**titThomas Kitaba note**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **C programing** | | | | | | | |
| 1 | Fire up the terminal from VS code  **gcc filename.c"** to compile the program  **.\a.exe or a./out** to run the program from the terminal  The default executable output of gcc is "a.out but you can change it by adding -o **outputfile** at the end  **gcc *filename* -o outputfile**  **gcc calc.c -o calc** | | | | | | | |
|  | Date time  USING DATIME FUNCTION  from datetime import datetime  # to get every thing  **current\_date\_and\_time** = datetime.now()  **output**: The current date and time is 2022-07-12 10:22:00.776664  # to get time  **time\_now** = datetime.now().strftime("%H:%M:%S")  **output** : The current date and time is 10:30:37s  # to get year month and day separetly  year: current\_time.year  month: current\_time.month  day: current\_time.day  USING DATIME FUNCTION  import time  current\_time = time.ctime()  output:Tue Jul 12 10:37:46 2022  To extract the current time, you also have to use the strftime() function:  import time  current\_time = time.strftime("%H:%M:%S")  print("The current time is", current\_time)  # The current time is 10:42:32   * **Convert datetime string to dateobject**   date\_str = "2021/01/20"  date\_object = datetime.**strptime**(date\_str, "%Y/%m/%d")  **Date Calculation**  from datetime import date, timedelta  start\_date = date(2008, 8, 15)  end\_date = date(2008, 9, 15) # perhaps date.now()  date\_diffrence = end\_date - start\_date # returns timedelta  for i in range(date\_diffrence.days + 1):  day = start\_date + timedelta(days=i)  print(day)  The [Python timedelta](https://pynative.com/python-timedelta/) object considers 24 hours as one day, and For calendar days, you’ll need to round down to the nearest day by removing the partial day on both sides. I.e., we need to set hour, minute, and seconds to zero in both datetime.  from datetime import datetime  # datetime in string format  str\_dt1 = '2021/10/20 09:15:32.36980'  str\_dt2 = '2022/2/20 04:25:42.120450'  # convert string to datetime  print(type(dt1)) #output: <class 'datetime.datetime'>  rounded\_dt1 = dt1.replace(hour=0, minute=0, second=0, microsecond=0)  rounded\_dt2 = dt2.replace(hour=0, minute=0, second=0, microsecond=0)  delta = (rounded\_dt2 - rounded\_dt1)  print(delta.days)  Set FLASK\_APP =application.app :-[Set Flask app defaoult]  **Strcmp(a,b) == 0** (0 = same, >0 after b, < 0 before b) | | | | | | | |
| 2 | **Email Notification flask using itsdengerous plus flask\_mail**  **Step 1: from itsdangerous import URLSafeTimedSerializer**  **Step 2: s = URLSafeTimedSerializer (app.config[‘SECRET\_KEY’])**  **URLSafeTimedSerializer (“my secret key”)**  **Step 3: Then you can start generating tokens**  **email = request.form[“user-email”]**  **token = s.dumps( email, salt=’email-confirm’ ) 🡪 this will change the email into token to be send to an external email**  **step 4: Start coding the route that performs the confirmation**  **app.route(“/confirmemail/<token>”)**  **def confimMail(token):**  **try:  email = s.loads (token, salt=’email-confirm’, max\_age= 60)**  **in our case { change email confirmed to true and redirect to login screen }**  **except signatureExpired: 🡪 signatureExpired must be imported from itsdangerous**  **return ‘token expired’**  **in ourcase { if email confiremed false but user tried to login then provide with send confirmation button}**  **step 5: Send the token to the external email**  **msg = Message(‘Tom-Diary Confirmation’, sender='tom@mail.com’, recipients=[email])**  **link = url\_form(“confirmemail”, token=token, external=True) 🡪 import url\_for from flask**  **msg.body = ‘Your link is {}'.format(link)**  **mail.send(msg)** | | | | | | | |
|  | **Strcpy(char \* desitination, char \* source)**  **copies the string pointed by source (including the null character) to the destination|** | | | | | | | |
|  | INSERT TO A LIST  In [1]: ls = [1,2,3]  In [2]: ls.insert(0, "new")  In [3]: ls  Out[3]: ['new', 1, 2, 3] | | | | | | | |
| 4 | **SQL**  **CREATE TABLE users (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, username TEXT NOT NULL, hash TEXT NOT NULL, cash NUMERIC NOT NULL DEFAULT 10000.00);**  **CREATE TABLE userstocks (uid INTEGER NOT NULL, sid INTEGER NOT NULL, transactions TEXT NOT NULL, FOREIGN KEY(uid) REFERENCES users(id), FOREIGN KEY(sid) REFERENCES stocks(id) );**  **CREATE TABLE stocks (id INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, symbol TEXT NOT NULL, name TEXT , shares INTEGER NOT NULL, price NUMERIC NOT NULL DEFAULT 0.00);** | | | | | | | |
| 5 | DELETE FROM stocks WHERE name = 'Alfred Schmidt';S  UPDATE stocks SET name = 'Alfred Schmidt', City= 'Frankfurt' WHERE CustomerID = 1; | | | | | | | |
| 6 | INSERT INTO table\_name (column1, column3, ...)VALUES (value1, value3, ...);  Insert into stocks (**symbol, name, shares, price) values (“TOM”, “thomasinc”, 10, 100);** | | | | | | | |
| 7 | **Sql Transaction**  **db.execute("BEGIN TRANSACTION")**  **rows = db.execute("SELECT likes FROM posts WHERE id = ?", id);**  **db.execute("UPDATE posts SET likes = ? WHERE id = ?", likes +1, id);**  **db.execute (commit)** | | | | | | | |
|  | DELETE FROM users WHERE id = 1; | | | | | | | |
|  | GET LAST RECORD= 1: SELECT \* FROM table ORDER BY column DESC LIMIT 1;  2: SELECT \* FROM TABLE WHERE I­­D = (SELECT MAX(ID) FROM TABLE);  working:  inserted\_stock\_ld = db.execute("SELECT \* FROM stocks ORDER BY id DESC LIMIT 1") | | | | | | | |
|  | * **DROP COLUMN:** ALTER TABLE <TABLENAME> DROP COLUMN <COLUMNNAME>; * **ADD NEW COLUMN**= ALTER TABLE userstocks ADD transaction\_type TEXT NOT NULL; * **Add multiple columns** = BEGIN TRANSACTION;   ALTER TABLE tblName ADD ColumnNameA TEXT DEFAULT '';  ALTER TABLE tblName ADD ColumnNameB TEXT DEFAULT '';  ALTER TABLE tblName ADD ColumnNameC TEXT DEFAULT '';  COMMIT  **UPDATE employees** SET last\_name = 'Johnson' WHERE employee\_id = 1; | | | | | | | |
|  | {% if something\_random == 15 %}  <h1>Hello, this equals 15</h1>  {% elif something\_random == 16 %}  <h1>Hello, this equals 15</h1>  {% else %}  <h1>This does not equal 15</h1>  {% endif %} | | | | | | | |
|  | query = "SELECT \* FROM users WHERE id = ?"  id = (int(session["user\_id"]))  user= db.execute(query, id) | | | | | | | |
|  | **JOIN: SELECT student.first\_name, student.last\_name,  course.name**  **FROM student JOIN student\_course**  **ON student.id = student\_course.student\_id**  **JOIN course**  **ON course.id = student\_course.course\_id;**  **DELETE TABLE:**  **fire accident**  **Drop table users;** | | | | | | | |
|  | Convert datatypes in sqlite3 :  CAST(**myStringNum** **as** REAL)  CAST(**myStringNum as** INT)  CAST(**myStringNum** **as** DOUBLE) | | | | | | | |
|  | Use dollar format {{"${:,.2f}".format(444.55) }} | | | | | | | |
|  | **mysql**  **shell cmd: mysql –e –u username –p password=mypassword “SHOW DATABASES;”**  **shell cmd: mysql –e “SHOW DATABASES;”**  **sql stmt: SHOW DATABASES LIKE 'DATABASE\_NAME';**  **sql stmt: CREATE DATABASE $DATABASE\_NAME;**  **-- This is a single-line comment**  **/\* This is a**  **multi-line comment \*/**  **Note: the difference between backticks or grave accents (`) and single quotes (' ')**  **backticks (`) are used for specifying identifiers, especially when they include reserved keywords or special characters.**  **single quotes (' ') are used to delimit string literals or character values in SQL statements**  **Identifiers are used to represent the names of database objects, such as tables, columns, views, indexes, or aliases. In SQL** | | | | | | | |
|  | **BOOTSTRAP** | | | | | | | |
|  | BOOTSTRAP   * To turn link to button ----- <a class = “btn btn-primary” href=”#” role = “button”> link </a> * To turn link to Google search button ---- < a class = “btn btn-light” role = “button” href = “#”> link2 </a> * To fill the screen------ <div class= “container-fluid”> content goes here </div>   Links ----- <a class= “nav-link”> link </a>   * To center element   <div class=”position-relative”>  <div class="position-absolute top-50 start-0 translate-middle"></div>  </div> | | | | | | | |
|  |  | | | | | | | |
|  | CSS | | | | | | | |
|  | .classname::before{ filter: blur(20px);        transform: translateY(30px) scale(1);} | | | | | | | |
|  | **Adding Counters**  <div class= “my-list”>  <ol>  <li class = “list”> list 1 <li>  <li class = “list”> list <li>  </ol>  </div> | | | | | | . my-list ol {  Counter-reset: name-of-mylist;  List-style: none;}  .my-list li:: before{  Counter-increment: name-of-mylist;  Content: counter(name-of-mylist);  } | |
|  | Margin: auto; You can set the margin property to auto to **horizontally center the element within its container**.  = For absolutely positioned elements whose containing block is based on a block container element, the percentage is calculated with respect to the width of the padding box of that element  .container{    width: 100%;    max-height: 50vh;    overflow: scroll;} | | | | | | | |
|  | From transparent to color:  Background: linear-gradient(to bottom,    card - content {  padding : 1.5em ;  background : linear - gradient (  hsl (0 0% 0% /0 ) ,  hsl ( 20 0 % 0 % / 0.3 ) 20 % ,  hsl (0 0% 0% / 1 )  )  ) ; | | | | | | | |
|  | Max-width: 100%; will certainly text from over flowing | | | | | | | |
|  |
|  |  | | | | | | | |
|  |  | | | | | | | |
|  |  | | | | | | | |
|  |  | | | | | | | |
|  |  | | | | | | | |
|  | **BOOTSTRAP** | | | | | | | |
|  | Containers are used to pad the content inside of them, and there are two container classes available:   1. The .container class provides a responsive **fixed width container** 2. The .container-fluid class provides a **full width container**, spanning the entire width of the viewport   You can also use the .container-sm|md|lg|xl classes to determine when the container should be responsive.  The max-width of the container will change on different screen sizes/viewports: Grid Classes The Bootstrap 5 grid system has six classes:   * .col- (extra small devices - screen width less than 576px) * .col-sm- (small devices - screen width equal to or greater than 576px) * .col-md- (medium devices - screen width equal to or greater than 768px) * .col-lg- (large devices - screen width equal to or greater than 992px) * .col-xl- (xlarge devices - screen width equal to or greater than 1200px) * .col-xxl- (xxlarge devices - screen width equal to or greater than 1400px)   create a row (<div class="row">). Then, add the desired number of columns (tags with appropriate .col-\*-\* classes). The first star (\*) represents the responsiveness: sm, md, lg, xl or xxl, while the second star represents a number, which should add up to 12 for each row. | | | | | | | |
|  | <div class="row">   <div class="col-\*-\*"></div>   <div class="col-\*-\*"></div> </div> <div class="row">   <div class="col-\*-\*"></div>   <div class="col-\*-\*"></div>   <div class="col-\*-\*"></div> </div> | | | | | | | |
|  | **IMAGES**  The .img-thumbnail class shapes the image to a thumbnail (bordered):  The .rounded-circle class shapes the image to a circle.  The .rounded class adds rounded corners to an image:  Float an image to the left with the .float-start class or to the right with .float-end:  Center an image by adding the utility classes .mx-auto (margin:auto) and .d-block (display:block) to the image: | | | | | | | |
| **TABLES**  The .table-responsive class adds a scrollbar to the table when needed (when it is too big horizontally): | | | | | | | |
| TABLE | BG | ALERT | TEXT |  |  |  | |  |
| table-primary | bg-primary | alert-primary | text-primary |  | Blue: Indicates an important action | | | |
| .table-success | .bg-success | . alert -success | .text-success |  | Green: Indicates a successful or positive action | | | |
| .table-danger | .bg-danger | .alert-danger | .text-danger |  | Red: Indicates a dangerous or potentially negative action | | | |
| .table-info | .bg-info | .alert -info | .text-info |  | Light blue: Indicates a neutral informative change or action | | | |
| .table-warning | .bg-warning | .alert -warning | .text-warning |  | Orange: Indicates a warning that might need attention | | | |
| .table-active | .bg-active | .alert -active | .text-active |  | Grey: Applies the hover color to the table row or table cell | | | |
| .table-secondary | .bg-secondary | .alert-secondary | .text-secondary |  | Grey: Indicates a slightly less important action | | | |
| .table-light | .bg-light | .lert-light | .text-light |  | Light grey table or table row background | | | |
| .table-dark | .bg-dark | .alert-dark | .text-dark |  | Dark grey table or table row background | | | |
|  |  |  |  |  |  | | | |
|  |  |  |  |  |  | | | |
|  |  |  |  |  |  | | | |
|  |  |  |  |  |  | | | |
|  |  |  |  |  |  | | | |
|  |  | | | | | | | |
|  |  | | | | | | | |
|  |  | | | | | | | |
|  |  | | | | | | | |
|  |  | | | | | | | |
|  |  | | | | | | | |
|  |  | | | | | | | |
|  |  | | | | | | | |
|  |  | | | | | | | |
|  | Kali linux | | | | | | | |
|  | Websites: echo  Nano  Cat : open in terminal  List:  Touch: create files  Mkdir:  Rm rmdir foldername or rm –r foldername  Man  Help  Chown: change file owner  Chmode: change file mode | | | | | | | |
|  | If root folder access denied  $ Sudo su  $ chmode +777 /root | | | | | | | |
|  | Capture MITM data using tshark: tshark –w /capture  Write captured packet data to out folder: tshark –r /capture - - export -objects “smb, out”  Then cd into /out folder then you will find transferred files | | | | | | | |
|  | **Wireshark filters:**  **Frame contains pwd -** to find username and password from http page login  Ip\_addr==10.10.10.10  arp | | | | | | | |
|  | **bettercap**  View networks : **net.probe on**  View list of devices and their mac address: **net.show**  Set target ip for ARP spoffing: set arp.spoof.targets 192.168.1.9  Lauch the attack: arp.spoof on  Start the sniffer: net.sniff on | | | | | | | |
|  |  | | | | | | | |
|  |  | | | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **GIT** | | |
|  | **SETTING UP FOR THE FIRST TIME**  **Mkdir project1**  **Git init**  **CONFIGURING GIT FOR FIRST TIME**  = if git does not know you  **git config --global user.name " thomaskitaba "**  **git config --global user.email** [**thomas.kitaba@gmial.com**](mailto:thomas.kitaba@gmial.com)  then to check if you are correct or to check configuration  **git config –list** | | |
|  | = to see history of your git commits **git log -2(** to see the last 2 commits else you can use only  **git log**  **=** to check the directory you are in **pwd**  **=** to get out of VIM type in **:q= or** esc then write **:wq**  = to check on which branch you are working on  **git checkout source** | | |
|  | **= Remote**  is a duplicate instance of your repository on other remote server (like github)  To list all the remotes  **git remote**  usually the remote name will be **origin**  This will give you name of the remote then **git push origin master** | | |
|  | **CLONE REMOTE REPOSITORY TO YOUr LOCAL COMPUTER**  To download your repository and work locally you have to clone the repository first  **Git clone <url>**  **Git clone** <https://github.com/thomaskitaba/desktop-test.git>  **git clone** <https://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@github.com/thomaskitaba/alx-system_engineering-devops.git>  **git clone** [https://ghp\_qtanNwEmqthexzj8jgsEDt2urx6dFI44v5Q5@github.com/thomaskitaba/](https://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@github.com/thomaskitaba/alx-system_engineering-devops.git)alx-low\_level\_programming.git  **git clone** <https://yourtoken@github.com/yourusername/>alx-low\_level\_programming.git  or just simply use  **git clone** <https://github.com/yourusername/>alx-low\_level\_programming.git   * To configure Git to cache your credentials permanently, follow these steps:   git config --global credential.helper store  git commit --amend -m " Starting to code today, so cool"  git commit -m " Starting to code today, so cool"  # git clone <https://qtanNwEmqthexzj8jgsEDt2urx6dFI44v5Q5@github.com/thomaskitaba/alx-system_engineering-devops.git>  git clone https://qtanNwEmqthexzj8jgsEDt2urx6dFI44v5Q5@github.com/thomaskitaba/ alx-low\_level\_programming.git | | |
|  | **MAKE LOCAL FOLDER A REPOSITORY**  **= make folder on your computer a repository and push it to remote server**  **1**st Add it to the staging area  **Git add filename**  2nd commit the change  **git commit –m “ message description”**  or we can **ADD** and **COMMIT** at the same time using the command  **git commit –am “message description”**  NB- git commit –A ( commits and adds at the same time we use this most of the time)  Or 1st **git add –A 2nd git commit –m “msg description”** without using –A since we added all  **3rd**  go to github and make new **repository ( don’t initialize it with readme.txt)**  **Copy the link of the repository**  and associate the local repository with that address  **git remote add origin** <https://github.com/thomaskitaba/desktop-test.git>  **4th** push it to github  **Git push orign master**  **CORRECTING ERRORS**  **1 - Navigate to the repository.**  **2 Amend the message of the latest pushed commit: git commit --amend -m "New commit message."**  **3- Force push to update the history of the remote repository: git push --force <remoteName> <branchName>** | | |
|  | **BRANCHING** | | |
|  | * To create new branch **git checkout –b branch-name** * To change an existing branch **git checkout branch-name** * To merge branch to master **git merge boarders**   **Delete a branch:**   1. Delete a Merged Branch (-d flag):   Use git branch -d branch-name   1. Force Delete Unmerged Branch (-D flag):   to forcefully delete a branch, regardless of its merge status.  Use git branch -D branch-name | | **View Branch’s**   1. View local branches: **git branch** 2. View remote + local branches: **git branch -a** |
|  | **git diff** tells us the difference between the previous and our current file | | |
|  | **Merging via command line**  **If you do not want to use the merge button or an automatic merge cannot be performed, you can perform a manual merge on the command line. However, the following steps are not applicable if the base branch is protected.**  **Step 1:** From your project repository, check out a new branch and test the changes.  git checkout -b thomaskitaba2-main **main**  git pull https://github.com/thomaskitaba2/git-practice.git **main**  **Step 2:** Merge the changes and update on GitHub.   * **Check out to your master or main branch**   Cmd: git checkout main   * **Merger the branch you created with the master or main branch**   git merge --no-ff thomaskitaba2-main (--no-ff: no fastforward   * git add .   git commit –m “accepted pull request from thomaskitaba2”  git push origin main | | |
|  | **Github Issues:**   1. to resolve issue: follow this steps   get the number of the issue(from issues tab on GitHub ) -🡪 solve the issue if it is solvable🡪  🡪  **git commit –m “ resolved fixes #4”** (use closes #issue-number or fixes #issue-number inside a commit)   1. **To reference an issue.**   If you are on vsCode: just add **as per #issue-number** to your commit  If it is on github website:  Add- **#issue-number** without any special keyword to the commit message | | |
|  | Git add .   * Does not stage deleted file * Stages current and sub directory changes | Git add -A   * Also stages deleted file * Stages the whole repository | |
| git add -A is a shorthand for git add .; git add -u.  git add . stages all changes in the working directory, including new files.  git add -u stages modifications and deletions, but does not stage new files.  In other words, git add -A stages both new files and changes to already tracked files, while git add . only stages new files and changes to already tracked files.  The difference between git commit -m and git commit -am is as follows:  git commit -m "message" creates a new commit with the specified message and stages all changes in the working directory.  git commit -am "message" creates a new commit with the specified message and stages all changes to already tracked files in the working directory, but does not stage new files.  In other words, git commit -m stages all changes, including new files, while git commit -am only stages changes to already tracked files and does not stage new files. | | |
|  | **Isolating workspace**  **virtualenv** | | |
|  | -pip install virtualenv  -virtualenv tom-venv or  py -m venv your\_name  -source bin/activate or .\your\_name\Scripts\activate  ---- activate the virtual environment  -deactivate ------ to deactivate current virtual environment  - virtualenv -p python3 tom-venv ----- install python3 in your virtual environment | | |
|  |  | | |

|  |  |
| --- | --- |
|  | **VITE** |
|  | -create vite project **npm create vite@latest**  - name you vite project folder **myvite-project**  - select your framework **vinella**  - install dependencoies **npm install or npm i**  - run development server **npm run dev** |
|  | **Heroku**   1. **Pip install Heroku** 2. **Heroku login [ login to Heroku website]** 3. **Pip install virtualenv [create local miniserver to deploy your app ]** 4. **Virtualenv venv [ create virtual env in tom-diary folder]** 5. **Pip install -r requirements.txt** 6. **Pip install gunicorn [ a must to deploy your app on Heroku ]** 7. **Pip freeze > requirements.txt [records an environment's current package list into requirements. txt.]** 8. **Web gunicorn app [app.py]:app [ put this inside the Procfile file]** |
|  | **CSS** |
|  | **To make triangle**  .**classname** {  border-left: 1em solid transparent;  border-right: 1em solid transparent;  border-bottom: 1em solid red;  } |
|  |  |
|  | **JAVASCRIPT** |
|  | Vanilla JavaScript refers to **using plain Javascript without any additional libraries or frameworks**. |
|  | -**Get Element Width and Height**  Const element1 = document.querySelector(“.div1”)  Element1-height = element1. offsetWidth  elemet1-weidth = element1. offsetHeight |
|  |  |
|  | If we are manuplateing the DOM it is better to place the link to the **js** file at the end of the page  After **</body>** and before **</html>**  **<script defer src = “script.js “ </script>** |
|  | Functions:  Var myfunction = new function(“a”, “b”, “return a + b”) call: x = myfunction(2,5)  var y = function (a,b){return a+b} call x = y(8,5) |
|  | document.getElementById("myDIV").style.display = "none"; |
|  |  |
|  |  |
|  |  |
|  | **Settimeout:** |

|  |  |  |
| --- | --- | --- |
|  |  | |
|  |  | |
|  |  | |
|  | **JAVASCRIPT--------- JQUERY** | |
|  | Using jquery <script src = “folder/jquery.js “ type=”text/javascript “> </script>  <script src = “script.js “ type=”text/javascript “> </script>  If we are using the CDN verion  <script src = “ <http://djfalsdjfadjfkl> > </script> instead of folder/jquery  - Use a jQuery method to return the value of an input field. **$("input").val()** | |
|  | * $(element name) .action () same as document. getElementByTagName (element name)   **$**("div").**css**('background-color, **'orange'**); | |
|  | **==** operator does the type conversion of the operands before comparison  **===** operator compares the values as well as the data types of the operands. | |
|  | Never compare the objects with == operators.  Let car1 = { name=”atoz”} car2 = {name = “atoz”} car1 == car2 is FALSE  **Different objects with the same value are not equal** | |
|  |  | |
|  |  | |
|  | $("document") .ready(function(){  $("button") .click( (function(){  $("#img1") .css ('width', '500px');  });  }); | |
|  |  | |
|  |  | |
|  | $(element name) .action () | description |
|  | $(“#myid”).val() | Get value of an element with id myid |
|  | $(“a”).attr(href) | Get url address of a linl |
|  | $(“input”).val(“tom kit”) | change the value of an input field to "tom kit". |
|  |  |  |
|  | **JSON**  aList = [{'a':1, 'b':2}, {'c':3, 'd':4}]  jsonStr = json.dumps(aList)  print(jsonStry)  Output: **[{"a": 1, "b": 2}, {"c": 3, "d": 4}]**  to use jsonify from **flask** import **jsonify** | |
|  | **AJAX** | |
|  | AJAX is about loading data in the background and display it on the webpage, without reloading the whole page.  The 4 different states of request process  Initiating a request **( ready state 0)** 🡪 sending request🡪 recived response 🡪 response comes backto you 🡪 request iscomplete **( ready state 1 status = 200)**  **To check for error like if the server we are trying to connect to is down at the time of the request or if the server responds with an error code**  **we use the following error checking**  **option 1- if (myrequest.readyState == 4 && myrequest.status == 200)**  **or**  **option 2- if (myrequest.staus >= 200 && myrequest.status < 400);**  XMLhttprequest: a builin tool to will establish a connection with the url we provide and enable recive or send data  Step1- var ourRequest = **new** XMLHttpRequest instantiate xmlhttprequest  Step 2- ourRequest.open(‘GET’, ‘url address’) open the connection  Step 3- ourRequest.onload= function(){ what to do with the data we get }  ourRequest.onload= function(){  Var recivedData = JSON.parse(**ourRequest.responseText ) parse it so that the responsetText is converted to object**  console.log( }  Step 4- ourRequest.send(); send the request  SAMPLE WORKING CODE  var reciveddata;  function jsonTest(){    //create new instanse of XMLHttpRequest    var myrequest = new XMLHttpRequest();    // open connection to a server to send and recive data    // by providing   what you want to do, and from where you want it    myrequest.open('GET', 'https://learnwebcode.github.io/json-example/animals-1.json');    // load the data recived from the external location    myrequest.onload = function(){      var reciveddata = JSON.parse(myrequest.responseText);      console.log(reciveddata)      document.getElementById("object").innerHTML = reciveddata;    };    //send the request to the server    myrequest.send();  }  +++++++++++++++++++++++++++++++++++++++++++++++  **Simple Steps to Remember the AJAX code**  $(document).ready(function() {  $(**'form'**).on(**'submit'**, function(event) {  $.ajax({  data : {  firstName : $(**'#firstName'**).val(),  lastName: $(**'#lastName'**).val(),  },  type : **'POST'**,  url : **'/process'**  })  .done(function(data) {  $(**'#output'**).text(data.output).show();  });  event.preventDefault();  });  });  For beginners, It is difficult for the above Ajax part to remember. The following steps you should keep in mind.  Step - 1. Call the document load function.  $(document).ready(function() {  Step 2 : Call the action on submit evenet :  $('form') .on ( 'submit' , function(event){  Step 3: Call the Ajax Function  $.ajax( {  Step 4: Create here the data for the field values. Request Method 'POST' OR 'GET' and Url for Processing the form.  data : {  key1: $('attribute').val(),  key2: $('attribute').val(),  .  .  keyn: $('attribute').val()  }  type : 'POST' OR 'GET'  url : '/example'  } )    Step 5: Call the success function.  .done(function(data) {    Step 6 : Wrtite the code for the data retured from the Server url: '/example'  });  Step 7 : Prevent the enitre submit event to do default action  event.preventDefault();  } ); | |
|  | **NETWORKING** | |
|  | Private networks   * Class A: 10.0.0.0 — 10.255.255.255 * Class B: 172.16.0.0 — 172.31.255.255 * Class C: 192.168.0.0 — 192.168.255.255   127.0.0.0 reserved for experiment | |
|  | Types of cables   1. Coaxial cable- av jack , were used as a cable to give power to devices as electric cabel   rj 6 - decoder to dish video and audio  rg – 8 – (larger diameter) radiaio stations good for audio bad for video  rg-59 – similar to rg 6 but thiner   1. twisted pair-  * shielded twisted pair- for transmition short distance * unshielded twisted pair- interference is higher than stp   used for analogue telephony and highspeed computer connectin in LAN   1. Fiber optics : for long distance networking at global scale      1. Ethernet - computer to router to switch   rg45  fast Ethernet – 100 base x (100 mbps)  cat 5 e - standard version for the ethernet 1gb for browsing  cat 6 - 10 times cat 5 for online streaming and gaming  cat 6 a – augemented resistant to interference  cat 7 - 100gbs  cat 8 – faster than cat 7 for less than 30meters | |
|  | **Console Cables:**  are used to connect Cisco and other networking devices to terminals or pcs for configuration and troubleshooting | |
|  | **Rules of connecting cables ( NB – ass long as you are consistent with your you cabling you can leave the rules)**  **1 two like devices ( devices on the same osi model ) use cross over cable**  **2 devices on different OSI layer should be connected using straight through cable**  **3 rollover cables are used for console cables**  **network layer ( computer 🡪 server 🡪 router) these devices are smarter**  **data-link layer ( switch )**  **physical layer ( HUB)**  **2.0.1 CISCO Packet Tracer File Types**  Packet Tracer has the ability to create four different types of files. These file types are used for different purposes and include: .pka, .pkt, .pksz, and .pkz.  **2.0.2 Cisco Packet Tracer Assessments**  This is a media player component. Select the play / pause button to watch or listen.  There are two types of assessment items that take advantage of Packet Tracer. These include Packet Tracer Media Objects (PTMO) and Packet Tracer Skills Assessments (PTSA).  Types of date:   1. **Volunteered data -**This is created and explicitly shared by individuals, such as social network profiles. This type of data might include video files, pictures, text, or audio files. 2. **Observed data -** This is captured by recording the actions of individuals, such as location data when using cell phones. 3. **Inferred data -** This is data such as a credit score, which is based on analysis of volunteered or observed data.   There are three common methods of signal transmission used in networks:   * **Electrical signals -**Transmission is achieved by representing data as electrical pulses on copper wire. * **Optical signals -**Transmission is achieved by converting the electrical signals into light pulses. * **Wireless signals -**Transmission is achieved by using infrared, microwave, or radio waves through the air.   Bandwidth is the capacity of a medium to carry data. Digital bandwidth measures the amount of data that can flow from one place to another in a given amount of time.  Latency refers to the amount of time, including delays, for data to travel from one given point to another.  Throughput measurements do not take into account the validity or usefulness of the bits being transmitted and received  P2P applications require that each end device provide a user interface and run a background service.  The quality and speed of the DSL connection depends mainly on the quality of the phone line and the distance from the central office of your phone company The farther you are from the central office, the slower the connection.  The quality and speed of the DSL connection depends mainly on the quality of the phone line and the distance from the central office of your phone company The farther you are from the central office, the slower the connection.  **DSL** runs over a telephone line, with the line split into three channels. One channel is used for voice telephone calls. This channel allows an individual to receive phone calls without disconnecting from the internet.  We can connect with our ISP   1. DSL: with existing telephone land line [ phone + internet data ] 2. Cable: uses coaxial cable usually provided by cable tv [phone + internet + television] 3. Cellular: using mobile phone networks [ can be mettered ] 4. Satellite: satellite modem might be required   Encapsulation:  +++++  Advantages of a single local segment:   * Appropriate for simpler networks * Less complexity and lower network cost * Allows devices to be "seen" by other devices * Faster data transfer - more direct communication * Ease of device access   Disadvantages of a single local segment:   * All hosts are in one broadcast domain which causes more traffic on the segment and may slow network performance * Harder to implement QoS * Harder to implement security   ++++  **remote local segmet**  Advantages of remote local segment:   * More appropriate for larger, more complex networks * Splits up broadcast domains and decreases traffic * Can improve performance on each segment * Makes the machines invisible to those on other local network segments * Can provide increased security * Can improve network organization   Disadvantages:   * Requires the use of routing (distribution layer) * Router can slow traffic between segments * More complexity and expense (requires a router) | |
|  | **Browser tips and tricks** | |
|  | **Enable the Profile Picker:**  If you are interested in trying out the new Profile Picker feature, follow these steps.   * Launch Google Chrome browser. * In the address bar, input “**chrome://flags/#enable-new-profile-picker**“. * A new window opens up with the “**New Profile Picker**” option. * Click on the “**Enable**” button and then choose the “**Relaunch**” option from the bottom of the screen. * Now, you can see all the profiles that exist on your Chrome Browser. * If you want to**‘Add another profile,’**then click on the “**Add person**” option from the bottom of the screen. * Type the new profile’s name and choose a new avatar for the profile picture. * If you want to add the profile on your desktop, tick the checkbox beside “**Create a desktop shortcut for this user**” option. | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  | **FINDING INFO Hacking** | |
|  | [WWW.github.com/sundowndev/phoneinfoga](http://WWW.github.com/sundowndev/phoneinfoga)  documentation---🡪 installation | |
|  | docker pull sundowndev/phoneinfoga:latest  docker run -it -p 8080:8080 sundowndev/phoneinfoga serve -p 8080  then you are good to go | |
|  | CMD | |
|  | Insert new user to windows : net tom 123456a /add  Stop windows service: net stop WsNisSvc  Download link in powershell: Invoke-WebReauest https://github.com/jbara2002/windows-defender-remover/releases/download/release\_def\_12\_0\_0\_0\_1/Defender.Remover.12.exe -Outfile kill\_defencer.exe  Get drive names of your pc: fsutil fsinfo drives | |
|  | **WSL - Ubuntu** | |
|  | Sudo su: to be admin for the time being  sudo usermod -aG sudo thomaskitaba : add your user to the sudo group:  followed by su – thomaskitaba: to make the change without restarting | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |