

Sample Answers ▾

1. Who has the started the assignment?

Do you have have hints or advice for other students?

Discussed in tutorial.

2. The assignment specification doesn't fully explain the assignment - what can I do?

A big part of the assignment is understanding how to use the backend code you've been given.
You'll spend more time doing this than implementing subset 0.

Read the API documentation (run the backend and look at: <http://localhost:5000>)

Read ansers on the forum.

Ansk questions on the forum.

3. You work on the assignment for an hour tonight. What do you need to do when you are finished?

1. Update your **diary.txt** with a line indicating you work for an hour and give a brief breakdown of what the work was:
(was it coding, debugging, testing ...)
2. push the latest version of your code to **gitlab.cse.unsw.edu.au** by running

```
$ git push
```

4. The back end isn't working for me and another students says I should run **git pull**, why?

When we fix any issues with the backend we'll push them to your **gitlab.cse.unsw.edu.au** repo.
git pull transfers any updates in your **gitlab.cse.unsw.edu.au** repo to you local git repo.

Note this could break if you change the back end code - do not change the back end code yourself.

5. How hard are the subsets?

Once you understand what you have to do, subset 0 is not that hard.

Subset 1 will require more coding but is quite doable and can get you a CR.

Subset 2 is more work but not that hard and can get you a DN.

Subsets 3 are 4 are much more difficult and will require lots of reading, thinking & coding, if you want a HD.

6. What is Flask?

What do I need know about Flask for the assignment?

Flask is a a very popular web framework based on Python.

It has powerful features available and is used in large high traffic applications at companies such as pinterest, linkedin, twilio and lyft.

It has been used for the assignment backend code.

You can not change the backend.

You do not need to know anything about Flask for the assignment.

7. What is sqlite?

What do I need know about sqlite for the assignment?

sqlite is a a very popular embedded database system. It is much simpler to deploy & use than a client/server database system and works well for low-medium traffic applications with few clients.
It has been used for the assignment backend code.

You can not change the backend.

You do not need to know anything about sqlite for the assignment.

8. I have heard Angular (or React or jQuery or ...) is great can I use it for the assignment?

No you can only use Javascript you write yourself.
You can't use Javascript libraries or frameworks.

I have heard bootstrap is great can I use it for the assignment?

Yes you can. You are permitted to use with attribution CSS written by someone else.
bootstrap includes some Javascript support including jQuery which you are permitted to include so bootstrap works. You are not permitted to use/call jQuery in your own code.

An answer on Stack Overflow includes 5 lines of Javascript which does something my code needs - can I use it?

Yes you can with clear attribution (include the Stack Overflow URL in a comment)

9. JavaScript is a single-threaded, synchronous programming language that relies on asynchronous behaviour for I/O. What does this mean?

JavaScript uses a single-threaded event loop to execute code. Rather than 'blocking', or pausing execution when it hits I/O calls (read, write to hardware, network) it instead provides a callback that will be executed when the I/O call completes.

This allows the program to continue execution (important for UI responsiveness among other things), but comes at a cost: we can't guarantee that the consecutive lines of code will be executed in the order they are written!

It's important to note: in the context of the JavaScript engine one thing is always happening at a time. BUT, JavaScript implicitly relies on the its environment to deal with tasks happening at the same time.

10. What is AJAX and why is it a important design principle?

Asynchronous JavaScript And XML. It's called that because it was a set of principles for developing a web application that can send and retrieve data from a server asynchronously (in the background) without interfering with the display and behaviour of the existing page. The reason it mentions XML is because when it was developed XML was how sites sent little bits of data that weren't full pages to the client.

These days we use another format; JSON but the principle still stands. The basic idea is simply a separation of the data a page displays and the display itself.

Basically rather than the server sending the html which renders into your newsfeed, it sends a framework to render a arbitrary newsfeed post and then your computer asks the server for individual posts.

why? It's good decoupled code, you can change what data is being sent and how it gets rendered independently. It also shifts to making the client computer do the heavy rendering while leaving the server to do quick data processing. This ends up being faster overall, computers can handle having to render a bunch of things more than server's can handle doing a lot of processing to form html simply because servers are getting hit up non stop.

11. Why is this block of code potentially problematic?

```
function getName() {  
  let name;  
  
  const timer = setTimeout(() => {  
    name = 'Andrew';  
  }, 10);  
  
  return name;  
}
```

The `setTimeout` call is asynchronous; this means it will be pushed to a future iteration of the event loop. Meanwhile, the `return` statement is synchronous and will occur immediately.

The result will be that the function will return `name` as undefined before the `setTimeout` is ever executed.

12. What options are there to manage asynchronous execution in JavaScript?

Asynchronous behaviour is managed in three main ways:

- **Callbacks:** Traditional way to handle the 'then' versus 'now' problem. Callbacks are simply functions that we can attach to an event. When the event is triggered, the callback is run. Callbacks can however get messy to reason with when more complex event interaction is required.
- **Promises:** As of ES6+, promises have allowed programmers to tie asynchronous calls together in a synchronous way. Promises return Promises so they can be chained together. See MDN or course notes for more.
- **async/await:** As of ES7+, there's been an extension to the Promise API to allow a 'simpler' syntax for Promises. This is `async/await`. See MDN or course notes for more.

13. Familiarity with `fetch` is important for the assignment. Below is short overview of how it works. Make sure you go through this with your tutor!

```

/*
  The Fetch API takes in a url, with
  an optional second parameter,
  this includes options like additional headers,
  and body data.
*/

// An example GET request
const getGoogle = fetch('https://www.google.com')

// A successful fetch returns a promise of a response stream object.
const googleHTML = getGoogle.then(response => response.text())

// Now if we wanted to do something with the googleHTML
googleHTML.then(html => {
  console.log(html)
})

// A common use case is fetching JSON
fetch('https://someurlthatreturnsJSON.com/api')
  .then(response => response.json())
  .then(json => {
    // do something with the JSON
    console.log(json)
  })

// More complex use cases involve adding headers
// or changing the request type
const blog = {
  title: 'Cool Post',
  description: 'Last year I did some stuff in San Francisco, it was so fun!'
}

fetch('https://someurlthatwecanPOSTJSONto/api/post', {
  method: 'POST',
  body: blog,
  headers: {
    'Content-Type': 'application/json'
  }
}).then(response => {
  // do something if successful
})

```

14. Images in HTML can have a src URL or a src 'data' URL. What's the difference?

URL based images will fetch external resources and populate with the image (assuming it's a valid image) when loaded.

Data URL based images use a raw encoding of the image's bytes inline. The assignment relies on data based images. Data URLs can be used for many different file formats (not just images).

15. What is callback hell and what can we do to prevent it?

when you have callbacks waiting for callbacks waiting for callbacks due to a need to do a set of async operations in order. We can fix this by using promises which neaten the syntax and provide useful utilities like `promise.all` and `promise.race`. see <https://js.alhinds.com/content/async/> for more detail.

16. Refer to the js repo and observe the "blocked". Why is the UI locking up? Can promises fix this?

The ui locks up because js only allows 1 thing to run at a time, the slow function takes up this only slot so the key press events have to wait.

Promises would not fix this as this is compute bound, this is rare in js but if your issue is heavy processing promises don't help. That being said web workers may be able to let you shift a computationally expensive operation to the background so you can still react to events

17. Refer to the js repo and observe the code in "promises". Observe the difference in using promise.all and doing the requests in tandem (without promise.all). Why does the first request take so much longer then all further requests? What is promise.all and what does it do?

The first request is always the longest because once the browser gets the data it then caches various details and information to make further requests quicker.

promise.all executes multiple promises in tandem and waits for them all to complete before finishing. This is convenient and a lot faster then running them one at a time. in addition it has a fail fast feature where if one promise fails the entire list fails allowing you to fail early.

18. Refer to the js repo and observe the code in "timeouts".

setTimeout always adds your handler onto the event queue, if it's 0 seconds it does right on there but wither longer delays it'll wait before putting it on. Regardless you have to wait for the current handler to finish before you can handle the next one on the queue.

19. Discuss with your tutor any questions about the assignment

Remember that the api docs that are live at localhost:5000 are incredibly useful and should be a first point of contact for any backend related questions.

also pull often because fixes are going out every now and then for little bugs and issues in the backend

Revision questions

The remaining tutorial questions are primarily intended for revision - either this week or later in session. Your tutor may still choose to cover some of the questions time permitting.

20. Take the remaining time in the tute and see if you can build a a simple version of [regexr](#). Simply have a textbox in which users can type text and a input field in which users can insert a regular expression. Then highlight any matching text in the textbox live. (no boiler plate code is provided this time, see if you can set up a little work space on your own!)

COMP[29]041 18s2: Software Construction is brought to you by
the [School of Computer Science and Engineering](#) at the [University of New South Wales](#), Sydney.

For all enquiries, please email the class account at cs2041@cse.unsw.edu.au

CRICOS Provider 00098G