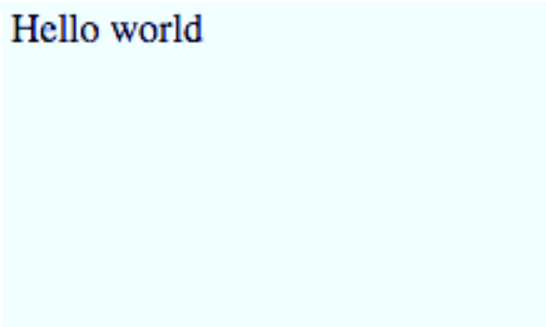


Basi dell'HTML (Hyper Text Markup Language)

Avendo capito cos'è l'HTML e come applicarlo, ora possiamo passare ad una maggiore infarinatura riguardo le basi. Iniziamo con una semplice riga di testo, e vediamo il risultato.

Hello world

In primis, guardiamo il risultato che ci si pone davanti, che sarà molto probabilmente simile a questo



Hello world

Fatto ciò, vedremo la nostra scritta in output, molto semplice e intuitivo. Ma cosa succede se iniziamo a lavorare con i tag? Per esempio se vogliamo mettere un titolo al nostro testo, possiamo usare il tag `<h1>`. Ecco un esempio

```
<h1> Titolo </h1>  
Titolo
```

Il risultato dovrebbe essere qualcosa di molto simile a questo



Prova

Prova

Abbiamo capito quindi come dare un minimo di editing al nostro testo, senza dover usare il CSS (codice di stile), ma semplicemente i tag dell'HTML.

Paragrafi ed i vari titoli in HTML

Ci sono diversi modi per gestire il testo in HTML, su questa pagina guarderemo come lavorare con titoli e paragrafi.

Paragrafi

Ci sono vari tipi di interazioni per gestire il testo nell'HTML, uno di questi è un paragrafo. Il tag per usarli è `<p>`. Questi cosiddetti paragrafi consentono di gestire il testo in blocchi, e di facilitare la sua lettura. Proviamo ad usarlo nel nostro editor.

```
<p>
```

La differenza nell'inserimento
dei paragrafi si vede proprio
nell'organizzazione

```
</p>
```

```
<p>
```

del nostro testo,
consentendo una facile
e chiara lettura

```
</p>
```

```
<p>
```

semplicemente paragrafando
il nostro testo

```
<p>
```

Questo è il risultato che dovresti vedere su schermo

La differenza nell'inserimento dei paragrafi si vede proprio nell'organizzazione
del nostro testo, consentendo una facile e chiara lettura
semplicemente paragrafando il nostro testo

Tipi di titolo

Come anticipato, ci sono vari tipi di titolo, variano dalla grandezza e servono per paragrafare una pagina web con titoli e sottotitolo, per provare qui sotto un esempio di tutti i vari spessori, con un esempio molto semplice.

- `<h1> Tipo numero 1 </h1>`
- `<h2> Tipo numero 2 </h2>`
- `<h3> Tipo numero 3 </h3>`
- `<h4> Tipo numero 4 </h4>`
- `<h5> Tipo numero 5 </h5>`
- `<h6> Tipo numero 6 </h6>`

Copiato ed incollato nel nostro editor, questi tag dovrebbero dare un output simile al seguente

Tipo numero 1

Tipo numero 2

Tipo numero 3

Tipo numero 4

Tipo numero 5

Tipo numero 6

I tag html, head e body in HTML

Sapendo che l'HTML è un linguaggio a tag, vediamo quali sono i tag principali. Tra questi ci si possono identificare il tag `html`, il tag `head`, ed il tag `body`.

Il tag html

Partendo dal più importante da comprendere, vediamo il tag **html**. Esso segnala al browser che quello che sta per leggere, è un file in HTML.

```
<html>  
Ciao  
</html>
```

Il tag **html** rappresenta anche l'inizio del suddetto file HTML, ed in più è anche il contenitore di tutti gli altri elementi in HTML, che vengono definiti con i già noti tag.

Il tag head ed il tag body

Avendo capito il funzionamento del tag **html**, possiamo passare ai due tag che suddividono il file in due parti. Il primo in ordine di scrittura, cioè il tag **head**, è il contenitore per tutti gli elementi HEAD, che sono

- <title>
- <style>
- <base>
- <link>
- <meta>
- <script>
- <noscript>

L'unico necessario in un file HTML è <title>, che appunto dà il titolo al nostro documento. Quello che per spiegarci si vede qui. (Specifichiamo che il tag **head** nel nuovo aggiornamento dell'HTML non richiede di essere scritto)

How to code - Impara a programmare in vari linguaggi già da adesso!

Il tag **body** invece contiene il 'corpo' del nostro file HTML, e può contenere tutti gli elementi HTML. Facciamo un esempio su qualcosa che già conosciamo dagli articoli precedenti, ed in più capiamo come utilizzare il tag **title**

```
<html>
<head>
<title>Titolo del documento</title>
</head>
<body>
<h2>Titolo della pagina</h2>
<p>Un paragrafo qualunque</p>
</body>
</html>
```

Ed il nostro risultato sarà

Titolo della pagina

Un paragrafo qualunque

Come andare a capo o tracciare una linea in HTML

Due dei tag più utili nelle generalità e nell'ordinamento delle nostre pagine web sono **
** ed **<hr>**. Vediamo cosa sono ed a cosa servono

Il tag **
**

Mandare a capo il testo in HTML richiede un semplice tag, ed ora vedremo insieme come utilizzarlo. La sua applicazione è molto semplice, ma i suoi utilizzi sono infiniti e sicuramente molto utili per una corretta impaginazione del nostro file HTML.

Presupponiamo un esempio pratico

```
<html>
<body>
<p>Il nostro testo non verrà mai mandato a capo senza il tag br</p>
<p> Mentre usandolo avremo un <br> perfetto <br> controllo di esso </p>
</body>
</html>
```

Ed ecco il nostro risultato

Il nostro testo non verrà mai mandato a capo senza il tag `br`

Mentre usandolo avremo un
perfetto
controllo di esso

Il tag `<hr>`

Un altro tag molto utile per organizzare il nostro file HTML è senza dubbio **hr**, che consente di tracciare una riga verticale che attraversa tutto il tag in cui è contenuta. Per fare un esempio, vediamo questo codice

```
<html>
<body>
<p>Il nostro testo non verrà mai mandato a capo senza il tag br</p>
<hr>
<p> Mentre usandolo avremo un <code>br</code> perfetto <code>br</code> controllo di esso </p>
</body>
</html>
```

Vediamo come il tag **hr** è contenuto nel tag **body**. Ciò vuol dire che la riga tracciata da esso, attraverserà tutto il **body**, cioè tutta la nostra pagina web. Ed ecco come apparirà il nostro risultato. Ovviamente ricordiamo che ogni cosa può essere provata con il proprio editor copiando ed incollando il codice qui sopra

Il nostro testo non verrà mai mandato a capo senza il tag br

Mentre usandolo avremo un
perfetto
controllo di esso

Gli attributi dei tag in HTML

I tag HTML possono contenere diversi tipi di attributi. Ricordiamo che ogni tag può averne più di uno, senza problemi. Vediamo alcuni esempi di tag con attributi.

Gli attributi. Come si usano?

Gli attributi vengono inseriti nel richiamo del tag HTML. Per fare un esempio

```
<tag attributo="esempio"> </tag>
```

Gli attributi più utilizzati sono **style**, **id** e **class**.

- **style** lo vediamo nel dettaglio sull'apposito articolo riguardante il [CSS](#).
- **id** serve ad assegnare un nominativo al tag, che ci servirà per codici come [JS](#) e [PHP](#), ma anche per impostazioni grafiche più specifiche in [CSS](#).
- **class** assegna una 'classe' appunto, ad un insieme di tag accomunati. Stesse funzioni di **id**, ma consente di agire su un gruppo di elementi.

Capito ciò, si può iniziare con un'infarinatura di CSS, che consentirà dopo una buona comprensione, di creare ottime grafiche e stili di webpage.

Si può vedere qui sotto un esempio di assegnazione di classe in HTML, che quindi apparterrà al gruppo di elementi "paragrafo".

```
<p class="paragrafo"> </p>
```

Come scrivere in CSS

Il CSS è un linguaggio usato dai web designer per dare particolari grafiche alla webpage, con noi avrete una guida su come programmare in CSS dettagliata ed articolata. Vi ritroverete a saperlo usare in men che non si dica.

Per esempio possiamo usarlo per dare determinate colorazioni ad un testo, o un particolare colore di sfondo, il posizionamento di una riga di testo, e via dicendo.

Consigliamo una buona conoscenza di [HTML](#) prima di procedere

Il CSS è strutturato in questa maniera

```
p
{
color:red;
}
```

Questo esempio qui sopra, semplicemente darà a tutti i tag <p> il colore del testo in rosso.

```
<p> Testo rosso </p>
```

Quindi per avere poi il testo rosso basterà utilizzare il tag <p>. Il risultato verrà semplicemente così Testo rosso

Si possono usare diverse combinazioni di caratteri per specificare se ciò che vogliamo modificare deve essere un gruppo di tag, un singolo elemento tramite id, o un gruppo di elementi tramite classe. Vediamo cosa li differenzia.

- **p {...}** = Questa combinazione, si riferirà ai tag.
- **.p{...}** = Questa combinazione invece, si riferirà alla classe “p”
- **#p{...}** = Stesso funzionamento del sopracitato, ma si applica solo al singolo elemento con il corrispondente id, che appunto sarà “p”.

Come usare il CSS in linea, interno e esterno

Il web design consiste nel dare ad una pagina web tramite CSS e HTML un determinato aspetto. Vediamo insieme come utilizzarlo in un file HTML, o in un file CSS. I tre metodi sono In linea, Interno e Esterno

Tra questi modi c'è una gerarchia, che è la seguente (In linea>Interno>Esterno). Tale gerarchia consente di dare delle 'precedenze', cioè se un tag ha una proprietà a livello **Interno** (*color:red;*), ed una proprietà contrastante a livello **In linea** (*color:black;*), si applicheranno le proprietà del livello **In linea** (*testo di colore nero*). Nei nostri esempi useremo questa proprietà:

```
color:red;
```


Quindi i tre modi sono

- **In linea**
- **Interno**
- **Esterno**

In linea

Il CSS in linea è una pratica alquanto sconsigliata, per via della sua scomodità nella modifica, perché essendo applicata al singolo tag, se volessimo farne diversi con le stesse caratteristiche, dovremo modificare il codice per ognuno di essi. Ovviamente l'effetto sarà lo stesso, cioè non ci saranno differenze ne sulla sintassi delle proprietà ne tantomeno sul risultato ottenuto.

La sua applicazione è la seguente

```
<p style="color:red;">Testo rosso</p>
```

L'attributo **style** consente di applicare delle proprietà in CSS.

Interno

Il CSS interno prevede l'inserimento del tag `<style>`, per poter utilizzare codice CSS appunto interno al file, per esempio

```
<html>
<head>
<style>
p
{
color:red;
}
</style>
</head>
</html>
```

Il codice visto prima, assegnato ad un insieme di tag (`<p>`).

Esterno

Il CSS esterno, come dice il nome, si trova in un file esterno rispetto al file `.html`, infatti si troverà dentro un apposito file `.css`! Un passaggio molto importante in questo caso però, è 'collegare' i nostri due file. Vediamo una potenziale struttura di entrambi i file

ESEMPIO.HTML

```
<html>
<head>
<link rel="stylesheet" type="text/css" href="esempio.css">
</head>
<body>
<p> Paragrafo </p>
</body>
</html>
```

ESEMPIO.CSS

```
p
{
color:red;
}
```

I colori in CSS, text color e background color

I colori sono specificati secondo diversi tipi di valore, che possono essere RGB, HEX, RGBA, il nome del colore e molti altri.

Nomi dei colori in CSS

L'HTML supporta la possibilità di selezionare colori inserendo la denominazione. Ne supporta ben 140 di standard. E' stato anche il nostro esempio nell'articolo precedente, che parla di [dove e come inserire il CSS](#).

```
<h1 style="color:blue;">Colore blu</h1>
<h2 style="color:green;">Colore verde</h2>
```

Il risultato sarà questo

Colore blu

Colore verde

Background color

Il **background color** è lo sfondo del nostro tag, che può essere cambiato. Vediamo come colorarlo. La proprietà è **background-color**. Ecco il suo funzionamento

```
<h1 style="color:black;background-color:red;">ProvaProvaProva</h1>  
<h2 style="color:white;background-color:darkblue;">ProvaProvaProva</h2>
```

Il risultato sarà il seguente



Valori del colore

I colori in CSS, come abbiamo detto, possono essere espressi in diverse maniere, per fare un esempio, vediamo il colore verde

```
<h1 style="background-color:#008000;"> #008000 </h1>  
<h1 style="background-color:green;"> green </h1>  
<h1 style="background-color:rgb(0, 128, 0);"> rgb(0, 128, 0) </h1>  
<h1 style="background-color:hsl(120, 100%, 25%;"> hsl(120, 100%, 25%) </h1>
```

#008000

green

rgb(0, 128, 0)

hsl(120, 100%, 25%)

Bordi in CSS

In CSS i bordi consentono di stilizzare i tag con dei bordi, di cui si può definire tipo, spessore e colore. Iniziamo subito a vederli, allora.

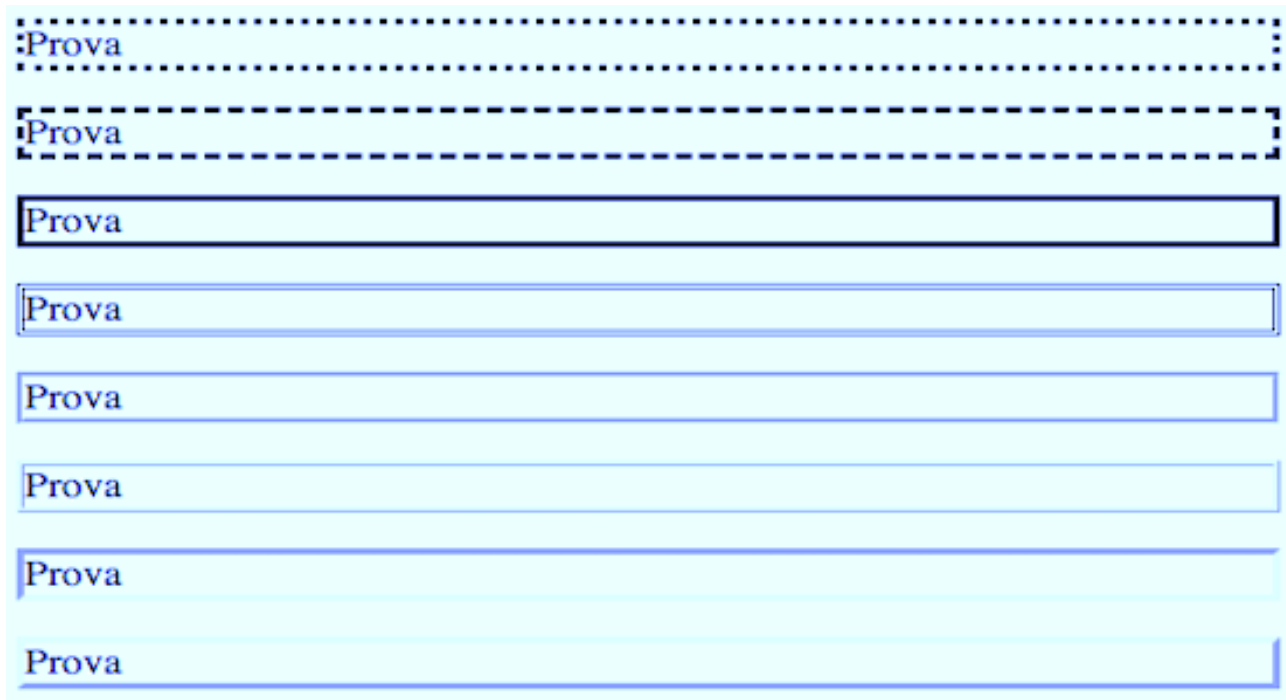
Border-style

Lo stile è molto importante da capire, perché senza definirlo, nessun'altra proprietà riguardo i bordi avrà effetto.

La proprietà che utilizzeremo sarà **border-style**.

- `<p style="border-style:dotted;"> Prova </p>`
- `<p style="border-style:dashed;"> Prova </p>`
- `<p style="border-style:solid;"> Prova </p>`
- `<p style="border-style:double;"> Prova </p>`
- `<p style="border-style:groove;"> Prova </p>`
- `<p style="border-style:ridge;"> Prova </p>`
- `<p style="border-style:inset;"> Prova </p>`
- `<p style="border-style:outset;"> Prova </p>`

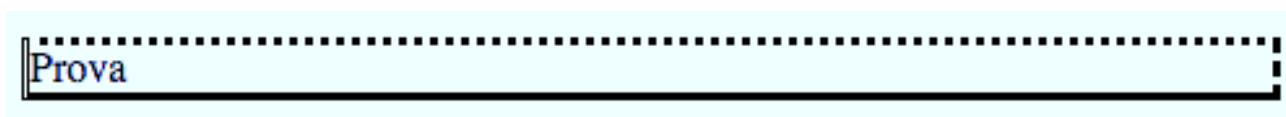
Il nostro risultato sarà il seguente



Specifichiamo anche che si possono dare più valori alla proprietà , per esempio

```
<p style="border-style: dotted dashed solid double;"> Prova </p>
```

L'ordine è **sopra, destra, sotto, sinistra**

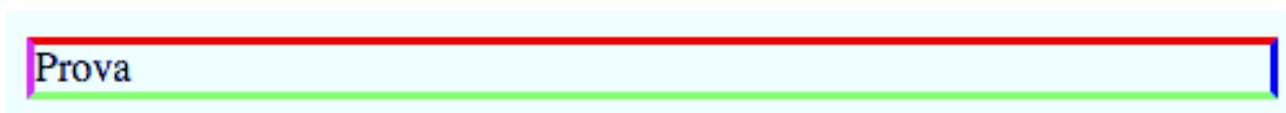


Colorazione dei bordi

I colori come abbiamo detto nell'articolo precedente ([se te lo fossi perso](#)), sono definiti in diverse maniere (RGB, HEX, ecc..).

Come lo stile, la proprietà **border-color** può avere fino a 4 valori, e come abbiamo detto, necessita la proprietà **border-style** per funzionare.

```
<p style="border-style:solid; border-color: red blue #90f542 rgb(233, 38, 240);"> Prova </p>
```



Spessore ed arrotondamento dei bordi e proprietà individuali in CSS

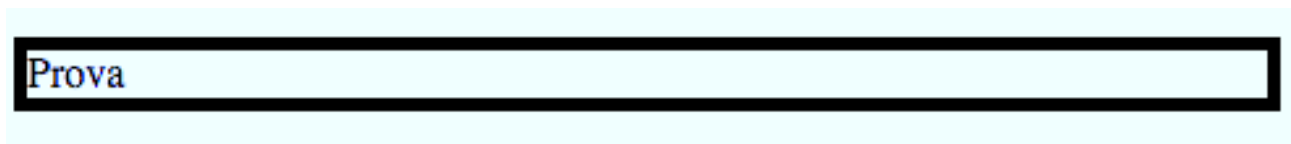
Spessore dei bordi

La proprietà **border-width** specifica lo spessore dei 4 bordi. Come abbiamo detto nell'articolo precedente, funziona solo dopo aver specificato la proprietà [border-style](#). Può essere espresso in diversi valori, ma per convenzione useremo l'unità px (pixel).

Vediamone un esempio

```
<head>
<style>
p
{
border-style:solid;
border-width:5px;
}
</style>
</head>
<body>
<p> Prova </p>
</body>
```

Il nostro risultato sarà il seguente




Arrotondamento dei bordi

La proprietà **border-radius** specifica appunto quanto un bordo debba essere arrotondato. Il valore può essere espresso anch'esso in diversi valori, noi useremo il pixel.

```
<head>
<style>
p
{
border-style:solid;
border-width:5px;
border-radius:10px;
}
</style>
```

```
</head>
<body>
<p> Prova </p>
</body>
```



La differenza è alquanto notevole.

Proprietà individuali

Le proprietà individuali sono delle proprietà che vengono assegnate ad un bordo o angolo specifico e ben definito. Per esempio con la proprietà **border-radius**, si possono ottenere varie proprietà individuali, tra cui **border-top-left-radius**, **border-bottom-right-radius**, per applicare le dimensioni rispettivamente per il bordo superiore sinistro e per quello inferiore destro. Invece con una proprietà come **border-width** è possibile ottenere proprietà come **border-top-width**, che dimensionerà solo il bordo superiore.

```
<head>
<style>
p
{
border-style:solid;
border-bottom-width:5px;
border-top-left-radius:10px;
border-bottom-right-radius:10px;
}
</style>
</head>
<body>
<p> Prova </p>
</body>
```

