


COMP1531 | T09B / H17B

Week 1

william.huynh3@unsw.edu.au

⋮	Console	What's New
⊘	top	Filter
>	$2 + 2$	
<	4	
>	$"2" + "2"$	
<	"22"	
>	$2 + 2 - 2$	
<	2	
>	$"2" + "2" - "2"$	
<	20	



UNSW
SYDNEY

Learning objectives

- Getting to know your groups + group activity !
- Going from C to JavaScript
(printing, variables, arrays, loops, functions)
- Quick Admin stuff



Admin: How the course works

Lectures	-> Youtube
Help Sessions	-> Teams
Group Assignment	-> Teams
Labs & Project	-> Gitlab
Forum	-> Ed

All of the above -> WebCMS



Admin: Attendance & Participation

Tutorial Attendance recorded starting **Week 2**

Tutorial **participation** and **attendance** contributes to your **project mark**

Low Attendance	Scaling Down
Regular Attendance	No Scaling
Great Attendance + Participation	Scaling Up

T09B Teams

5316642	Amy Clarke	T09B_BOOST
5387884	Andy Ng	T09B_BOOST
5271790	Meyliana Ng	T09B_BOOST
5420321	Daniel Chen	T09B_BOOST
5356417	Yuichiro Nakahira	T09B_BOOST
5419773	Ronald Chiang	T09B_CRUNCHIE
5419787	Brian Ngo	T09B_CRUNCHIE
5417461	Denton Nguyen	T09B_CRUNCHIE
5207959	Daniel Fan	T09B_CRUNCHIE
5404557	Sanskar Sahu	T09B_CRUNCHIE
5363946	Haadiya Mustafa	T09B_DREAM
5376765	Samuel Lim	T09B_DREAM
5421256	Sunny Chen	T09B_DREAM
5417001	Ansh Gulati	T09B_DREAM
5418955	Aedan Keldoulis	T09B_DREAM
5390758	Advaita Nitturkar	T09B_EGGS
5421946	Benjamin Stacey	T09B_EGGS
5425354	Jeyanth Goringe	T09B_EGGS
5417310	Rishi Wig	T09B_EGGS
5413747	Josh Fotheringham	T09B_EGGS

Get to know your Team!

- 🙋 Introduce yourselves !
- ➡️ Exchange contact details + comms
(Discord, Facebook, MS Teams, etc...)
- 100 Set your expectations
(HD / D / C / P)

Groups will be finalised **Friday night, 16th September**

H17B Teams

5416492	Samuel Lim	H17B_AERO
5416493	Lachlan Liu	H17B_AERO
5417542	Jonathan Lai	H17B_AERO
5406774	Nuren Alahee	H17B_AERO
5316690	Helen Song	H17B_AERO
5418326	Daniel Shi	H17B_BOOST
5417290	James Lu	H17B_BOOST
5419389	Jake Song	H17B_BOOST
5421898	Dulini Galhena	H17B_BOOST
5111130	Jingqi Wang	H17B_CRUNCHIE
5419375	Jeffrey Lu	H17B_CRUNCHIE
5413842	Robert Han	H17B_CRUNCHIE
5419703	Zachary Ecob	H17B_CRUNCHIE
5258425	Danny Dien	H17B_CRUNCHIE
5412365	Luqing Wang	H17B_DREAM
5397747	Cho Sun	H17B_DREAM
5401098	Xiaoyu Su	H17B_DREAM
5428082	Huu Phuc Tran	H17B_DREAM
5420097	Morgan Au	H17B_DREAM
5265211	Sebastian Williams	H17B_EGGS
5310467	Sam Mears	H17B_EGGS
5397993	Byron Petselis	H17B_EGGS
5421468	Kalaish Stanley	H17B_EGGS
5363988	Andrew Ha	H17B_EGGS

Get to know your Team!

- 🙋 Introduce yourselves !
- ➡️📱 Exchange contact details + comms
(Discord, Facebook, MS Teams, etc...)
- 100 Set your expectations
(HD / D / C / P)

Groups will be finalised **Friday night, 16th September**



Team Problem Solving



How many hours of **Zoom calls** happen in Sydney **everyday**?

Come up with an **exact** number



Tutorial Activities - Getting Started

COMP1531 > 22T3 > tutorials > Repository

master

tutorials / tut01 / README.md

Find file

Blame

History

Permalink



updated tut01

Khiet Tam Nguyen authored 10 hours ago

63090ac8



Click 'tutorials' !



M*

README.md



3.66 KiB



Open in Web IDE



Replace

Delete



Tutorial 1

- Tutorial 1
 - General notes:
 - A. Introduction and Problem Solving
 - B. From C to Javascript: A quick overview
 - Part 1 - Basic Syntax, Input/Output, Conditionals & Control Flow
 - Part 2 - Looping through Arrays
 - Part 3 - Functions
 - C. Git Fundamentals
 - Further JavaScript Practice



Tutorial Activities - Getting Started

1. Click 'Clone'



2. Under 'Clone with SSH', Click the clipboard icon



COMP1531 > 22T3 > tutorials > Repository

master

tutorials /

+

History

Find file

Web IDE

↓

Clone



mass updated tutorials

Khiet Tam Nguyen authored 10 hours ago

Name

Last commit

tut01

mass updated tutorials

tut02

mass updated tutorials

tut03

mass updated tutorials

tut04

mass updated tutorials

tut05

mass updated tutorials

Clone with SSH

gitlab@gitlab.cse.unsw.EDU.AU:COMP



Clone with HTTPS

https://gitlab.cse.unsw.EDU.AU/COM



Open in your IDE

Visual Studio Code (SSH)

Visual Studio Code (HTTPS)

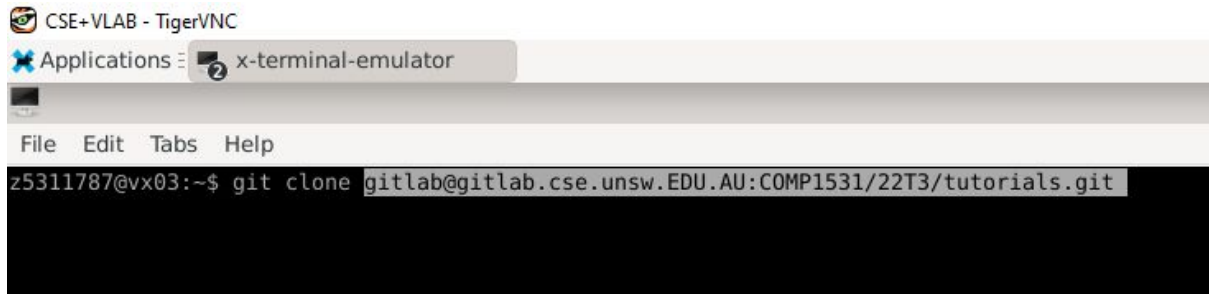
IntelliJ IDEA (SSH)

IntelliJ IDEA (HTTPS)

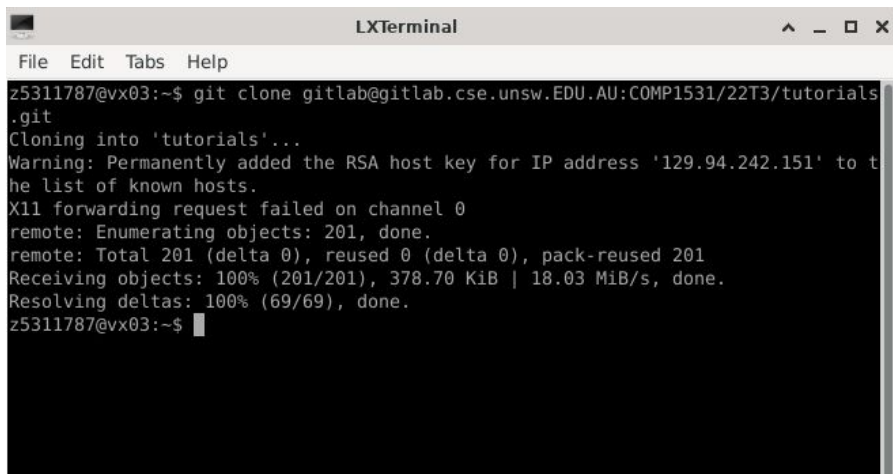
10 hours ago



Tutorial Activities - Getting Started



```
CSE+VLAB - TigerVNC
Applications ▾ x-terminal-emulator
File Edit Tabs Help
z5311787@vx03:~$ git clone gitlab@gitlab.cse.unsw.EDU.AU:COMP1531/22T3/tutorials.git
```



```
LXTerminal
File Edit Tabs Help
z5311787@vx03:~$ git clone gitlab@gitlab.cse.unsw.EDU.AU:COMP1531/22T3/tutorials
.git
Cloning into 'tutorials'...
Warning: Permanently added the RSA host key for IP address '129.94.242.151' to t
he list of known hosts.
X11 forwarding request failed on channel 0
remote: Enumerating objects: 201, done.
remote: Total 201 (delta 0), reused 0 (delta 0), pack-reused 201
Receiving objects: 100% (201/201), 378.70 KiB | 18.03 MiB/s, done.
Resolving deltas: 100% (69/69), done.
z5311787@vx03:~$
```

Take a look at `welcome.c`

How can we convert this `C` code to JavaScript?

Create a new file called `welcome.js`

To run the program:

`node welcome.js`

```
1  /*  
2   * [welcome.c]  
3   *  
4   * A simple program to print a welcome message  
5   * and print 10 numbers with information about  
6   * whether they are odd or even.  
7   */  
8  #include <stdio.h>  
9  
10 #define SIZE 10  
11  
12 int main(void) {  
13  
14     char message[] = "Welcome to COMP1531!";  
15     printf("%s\n", message);  
16  
17     printf("Numbers from 1 to %d\n", SIZE);  
18     for (int num = 1; num <= SIZE; num++) {  
19         if (num % 2 == 0) {  
20             printf("EVEN: %d\n", num);  
21         } else {  
22             printf("ODD: %d\n", num);  
23         }  
24     }  
25     return 0;  
26 }  
27
```

```

1  ✓ /*
2      * [welcome.c]
3      *
4      * A simple program to print a welcome message
5      * and print 10 numbers with information about
6      * whether they are odd or even.
7      */
8  #include <stdio.h>
9
10 #define SIZE 10
11
12 ✓ int main(void) {
13
14     char message[] = "Welcome to COMP1531!";
15     printf("%s\n", message);
16
17     printf("Numbers from 1 to %d\n", SIZE);
18     for (int num = 1; num <= SIZE; num++) {
19         if (num % 2 == 0) {
20             printf("EVEN: %d\n", num);
21         } else {
22             printf("ODD: %d\n", num);
23         }
24     }
25     return 0;
26 }
27

```

```

1  ✓ /*
2      * [welcome.js]
3      *
4      * A simple program to print a welcome message
5      * and print 10 numbers with information about
6      * whether they are odd or even.
7      */
8
9  ✓ // General rule: declare variables as constant, if you need
10 // to change the value later then change 'const' to 'let'
11 const SIZE = 10;
12
13 // Can use single quotes
14 const message = 'Welcome to COMP1531!';
15 // Don't need '\n'
16 console.log(message);
17
18 console.log('Numbers from 1 to ' + SIZE);
19
20 ✓ for (let num = 1; num <= SIZE; num++) {
21     // Triple equals (===) instead of double (==)
22     ✓ if (num % 2 === 0) {
23         console.log('EVEN: %d', num);
24     } else {
25         console.log('ODD: %d', num);
26     }
27 }

```



To summarise!

`const` → For variables that will **not change** (*immutable*)

`let` → For variables that will **change** (*mutable*)

Always use **const** unless you need **let**

`var` → **Never use this !!!**

`printf` → Doesn't automatically add `'\n'`

`console.log` → Automatically adds `'\n'`



Time to write some JavaScript!

1. Create a new file called `shopping.js`
2. Create an array of some `grocery` items!
3. Find **as many ways** of **looping through** the array as possible!
(Hint: there's at least 3 ways of looping through an array. Try to find them all!)

```
1  ✓ const shoppingList = [
2    |   'potato', 'ketchup', 'milk',
3    |   'eggs', 'flour'
4  ];
5
6  /**
7   * C-style
8   */
9  ✓ for (let i = 0; i < shoppingList.length; i++) {
10 |   console.log(shoppingList[i]);
11 | }
12
13 /**
14  * for-in
15  */
16  ✓ for (const i in shoppingList) {
17 |   console.log(shoppingList[i]);
18 | }
19
20 /**
21  * for-of
22  */
23  ✓ for (const element of shoppingList) {
24 |   console.log(element);
25 | }
```

← Uses integer based index

← Uses integer based index (*but less disgusting*)

← The best way to do loops
For each **element** in the **shoppingList**

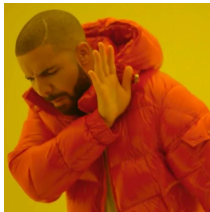
✓ To summarise!

We declare arrays like so:

```
1  ✓ const shoppingList = [  
2    'potato', 'ketchup', 'milk',  
3    'eggs', 'flour'  
4  ];
```

Always use **for...of** loops !

(Unless you really need an index loop)



```
6  /**  
7   * C-style  
8   */  
9  ✓ for (let i = 0; i < shoppingList.length; i++) {  
10 |   console.log(shoppingList[i]);  
11 | }  
20 /**  
21  * for-of  
22  */  
23 ✓ for (const element of shoppingList) {  
24 |   console.log(element);  
25 | }
```


Take a look at `cube.c`

```
1  /**
2   * [cube.c]
3   */
4
5  #include <stdio.h>
6
7  int cube(int x);
8
9  int main(void) {
10
11     int number = 5;
12     int result = cube(number);
13     printf("%d^3 = %d\n", number, result);
14
15     return 0;
16 }
17
18 int cube(int x) {
19     return x * x * x;
20 }
```

How can we convert this **C** code to JavaScript?

Create a new file called `cube.js`

We don't have function declarations in JavaScript

```
1  /**
2   * [cube.c]
3   */
4
5  #include <stdio.h>
6
7  int cube(int x);
8
9  int main(void) {
10
11     int number = 5;
12     int result = cube(number);
13     printf("%d^3 = %d\n", number, result);
14
15     return 0;
16 }
17
18 int cube(int x) {
19     return x * x * x;
20 }
```

```
1  function cube(x) {
2      return x ** 3;
3  }
4
5  const number = 5;
6  const result = cube(number);
7
8  console.log(`${number}^3 = ${result}`);
```

✓ To summarise!

`function` → The keyword we use to implement functions in JavaScript

```
1  function hello() {  
2    |  console.log('hello!')  
3  }
```

~~`int main (void)`~~ → There are no main functions in JavaScript !

```
int main(void) {  
  printf("hi\n");  
  return 0;  
}
```



```
1  
2  
3  console.log('hi');  
4  
5
```



Labs

Our lab assistant is Cam (*please say hi!*)

Will be 2 hours time where you can work on the labs

Later in the term, will be used for project check-ins and project help !

Lab attendance is not mandatory, unless there is a project check-in planned for that lab (*will be announced*)