**CodeIgniter 3 :**

**\_\_construct():**

In CodeIgniter 3, the **\_\_construct()** method is commonly used in controllers and models to perform initialization tasks when an instance of the class is created. This method is part of the standard PHP object-oriented programming (OOP) syntax and can be overridden in your CodeIgniter classes to set up necessary resources, configurations, or other tasks that need to be performed before the controller or model methods are executed.

Here's how the **\_\_construct()** method can be used in CodeIgniter 3:

**In Controllers:**

When you create a custom controller in CodeIgniter, you can define a **\_\_construct()** method in your controller class. This is useful for tasks like loading libraries, helpers, and setting up common data that will be used across multiple methods in the controller.

**class MyController extends CI\_Controller**

**{**

**public function \_\_construct()**

**{**

**parent::\_\_construct();**

**// Load necessary libraries, models, or perform other setup tasks here**

**$this->load->model('my\_model');**

**$this->load->library('session');**

**}**

**public function index()**

**{**

**// Your controller logic**

**}**

**}**

In this example, the \_\_construct() method is used to load a model and a session library, making them available for use in all methods of the controller.

**In Models:**

Similar to controllers, you can also define a **\_\_construct()** method in your custom model classes. This can be useful for tasks like initializing database connections or performing setup operations.

class My\_model extends CI\_Model

{

public function \_\_construct()

{

parent::\_\_construct();

// Initialize database connection or perform other setup tasks here

$this->load->database();

}

public function get\_data()

{

// Your model logic

}

}

class My\_model extends CI\_Model

{

public function \_\_construct()

{

parent::\_\_construct();

// Initialize database connection or perform other setup tasks here

$this->load->database();

}

public function get\_data()

{

// Your model logic

}

}

In this example, the **\_\_construct()** method is used to load the database library, ensuring that the database connection is available for use in model methods.

By using the **\_\_construct()** method, you can centralize common initialization tasks, which can help keep your code organized and reduce redundancy in your controllers and models. Remember to call **parent::\_\_construct();** within your **\_\_construct()** method to ensure that the parent class's constructor is executed properly.