



**EBook Gratis**

# APRENDIZAJE requirejs

Free unaffiliated eBook created from  
**Stack Overflow contributors.**

**#requirejs**

# Tabla de contenido

<b>Acerca de</b> .....	<b>1</b>
<b>Capítulo 1: Empezando con requirejs</b> .....	<b>2</b>
Observaciones.....	2
Examples.....	2
Hola Mundo.....	2
Explicación.....	3
Hola mundo con dependencias anidadas.....	3
Explicación.....	4
Usando el punto de entrada principal de datos.....	5
Incorporación de bibliotecas no-AMD.....	6
<b>Creditos</b> .....	<b>7</b>

---

## Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [requirejs](#)

It is an unofficial and free requirejs ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official requirejs.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to [info@zzzprojects.com](mailto:info@zzzprojects.com)

# Capítulo 1: Empezando con requirejs

## Observaciones

RequireJS es una implementación de la [API del módulo asíncrono \(AMD\)](#) utilizada para cargar archivos javascript de forma asíncrona (es decir, sin bloquear) y administrar las dependencias entre varios archivos javascript. También incluye una herramienta de optimización para combinar y minimizar los archivos de script para la implementación, a la vez que permite al desarrollador mantener un código lógicamente separado.

## Examples

### Hola Mundo

El siguiente ejemplo mostrará una instalación y configuración básicas de RequireJS.

Cree un nuevo archivo HTML llamado `index.html` y pegue el siguiente contenido:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Hello RequireJS</title>
  <script type="text/javascript"
src="http://requirejs.org/docs/release/2.3.2/minified/require.js"></script>
</head>
<body>
  <!-- place content here --->
  <script>
    requirejs(["scripts/say"], function(say) {
      alert(say.hello());
    });
  </script>
</body>
```

Cree un nuevo archivo JS en `scripts/say.js` y pegue el siguiente contenido:

```
define([], function(){
  return {
    hello: function(){
      return "Hello World";
    }
  };
});
```

La estructura de su proyecto debe verse así:

```
- project
  - index.html
  - scripts
    - say.js
```

Abra el archivo `index.html` en un navegador y lo alertará con 'Hello World'.

---

## Explicación

### 1. Cargue el archivo de script `require.js`.

```
<script type="text/javascript"
src="http://requirejs.org/docs/release/2.3.2/minified/require.js"></script>
```

### 2. Cargue el módulo `say` forma asíncrona desde la carpeta de `scripts`. (Tenga en cuenta que *no necesita la extensión `.js` cuando se hace referencia al módulo*). El módulo devuelto se pasa a la función provista donde se invoca a `hello()`.

```
<script>
  requirejs(["scripts/say"], function(say) {
    alert(say.hello());
  });
</script>
```

### 3. El módulo `say` devuelve un solo objeto con una función `hello` definida.

```
define([], function(){
  return {
    hello: function(){
      return "Hello World";
    }
  };
});
```

## Hola mundo con dependencias anidadas

El siguiente ejemplo expande `Hello World` al demostrar múltiples dependencias usando la función `define()`.

Cree un nuevo archivo HTML llamado `index.html` y pegue el siguiente contenido:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Hello RequireJS</title>
  <script type="text/javascript"
src="http://requirejs.org/docs/release/2.3.2/minified/require.js"></script>
</head>
<body>
  <!-- place content here --->
  <script>
    requirejs(["scripts/say"], function(say) {
      console.log(say.hello("english"));
      console.log(say.hello("spanish"));
      console.log(say.hello("french"));
    });
  </script>
```

```
</body>
```

Cree un nuevo archivo JS en `scripts/say.js` y pegue el siguiente contenido:

```
define(["scripts/translations"], function(translations){
  return {
    hello: function(language){
      return translations[language].hello + " " + translations[language].world;
    }
  };
});
```

Cree un nuevo archivo JS en `scripts/translations.js` y pegue el siguiente contenido:

```
define([], function(){
  return {
    "english": {
      hello: "Hello",
      world: "World"
    },
    "spanish": {
      hello: "Hola",
      world: "Mundo"
    },
    "french": {
      hello: "Bonjour",
      world: "Le Monde"
    }
  };
});
```

La estructura de su proyecto debe verse así:

```
- project
  - index.html
  - scripts
    - say.js
    - translations.js
```

Abra el archivo `index.html` en un navegador y la consola mostrará:

```
Hello World
Hola Mundo
Bonjour Le Monde
```

## Explicación

1. Cargue el archivo de script `require.js`.

```
<script type="text/javascript"
src="http://requirejs.org/docs/release/2.3.2/minified/require.js"></script>
```

2. Cargue el módulo `say` forma asíncrona desde la carpeta de `scripts` . (Tenga en cuenta que no necesita la extensión `.js` cuando se hace referencia al módulo). El módulo devuelto se pasa a la función provista donde se invoca a `hello()` .

```
<script>
  requirejs(["scripts/say"], function(say) {
    console.log(say.hello("english"));
    console.log(say.hello("spanish"));
    console.log(say.hello("french"));
  });
</script>
```

3. El módulo `say` devuelve un solo objeto con una función `hello` definida, pero en este caso hemos definido una dependencia ( `scripts/translations` ) y extraeremos las traducciones desde allí.

```
define(["scripts/translations"], function(translations){
  return {
    hello: function(language){
      return translations[language].hello + " " + translations[language].world;
    }
  };
});
```

4. El módulo de `translations` simplemente devuelve un objeto con varias traducciones de palabras.

```
define([], function(){
  return {
    "english": {
      hello: "Hello",
      world: "World"
    },
    "spanish": {
      hello: "Hola",
      world: "Mundo"
    },
    "french": {
      hello: "Bonjour",
      world: "Le Monde"
    }
  };
});
```

## Usando el punto de entrada principal de datos

Un solo punto de entrada a su aplicación es posible con RequireJS usando el atributo de `data-main` dentro de la etiqueta `<script>` .

```
<script type="text/javascript" data-main="scripts/main"
src="http://requirejs.org/docs/release/2.3.2/minified/require.js"></script>
```

En la carga, RequireJS buscará el atributo `data-main` e inyectará la etiqueta de script principal en

el DOM con el conjunto de atributos `async`. Es este archivo el que querrá hacer cualquier configuración antes de iniciar su aplicación.

### Por ejemplo:

```
// contents of scripts/main.js
require.config({
  waitSeconds: 10,
  paths: {
    jquery: 'libs/jquery-1.4.2.min'
  }
});

requirejs(["jquery", "libs/say"], function($, say) {
  var $body = $('body');
  $body.append( $('<p/>').text(say.hello("english")) );
  $body.append( $('<p/>').text(say.hello("spanish")) );
  $body.append( $('<p/>').text(say.hello("french")) );
});
```

## Incorporación de bibliotecas no-AMD

No todas las bibliotecas están definidas de una manera que sea compatible con AMD y la función `define()` RequireJS. Los autores han abordado esto al incluir una directiva `shim` para configurar esas dependencias.

Un ejemplo es el uso del complemento de diseño de interfaz de usuario jQuery. Ese plugin depende de jQuery. Puedes configurarlo así:

```
requirejs.config({
  paths: {
    'jquery': '../path/to/jquery.min',
    'jquery.layout': '../path/to/jquery.layout.min'
  },
  shim: {
    'jquery.layout': {
      deps: ['jquery']
    }
  }
});
```

Y luego usarlo en un módulo de diseño como este:

```
define(['jquery', 'jquery.layout'], function ($, layout) {
  $('body').layout();
});
```

Lea [Empezando con requirejs en línea](https://riptutorial.com/es/requirejs/topic/8791/empezando-con-requirejs): <https://riptutorial.com/es/requirejs/topic/8791/empezando-con-requirejs>



---

# Creditos

S. No	Capítulos	Contributors
1	Empezando con requirejs	<a href="#">Community</a> , <a href="#">Michael Sacket</a>