erat vel,...</p>
    <p> altro paragrafo! con un'immagine </p>
    <br><br><br><br><br><br><br><br>
    <div id="id1"> Questo div è un pò più in basso e ha un id associato
</div>
  </body>
</html>
```



# <body>

— — —

- **<li>**: elemento di una lista
- **<ol>**: lista ordinata
- **<ul>**: lista non ordinata
- **<div>**: container generico
- **<span>**: elemento nullo, container in-line
- **<table>**: tabelle
- **<form>**: form
- **<input>**: vari input (es .caselle di testo, select, options, checkbox)
- **<textarea>**: caselle di testo multi-linea
- **<button>**: pulsanti

Lista degli elementi HTML:

<https://www.w3schools.com/tags/default.asp>

```
<!DOCTYPE html>
<html>
  <head>
    <title>Test3</title>
  </head>
  <body>
    <ol>
      <li>Primo punto</li>
      <li>Secondo punto</li>
      <li><a href="https://www.google.com"
target="_blank">Terzo punto</a></li>
    </ol>
    <ul>
      <li>Primo punto</li>
      <li>Secondo punto</li>
      <li><a href="https://www.google.com"
target="_blank">Terzo punto</a></li>
    </ul>
    <div>Contenuto div 1</div>
    <div>Contenuto div 2</div>
    <span>Contenuto span 1</span>
    <span>Contenuto span 2</span>
  </body>
</html>
```



# Esempio tabella

— — —

Col1	Col2	Col3
Pippo	cane	Disney
Paperino	papero	Disney

```
<!DOCTYPE html>
<html>
  <head>
    <title>Test3</title>
  </head>
  <body>
    <table>
      <tr>
        <th>Col1</th>
        <th>Col2</th>
        <th>Col3</th>
      </tr>
      <tr>
        <td>Pippo</td>
        <td>cane</td>
        <td>Disney</td>
      </tr>
      <tr>
        <td>Paperino</td>
        <td>papero</td>
        <td>Disney</td>
      </tr>
    </table>
  </body>
</html>
```



## Esempio form

— — —

# Form

Questa è un esempio di form

Nome:

Cognome:

☒ Dichiaro di accettare le condizioni

```
<!DOCTYPE html>
<html>
  <head>
    <title>Form</title>
  </head>
  <body>
    <h1>Form</h1>
    <p>Questa è un esempio di form</p>
    <form action="form.html" method="get">
      <label for="fname">Nome:</label>
      <input type="text" id="fname" name="fname"><br>
      <label for="lname">Cognome:</label>
      <input type="text" id="lname" name="lname"><br><br>
      <input type="checkbox" checked="true" name="terms"
id="termini" value="terms">
      <label for="termini">Dichiaro di accettare le
condizioni</label><br><br>
      <input type="submit" value="Submit">
      <input type="reset" value="Cancel">
    </form>
  </body>
</html>
```



# Cosa succede quando il browser carica una pagina HTML?

— — —

- La pagina è processata in modo progressivo, ovvero gli elementi vengono analizzati e visualizzati progressivamente
- Il browser traduce il documento HTML in un oggetto Document (DOM).
- Il DOM permette di interagire con gli elementi della pagina
- In Javascript possiamo accedere al DOM dalla proprietà **document** della variabile globale **window**

```
<!DOCTYPE html>
<html>
  <head>
    <title>Script Example</title>
    <script>
      console.clear();
      let a=10;
      const el = window.document.getElementById("id1");
      console.log("el", el);
    </script>
    <script defer>
      const el1 = window.document.getElementById("id1");
      console.log("el1", el1);
      console.log("a", a)
    </script>
  </head>
  <body>
    <script>
      const el2 = window.document.getElementById("id1");
      console.log("el2", el2);
    </script>
    <h1>Esempio Script</h1>
    <div id="id1">10</div>
    <script>
      const el3 = window.document.getElementById("id1");
      console.log("el3", el3);
      console.log("el3 inner", el3.innerHTML);
    </script>
  </body>
</html>
```



# Events attributes & DOM access

```
<!DOCTYPE html>
<html>
  <head>
    <title>Evento</title>
  </head>
  <body>
    <h1>Esempio di evento</h1>
    <p>Questa è un esempio di pagina con script che
usano gli eventi degli elementi html</p>
    <div><b>Contatore:</b></div>
    <div id="counter">0</div>
    <button onclick="reset();">reset</button>
    <button id="button1">increment</button>
    <script>
      let c = 0;
      const counter =
window.document.getElementById("counter");
      const button1 =
window.document.getElementById("button1");
```

```
setInterval(()=>{
  c++;
  if(counter){
    counter.innerHTML=c;
  }
}, 1000);
button1.addEventListener("click", ()=>{
  c+=100;
  if(counter){
    counter.innerHTML=c;
  }
});

function reset(){
  c = 0;
  if(counter){
    counter.innerHTML=c;
  }
}

</script>
</body>
</html>
```

JS

# CSS

— — —

Cascade Style Sheet

**Separazione contenuto/formattazione:** Utilizzato per definire le regole di formattazione di una pagina html.





# CSS

breve storia

---

- CSS1 - Nato nel 1996 per opera del W3C come soluzione all'esigenza di poter cambiare lo stile grafico delle pagine web. Fino ad allora i principali browser (Netscape, IE) utilizzavano una sintassi non compliant allo standard HTML per modificare l'aspetto delle pagine web.
- 1998 - CSS2 - Superset di CSS1, aggiunte nuove funzionalità come posizionamenti assoluti e relativi
- 2004-2011- CSS2.1 (maggiore supporto ai device palmari) Fix di errori e rimozione di regole non supportate pienamente dai browser
- CSS3 - Regole che estendono CSS2 . Il primo draft è del 1999, a differenza di CSS1 e CSS2/2.1 non è un unico standard ma un insieme di moduli le cui specifiche sono portate avanti in parallelo e possono avere un livello di stabilità differente



# CSS3

moduli



- `css3-background`
- `css3-box`
- `css-cascade-3`
- `css3-color`
- `css3-content`
- `css-fonts-3`
- `css3-gcpm`
- `css3-layout`
- `css3-mediaqueries`
- `mediaqueries-4`
- `css3-multicol`
- `css3-page`
- `selectors-3`
- `selectors-4`
- `css3-ui`



# CSS syntax

— — —

La sintassi di CSS è relativamente semplice, un documento CSS definisce un insieme di regole di stile dove ogni regola è costituita da:

- Uno o più selettori utilizzati per identificare gli elementi del documento HTML a cui si applicano le regole di stile definite nel blocco di dichiarazione
- Un blocco di dichiarazione racchiuso tra parentesi graffe che contiene un set di dichiarazioni. Ciascuna dichiarazione è espressa tramite una proprietà e il suo valore:
  - Le proprietà sono definite dallo standard
  - I valori dipendono dalle proprietà (es. numeri, stringhe, etc..)

```
/* This is a comment */  
selector{  
    property: value;  
    property: value;  
    ..  
    ..  
    property: value;  
}
```



# CSS selettori

---

Selezionano gli elementi a cui applicare lo stile. Alcuni esempi

- `.<class>`: Seleziona tutti gli elementi che hanno l'attributo class uguale a `<class>`
- `#<id>`: Seleziona l'elemento che ha l'attributo id uguale a `<class>`
- `*`: seleziona tutti gli elementi del documento
- `<element>[,<elementN>]`: seleziona tutti gli elementi con nome del tag uguale a uno dei selettori separati da virgola
- `<element> <element2>`: seleziona tutti gli elementi di tipo `<element2>` all'interno di un tag di tipo `<element>`
- `<element> > <element2>`: seleziona tutti gli elementi figli di tipo `<element2>` che hanno il padre di tipo `<element>`

Lista dei possibili selettori:

[https://www.w3schools.com/cssref/css\\_selectors.asp](https://www.w3schools.com/cssref/css_selectors.asp)

```
.my-class{...}
```

```
#my-id-1{....}
```

```
*{....}
```

```
div, span{...}
```

```
div p {....}
```

```
div>p{....}
```



# CSS blocchi di dichiarazione e proprietà

---

- I blocchi di dichiarazione definiscono le proprietà di stile da applicare agli elementi selezionati dal corrispondente selettore.
- Le proprietà sono definite dallo standard, alcune sono applicabili a tutti gli elementi altre sono specifiche di alcuni elementi
- Alcune proprietà sono ereditabili mentre altre no
- I valori delle proprietà possono essere di diversi tipi
- Le proprietà hanno un valore di default!

Alcuni esempi:

- background-color: imposta il colore di sfondo di un elemento
- color: imposta il colore del testo di un elemento
- display: Specifica come deve essere visualizzato un elemento

Riferimenti: <https://www.w3schools.com/cssref/default.asp> oppure  
<https://www.w3.org/TR/CSS22/propidx.html>

```
.my-class{
  color: #ff0000; /*red*/
}

#my-id-1{
  font-size: 22px;
  font-weight: bold;
  font-family: "Verdana", "Roboto", serif
  width: 150px;
  color: rgb(0,0,0);
}

*{
  background-color: yellow;
}
```



# CSS unità

---

Per esprimere la grandezza di un elemento in CSS si possono usare diversi tipi di unità di misura, alcuni relativi altri assoluti. Le unità di misura si utilizzano per le proprietà che esprimono una grandezza (es. width, height, border-size, font-size etc.)

Esempi di unità di misura:

- Assolute:
  - cm: centimetri
  - mm: millimetri
  - px: pixel
  - in: inches
- Relative:
  - em: relativa alla grandezza del font dell'elemento
  - rem: relativa alla grandezza del font dell'elemento root
  - vw: relativa alla larghezza della finestra del browser
  - vh: relativa all'altezza della finestra del browser
  - %: relativa alla grandezza dell'elemento parent

```
div{  
    width: 300px; /* tutti i div */  
}  
  
div>p{  
    width: 50%; /* 50% del parent =>  
150px */  
}
```



# CSS - inserimento in un documento HTML

---

Ci sono più modi per aggiungere del codice CSS ad un documento HTML:

- Caricando un file esterno tramite un elemento **link** con attributo **rel** di tipo **style**
- Scrivendo il CSS direttamente all'interno del documento HTML:
  - In un elemento di tipo **style**
  - all'interno dell'**attributo style** di uno specifico elemento (**inline-style**)

```
<head>
  <link rel="stylesheet" href="styles.css">
  <style>
    h1{
      color: #000000;
    }
  </style>
</head>
<body>
  <h1>Un titolo</h1>
  <h1 style="color:#ffffff;
background-color:#000000">Un altro titolo</h1>
</body>
```



# CSS precedenze

— — —

## Attenzione!

- CSS ha delle regole di precedenza nell'applicare le regole di stile che possono causare la sovrascrittura di una regola nel caso un elemento sia selezionabile da più selettori:
  - Regole che dipendono dalla posizione delle regola nel documento di stile (Come per HTML un documento CSS viene processato progressivamente)
  - Regole associate al tipo di selettore (**specificity**)
- Al contempo questo però ci permette di creare un meccanismo di “ereditarietà” che può risultare molto utile.

CSS Selector	Precedenza
Stili ereditati	+ (Bassa)
*	++
element	+++
attribute	++++
class	+++++
id	++++++
inline-style	+++++++
Selettori combinati	Dipende - dalla combinazione dei selettori





# CSS esempio precedenza

```
<body>
  <h1>Test stile CSS</h1>

  <p>Questo è un paragrafo</p>

  <a href="index.html">Torna alla
home!</a>

  <div> <p>Quest è un paragrafo in un
div</p></div>

  <div><p class="pGray">Quest è un
paragrafo in un div che ha una
classe</p></div>

  <div> <span><p>Quest è un paragrafo in
un div in un span</p></span></div>
</body>
```

```
h1 {
  font-size: 40px;
}
a, a:visited{
  color: #ff00ff;
  text-decoration: none;
}
a:hover{
  color: #00ff00;
}
.pGray{
  color: #cccccc;
}
div p{
  color: #0000ff;
}
div>p{
  color: #ff0000;
}
```

## Test stile CSS

Questo è un paragrafo

Torna alla home!

Quest è un paragrafo in un div

Quest è un paragrafo in un div che ha una classe

Quest è un paragrafo in un div in un span



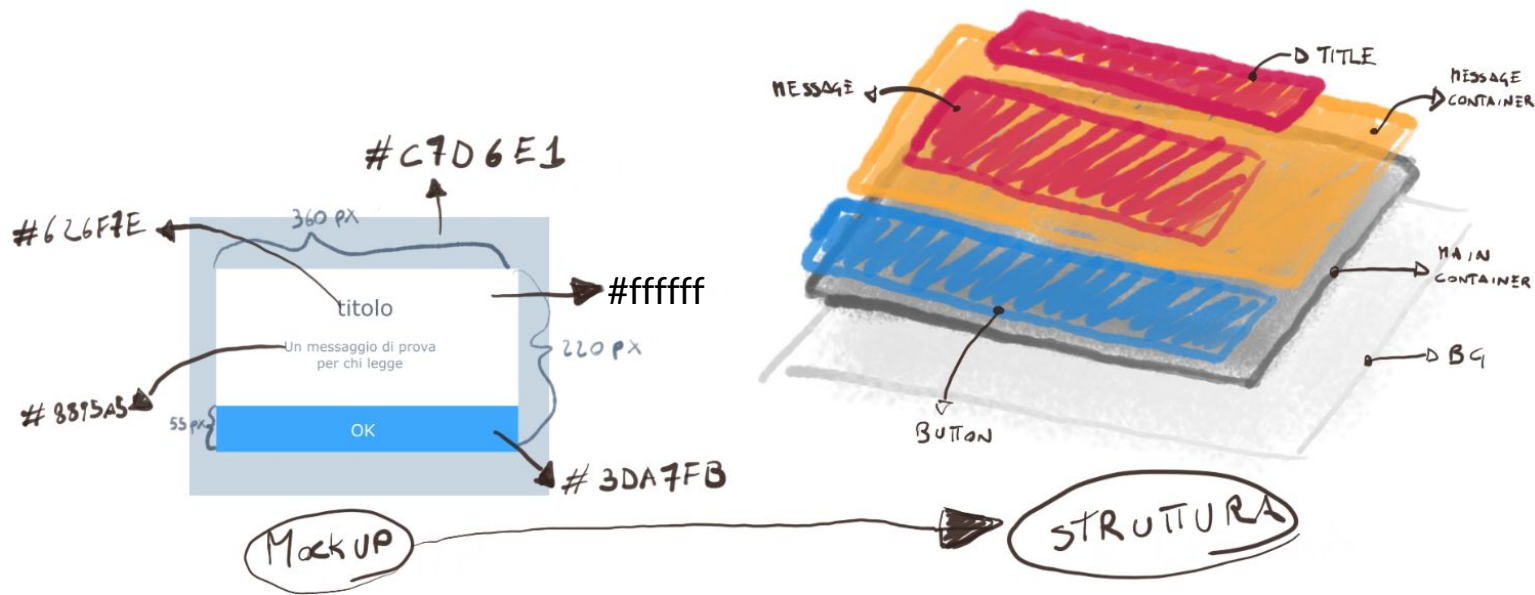
# CSS e Javascript

— — —

- Il DOM permette di accedere anche agli stili applicati ad un elemento!
- I nomi delle proprietà CSS in JS seguono la sintassi camelCase, per motivi di compatibilità
- Mapping proprietà CSS -> JS:  
[https://developer.mozilla.org/en-US/docs/Web/CSS/CSS\\_Properties\\_Reference](https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Properties_Reference)

```
<!DOCTYPE html>
<html>
  <head>
    <title>Script Example</title>
    <style>
      #id1{
        color: #ffffff;
        background-color: #000000;
      }
    </style>
  </head>
  <body>
    <h1>Esempio Script</h1>
    <div id="id1">10</div>
    <script>
      const el = window.document.getElementById("id1");
      el.style.backgroundColor = "#cccccc";
    </script>
  </body>
</html>
```

# Come approcciare la progettazione di un componente HTML?





# Esempio

---

```
<body>
  <div class="main-container">
    <div class="message-container">
      <div class="titolo">titolo</div>
      <div class="message">
        Un messaggio di prova per chi
        legge
      </div>
    </div>
    <button class="blue">OK</button>
  </div>
</body>
```

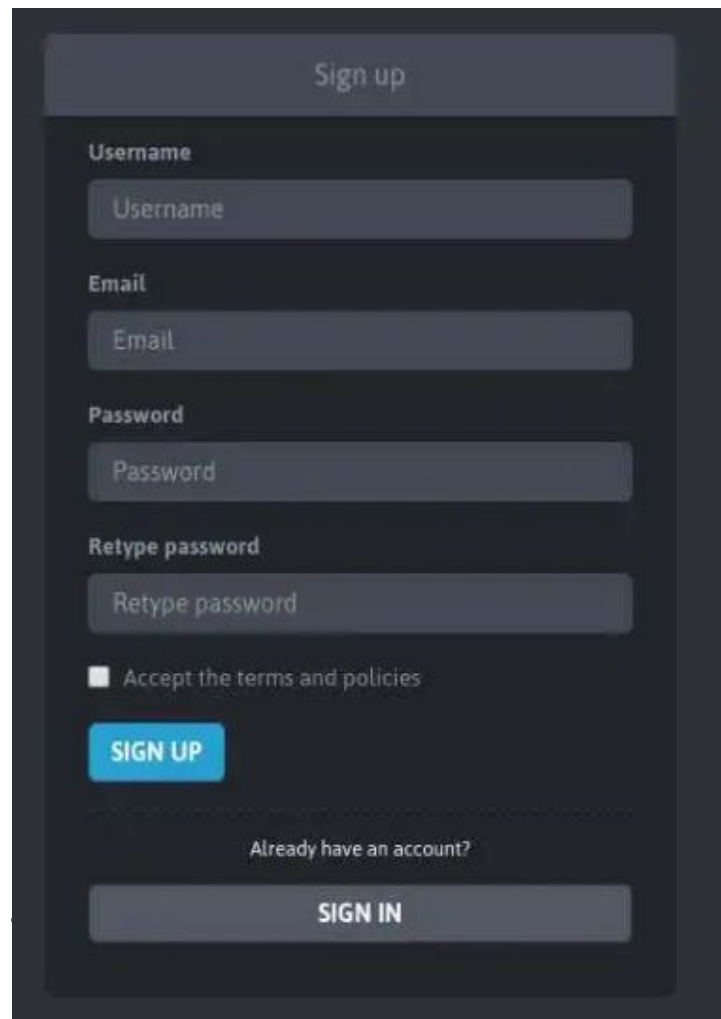
```
<style>
  body{
    background-color: #c7d6e1;
    padding: 30px;
  }
  .main-container{
    background-color: #ffffff;
    width: 360px;
    height: 220px;
    margin: 0;
  }
  .message-container{
    width: 100%;
    height: 165px;
    margin: 0;
    padding: 0;
    text-align: center;
  }
  .titolo{
    display: inline-block;
    margin-top: 20px;
    font-size: 20px;
    font-family: "Arial";
    color: #626f7e;
    width: 100%;
  }
}
```

```
.message{
  display: inline-block;
  font-size: 14px;
  font-family: Arial;
  color: #8895a5;
  text-align: center;
  width: 50%;
  padding: 10px;
  margin-top: 20px;
}
button.blue{
  height: 55px;
  width: 100%;
  border: none;
  background-color: #3da7fb;
  color: #ffffff;
}
button.blue:hover{
  background-color: #85c9ff;
  cursor: pointer;
}
button.blue:focus{
  border: none;
  outline: none;
}
button.blue:active{
  background-color: #3da7fb;
}
</style>
```

JS

# Esercizio

Provare a creare un componente di registrazione che segua il seguente mockup



A dark-themed mockup of a sign-up form. At the top, a grey bar contains the text "Sign up". Below this, the form consists of several sections: "Username" with a text input field, "Email" with a text input field, "Password" with a text input field, and "Retype password" with a text input field. Below the password fields is a checkbox labeled "Accept the terms and policies". A blue button with the text "SIGN UP" is positioned below the checkbox. At the bottom, the text "Already have an account?" is displayed above a grey button with the text "SIGN IN".

Sign up

Username

Username

Email

Email

Password

Password

Retype password

Retype password

☐ Accept the terms and policies

SIGN UP

Already have an account?

SIGN IN

JS

# Materiale della lezione

— — —

<https://plnkr.co/edit/v7GvWPy907HgWBGv>