

ΜΟΝΑΔΑ ΑΡΙΣΤΕΙΑΣ
ΑΝΟΙΧΤΟΥ ΛΟΓΙΣΜΙΚΟΥ

ΧΑΡΟΚΟΠΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ



ΜΟΝΑΔΕΣ ΑΡΙΣΤΕΙΑΣ
ΑΝΟΙΧΤΟΥ ΛΟΓΙΣΜΙΚΟΥ

CodeIgniter



An open-source PHP web application framework.
Home page: <http://ellislab.com/codeigniter>



ψηφιακή ΕΡΑ66α
Όλα είναι δυνατό
Εθνικό Πρόγραμμα
"Ψηφιακή Συγκλήση"



ΕΥΡΩΠΑΪΚΗ ΕΝΩΣΗ
ΕΥΡΩΠΑΪΚΟ ΤΑΜΕΙΟ
ΠΕΡΙΦΕΡΕΙΑΚΗΣ ΑΝΑΠΤΥΞΗΣ



ΕΣΠΑ
2007-2013
πρόγραμμα για την ανάπτυξη



ΕΕΛ/ΛΑΚ
Εταιρεία Εκδόσεων Ληροφορικής /
Ληροφορική Ανάπτυξη Κωδίκων

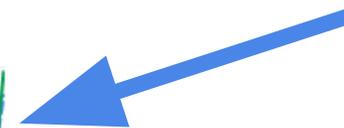
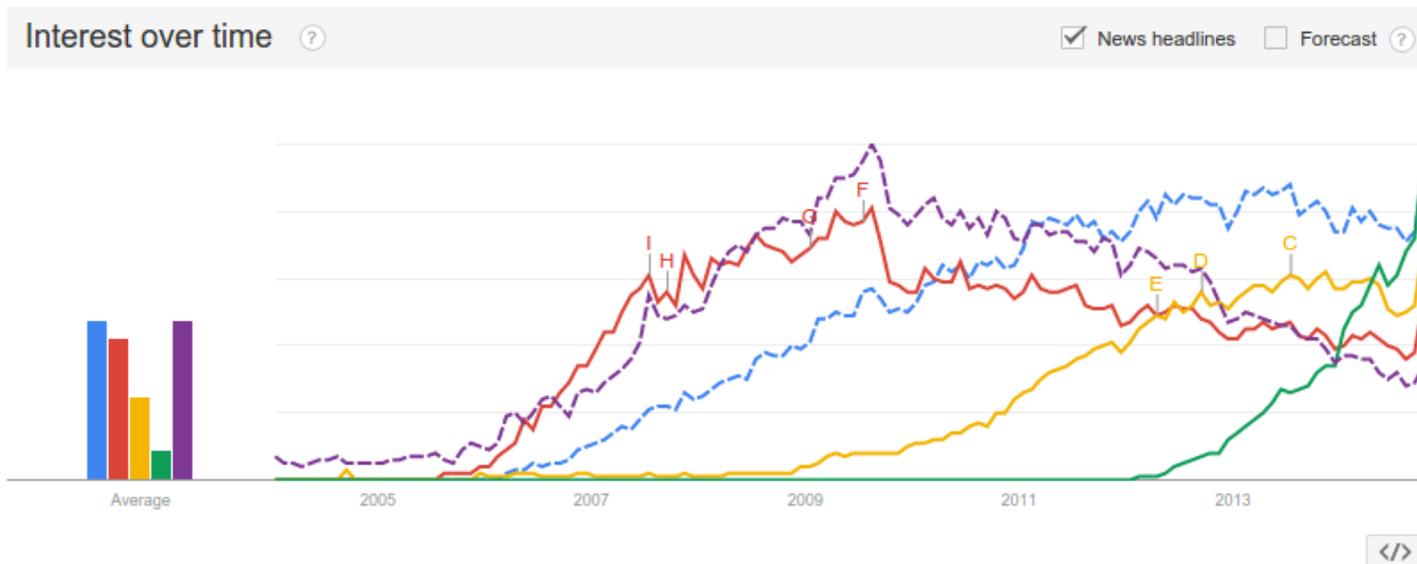


PHP frameworks

Topics Subscribe Share

CodeIgniter Web Development	cakephp Search term	yii Search term	laravel Search term	Zend Framework Web Framework
---------------------------------------	-------------------------------	---------------------------	-------------------------------	--

Beta: Measuring search interest in *topics* is a beta feature which quickly provides accurate measurements of overall search interest. To measure search interest for a specific *query*, select the "search term" option. [?](#)



List of issues

- What is **CodeIgniter**?
- Why use **CodeIgniter**?
- Application flow
- References class
- Active Records
- Links

What is CodeIgniter?

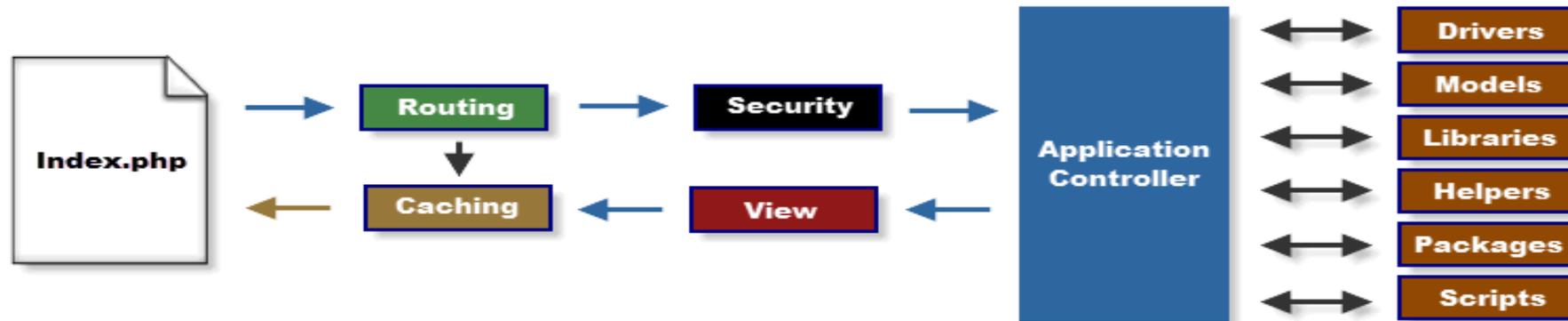
CodeIgniter is an open source rapid development web application framework, for use in building dynamic web sites with PHP.

- Lightweight
- Fast performance
- Very little config

Why use CodeIgniter?

- MVC architecture
- Little server requirements
 - PHP version 5.1.6 or newer.
 - A Database. Supported databases are MySQL (4.1+), MySQLi, MS SQL, Postgres, Oracle, SQLite, and ODBC.
- Easy to understand and extend
- Support many database engines
- Large and active user community
- Excellent documentation

Application flow > Flowchart



- The index.php serves as the front controller, initializing the base resources needed to run **CodeIgniter**.
- The Router examines the HTTP request to determine what should be done with it.
- If a cache file exists, it is sent directly to the browser, by passing the normal system execution.
- Security. Before the application controller is loaded, the HTTP request and any user submitted data is filtered for security.
- The Controller loads the model, core libraries, helpers, and any other resources needed to process the specific request.
- The finalized View is rendered then sent to the web browser to be seen. If caching is enabled, the view is cached first so that on subsequent requests it can be served.

Application flow > URL and Folder struct

Segment based

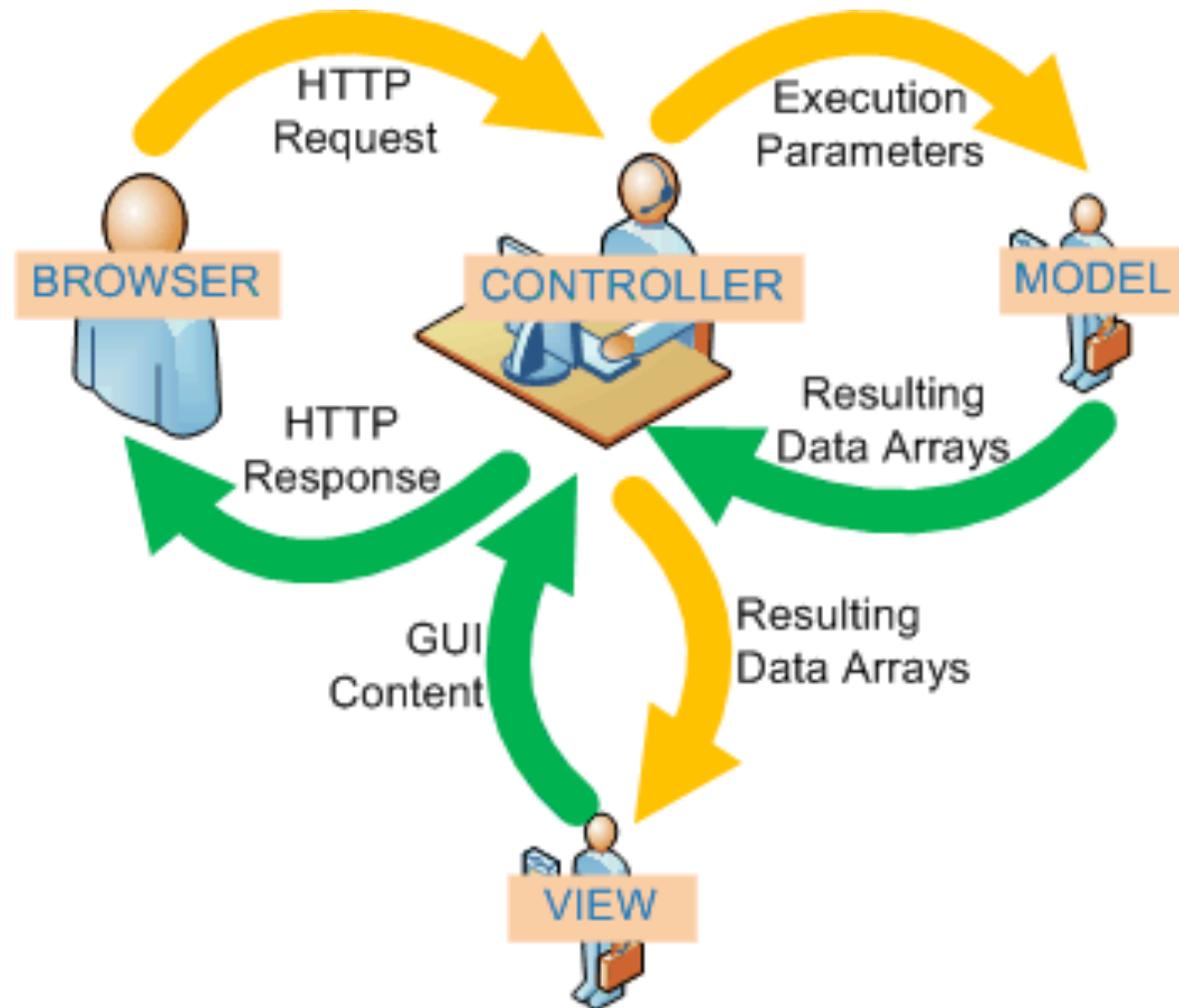
http://example.com/index.php/news/local/metro/crime_is_up

```
$this->uri->segment(n)
```

Supports query string URL

```
application/  
├── cache  
├── config  
├── controllers  
├── core  
├── errors  
├── helpers  
├── hooks  
├── language  
│   └── english  
├── libraries  
├── logs  
├── models  
├── third_party  
└── views
```

MVC



Application flow > MVC > Controller

```
<?php

class Products extends CI_Controller{
    public function __construct(){
        parent::__construct();
        $this->load->model('Product_M');
    }

    public function index(){
        $data['products'] = $this->Product_M->get_all();

        $this->load->view('product', $data);
    }

    public function get_byid($id){
        $data['product'] = $this->Product_M->get_byid($id);
        $this->load->view('product_detail', $data);
    }
}
```

Application flow > MVC > Model

```
<?php

class Product_M extends CI_Model{
    public function __construct(){
        parent::__construct();
        $this->load->database();
    }

    public function get_all(){
        $query = $this->db->get('products');
        return $query->result_array();
    }

    public function get_byid($id){
        $query = $this->db->where(array('product_id'=>$id))
        ->get('products');
        return $query->result_array();
    }
}
```



Application flow > MVC > View

```
<html>
<body>
<?php
    foreach($products as $key => $value){
        echo '<a href="product/' . $value['product_id'] . '">' . $value['product_name'] . '</a>';
        echo '<p>Color: ' . $value['color'] . '</p>';
        echo '<p>Price: ' . $value['price'] . '$</p>';
    }
?>
<?php $this->load->view('footer') ?>
</body>
</html>
```



References class > Libraries

Class Reference

- ⇒ Benchmarking Class
- ⇒ Calendar Class
- ⇒ Cart Class
- ⇒ Config Class
- ⇒ Email Class
- ⇒ Encryption Class
- ⇒ File Uploading Class
- ⇒ Form Validation Class
- ⇒ FTP Class
- ⇒ HTML Table Class
- ⇒ Image Manipulation Class
- ⇒ Input Class
- ⇒ Javascript Class
- ⇒ Loader Class
- ⇒ Language Class
- ⇒ Migration Class
- ⇒ Output Class
- ⇒ Pagination Class
- ⇒ Security Class
- ⇒ Session Class
- ⇒ Trackback Class
- ⇒ Template Parser Class
- ⇒ Typography Class
- ⇒ Unit Testing Class
- ⇒ URI Class
- ⇒ User Agent Class
- ⇒ XML-RPC Class
- ⇒ Zip Encoding Class

References class > Helpers

Helper Reference

- ⇒ Array Helper
- ⇒ CAPTCHA Helper
- ⇒ Cookie Helper
- ⇒ Date Helper
- ⇒ Directory Helper
- ⇒ Download Helper
- ⇒ Email Helper
- ⇒ File Helper
- ⇒ Form Helper
- ⇒ XML Helper
- ⇒ HTML Helper
- ⇒ Inflector Helper
- ⇒ Language Helper
- ⇒ Number Helper
- ⇒ Path Helper
- ⇒ Security Helper
- ⇒ Smiley Helper
- ⇒ String Helper
- ⇒ Text Helper
- ⇒ Typography Helper
- ⇒ URL Helper

Active Records

- Selecting Data

```
$query = $this->db->get('mytable', 10, 20);  
// Produces: SELECT * FROM mytable LIMIT 20, 10
```

- Inserting Data

```
$data = array(  
    'title' => 'My title' ,  
    'name' => 'My Name' ,  
    'date' => 'My date');  
$this->db->insert('mytable', $data);
```

```
// Produces: INSERT INTO mytable (title, name, date) VALUES ('My title', 'My name', 'My date')
```

- Updating Data

- Deleting Data

Useful Libraries & Helpers

- Pagination
- XSS Filtering
- Form Validation
- File Upload
- Session encryption
- URL helper
- Email

Links

- CodeIgniter User Guide
- CodeIgniter From Scratch
- Codeigniter REST server
- Form Helper
- JSON Validator
- CURL library