

# Codelgniter è un Framework per creare Applicazioni

Codelgniter è uno strumento per persone che vogliono realizzare applicazioni usando PHP. Il suo obiettivo principale è quello di permettere di sviluppare progetti molto velocemente invece di riscrivere tutto il codice ogni volta da capo, mettendo a disposizione una ricca serie di librerie per risolvere i compiti più comuni, così come una semplice interfaccia ed una struttura logica per accedere a queste librerie. Codelgniter permette di focalizzare la propria creatività sul progetto, minimizzando il codice necessario per un determinato task.

## Codelgniter è gratuito

Codelgniter è licenziato sotto Apache/BSD-style open source license così è possibile utilizzarlo comunque si desideri. E' possibile leggere più informazioni circa la liceza [qui](#) (inglese).

## Codelgniter gira su PHP 4

Codelgniter è scritto per essere compatibile con PHP4. Benché apprezziamo i grandi vantaggi che PHP 5 ha introdotto nella gestione degli oggetti, abbiamo realizzato soluzioni che ovviassero ai problemi di compatibilità (ereditarietà...), questo perché al tempo in cui CI è stato scritto PHP 5 non era molto diffuso, ciò avrebbe influito sul numero dei potenziali utilizzatori del nostro framework. I maggiori fornitori di OS come RedHat si sono mossi lentamente nel supportare PHP 5, così abbiamo pensato che non c'era un vero interesse della comunità PHP se Codelgniter fosse scritto in PHP 5.

Nota: Codelgniter girerà anche su PHP 5. Semplicemente non usufruirà dei vantaggi nativi in questa versione.

## Codelgniter è leggero

Veramente leggero. Il cuore del sistema richiede solo poche e piccole librerie. Questo è in contrasto con molti frameworks che richiedono significative risorse. Librerie aggiuntive sono caricate solo a richiesta, in base alle necessità, questo permette di avere un sistema di base molto snello e abbastanza veloce.

## Codelgniter è veloce

Veramente veloce. Sfidiamo chiunque a trovare un framework che abbia performance migliori di

## Un primo sguardo a CodeIgniter

Scritto da Administrator

Mercoledì 20 Gennaio 2010 12:53 - Ultimo aggiornamento Sabato 20 Febbraio 2010 15:20

---

CodeIgniter.

### CodeIgniter Usa M-V-C

CodeIgniter usa l'approccio Model-View-Controller approach, il quale permette una forte separazione tra la parte logica dell'applicazione e quella dell'interfaccia. Questo è particolarmente utile in progetti lavorano con template dove la presenza di codice è ridotta al minimo. L'approccio MVC verrà spiegato più in dettaglio nella sezione apposita

### CodeIgniter Genera URLs Chiari

Gli URLs generati da CodeIgniter sono chiari e adatti per i motori di ricerca. Invece di usare l'approccio standard "query string" per URLs creati dinamicamente, CodeIgniter utilizza l'approccio "segment-based":

```
{codecitation class="brush: php"}
```

```
example.com/news/article/345
```

```
{/codecitation}
```

Nota: di default il file `index.php` è incluso nell'URL, ma è possibile rimuoverlo usando un semplice `.htaccess` file.

### CodeIgniter Packs a Punch

CodeIgniter viene in aiuto con un'ampia serie di librerie che facilitano lo sviluppatore nella realizzazione dei task più comuni, come l'accesso ai database, spedire email, validare i dati di un form, mantenere le sessioni, manipolare immagini, lavorare con i dati XML-RPC e molto altro ancora.

### CodeIgniter è Estensibile

## Un primo sguardo a Codelgniter

Scritto da Administrator

Mercoledì 20 Gennaio 2010 12:53 - Ultimo aggiornamento Sabato 20 Febbraio 2010 15:20

---

Il systema puà essere facilmente esteso attraverso l'uso le librerie plugin e helper, oppure attraverso l'estensione di classi o system hooks.

### Codelgniter Non Richiede Un Template Engine

Benché Codelgniter *sia dotato* di un semplice template engine che può essere usato opzionalmente, non ti forza ad usarne uno. I Template engines semplicemente possono non uguagliare le performace di PHP nativo, e la sintassi che è necessario imparare per usare un template engine è solo leggermente più facile che imparare le basi del PHP. Considera questo blocco di codice PHP:

```
{codecitation class="brush: php"}
```

```
<ul>
```

```
<?php foreach ($addressbook as $name): ?>
```

```
<li><?=$name></li>
```

```
<?php endforeach; ?>
```

```
</ul>
```

```
{/codecitation}
```

Confrontalo con questo pseudo-codice usato da un template engine:

## Un primo sguardo a Codelgniter

Scritto da Administrator

Mercoledì 20 Gennaio 2010 12:53 - Ultimo aggiornamento Sabato 20 Febbraio 2010 15:20

---

```
{codecitation class="brush: php"}
```

```
<ul>
```

```
{foreach from=$addressbook item="$name" }
```

```
<li>{$name}</li>
```

```
{/foreach}
```

```
</ul>
```

```
{/codecitation}
```

Certo, il template engine è un po' più chiaro, ma il prezzo che si paga ricade sulle performace, perché il codice deve comunque essere riconvertito in PHP per funzionare. Uno dei nostri primi obiettivi è *massimizzare le performance*, quindi noi scegliamo di non richiedere l'utilizzo di un template engine.

## Codelgniter è Completamente Documentato

I programmatori amano il codice e odiano scrivere documentazione. Noi non siamo differenti, certamente, ma la documentazione è **importante** almeno tanto quanto il codice, per questo ci impegnamo a farlo. Il nostro codice è estremamente chiaro e ben commentato.

## Codelgniter ha una Importate Community di Utenti

E' a disposizione una comunità di utenti attivi partecipanti ai nostri [forum](#) (in inglese: [Community Forums](#))

## Un primo sguardo a Codelgniter

Scritto da Administrator

Mercoledì 20 Gennaio 2010 12:53 - Ultimo aggiornamento Sabato 20 Febbraio 2010 15:20

---

).