



**EBook Gratis**

# APRENDIZAJE

## v8

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

**#v8**

# Tabla de contenido

Acerca de.....	1
<b>Capítulo 1: Empezando con v8.....</b>	<b>2</b>
Observaciones.....	2
Versiones.....	2
lanzamientos v8.....	2
Examples.....	2
Instalación o configuración.....	2
<b>Capítulo 2: Devolución de llamadas débiles.....</b>	<b>4</b>
Observaciones.....	4
Examples.....	4
Ejecución del código especificado por el usuario cuando un objeto se recolecta basura.....	4
<b>Capítulo 3: Empezando con v8.....</b>	<b>6</b>
Examples.....	6
Ejecutando v8 en un archivo.....	6
Funciones y objetos incorporados útiles en d8.....	6
<b>Creditos.....</b>	<b>9</b>

---

## Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [v8](#)

It is an unofficial and free v8 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 v8.

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 v8

## Observaciones

Esta sección proporciona una descripción general de qué es v8 y por qué un desarrollador puede querer usarlo.

También debe mencionar cualquier tema importante en v8, y vincular a los temas relacionados. Dado que la Documentación para v8 es nueva, es posible que deba crear versiones iniciales de esos temas relacionados.

## Versiones

### lanzamientos v8

Versión	Fecha de lanzamiento
0.1.0	2008-07-03
1.0.0	2009-02-09
2.0.0	2009-11-18
3.0.0	2010-12-07
3.10.0	2012-10-30
3.20.0	2013-06-28
4.2.1	2015-01-14
4.10.1	2016-01-16

## Examples

### Instalación o configuración

v8 usa Google `depot_tools` para obtener la fuente y construir la biblioteca. Para instalar `depot_tools`, siga las instrucciones aquí:

<https://www.chromium.org/developers/how-tos/install-depot-tools>

Obtener la fuente v8 ejecutando

```
/path/to/depot_tools/fetch v8
```

Después de ejecutar este comando, para extraer nuevas versiones en el futuro, ejecute

```
/path/to/depot_tools/gclient sync
```

Las instrucciones de compilación se proporcionan aquí:

<https://github.com/v8/v8/wiki/Building%20with%20GN>

Lea **Empezando con v8 en línea**: <https://riptutorial.com/es/v8/topic/7639/empezando-con-v8>

# Capítulo 2: Devolución de llamadas débiles

## Observaciones

Las devoluciones de llamada débiles se utilizan principalmente para limpiar objetos C++ incrustados en `InternalField` de un `v8::Object` creado a partir de un `v8::ObjectTemplate`. Cuando el objeto de JavaScript se recolecta como basura, muchas veces también se debe eliminar el objeto de C++. Al establecer una devolución de llamada débil, puede recibir una notificación de que un objeto javascript se ha recolectado como basura y tomar las medidas apropiadas.

Es *MUY* importante recordar que la recolección de basura *NO* es determinista. Su programa puede salir con objetos con devoluciones de llamada de referencia débiles registradas que nunca se llaman. Estas devoluciones de llamada son importantes para un programa de larga duración que se comporte correctamente, pero no se debe confiar en él para liberar recursos de ruta crítica de manera consistente o rápida.

Para que el recolector de basura sepa cuándo debe ejecutarse, debe informarle sobre la cantidad de espacio que utilizan sus objetos C++ a través de la llamada

`v8::Isolate::AdjustAmountOfExternalAllocatedMemory`. El parámetro para esta llamada es el *cambio* en bytes, por lo que cuando lo asignas, a menudo envías en `sizeof(T)` y cuando limpias en tu devolución de llamada de referencia débil, envías en `-sizeof(T)`.

## Examples

Ejecución del código especificado por el usuario cuando un objeto se recolecta basura.

```
/**
 * Runs user-specified code when the given javascript object is garbage collected
 */
template<class CALLBACK_FUNCTION>
void global_set_weak(v8::Isolate * isolate, const v8::Local<v8::Object> & javascript_object,
CALLBACK_FUNCTION function)
{
    struct SetWeakCallbackData{
        SetWeakCallbackData(CALLBACK_FUNCTION function, v8::Isolate * isolate, const
v8::Local<v8::Object> & javascript_object) :
            function(function) {
                this->global.Reset(isolate, javascript_object);
            }
        // function to call for cleanup
        CALLBACK_FUNCTION function;

        // this is the weak reference
        v8::Global<v8::Object> global;
    };

    // This must be dynamically allocated so it sticks around until the object
    // is garbage collected. It cleans itself up in the callback.
    auto callback_data = new SetWeakCallbackData(function, isolate, javascript_object);
```

```
// set the callback on the javascript_object to be called when it's garbage collected
callback_data->global.template SetWeak<SetWeakCallbackData>(callback_data,
  [](const v8::WeakCallbackInfo<SetWeakCallbackData> & data) {
    SetWeakCallbackData * callback_data = data.GetParameter();
    callback_data->function(); // run user-specified code
    callback_data->global.Reset(); // free the V8 reference
    delete callback_data; // delete the heap variable so it isn't leaked
  }, v8::WeakCallbackType::kParameter);
}
```

Lea Devolución de llamadas débiles en línea: <https://riptutorial.com/es/v8/topic/7655/devolucion-de-llamadas-debiles>

# Capítulo 3: Empezando con v8

## Examples

### Ejecutando v8 en un archivo

Usa el shell d8 para ejecutar el motor v8. Utilice el siguiente patrón para ejecutar en un archivo:

```
/path/to/d8 [flags] [file].js
```

Por ejemplo:

```
./d8 --log-gc script.js
```

ejecutará d8 en script.js con el registro de recolección de basura habilitado

### Funciones y objetos incorporados útiles en d8

Además de las bibliotecas definidas en la especificación del lenguaje EcmaScript y la especificación de la API de internacionalización de EcmaScript, d8 también implementa las siguientes funciones y objetos.

- `print(args...): function` . Imprimir a la `stdout` .
- `printErr(args...): function` . Imprimir a `stderr` .
- `write(args...): function` . Igual que el `print` pero sin nueva línea al final.
- `read(filename): function` . Leer el texto del archivo y devuelto como `String` .
- `readbuffer(filename): function` . Leer binario desde archivo y devuelto como `ArrayBuffer` .
- `readline(): function` . Lea la línea de `stdin` . Use `'\n'` para entrada de líneas múltiples.
- `load(filename): function` . Cargar y ejecutar el archivo JavaScript.
- `quit([exitCode]): function` . Salir con el código de salida opcional.
- `version(): function` . Devuelve el código de la versión como `String` .
- `os: object` . Utilidades relacionadas con el sistema operativo, solo disponibles para POSIX.
  - `os.system(command): function` : Ejecutar comando del sistema.
  - `os.chdir(path): function` : cambia el directorio actual.
  - `os.setenv(name, value): function` : establecer la variable de entorno.
  - `os.unsetenv(name): function` : Desactivar variable de entorno.
  - `os.umask(alue) function` : llama a la llamada del sistema `umask` y devuelve la antigua `umask`.
  - `os.mkdirp(path[, mask]): function` : crea un directorio. La máscara (si está presente) se utiliza con la `umask` actual. Los directorios intermedios se crean si es necesario.
  - `os.rmdir(path): function` : eliminar directorio.
- `performance: object` : usar para análisis de performance.
  - `performance.now(): function` . Devuelve una marca de tiempo como doble, medida en milisegundos.
- `Worker: object` : modificado de [HTML5 Web Worker](#) (vea el ejemplo a continuación para más



## detalles)

- `Realm: object` : crear y administrar un entorno aislado (reino).
  - `Realm.create(): function` : `Realm.create(): function` un nuevo reino con un token de seguridad distinto y devuelve su índice.
  - `Realm.createAllowCrossRealmAccess(): function` : `Realm.createAllowCrossRealmAccess(): function` un nuevo reino con el mismo token de seguridad que el reino actual y devuelve su índice.
  - `Realm.current(): function` : devuelve el índice del reino actualmente activo. El índice del reino global es 0.
  - `Realm.global(i): function` : Devuelve el objeto global del reino i.
  - `Realm.owner(globalObj): function` : Devuelve el índice del reino que creó `globalObj`.
  - `Realm.eval(i, s): function` : Evalúa `s` en el reino `i` y devuelve el resultado.
  - `Realm.switch(i): function` : Cambia al reino `i` para entradas interactivas consecutivas.
  - `Realm.dispose(i): function` : Desecha la referencia al reino `i`.
  - `Realm.shared: object` : un elemento de acceso para un único valor compartido entre reinos.

Las implementaciones detalladas y los comentarios se pueden encontrar en [d8.h](#) , [d8.cc](#) , [d8-posix.cc](#) y [d8-windows.cc](#) .

## Ejemplo:

```
print("Hello World!");
write("Hello ");
write("again!\n");
printErr("Nothing went wrong.");

write("Your name: ");
var name = readline();
print("Hello, ", name, "!");

load("external.js");

var string = read("text.txt");
var buffer = readbuffer("binary.bin");

print("Version: ", version());

quit(0); // bye
```

## Ejemplo de trabajador:

### main.js

```
var workerScript = read("worker.js");
var worker = new Worker(workerScript);
worker.postMessage(12);
```

### worker.js

```
onmessage = ev => {
  print(ev); // 12
}
```

```
};
```

## Ejemplo de reino:

```
print(Realm.current()); // 0

var rIndex = Realm.create();
print(rIndex); // 1

Realm.eval(rIndex, "var x = 100");
Realm.eval(rIndex, "print(x)"); // 100
var result = Realm.eval(rIndex, "x * 2");
print(result); // 200

Realm.eval(rIndex, "var rIndex2 = Realm.create()");
Realm.eval(rIndex, "print(rIndex2)"); // 2
Realm.eval(rIndex, "print(Realm.owner(this))"); // 1

try {
  var childGlobal = Realm.global(rIndex);
  print(childGlobal); // error
} catch (e) {
  print("Global object cannot be read/written cross-realm.");
}

var rIndex3 = Realm.createAllowCrossRealmAccess();
Realm.eval(rIndex, "var x = 50");
var childGlobal = Realm.global(rIndex3);
childGlobal.x++;
Realm.eval(rIndex3, "print(x)"); // 51

try {
  Realm.dispose(rIndex3);
  Realm.eval(rIndex3, "print(x)"); // error
} catch (e) {
  print("The realm is dereferenced");
}

Realm.shared = "Hello from another world";
Realm.eval(rIndex, "print(Realm.shared)");
```

Lea Empezando con v8 en línea: <https://riptutorial.com/es/v8/topic/7745/empezando-con-v8>

---

# Creditos

S. No	Capítulos	Contributors
1	Empezando con v8	<a href="#">Community</a> , <a href="#">Paul Sweatte</a> , <a href="#">RamenChef</a> , <a href="#">xaxxon</a>
2	Devolución de llamadas débiles	<a href="#">RamenChef</a> , <a href="#">tbodt</a> , <a href="#">xaxxon</a>