CS213: Software Systems Laboratory

Autumn 2023-24

Koteswararao Kondepu

k.kondepu@iitdh.ac.in

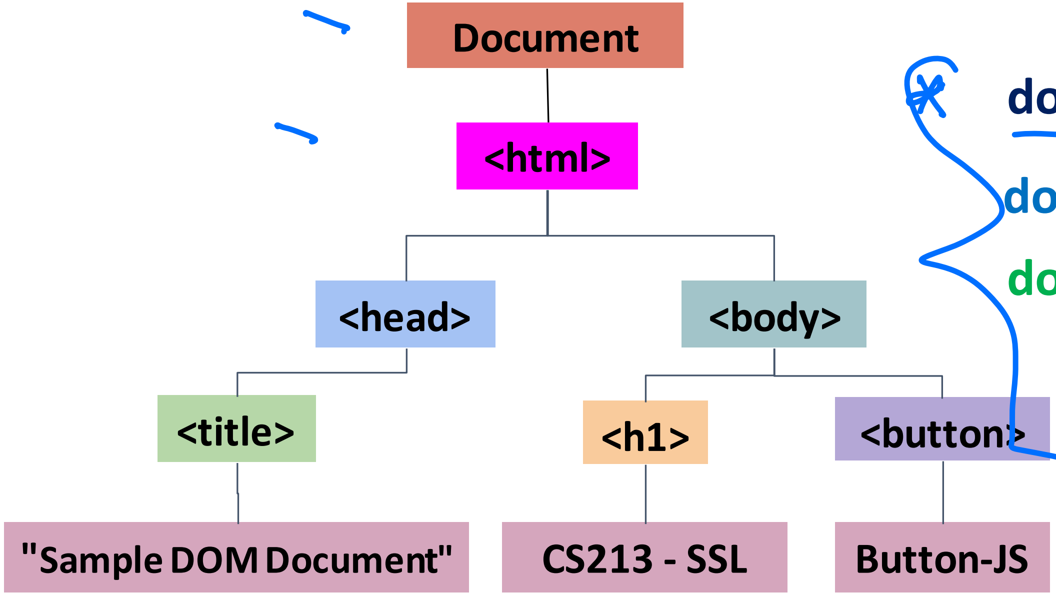
Recap

JavaScript

- ○ Introduction Java Script (JS)
- ○ JavaScript Linking ✓
- ○ JavaScript Document Object Model (DOM)
- ○ JavaScript DOM Tree ✓
- ○ JavaScript DOM Nodes/Objects

- Interactive/dynamic content

dynamically access and update the content, structure, and style of a document



```
document.getElementById(id); ✓
document.getElementsByTagName(tagName); ✓
document.getElementsByClassName(className);
document.querySelector(cssQuery);
document.querySelectorAll(cssQuery);
```

Outline

- ○ DOM Event – Remove
- ○ DOM Event – Propagation
- ○ DOM Event – ~~Propagation~~ Delegation ✓

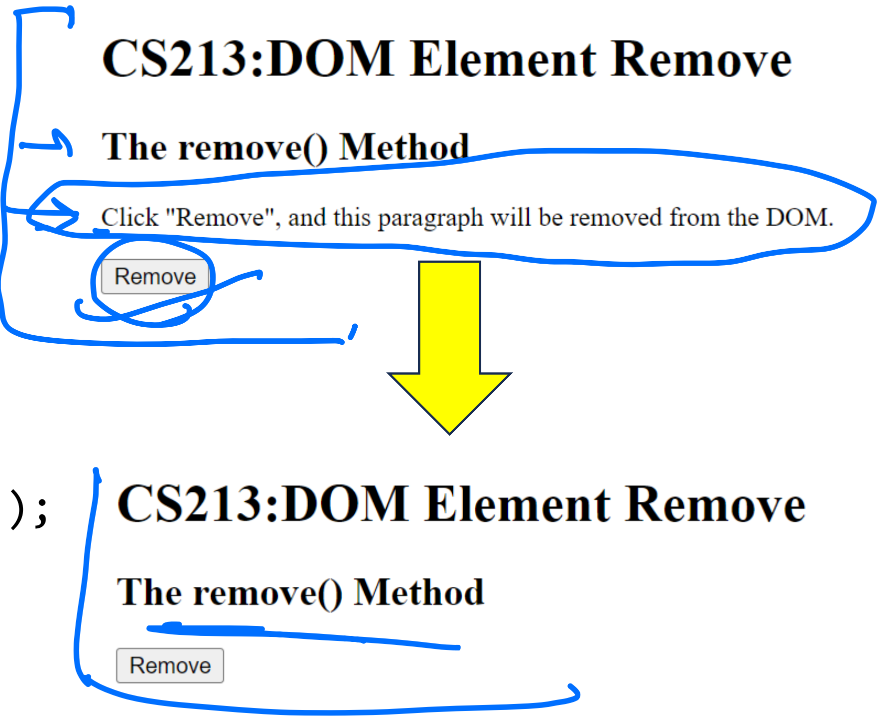
*

- ○ MySQL ✓
- ○ Version Control ✓
- ○ Stats of CS213

DOM Element - Remove

- Remove an element from the document.
- The remove() method removes an element (or node) from the document.

```html
<!DOCTYPE html>
<html>
<body>
<h1>CS213:DOM Element Remove </h1>
<h2>The remove() Method</h2>
<p id="demo">Click "Remove", and this paragraph
will be removed from the DOM.</p>
<button onclick="myFunction()">Remove</button>
<script>
function myFunction() {
 const element = document.getElementById("demo");
 element.remove();
}
</script>
</body>
</html>
```



CS213:DOM Element Remove

The remove() Method

Click "Remove", and this paragraph will be removed from the DOM.

CS213:DOM Element Remove

The remove() Method

DOM Element - Propagation

- When an element receives an event, and that event propagates or bubbles up to its parent and ancestor elements in the DOM tree until it gets to the root element.

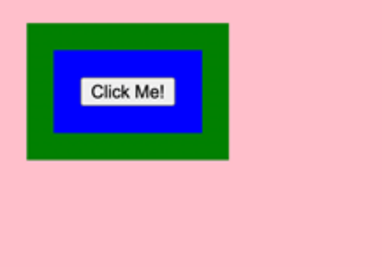
HTML

```html
<body>
 <div>
  <span>
   <button>Click Me!</button>
  </span>
 </div>
</body>
```

<html>

CSS

```css
body {
  padding: 20px;
  background-color: pink;
}

div {
  padding: 20px;
  background-color: green;
  width: max-content;
}

span {
  display: block;
  padding: 20px;
  background-color: blue;
}
```



output

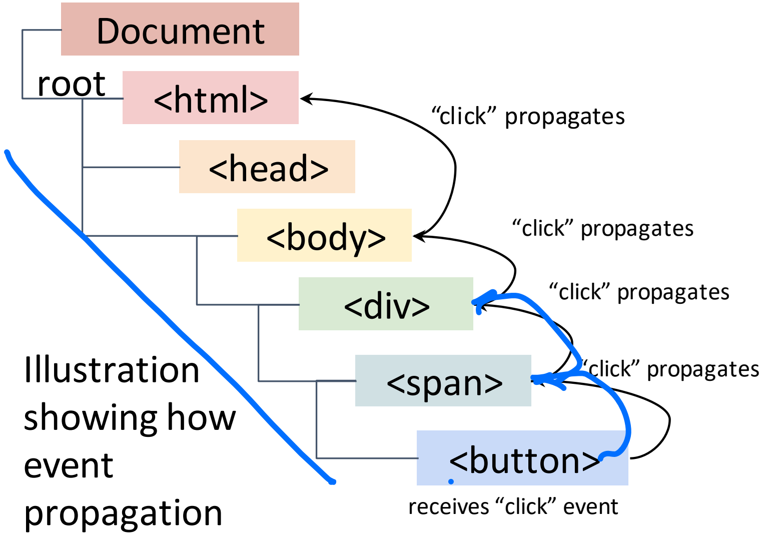


Illustration showing how event propagation works

DOM Element - Propagation

```
const body = document.getElementsByTagName("body")[0]
const div = document.getElementsByTagName('div')[0]
const span = document.getElementsByTagName("span")[0]
const button = document.getElementsByTagName("button")[0]

body.addEventListener('click', () => {
  console.log("body was clicked")
})
div.addEventListener('click', () => {
    console.log("div was clicked")
})
span.addEventListener('click', () => {
  console.log("span was clicked")
})
button.addEventListener('click', () => {
  console.log("button was clicked")
})
```

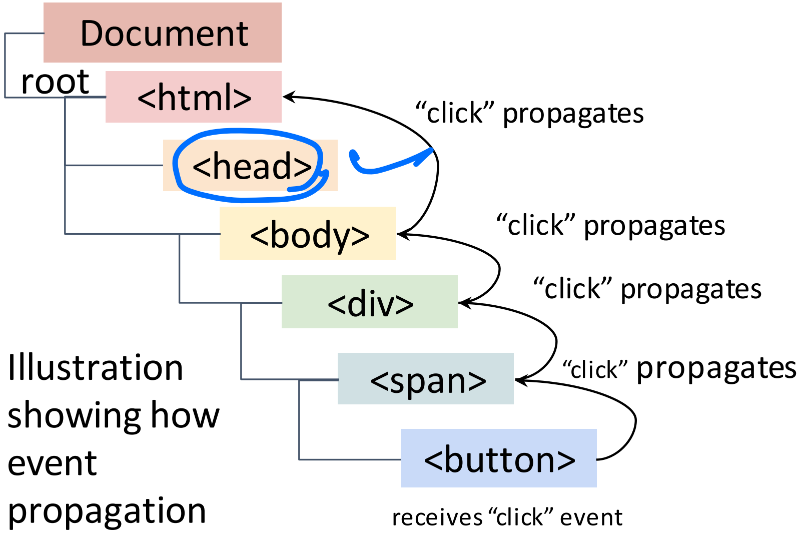
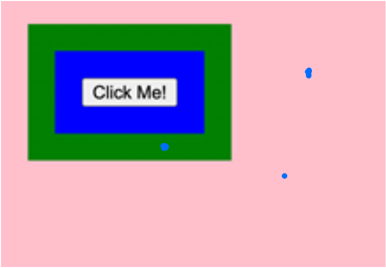


Illustration showing how event propagation works



click on the pink background (i.e., body)

🟥 output

```
body was clicked
```

click on the green background (i.e., div)

🟩 output

```
div was clicked
```

Note: To prevent event propagation, you use the **stopPropagation** method of the event object.

DOM Element - Delegation

- Delegates the handling of an event to a different element instead of the actual element that received the event.
- If we have a lot of elements handled in a similar way, then instead of assigning a handler to each of them – we put a single handler on their common ancestor.

```
const div =
document.getElementsByTagName("div")[0]
div.addEventListener("click", (event) => {
  if(event.target.tagName === 'BUTTON') {
    console.log("button was clicked")
  }
})
```

Note: With this code, when you click the button, the event bubbles up to the div which handles the event.

```
<div>
  <button>Button 1</button>
  <button>Button 2</button>
  <button>Button 3</button>
</div>
const buttons = document.querySelectorAll('button')
buttons.forEach(button => {
  button.addEventListener("click", (event) => {
    console.log(event.target.innerText)
  })
})
```

Note: querySelectorAll returns a **NodeList** which we can use the **forEach** method **getElementsByTagName** returns an HTMLCollection which doesn't have the forEach method.

My Structured Query Language (MySQL)

- A database system used for developing web-based software applications.
- A relational database management system (RDBMS).
- Supports standard SQL (Structured Query Language)
- Free to download and use.
- Written in C, C++.

Features

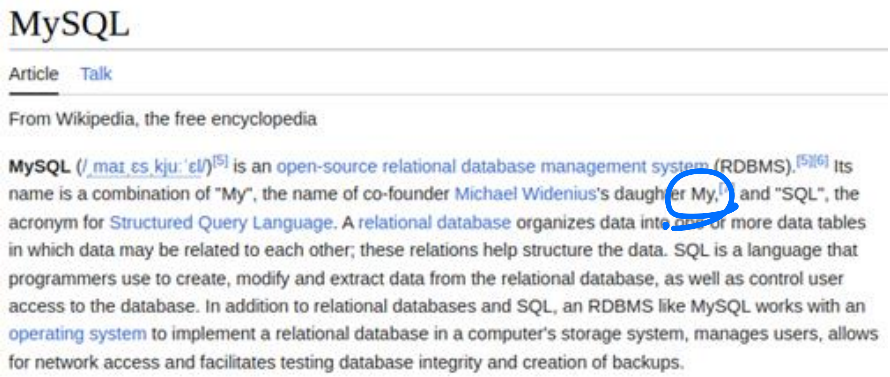
- ❑ Portability — Can be installed and run on any type of Hardware and OS.
- ❑ Security — Creates a secured database
- ❑ Connectivity — Connects various types of Network client using different protocols.

Installation

- ❑ For Windows — [Tutorial Link]
- ❑ For Ubuntu — [Tutorial Link]

MySQL Tutorial

- ❑ [Link]

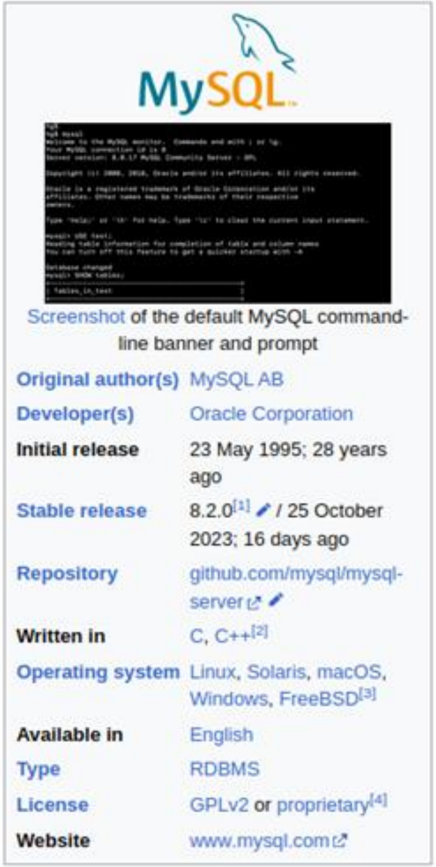


MySQL

Article Talk

From Wikipedia, the free encyclopedia

MySQL (/ˌmaɪ ˌɛsˌkjuːˈɛl/)[5] is an open-source relational database management system (RDBMS).[5][6] Its name is a combination of "My", the name of co-founder Michael Widenius's daughter My,[7] and "SQL", the acronym for Structured Query Language. A relational database organizes data into one or more data tables in which data may be related to each other; these relations help structure the data. SQL is a language that programmers use to create, modify and extract data from the relational database, as well as control user access to the database. In addition to relational databases and SQL, an RDBMS like MySQL works with an operating system to implement a relational database in a computer's storage system, manages users, allows for network access and facilitates testing database integrity and creation of backups.



Screenshot of the default MySQL command-line banner and prompt

Original author(s)	MySQL AB
Developer(s)	Oracle Corporation
Initial release	23 May 1995; 28 years ago
Stable release	8.2.0[1] / 25 October 2023; 16 days ago
Repository	github.com/mysql/mysql-server
Written in	C, C++[2]
Operating system	Linux, Solaris, macOS, Windows, FreeBSD[3]
Available in	English
Type	RDBMS
License	GPLv2 or proprietary[4]
Website	www.mysql.com

source: https://en.wikipedia.org/wiki/MySQL

MySQL (1)

SQL	MySQL
Database language for storing, manipulating and retrieving data in a relational database	Open source Relational Database Management System (RDBMS) that allows managing relational databases.
A database language	A software
Helps to manage the data in the relational database	Helps to manage relational databases using SQL
Data Definition Language (DDL) — To create, alter, and delete database objects like table, views, index etc.	CREATE, ALTER, DROP, etc.
Data Manipulation Language (DML) — To insert, delete, update, and retrieve the stored records from the table.	SELECT …, INSERT …, DELETE …, UPDATE …, etc.
Transaction Control Language (TCL) — Used to control the transaction.	COMMIT, ROLLBACK, SAVEPOINT, etc.
Data Control Language (DCL) — To manipulate permission or access rights to the table etc.	GRANT, REVOKE, etc.

MySQL : Data Types

Numeric Data Types

- ❑ INTEGER or INT — Upto 11 digit number without decimal
- ❑ SMALLINT — Upto 5 digit number without decimal
- ❑ FLOAT (M,D)
- ❑ DECIMAL (M,D)
- ❑ NUMERIC (M,D)

Real number with digit length (M) with decimal places (D)

Date & Time Data Types

- ❑ DATE — Stores date in YYYY-MM-DD format
- ❑ TIME — Store time in HH:MM:SS format

String Data Type

- ❑ CHAR (size) — Fixed length string upto 255 characters.
- ❑ VARCHAR (size) — A variable length string up to 255 characters.

Note: Char, Varchar, Date and Time values should be enclosed with single (‘’) or double (“”) quotes.

mysql>

MySQL – Database handling

- **Creating a Database**

create database <DB-NAME>;

- **Getting List of Databases**

show databases;

- **Selecting a Database**

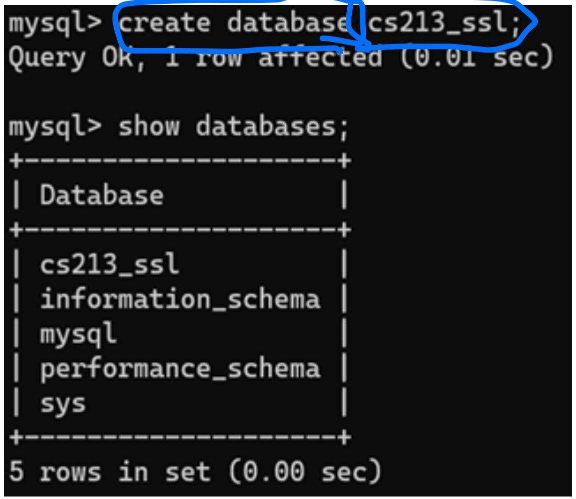
use <DB-NAME>;

use cs213_ssl

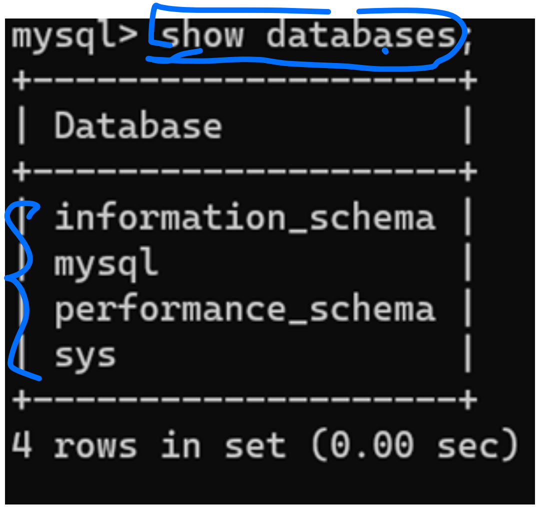
- **Deleting a Databases**

drop database <DB-NAME>;

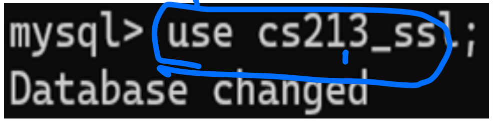
cs213_ssl



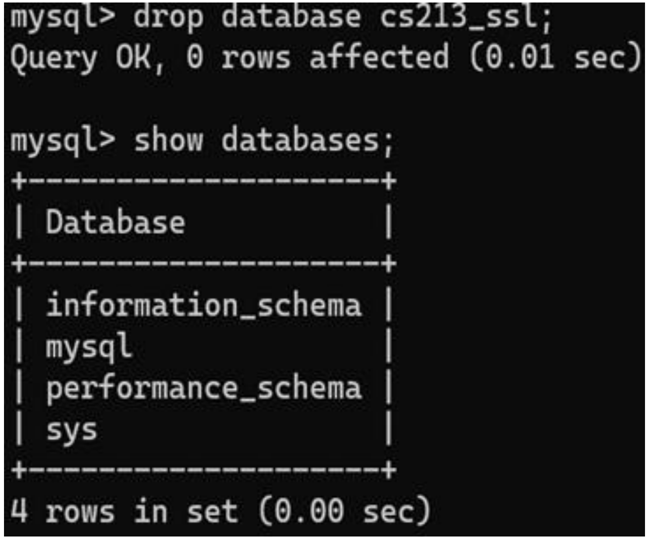
```
mysql> create database cs213_ssl;
Query OK, 1 row affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| cs213_ssl          |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0.00 sec)
```



```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)
```



```
mysql> use cs213_ssl;
Database changed
```

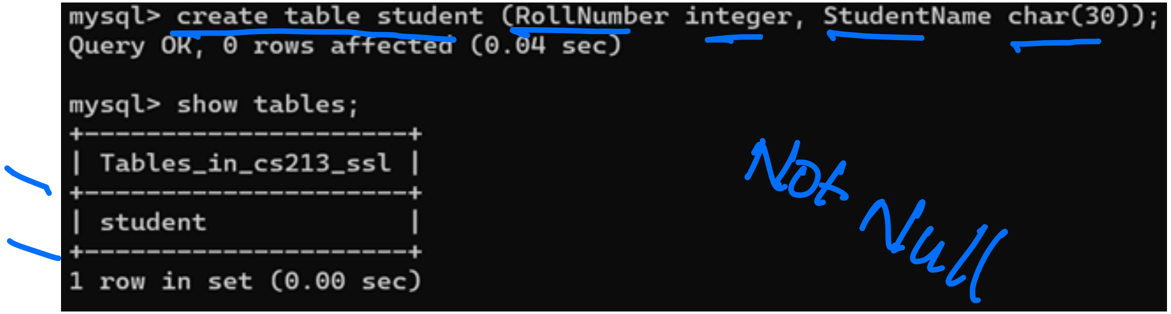


```
mysql> drop database cs213_ssl;
Query OK, 0 rows affected (0.01 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
4 rows in set (0.00 sec)
```

MySQL Table Handling

- **Creating a Table**

* **create table <TABLE-NAME> (<Col name1> <data type> [(size)], …);**



```
mysql> create table student (RollNumber integer, StudentName char(30));
Query OK, 0 rows affected (0.04 sec)

mysql> show tables;
+----------------------+
| Tables_in_cs213_ssl |
+----------------------+
| student              |
+----------------------+
1 row in set (0.00 sec)
```

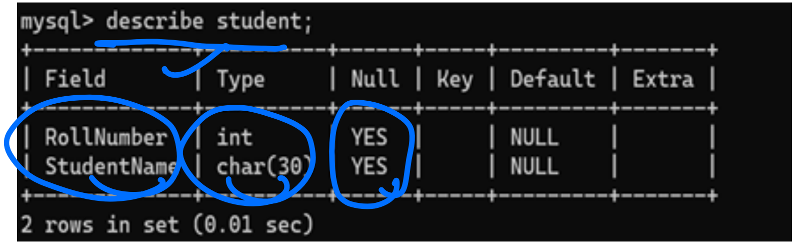
Not Null

- **Getting list of Tables**

show tables;

- **Deleting a Table**

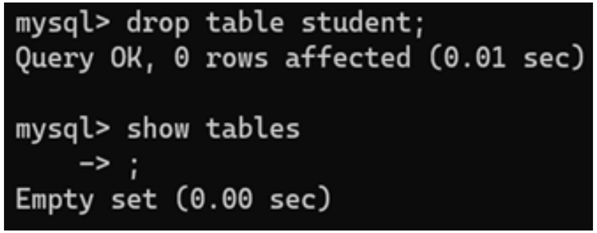
drop table <TABLE-NAME>;



```
mysql> describe student;
+-------------+----------+------+-----+---------+-------+
| Field       | Type     | Null | Key | Default | Extra |
+-------------+----------+------+-----+---------+-------+
| RollNumber  | int      | YES  |     | NULL    |       |
| StudentName | char(30) | YES  |     | NULL    |       |
+-------------+----------+------+-----+---------+-------+
2 rows in set (0.01 sec)
```

- **Viewing a Table Structure**

describe <TABLE-NAME>;

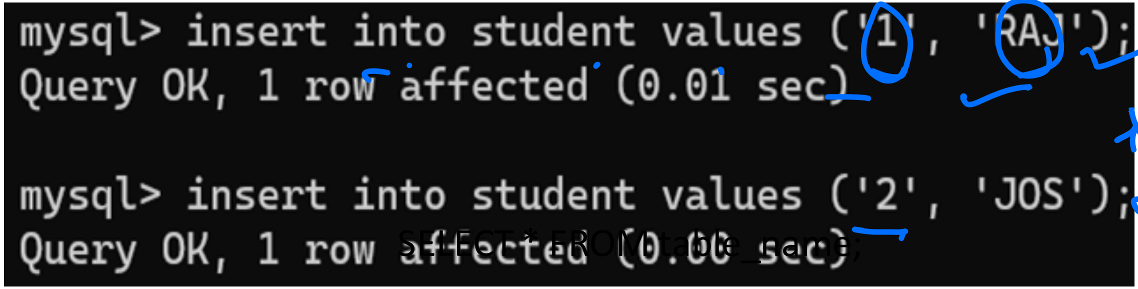


```
mysql> drop table student;
Query OK, 0 rows affected (0.01 sec)

mysql> show tables
    -> ;
Empty set (0.00 sec)
```

MySQL – Table Handling (1)

- **Inserting records into a table**

insert into <TABLE-NAME> VALUES (value1, value2, ….);

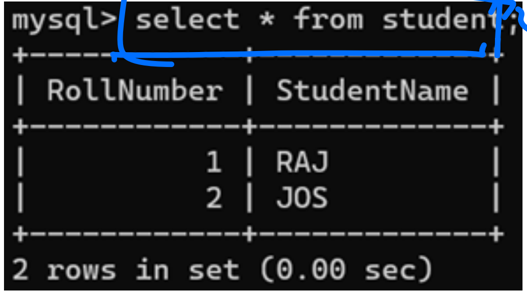


```
mysql> insert into student values ('1', 'RAJ');
Query OK, 1 row affected (0.01 sec)

mysql> insert into student values ('2', 'JOS');
Query OK, 1 row affected (0.00 sec)
```

SELECT * FROM table_name;

SELECT column1, column2, ... FROM table_name;

SELECT * FROM table_name;



```
mysql> select * from student;
+------------+-------------+
| RollNumber | StudentName |
+------------+-------------+
|          1 | RAJ         |
|          2 | JOS         |
+------------+-------------+
2 rows in set (0.00 sec)
```

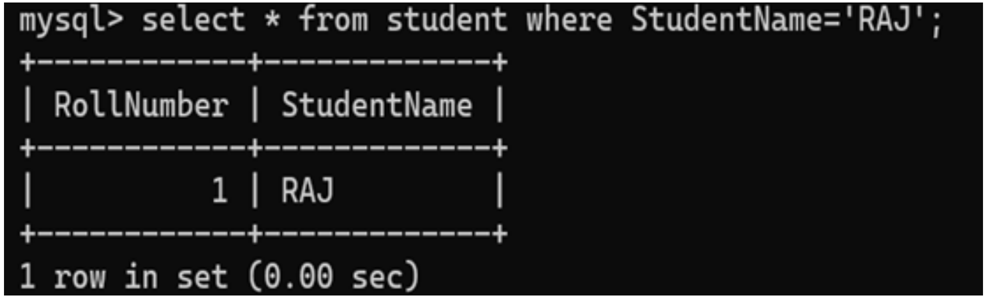
SELECT column1, column2, ...
FROM table_name
WHERE condition1 AND condition2
AND condition3 ...;

OR

Roll

WHERE column_name IS NULL;

WHERE column_name IS NOT NULL;



```
mysql> select * from student where StudentName='RAJ';
+------------+-------------+
| RollNumber | StudentName |
+------------+-------------+
|          1 | RAJ         |
+------------+-------------+
1 row in set (0.00 sec)
```

MySQL

```
SELECT MIN(column_name)/  MAX(column_name)/ AVG(column_name)/  SUM (column_name)
FROM table_name
WHERE condition;
```

```
SELECT MIN(column_name)
FROM table_name
WHERE condition;
```

```
SELECT MAX(column_name)
FROM table_name
WHERE condition;
```

MySQL - JOINS

- Used to combine rows from two or more tables, based on a related column between them
- **Types**
- INNER JOIN: Returns records that have matching values in both tables
- LEFT JOIN: Returns all records from the left table, and the matched records from the right table
- RIGHT JOIN: Returns all records from the right table, and the matched records from the left table
- CROSS JOIN: Returns all records from both tables

```
SELECT column_name(s)
FROM table1
INNER JOIN table2
ON table1.column_name = table2.column_name;
```

```
SELECT column_name(s)
FROM table1
LEFT JOIN table2
ON table1.column_name = table2.column_name;
```

```
SELECT column_name(s)
FROM table1
RIGHT JOIN table2
ON table1.column_name = table2.column_name;
```

```
SELECT column_name(s)
FROM table1
CROSS JOIN table2;
```

MySQL: Constraints

NOT NULL	Ensures that a column cannot have a NULL value
UNIQUE	Ensures that all values in a column are different
PRIMARY KEY	A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
FOREIGN KEY	Prevents actions that would destroy links between tables
CHECK	Ensures that the values in a column satisfies a specific condition
DEFAULT	Sets a default value for a column if no value is specified
CREATE INDEX	Used to create and retrieve data from the database very quickly

```
CREATE TABLE Persons (
  ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255) NOT NULL,
  Age int );
```

```
CREATE TABLE Persons (
  ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int,
  UNIQUE (ID));
```

```
CREATE TABLE Persons (
  ID int NOT NULL,
  LastName varchar(255) NOT NULL,
  FirstName varchar(255),
  Age int,
  PRIMARY KEY (ID) );
```

MySQL: Constraints

NOT NULL	Ensures that a column cannot have a NULL value
UNIQUE	Ensures that all values in a column are different
PRIMARY KEY	A combination of a NOT NULL and UNIQUE. Uniquely identifies each row in a table
FOREIGN KEY	Prevents actions that would destroy links between tables
CHECK	Ensures that the values in a column satisfies a specific condition
DEFAULT	Sets a default value for a column if no value is specified
CREATE INDEX	Used to create and retrieve data from the database very quickly

```
CREATE TABLE Orders (
    OrderID int NOT NULL,
    OrderNumber int NOT NULL,
    PersonID int,
    PRIMARY KEY (OrderID),
    FOREIGN KEY (PersonID) REFERENCES Persons(PersonID) );
```

```
ALTER TABLE Orders
ADD FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);
```

```
ALTER TABLE Orders
DROP FOREIGN KEY FK_PersonOrder;
```

Version Control

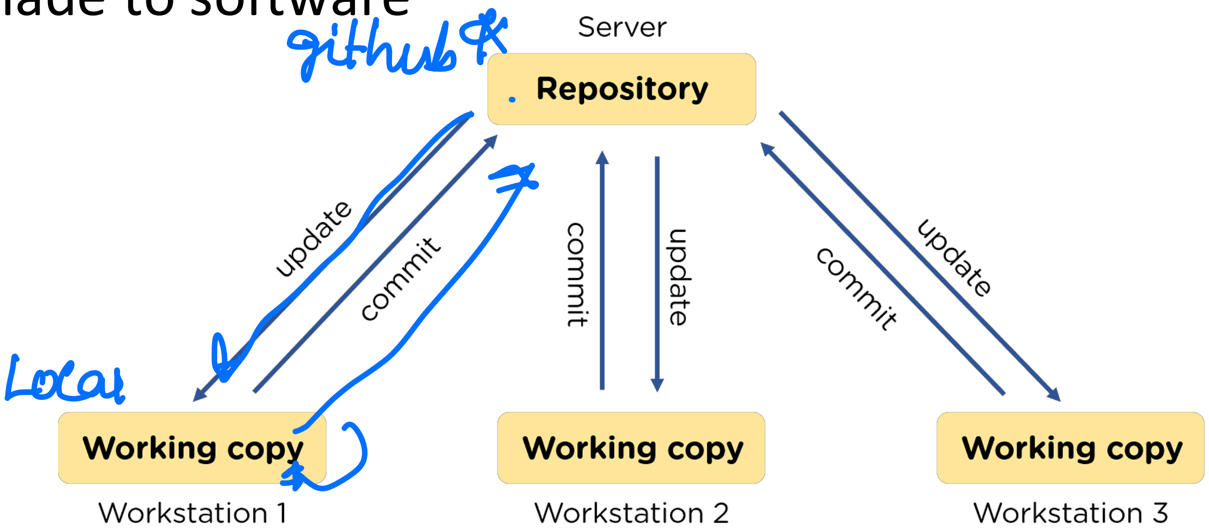
Version control, also known as source control, is the practice of tracking and managing changes to software code.

Benefits of version control systems

1. A complete long-term change history of every file.
2. Branching and merging : Helps team members work concurrently.
3. Traceability : Tracing changes made to software

git

mercurial



Git and GitHub

Overview

Git is a popular version control system. It was created by Linus Torvalds in 2005, and has been maintained by Junio Hamano since then.

What does Git do?

- Manage projects with ***Repositories***
- Clone a project to work on ***a local copy***
- Control and track changes with ***Staging and Committing***

GitHub is a web-based hosting service for git repositories

One can use git without Github,but you cannot use GitHub without Git

Git

Setup

Windows

Download the package from https://git-scm.com/download/win

Linux

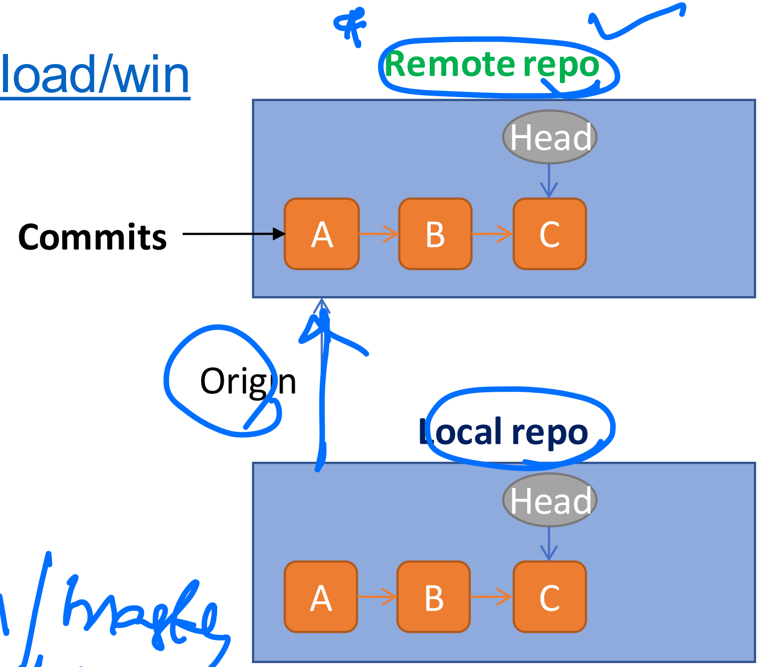
Use the command *sudo apt install git-all*

Use git –version to check the version

‘Repository’ – term used to represent storage

There are 2 types of Repositories

- Local – On our machine
- Remote – GitHub.com

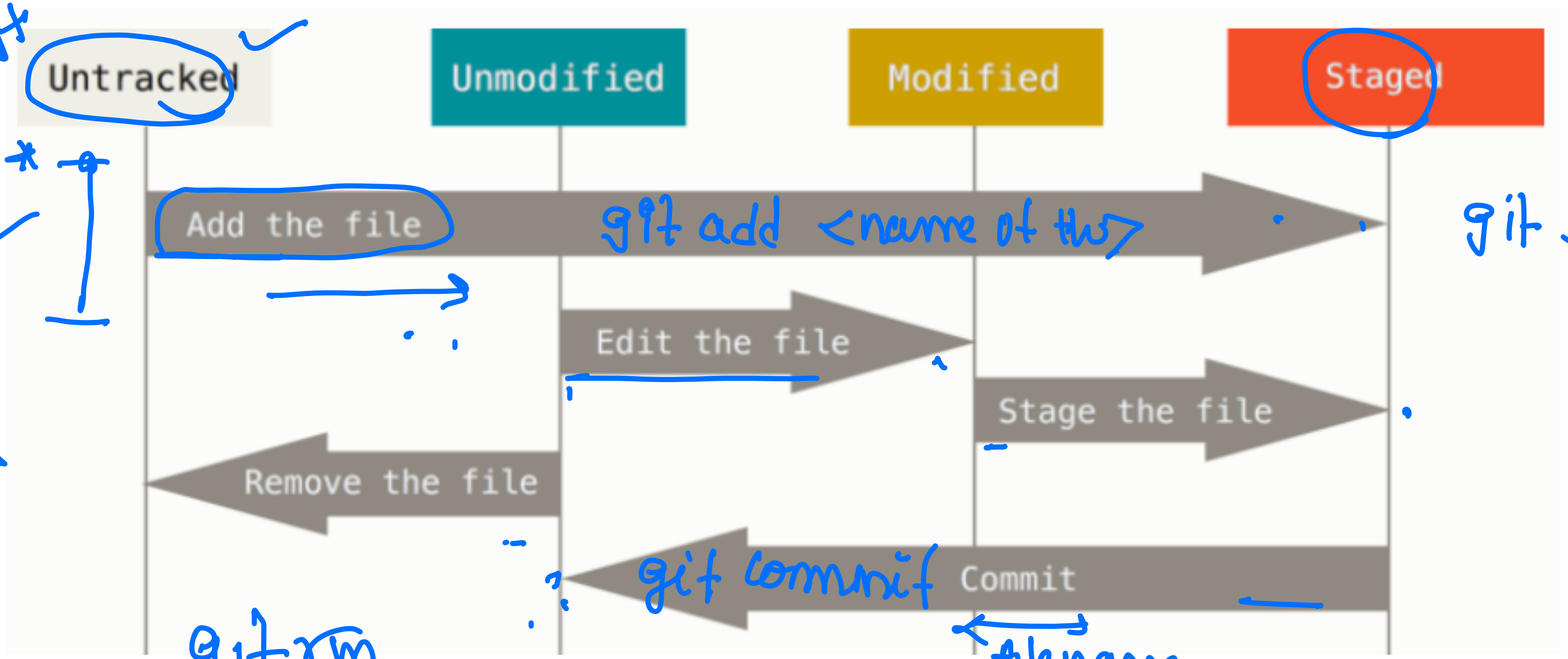


origin/master
origin/develo.

Git

Lifecycle of a File *

touch readme
git status



git status

git rm

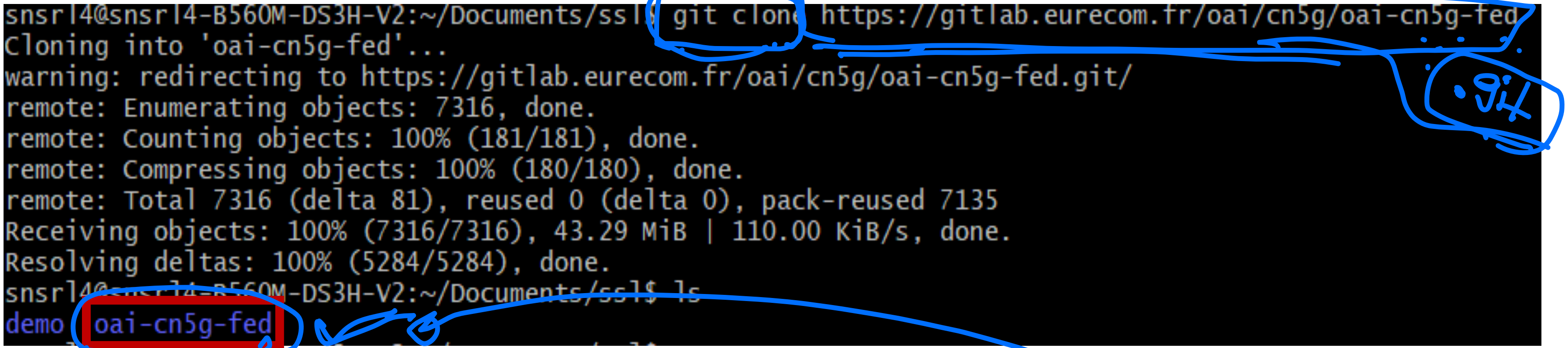
filename .

Git

Creating Repositories

- Cloning a Repository from a remote server to the local machine

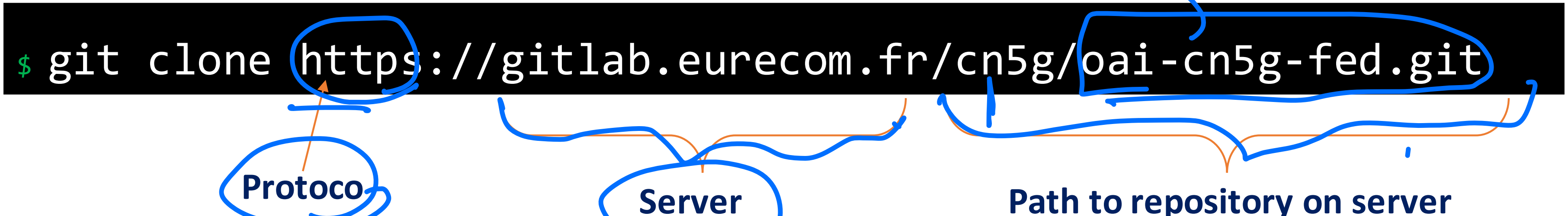
url



```
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl$ git clone https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed
Cloning into 'oai-cn5g-fed'...
warning: redirecting to https://gitlab.eurecom.fr/oai/cn5g/oai-cn5g-fed.git/
remote: Enumerating objects: 7316, done.
remote: Counting objects: 100% (181/181), done.
remote: Compressing objects: 100% (180/180), done.
remote: Total 7316 (delta 81), reused 0 (delta 0), pack-reused 7135
Receiving objects: 100% (7316/7316), 43.29 MiB | 110.00 KiB/s, done.
Resolving deltas: 100% (5284/5284), done.
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl$ ls
demo  oai-cn5g-fed
```

.git

Git clone is used to clone remote repositories from remote servers such as GitHub

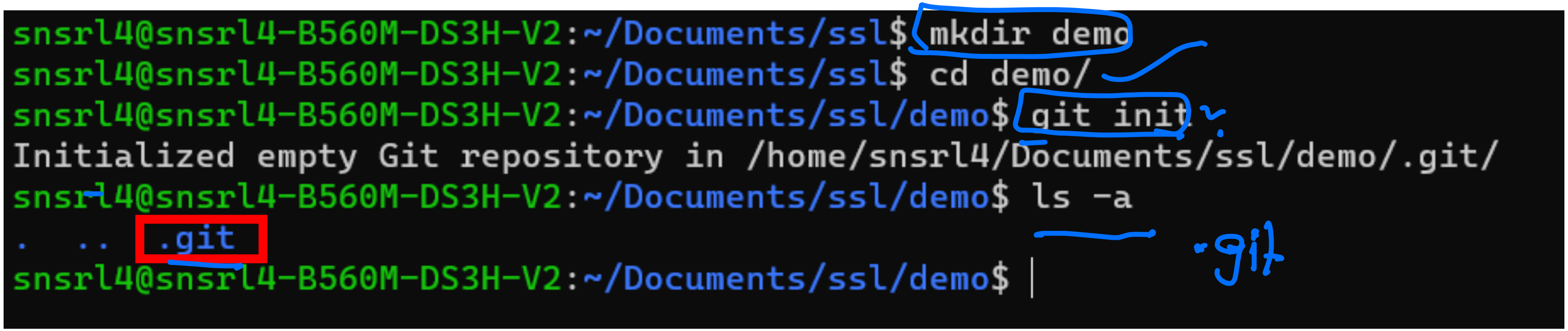


```
$ git clone https://gitlab.eurecom.fr/cn5g/oai-cn5g-fed.git
```

Git

Creating Repositories

2. git init for initializing a local repository



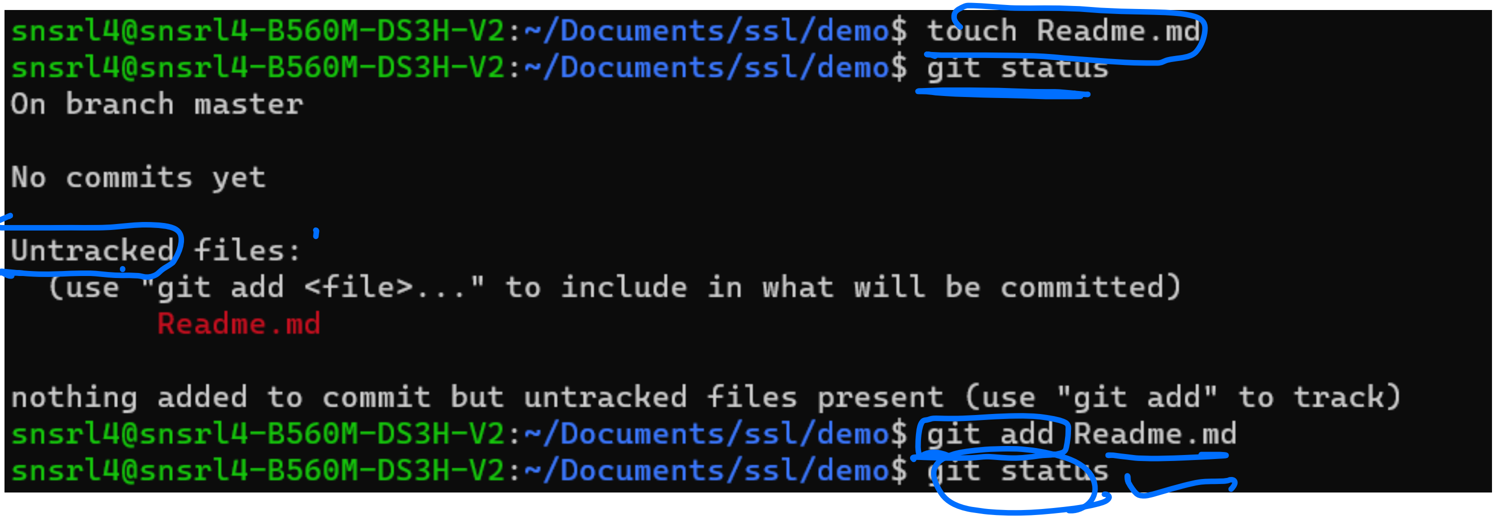
```
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl$ mkdir demo
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl$ cd demo/
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/demo$ git init
Initialized empty Git repository in /home/snsrl4/Documents/ssl/demo/.git/
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/demo$ ls -a
.  ..  .git
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/demo$ 
```

git init converts a directory to Git local repo

Git

Adding files to Repo

git add



```
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/demo$ touch Readme.md
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/demo$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
        Readme.md

nothing added to commit but untracked files present (use "git add" to track)
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/demo$ git add Readme.md
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/demo$ git status
```

git status shows the status of untracked files

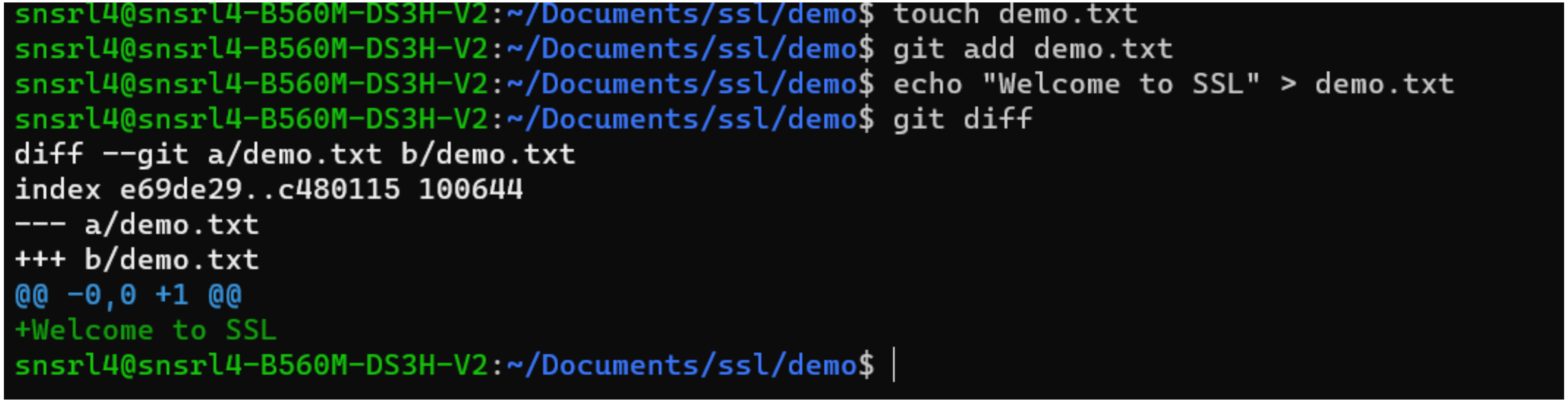
git add is used to add files to the repository

When changes are made to a added file, we need to add the file again using the command **git add**

Git

Checking the difference between the staged files and the current files

git diff *



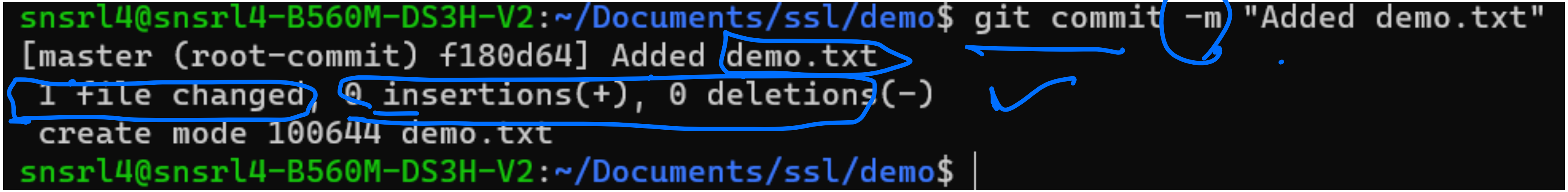
```
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/demo$ touch demo.txt
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/demo$ git add demo.txt
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/demo$ echo "Welcome to SSL" > demo.txt
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/demo$ git diff
diff --git a/demo.txt b/demo.txt
index e69de29..c480115 100644
--- a/demo.txt
+++ b/demo.txt
@@ -0,0 +1 @@
+Welcome to SSL
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/demo$
```

git diff shows the files that are modified from the last add.

Git

git commit for saving changes in local repository

git commit



```
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/demo$ git commit -m "Added demo.txt"
[master (root-commit) f180d64] Added demo.txt
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 demo.txt
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/demo$
```

‘commit’ changes i.e. save all the changes (adding a new file in this example) in the local repository

GitHub

Create a GitHub account, if you don’t have one.

Generate a SSH key on you local system and add the SSH key to your GitHub account

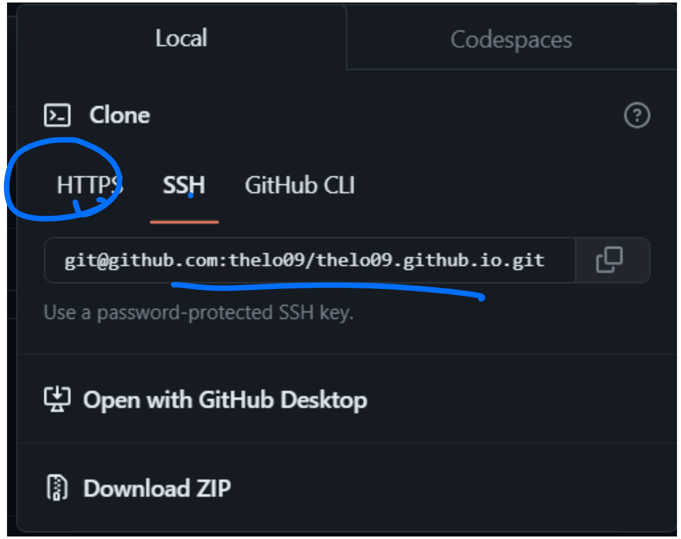
https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent

https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account

GitHub

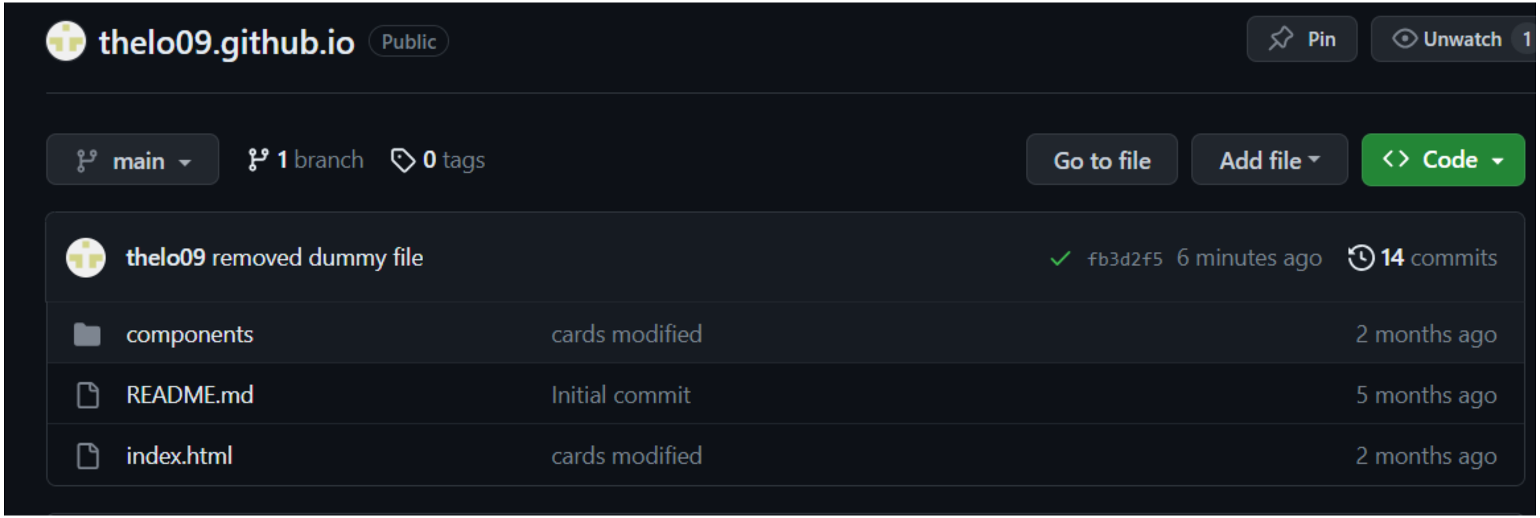
git pull ✓

- Git clone a repo
- Add a new file ✓
- Stage and commit the file to the repo
- Push the staged file
- Observe the changes



```
git@github.com:thelo09/thelo09.github.io.git
```

Use a password-protected SSH key.

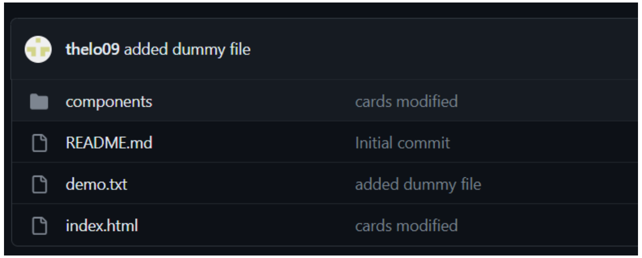


thelo09 removed dummy file ✓ fb3d2f5 6 minutes ago 14 commits

components	cards modified	2 months ago
README.md	Initial commit	5 months ago
index.html	cards modified	2 months ago

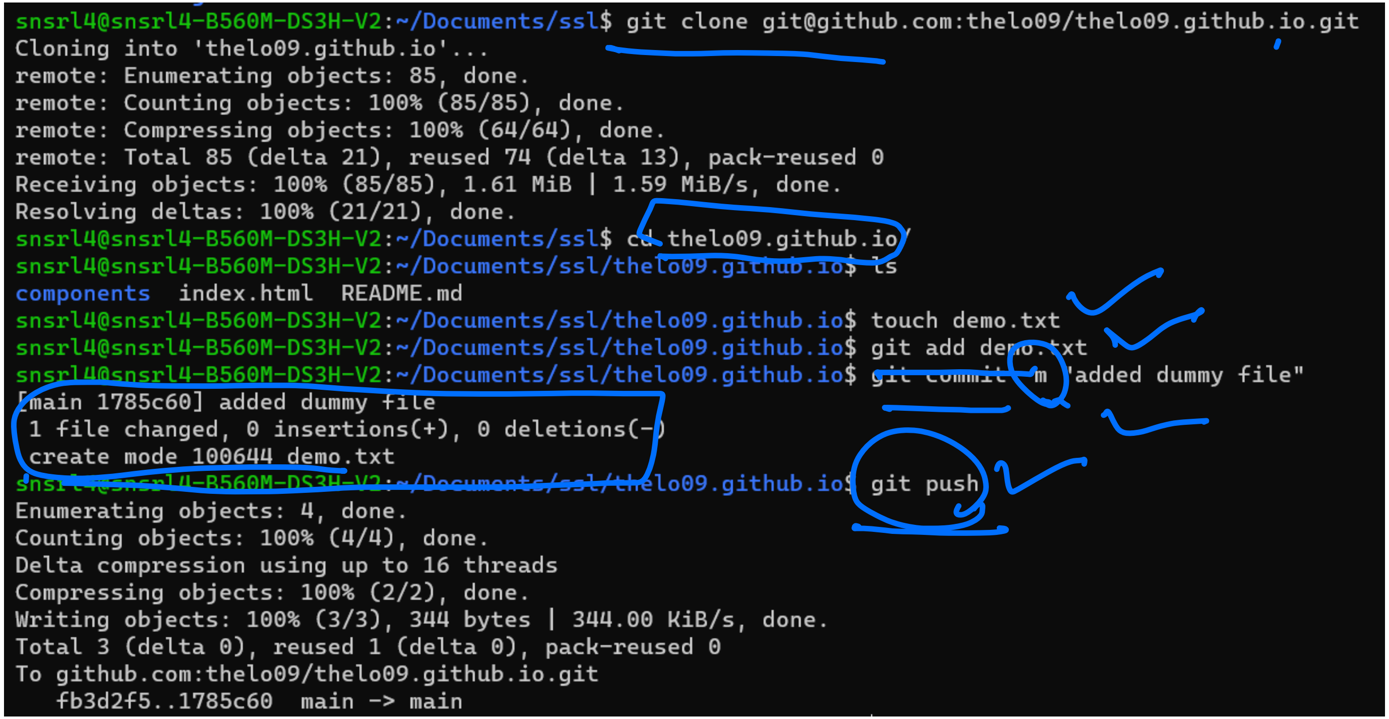
GitHub

- Git clone a repo
- Add a new file
- Stage and commit the file to the repo
- Push the staged file
- Observe the changes



thelo09 added dummy file

components	cards modified
README.md	Initial commit
demo.txt	added dummy file
index.html	cards modified



```
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl$ git clone git@github.com:thelo09/thelo09.github.io.git
Cloning into 'thelo09.github.io'...
remote: Enumerating objects: 85, done.
remote: Counting objects: 100% (85/85), done.
remote: Compressing objects: 100% (64/64), done.
remote: Total 85 (delta 21), reused 74 (delta 13), pack-reused 0
Receiving objects: 100% (85/85), 1.61 MiB | 1.59 MiB/s, done.
Resolving deltas: 100% (21/21), done.
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl$ cd thelo09.github.io/
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/thelo09.github.io$ ls
components  index.html  README.md
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/thelo09.github.io$ touch demo.txt
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/thelo09.github.io$ git add demo.txt
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/thelo09.github.io$ git commit -m "added dummy file"
[main 1785c60] added dummy file
 1 file changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 demo.txt
snsrl4@snsrl4-B560M-DS3H-V2:~/Documents/ssl/thelo09.github.io$ git push
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 16 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 344 bytes | 344.00 KiB/s, done.
Total 3 (delta 0), reused 1 (delta 0), pack-reused 0
To github.com:thelo09/thelo09.github.io.git
   fb3d2f5..1785c60  main -> main
```

List of Topics [C213]

Unix

- ○ *Basics: shell, file system, permissions, process hierarchy, process monitoring, ssh, rsync*
- ○ *Tools: grep, find, head, tail, tar, cut, sort, sed, awk*
- ○ *Bash scripting: I/O redirection, pipes, makefile, libraries and linking*

1hr

2+2

3T

Report and Presentation tools

- ○ *Tools: Latex , beamer*
- ○ *Drawing software (e.g., Inkscape, xfig open-office)*

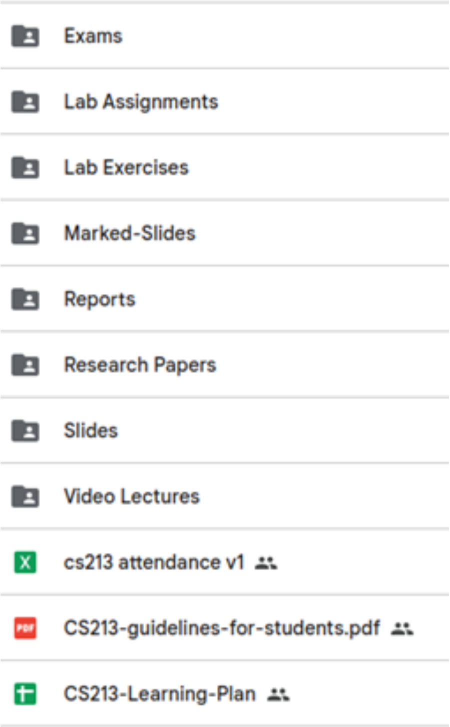
Web Design

- ○ *HyperText Markup Language (HTML)*
- ○ *Hypertext Preprocessor (PHP), Structured Query Language (SQL) and Java Script*

Programming Version Management

- ○ *SVN/Git: Version control, code proofing, documentation*

Statistics (1)



Exams

Lab Assignments

Lab Exercises

Marked-Slides

Reports

Research Papers

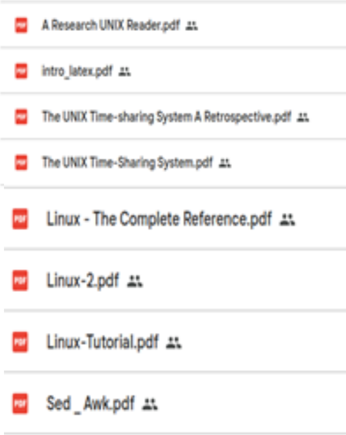
Slides

Video Lectures

cs213 attendance v1

CS213-guidelines-for-students.pdf

CS213-Learning-Plan



A Research UNIX Reader.pdf

intro_latex.pdf

The UNIX Time-sharing System A Retrospective.pdf

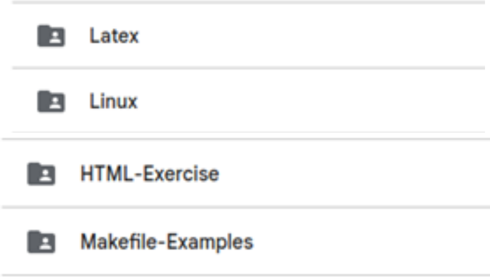
The UNIX Time-Sharing System.pdf

Linux - The Complete Reference.pdf

Linux-2.pdf

Linux-Tutorial.pdf

Sed _ Awk.pdf

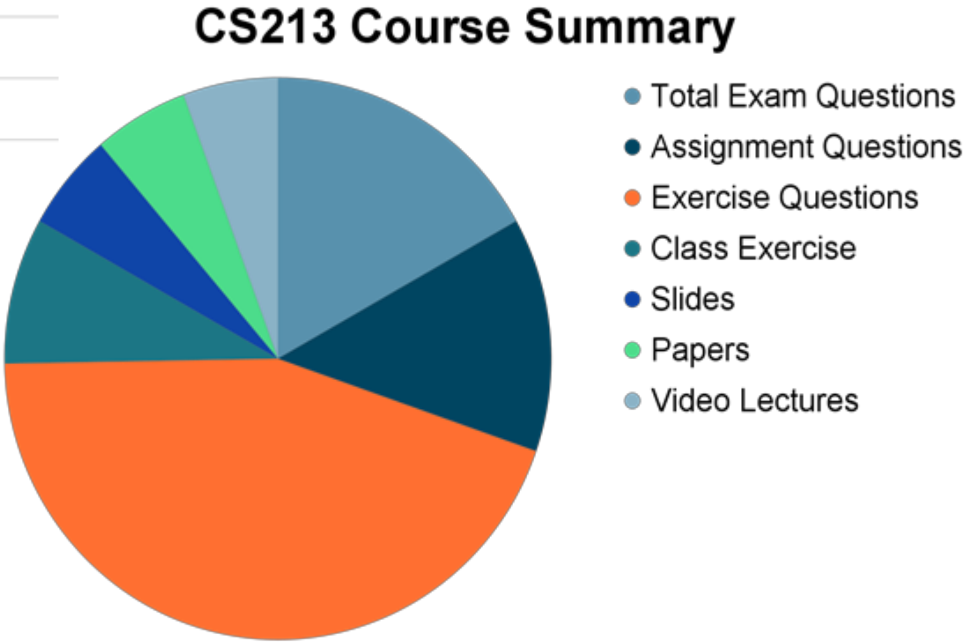


Latex

Linux

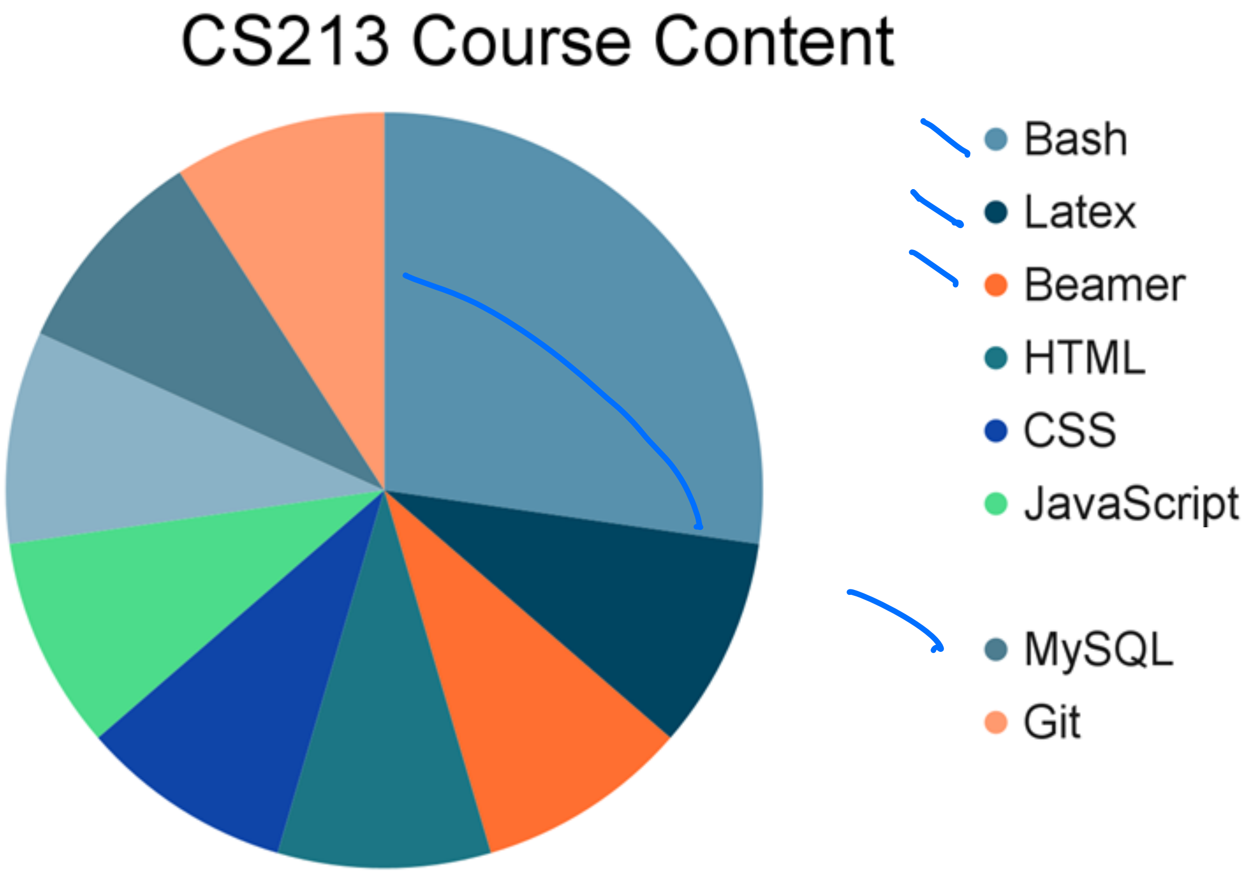
HTML-Exercise

Makefile-Examples



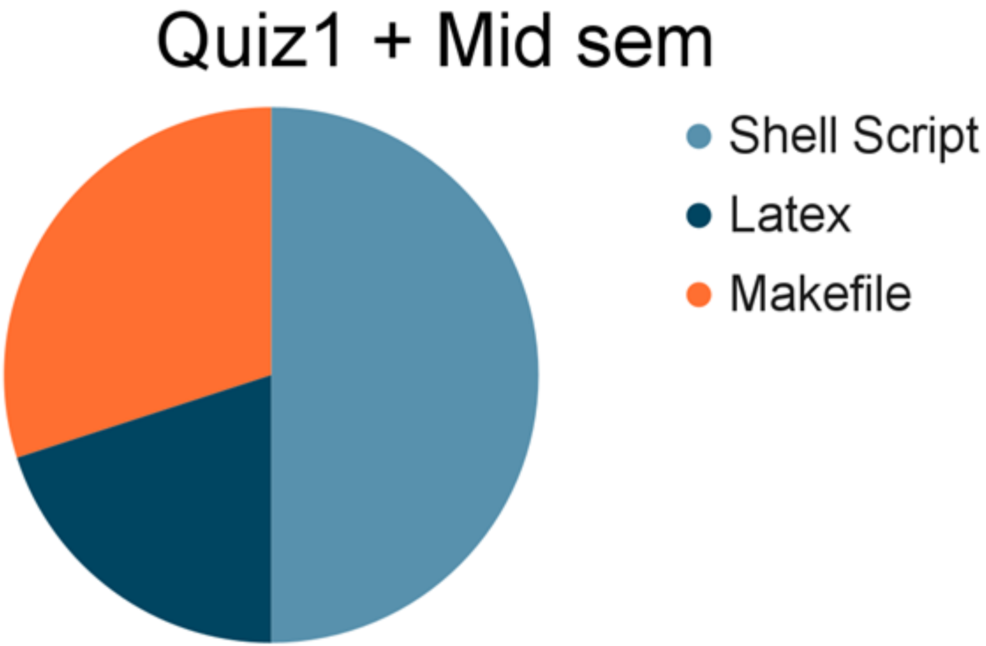
CS213 Course Summary

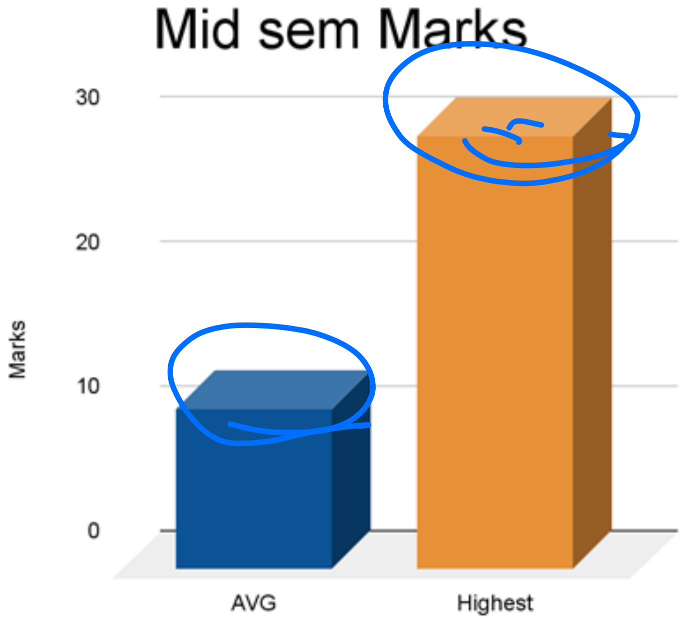
Statistics (2)



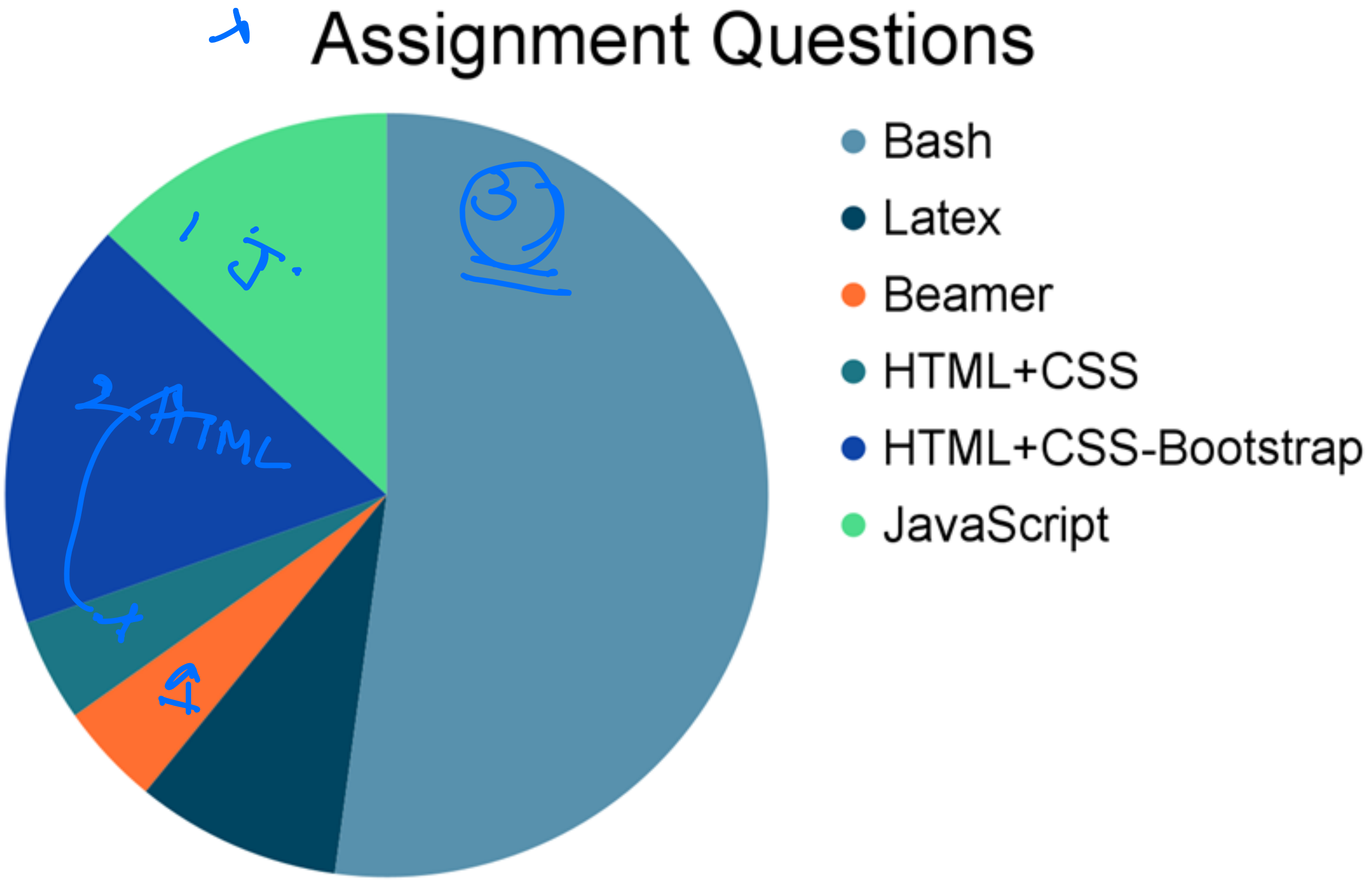
Statistics (3)

100



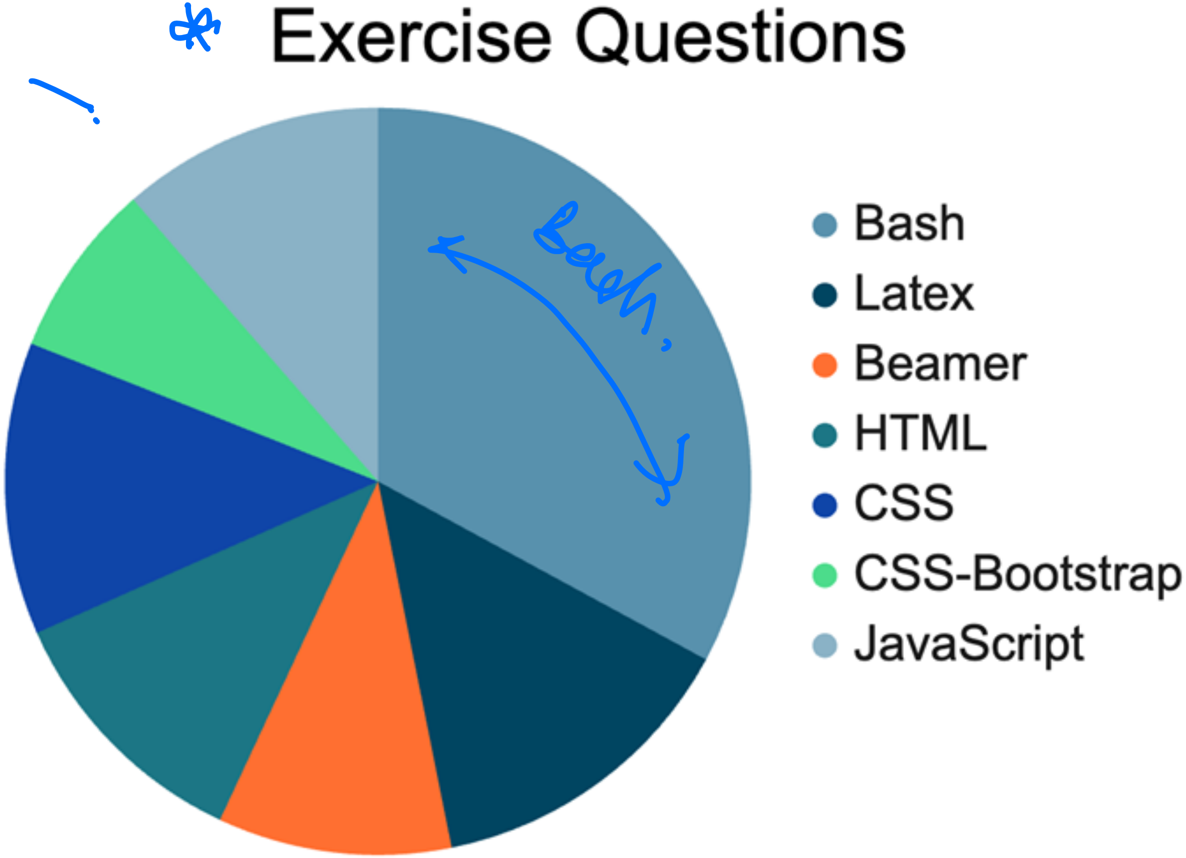


Statistics (4)



Statistics (5)

*



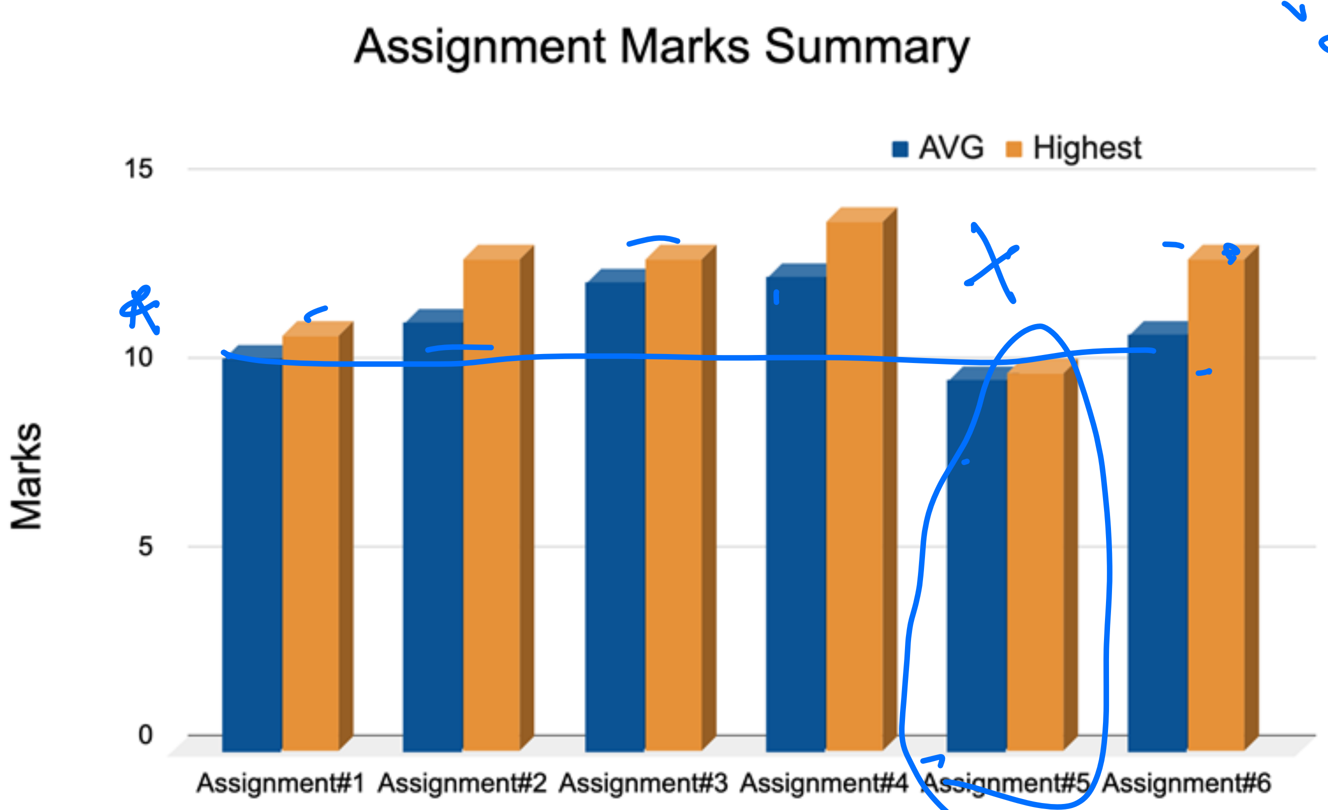
"Noted : Makefile

Statistics (6)

40%

5x2mark = 10

& AVG: 80 marks



thank you!

email:
k.kondepu@iitdh.ac.in