<https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/>

How is javascript asynchronous AND single threaded? 💫

Mar 14, 2016 • 5 minute read

Asynchronous programming is one of those programming paradigms that’s extremely difficult to fully understand, until you’ve done enough of it in practice. In an ideal world this shouldn’t be the case, so here’s yet another attempt to explain the concept of async programming, and why its different from parallel programming, in the context of javascript.

*Everything runs on a different thread except our code.*

At first glance, this sentence doesn’t seem to make a lot of sense. Isn’t everything we execute technically “our code”? Well, yes and no. Lets take a look at two examples of synchronous and asynchronous implementations of the same functionality. Synchronous implementation (python) :

import requests

r = requests.get('http://sohamkamani.com')

print r.text

print "I come after the request"

Async implementation (js) :

var request = require('request');

request('http://sohamkamani.com', function (error, response, body) {

console.log(body);

})

console.log('I come after the request');

Now, all the above code runs on the same thread, no doubt about it. But what were missing is that the request and requests libraries, make http requests that go to other servers. The time spent in sending the request, processing it server side, and returning the response, is not spent in our thread. Thats what the web server you sent the request to does.  
In our python implementation, we wait for all these processes to complete and receive the response before moving on to executing the next line of code. The async philosophy adopted by javascript and Node.js is fundamentally different in this regard. Instead of waiting for the response before executing the next bit of code, we declare what we *want* to happen once we receive our response, and move on to the next bit of code as usual.  
This is why "I come after the request" will always get printed to the console after the response in the case of our python code, and always get printed before the response for our javascript code[[1]](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#footnotes).

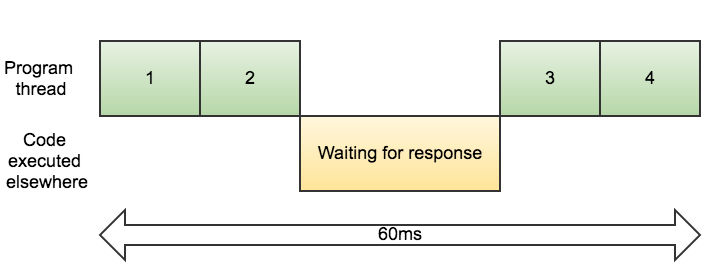
*What good does any of this do me?*

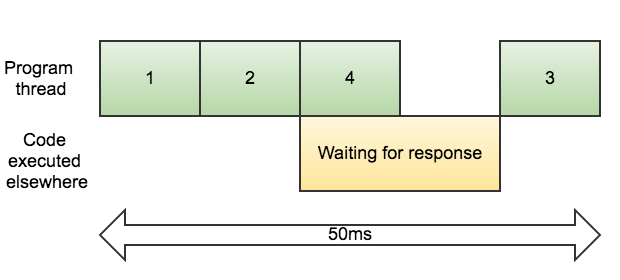
Both the snippets of code are exactly similar in their functionality :

1. import requests == var request = require('request');
2. r = requests.get('http://sohamkamani.com') == request('http://sohamkamani.com', ... )
3. print r.text == console.log(body);
4. print "I come after the request" == console.log('I come after the request');

Let us assume, for the sake of experimentation, that each of the 4 snippets of code above take ~10ms to execute. Since we are only here to see the power of async, we are not going to take the raw execution speed of either language into consideration, and assuming the synchronous parts of both examples to have the same execution time (of 10ms). We will also take two cases of waiting time into consideration, one of 20ms, and one of 5ms.

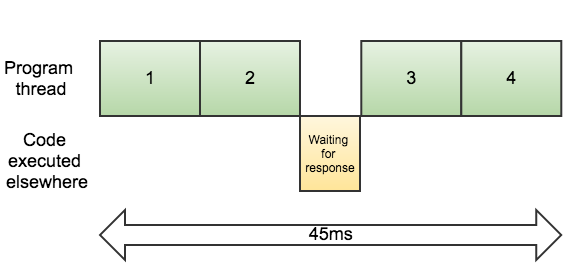
Case 1 (Waiting time = 20ms)

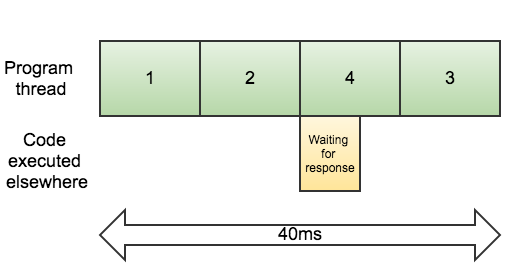
With synchronous execution :  


With asynchronous execution :  


Snippet 4 doesn’t have to wait for our response to arrive in order to execute, but snippet 3 does. Our javascript code handles this by defining tasks that need to wait inside the callback and other tasks outside of it. In the case of our python example, all code after we send the request is blocked until the response arrives.  
This gives us a *net loss of****10ms*** in this case for the synchronous implementation.

Case 2 (Waiting time = 5ms)

With synchronous execution :  


With asynchronous execution :  


Synchronous execution with a smaller waiting time doesn’t look much different from the last picture, but the asynchronous timing diagram is pretty interesting. We see that snippet 4 starts execution as usual during waiting time, but snippet 3 doesnt execute right after the waiting time is over. **This is because snippet 3 and snippet 4 are running on the same thread**and hence snippet 3 has to wait for 4 to finish before it can start. This is a much better illustration of what it means to be *single threaded and asynchronous*.

Final thoughts

*If async is so obviously the correct thing to do, then why should we bother with synchronous programming?*

The first thing that stands out in the javascript code snippet is that it’s *much less simple* than the corresponding python snippet, and so takes a bit more time to read, understand, and develop. In fact, there are many articles and [blog posts](https://www.sohamkamani.com/blog/2015/08/21/understand_node_without_losing_your_mind/) dedicated to managing async code, because without proper management, it can all get out of hand pretty quickly.

For rapid prototypes, or in cases where speed and timing is not the main concern, going the synchronous way can be more productive. On the other hand, if you’re planning to build an application with a lot of I/O and networking tasks, or with a lot of users, then the power of async really starts to shine.

Although async is not embedded in pythons “philosophy”, like it is with NodeJs, there are many libraries which let you leverage event driven and async programming, like the [Twisted](https://twistedmatrix.com/trac/)library.

*[1] Edit - Javascript always finishes the currently executing function first. Thanks @Twitchard for the correction.*

* [**Twitchard**](https://disqus.com/by/disqus_WMLowxoRoM/) • [2 years ago](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-2591782185)

"I come after the request" will \_always\_ come before the request is printed. Not "almost always." Even if you did

request(..., function (error, response, body) {  
console.log('foo');  
})  
callAComputationallyIntensiveSynchronousFunctionThatTakesSixHoursToExecute()  
console.log('bar')

'bar' will still come before 'foo' because Javascript always finishes the currently executing function first. An event will never interrupt a function.

* + [6](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default" \o ")
  + •
  + [Reply](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
  + •
  + Share ›
    - 

* + - [−](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default" \o "Collapse)

[](https://disqus.com/by/sohamkamani/)

[**Soham Kamani**](https://disqus.com/by/sohamkamani/) **Mod** [Twitchard](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-2591782185) • [2 years ago](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-2593374755)

Yep, you're absolutely right. I have edited the post. Thanks so much for the correction :)

* + - [1](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
    - •
    - [Reply](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
    - •
    - Share ›
      * 
    - [−](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)

[](https://disqus.com/by/timoxley/)

[**Tim Oxley**](https://disqus.com/by/timoxley/) [Twitchard](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-2591782185) • [2 years ago](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-2593351425)

yep, that's bad misinformation. Also note it won't just finish the \*current\* function, it will run to completion of all synchronous functions and I believe anything queued with process.nextTick... before the request callback is handled.

* + - •
    - [Reply](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
    - •
    - Share ›
      * 
      * [−](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)

[](https://disqus.com/by/sohamkamani/)

[**Soham Kamani**](https://disqus.com/by/sohamkamani/) **Mod** [Tim Oxley](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-2593351425) • [2 years ago](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-2593375539)

I have edited the post. Thanks for the correction :)

* + - * [1](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
      * •
      * [Reply](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
      * •
      * Share ›
        + 
  + [−](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)

[](https://disqus.com/by/nikhilponnuru/)

[**Nikhil Ponnuru**](https://disqus.com/by/nikhilponnuru/) • [a month ago](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-3743210483)

good article!

* + [4](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
  + •
  + [Reply](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
  + •
  + Share ›
    - 
  + [−](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)

[](https://disqus.com/by/tracker1/)

[**Tracker1**](https://disqus.com/by/tracker1/) • [2 years ago](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-2591802961)

Regarding the last comment on async looking a bit complex... This is part of why I really like babel and have using async/await...

import axios from 'axios';  
async function getData() {  
var result = await axios.get('[http://sohamkamani.com](http://disq.us/url?url=http%3A%2F%2Fsohamkamani.com%3A-LFCoezevRlY0nY-eynx-tfFCpQ&cuid=3764043)', {responseType:'text' /\*default is json\*/});  
console.log([result.data](http://disq.us/url?url=http%3A%2F%2Fresult.data%3Aj5R619kvWJKIgosVyqNADKfqU_k&cuid=3764043" \o "result.data));  
console.log('I come after the request');  
}

* + [2](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
  + •
  + [Reply](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
  + •
  + Share ›
    - 
  + [−](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)

[](https://disqus.com/by/manpreetsethi/)

[**Manpreet Sethi**](https://disqus.com/by/manpreetsethi/) • [2 years ago](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-2598826371)

Great article!

* + [1](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
  + •
  + [Reply](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
  + •
  + Share ›
    - 
  + [−](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)

[](https://disqus.com/by/artur_mostowiak/)

[**Artur Mostowiak**](https://disqus.com/by/artur_mostowiak/) • [a month ago](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-3744422413)

For me it is a better explanation than in Kyle's Simpson book YDKJS

* + •
  + [Reply](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
  + •
  + Share ›
    - 
  + [−](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)

[](https://disqus.com/by/itsabdulghani/)

[**Abdul Ghani**](https://disqus.com/by/itsabdulghani/) • [3 months ago](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-3632445176)

great article. so node.js is single-threaded because it is asynchronous? so that in terms of synchronous fashion it said multi-threading because of waiting? or is there any use of their multi-threading potential? (I thought that multi-threading could be used to handle more request as much as non-blocking i/o could handle request simultaneously (without waiting))

* + •
  + [Reply](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
  + •
  + Share ›
    - 
    - [−](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)

[](https://disqus.com/by/hkmaly/)

[**hkmaly**](https://disqus.com/by/hkmaly/) [Abdul Ghani](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-3632445176) • [3 months ago](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-3634620878)

It's more like node.js is single threaded DESPITE being asynchronous :).

* + - [2](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
    - •
    - [Reply](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
    - •
    - Share ›
      * 
  + [−](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)

[](https://disqus.com/by/disqus_qDA7kOq1HB/)

[**Tom**](https://disqus.com/by/disqus_qDA7kOq1HB/) • [6 months ago](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-3489133293)

Correct me if I'm wrong but isn't that a very misleading title, JavaScript isn't asynchronous, it's synchronous.

* + •
  + [Reply](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
  + •
  + Share ›
    - 
    - [−](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)

[](https://disqus.com/by/atilpai/)

[**Atil Pai**](https://disqus.com/by/atilpai/) [Tom](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-3489133293) • [5 months ago](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-3570839828)

Do your javascript advertisements on a random page load before the other DOM elements render???  
No! They do not. Thats why the model behind building Javascript is the ASYNC Non-Blocking model.

* + - •
    - [Reply](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
    - •
    - Share ›
      * 
      * [−](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)

[](https://disqus.com/by/disqus_qDA7kOq1HB/)

[**Tom**](https://disqus.com/by/disqus_qDA7kOq1HB/) [Atil Pai](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-3570839828) • [3 months ago](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-3648067369)

But we're not talking about the page render, but the actual JavaScript code that runs synchronously, albeit with some asynchronous features via callbacks.

* + - * •
      * [Reply](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
      * •
      * Share ›
        + 
  + [−](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)

[](https://disqus.com/by/kulkarni_baswaraj/)

[**Kulkarni Baswaraj**](https://disqus.com/by/kulkarni_baswaraj/) • [9 months ago](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-3346094732)

Thanks for this post !!!

* + •
  + [Reply](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
  + •
  + Share ›
    - 
  + [−](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)

[](https://disqus.com/by/ajoms/)

[**ajoms**](https://disqus.com/by/ajoms/) • [10 months ago](https://www.sohamkamani.com/blog/2016/03/14/wrapping-your-head-around-async-programming/#comment-3297115212)

Asynchronous JavaScript is simply "Time Sharing" Client-side Thread execution while waiting for external responses..Check "AJOMS" Web-site Example to see Thread Tme-Shared( Asynchronous JavaScript) execution in action.

* + •
  + [Reply](https://disqus.com/embed/comments/?base=default&f=sohamkamani&t_u=https%3A%2F%2Fwww.sohamkamani.com%2Fblog%2F2016%2F03%2F14%2Fwrapping-your-head-around-async-programming%2F&t_d=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&t_t=How%20is%20javascript%20asynchronous%20AND%20single%20threaded%3F%20%F0%9F%92%AB&s_o=default)
  + •
  + Share ›
    - 