



Appunti di HTML5

Conoscere e usare da subito l'html5 su tutti i browser.

[Home](#) [Scarica](#) [Contatti](#)

## Conoscere l'HTML5

Appunti di markup e come implementarlo anche sui browser che non lo supportano.



Questo documento non vuole essere una guida completa su tutti gli aspetti legati all'HTML5, ma una reference pratica incentrata sugli aspetti implementabili da subito affiancati da suggerimenti per rendere le nostre pagine compatibili su tutti i browser e tutte le piattaforme.

Versione PDF per lettura offline e su ebook [clicca qui](#)

## Indice

### Le basi

[Cosa è?](#)  
[Perché adesso?](#)  
[E' supportato?](#)  
[Le fondamenta](#)

### Le novità

[Il doctype](#)  
[Form](#)  
[CANVASS](#)  
[Video](#)

### Conclusioni

## Introduzione

### Cosa è

L'HTML5 è la nuova versione (attualmente in stato di bozza) del linguaggio HTML sul quale è basato il web. L'HTML5 è quindi un'evoluzione incentrata sull'introduzione di nuovi TAG non previsti dall'HTML4 che hanno lo scopo di standardizzare includendole nel linguaggio caratteristiche ormai ampiamente diffuse sul web (Esempio: contenuti multimediali e geolocalizzazione solo per citarne 2). Inoltre semplifica e snellisce elementi esistenti (Esempio: DOCTYPE e META).

### Perché adesso?

I recenti browser web già supportano l'HTML5, inoltre chiunque lavori o si interessi di sviluppo legato al web deve conoscere quello che nei prossimi anni diventerà lo standard del linguaggio principe del web.

Anche se la quota di utilizzatori di browser che non supportano l'HTML5 è sempre più esigua, esistono strumenti e accortezze per rendere le nostre pagine compatibili per chiunque.

### E' supportato?

L'HTML5 è supportato da tutti i browser più recenti, compresi i browser dei telefoni cellulari e tablet (iphone, android, etc).

Esistono però anche strumenti per non escludere (i pochi) utilizzatori di browser datati che potrebbero non interpretare correttamente le direttive HTML5.

### Le fondamenta delle pagine HTML5 (e differenze rispetto all'HTML4)

L'HTML è un linguaggio di markup. E' basato su tag che ne determinano le caratteristiche e le proprietà.

I tag fondamentali di una pagina HTML sono:

- Il DOCTYPE, che specifica al browser di che tipo di documento si tratta
- La sezione , dove viene specificato il titolo della pagina e che contiene informazioni di servizio come la codifica, la posizione dei file css, etc.
- Il body o corpo della pagina che è l'area con il contenuto vero e proprio.

## Le Novità

### Il doctype

Il doctype specifica il tipo di pagina al fine di permettere al browser una corretta interpretazione dei suoi elementi.

Esempio di doctype di una pagina HTML4

[Esempio di doctype di una pagina HTML4] Il doctype di una pagina HTML5 è semplicemente:

Esempio di doctype di una pagina HTML5

```
<!DOCTYPE html>
```

Subito dopo il doctype (come nell'HTML4) si apre il tag <html>

Tale tag è sostanzialmente uguale all'HTML4, l'unico attributo opzionale è la lingua, es: Eventuali attributi opzionali come xml:lang non ha senso specificarli in un documento HTML5.

[Esempio di layout di una pagina HTML (fonte [smashingmagazine.com](http://smashingmagazine.com))]

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8" />
<title>Smashing HTML5!</title>

<link rel="stylesheet" href="css/main.css" type="text/css" />

<!--[if IE]>
  <script src="http://html5shiv.googlecode.com/svn/trunk/html5.js"></script><![endif-->
<!--[if lte IE 7]>
  <script src="js/IE8.js" type="text/javascript"></script><![endif-->
<!--[if lt IE 7]>

  <link rel="stylesheet" type="text/css" media="all" href="css/ie6.css"/><![endif-->
</head>

<body id="index" class="home">
</body>
</html>
```

## Form

[Vedi la demo](#)
[Scarica l'esempio](#)

I form sono un componente essenziale di qualsiasi sito/applicazione web. Ecco la struttura di un classico form HTML4:

```
<form>
  <input type="text">
  <input type="submit" value="invia">
</form>
```

I dati da inserire possono essere di diversi tipi, si va dalle classiche caselle di testo a date, numeri, indirizzi email, URL, etc. Per venire incontro a queste esigenze nelle specifiche HTML5 sono stati previsti degli attributi che permettono di specificare che tipo di dati si aspetta un determinato campo di input.

I browser si stanno aggiornando per riconoscere tali attributi e facilitare gli utilizzatori nell'inserimento dei dati richiesti. La buona notizia è che un campo di input valorizzato con i nuovi attributi, se visualizzato in un vecchio browser che non conosce gli attributi HTML5 verrà renderizzato come una semplice casella di testo, l'utente non avrà l'aiuto fornito dai nuovi motori di rendering, ma potrà continuare a utilizzarlo come ha fatto fino a adesso.

Ecco i principali attributi previsti dall'HTML5:

### autofocus

Funzionalità che dice al browser, una volta caricata la pagina di posizionare il cursore su una determinata casella di testo (Un esempio su tutti la home page di google posiziona l'autofocus sulla casella di ricerca).

Tale funzionalità prima dell'avvento dell'HTML5 veniva realizzata tramite javascript.

I browser che non conoscono questa istruzione la ignorano, non si avrà l'autofocus sulla casella di testo, ma per tutto il resto non cambia niente.

Es:

```
<input type="text" autofocus />
```

### testo segnaposto

Testo preimpostato in una casella che quando prende il fuoco (viene cliccata/selezionata) scompare. Spesso viene usato per spiegare la funzione di quella casella di testo che lo ospita e che dati si aspetta (Es "Cerca nel sito").

I browser che non supportano tale istruzione la ignorano (gli utenti potranno comunque usare la casella di testo anche se senza l'aiuto del testo segnaposto)

Es:

```
<input type="text" placeholder="Testo segnaposto" />
```

### Obbligatorio

Per specificare che un determinato input è richiesto (non opzionale) è previsto l'attributo "required".

```
<input type="text" required />
```

### indirizzi email

Molto spesso si ha necessità di richiedere un indirizzo email nei vari form. Istruendo il browser che una determinata casella di testo si aspetta un indirizzo email

consente allo stesso di facilitare l'utente alla compilazione. (Attualmente il comportamento è diverso da browser a browser, alcuni come l'iphone mostrano una tastiera con i caratteri più usati negli indirizzi email, altri hanno altri comportamenti). I browser che non supportano tale istruzione la ignorano e per gli utenti sarà come una normale casella di testo senza aiuti.

Es:

```
E-mail: <input type="email" name="txtemail" />
```

### indirizzi web (URL)

Come per gli indirizzi email ha lo scopo di facilitare l'utente nell'inserimento degli url. I browser che non supportano questa istruzione la ignorano e per gli utenti sarà come una normale casella di testo senza aiuti.

Es:

```
La tua homepage: <input type="url" name="homepage" />
```

### selettori di date

L'isermimento delle date spesso è un operazione frustrante per l'utente, con tale istruzione i browser che la supportano cercheranno, (ognuno con la propria logica, non c'è un interfaccia standard al momento), di aiutare l'utente con calendarietti popup o altri sistemi propri del sistema utilizzato.

Es:

```
<input type="date" name="txtdata" />
```

(oppure data e ora)

```
<input type="datetime" name="txtdataora" />
```

Vedi la demo

Scarica l'esempio

Questo elenco di nuovi attributi non è completo, per visualizzare tutti i nuovi attributi previsti dall'html5 visitare: [http://www.w3schools.com/html5/tag\\_input.asp](http://www.w3schools.com/html5/tag_input.asp)  
Ancora i browser non li supportano tutti, ma cio' non pregiudica l'usabilità e il funzionamento del form e cominciare fin da ora ad arricchire i form con i nuovi attributi ha il vantaggio di essere pronti al momento che le nuove versioni dei browser li supporteranno.

Per vedere il tuo browser quali attributi supporta puoi visitare questa pagina:

<http://miketaylr.com/code/input-type-attr.html>

## CANVASS

L'elemento canvass introdotto con l'HTML5 è una specie di tela (di dimensioni definibili) che permette il rendering di elementi grafici, immagini e testo. Ecco come inserire un elemento canvass in una pagina web. Testo per chi non supporta il canvass Per "disegnare" nel canvass usiamo javascript (N.B. lo script deve essere DOPO l'elemento canvass) [Esempio disegno linea] In un elemento canvass possiamo disegnare:

- Linee
- Cerchi (o archi)
- Rettangoli
- Immagini
- Gradienti
- Testo

Quando inseriamo oggetti occorre ricordare che il punto con coordinate 0,0 è l'angolo in alto a sinistra. Ecco un esempio completo in cui in un elemento canvass vengono inseriti tutti gli oggetti sopra citati

```
<!DOCTYPE html>
<html lang="it">
<head>
  <meta charset="utf-8" />
  <title>Canvass HTML5</title>
</head>

<body style="background-color:#CECECE">
<h1>Esempio di utilizzo</h1>
<canvas id="prova" width="500" height="400" style="background-color:#FFFFFF; border: 1px solid #000000">
Se il browser non supporta il CANVAS viene visualizzato questo testo.<br />
</canvas>
<br />
il codice di questa pagina:
[DA FARE]
<script type="text/javascript">
  // queste prime 2 righe servono per creare i riferimenti al canvass con id="prova"
  var c=document.getElementById("prova");
  var cont=c.getContext("2d");
  //Disegno una Linea: verticale
  cont.moveTo(10,10);//definisce il punto iniziale
  cont.lineTo(10,110);//definisce il punto finale
  //Disegno una Linea: orizzontale
```

```

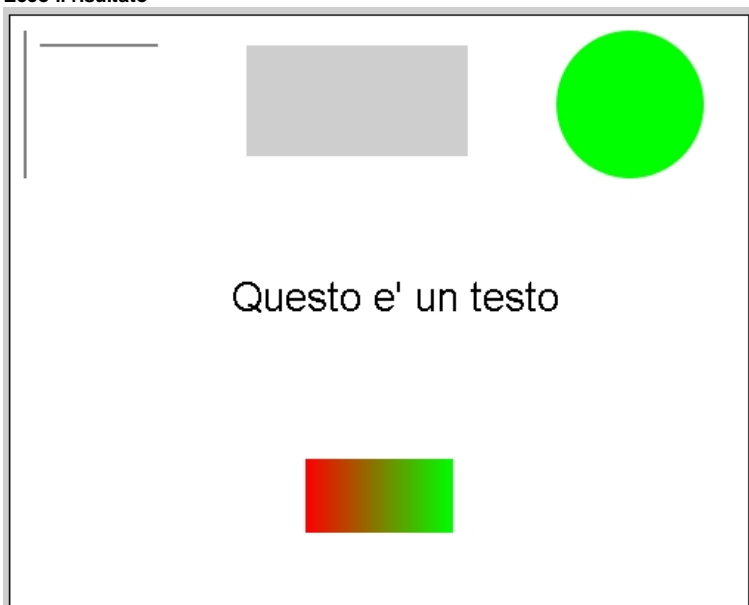
cont.moveTo(20,20);
cont.lineTo(100,20);
cont.stroke();
// Disegno un rettangolo
cont.fillStyle="#CCECECE"; //Colore di riempimento
cont.fillRect(160,20,150,75); // Dove: (coordinata X iniziale, coordinata Y iniziale, larghezza, altezza)
//Disegno un cerchio
cont.fillStyle="#CCCCC";
cont.beginPath();
cont.arc(420,60,50,0,Math.PI*2,true); // Dove: (coordinata X del centro, coordinata Y del centro, Raggio, Angolo
iniziale(usato per archi), angolo finale(usato per archi), true = orario
cont.closePath();
//TESTO
cont.fillStyle="#000000"; //Colore = nero
cont.font = "20pt Arial"; //stile del testo (default = 10x sans-serif)
cont.fillText("Questo e' un testo", 150, 200)
//IMMAGINI
var img=new Image();
img.src="HTML5_logo_75.png";
img.onload = function(){
    cont.drawImage(img,20,300);
};
//GRADIENTI
var grd=cont.createLinearGradient(200,300,300,300); // Coordinate iniziali e finali della linea che definisce il
gradiente
grd.addColorStop(0,"#FF0000");//Colore iniziale
grd.addColorStop(1,"#00FF00");//Colore Finale
cont.fillStyle=grd; //Definisco il gradiente appena creato come riempimento
cont.fillRect(200,300,100,50); //disegno un rettangolo che verrà riempito con il gradiente appena definito

cont.fill();
</script>

</body>
</html>

```

#### Ecco il risultato


[Vedi la demo](#)
[Scarica l'esempio](#)

Come mostrato nell'esempio il canvass viene "disegnato" tramite javascript, e quindi abbiamo la possibilità tramite opportuni script (che non fanno parte di questa guida) di animare il canvass stesso con animazioni e effetti. Ecco alcuni esempi:

- [Esempio di orologio realizzato con l'elemento canvass dell'html5](#)
- [Altri esempi di utilizzo del canvass](#)

## Video

Sempre più pagine web mostrano filmati e video vari. Fino a oggi mancava un'implementazione standard per incorporare un file video in una pagina web. Per venire incontro a questa esigenza nelle specifiche dell'HTML5 è stato introdotto il tag <video>.

Prima dell'HTML5 i video venivano mostrati con player basati su macromedia flash, che si occupava di gran parte degli aspetti tecnici del video come la codifica del filmato e dell'audio.

Oggi possiamo incorporare un filmato direttamente nella pagina web senza dover passare da flash, ma per farlo occorre conoscere gli standard video attualmente supportati, che sono i seguenti 3:

- Ogg
- WebM
- MP4

Ogni browser ha scelto di supportare 1 o più standard di codifica video e quindi il risultato è che non c'è un formato video compatibile con tutti i browser.

[Vedi tabella comparativa su wikipedia](#)

Il tag <video> viene incontro anche a questa problematica fornendo la possibilità di inserire più file video (in vari formati) nel solito tag, in maniera tale che il browser inizia dal primo, e se non lo supporta passa al secondo, poi al terzo e così via fino a quando non trova un filmato in un formato che può decodificare. Ecco un esempio di tag video

```
<video width="320" height="240" controls>
<source src = "nomefile.mp4" type = 'video/mp4; codecs="avc1.42E01E, mp4a.40.2"'>
<source src = "nomefile.webm" type = 'video/webm; codecs="vp8, vorbis"'>
<source src = "nomefile.ogv" type = 'video/ogg; codecs="theora, vorbis"'>
Messaggio per i browser che non supportano nessuno dei formati elencati sopra
</video>
```

E per i browser che non supportano il tag <video>?

La soluzione migliore e trasparente per i fruitori è inserire un classico player flash come flowplayer (flowplayer.org) e per chi non supporta neanche flash un eventuale messaggio di avviso.

## Conclusioni

L'HTML5 e il supporto a tale linguaggio da parte dei browser è in continuo sviluppo, ciò non toglie che cominciare a implementare le novità dell'ormai prossimo futuro possa portare dei benefici sia a breve che a lungo termine. Questo documento non vuole essere una guida completa, si concentra soprattutto sulle features implementabili da subito e compatibili con la quasi totalità dei browser presenti e quindi esso stesso sarà in continua evoluzione.

Aggiornamenti, segnalazioni e articoli di approfondimento saranno pubblicati sul blog di [inrealizzazione.it](http://inrealizzazione.it) (abbonatevi al feed RSS)

Per segnalazioni, commenti o qualsiasi cosa scrivete a [html5@inrealizzazione.it](mailto:html5@inrealizzazione.it)

Per qualsiasi informazione o segnalazione: [html5@inrealizzazione.it](mailto:html5@inrealizzazione.it)

[Back to top](#)