

Démarche pour set l'env sans Anaconda :

D'abord trouver le fichier de pip (ici c'est sur anaconda 3 mais si j'avais installé je l'aurais sur python au lieu d'anaconda) : C:\Users\c93884\AppData\Local\anaconda3\Scripts

Prendre ce chemin et le lancer « VARIABLE D'ENVIRONNEMENT » et l'ajouter au path.

Du coup là pip est connu sur VS CODE si je l'ajoute ici :

Après il faut exécuter cette commande pour qu'on set python à ce path :

Set-Alias -Name python -Value C:/Users/C93884/AppData/Local/anaconda3/python.exe (à adapter suivant le chemin vrm suivi)

Pour faire en sorte que cette commande s'exécute à chaque lancement de VSCode on crée un fichier profil personnel :

Un fichier de profil personnel, dans le contexte de PowerShell (et également dans d'autres environnements de ligne de commande), est un fichier script où vous pouvez définir des configurations personnalisées, des alias, des fonctions et d'autres commandes qui seront chargées automatiquement chaque fois que vous démarrez PowerShell.

```
Test-Path $Profile
```

```
New-Item -Path $Profile -ItemType file –Force
```

Dans le fichier du path indiqué sur le terminal, on écrit cette commande qui va s'exécuter à chaque fois du coup :

```
Set-Alias -Name python -Value C:/Users/C93884/AppData/Local/anaconda3/python.exe
```

J'avais que python d'installé : comment créer un virtual env ?

Je n'arrivai pas à trouver python que j'ai installé, alors j'ai cherché son path :

```
C:/Program Files/Python311/python.exe
```

Il fallait l'appeler comme ceci : & " C:/Program Files/Python311/python.exe"

Après ceci j'ai créé un nouvel environnement avec la commande : python -m venv nom

(ms j'ai appelé python avec & " C:/Program Files/Python311/python.exe" )

Après il faut kill le terminal courant et en ouvrir un nouveau ou est bien sourcé (Il y'aura (nom) en vert affiché à gauche)

```
"workbench.colorTheme": "Dracula Theme",
```

Pour installer les modules nécessaires, il vaut mieux les mettre dans un requirements.txt, et les installer tous avec une seule commande :

pip install -r .\requirements.txt

des fois si ça marche pas, essaie juste de réouvrir le bon folder

## GIT :

git status  
git add \*  
git commit -m "name\_that\_will\_appear\_on branch"  
git push  
git pull  
git restore  
git diff

pour voir une version de commit (nouvelle qui n' pas été pull ou ancienne qui a été modifiée après):

git fetch origin  
git checkout name of branch  
(exemple : git checkout 9a5fc970)

git checkout -b export\_main\_2 origin/export\_main (pour créer une branche locale export\_main\_2 qui est reliée à la branche distante export\_main)(on ne peut pas se mettre directe sur la branche distante sans créer la branche locale)(la branche locale reste sur mon pc sauf si je reclone le projet)

git for-each-ref --format='%(refname:short) -> %(upstream:short)' refs/heads/ (pour vérifier que les branches locales et distantes sont bien connectées)

git log  
git branch (pour voir dans quelle branche on est)

## Raccourcis VS CODE:

- CTRL+ D pour sélectionner toutes les instances du mot initialement sélectionné. On peut après les avoir sélectionné comme ceci les modifier tous au même temps.
- Je pose le curseur qlq part, je laisse ALT appuyé dessus et je peux poser mon curseur qlq part d'autre. En enlevant mon doigt de alt je peux modifier ou écrire simultanément sur toutes les places où j'ai posé le curseur.
- ALT+↓ pour ramener une ligne en bas.
- CTRL+maj+F pour chercher globalement un mot dans tous les docs
- ctrl +maj +p pour sélectionner une action (comme choisir l'interpréteur de python)

## **Fichier JSON spécial :**

**JSON (JavaScript Object Notation)** est un format de fichier léger et lisible par les humains utilisé pour stocker et échanger des données.

**Objet** : Un objet est une collection de paires clé-valeur entourée par des accolades {}.

**Tableau** : Un tableau est une liste ordonnée de valeurs entourée par des crochets [ ].

**Exemple :**

```
{
  "name": "John Doe",
  "age": 25,
  "email": "john.doe@example.com",
  "isVerified": true,
  "friends": [
    {
      "name": "Jane Smith",
      "age": 28
    },
    {
      "name": "Mike Johnson",
      "age": 22
    }
  ]
}
```

Pour accéder par exemple à l'âge de Mike Johnson, comment on fait?

D'abord les fichiers JSON ne sont pas directement exploitables, il faut d'abord les parser.

**Parser** signifie convertir une chaîne de caractères en une structure de données que le programme peut manipuler. Dans le contexte des fichiers JSON, parser une chaîne JSON signifie la convertir en un objet ou une structure de données propre au langage de programmation utilisé (comme un objet en JavaScript, un dictionnaire en Python, etc.).

```
Sur JavaScript : const obj = JSON.parse(jsonString);
const mikeAge = obj.friends[1].age;
console.log(mikeAge); // Affiche : 22
```

```
Sur Python :      obj = json.loads(json_string)
```

```
mike_age = obj["friends"][1]["age"]
print(mike_age) # Affiche : 22
```

## **NoSQL Spécial :**

Dans le SQL classique on utilise des tableaux contenant des lignes et des colonnes pour stocker les données.

**Concept 1** : Stocke des paires clé-valeur, un peu comme un dictionnaire.

**Concept 2** : Stocke des documents (souvent en JSON), chaque document peut avoir une structure différente.

- ❖ Pour appeler une fonction en commande d'un bouton sans que ça s'exécute par défaut lors de la création du bouton  
`(self.function(arg1,arg2) ou self.function() si la fonction n'a pas d'argument à part self)` on l'appelle comme ceci :  
`self.function ou lambda:self.function()`

La définition `def alert_change(self, _)` dans une classe en Python signifie que la méthode `alert_change` est définie pour prendre deux arguments : `self` et `_`.

- `self` est une référence à l'instance actuelle de la classe. Il est utilisé pour accéder aux attributs et méthodes de la classe.
- `_` est un argument qui peut être utilisé pour indiquer que la valeur n'a pas d'importance dans le contexte de cette méthode. C'est une convention en Python pour signaler que l'argument ne sera pas utilisé.

## **Différence client lourd- client léger :**

### **Client Lourd (Thick Client) :**

**Description** : Un client lourd est une application installée localement sur l'ordinateur de l'utilisateur.

#### **Avantages :**

- Fonctionnalités complètes.

- Performances rapides puisque l'application utilise les ressources locales.

#### **Inconvénients :**

- Nécessite des installations et des mises à jour locales.
- Peut consommer beaucoup de ressources système.

#### **Client Léger (Thin Client) :**

**Description :** Un client léger est une application qui fonctionne principalement via un navigateur web ou avec un minimum de ressources locales, en s'appuyant sur un serveur distant pour la plupart des traitements.

#### **Avantages :**

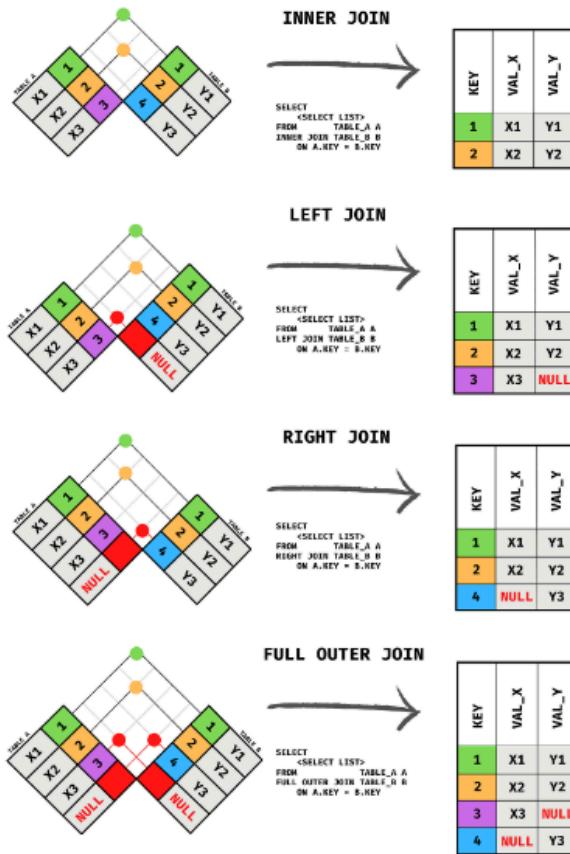
- Facile à déployer et à maintenir.
- Accessible depuis n'importe quel appareil avec un navigateur web.

#### **Inconvénients :**

- Peut être plus lent, dépendant de la connexion Internet.
- Moins de fonctionnalités par rapport au client lourd.

SQL:

Joint special :



#### SELECT DISTINCT

```
table1.* ,
table2.*

FROM table1
JOIN table2
ON matching_condition

WHERE constraint_expression
GROUP BY columns
HAVING constraint_expression
ORDER BY column
LIMIT count
```

- La clause SELECT DISTINCT en SQL est utilisée pour retourner des lignes uniques dans le résultat d'une requête. En d'autres termes, elle permet d'éliminer les doublons dans les résultats.

```
SELECT customer_ID, SUM(total_amount) AS "Total"
FROM orders
WHERE order_date BETWEEN '2022-01-01' AND '2022-03-31'
AND customer_city = 'New York'
GROUP BY customer_id
ORDER BY Total DESC;
```

Pour tout sélectionner : SELECT \*

Différence entre Group By et Order By:

## Différences Clés

- **Objectif:**
  - ORDER BY trie les résultats.
  - GROUP BY regroupe les résultats.
- **Utilisation des Fonctions d'Agrégation:**
  - ORDER BY ne nécessite pas de fonctions d'agrégation.
  - GROUP BY est souvent utilisé avec des fonctions d'agrégation pour calculer des statistiques sur des groupes.
- **Impact sur les Données:**
  - ORDER BY n'affecte pas le contenu des lignes retournées, seulement leur ordre.
  - GROUP BY réduit le nombre de lignes retournées en combinant celles qui ont des valeurs communes.

-La clause ORDER BY est utilisée pour trier les résultats d'une requête selon une ou plusieurs colonnes, en ordre croissant (ASC, par défaut) ou décroissant (DESC).

**Exemple**

Supposons que nous avons une table `Employes` :

Nom	Prenom	Age	Département
Dupont	Jean	35	Ventes
Martin	Marie	28	Marketing
Durand	Paul	40	IT
Dupont	Anne	25	Ventes

Pour trier les employés par âge en ordre croissant :

sql Copier le code

```
SELECT *
FROM Employes
ORDER BY Age ASC;
```

Résultat :

Nom	Prenom	Age	Département
Dupont	Anne	25	Ventes
Martin	Marie	28	Marketing
Dupont	Jean	35	Ventes
Durand	Paul	40	IT

- La clause `GROUP BY` est utilisée pour regrouper les lignes qui ont les mêmes valeurs dans des colonnes spécifiées en une seule ligne. Souvent, `GROUP BY` est utilisé en conjonction avec des fonctions d'agrégation (`COUNT`, `SUM`, `AVG`, `MAX`, `MIN`).

### Syntaxe:

**Exemple**

Pour compter le nombre d'employés dans chaque département :

```
sql
SELECT Departement, COUNT(*)
FROM Employes
GROUP BY Departement;
```

Résultat :

Departement	COUNT(*)
Ventes	2
Marketing	1
IT	1

### Module OPENCV et REPORTLAB:

- Le module `cv2` est la bibliothèque OpenCV (Open Source Computer Vision Library) pour Python. OpenCV est une bibliothèque open-source qui contient plus de 2 500 algorithmes optimisés pour des applications de vision par ordinateur et de traitement d'images. Elle est largement utilisée pour le développement de projets impliquant la reconnaissance d'images, la détection de visages, la segmentation d'images, le suivi d'objets, la réalité augmentée, etc.
- `reportlab` est une bibliothèque Python puissante pour générer des documents PDF directement à partir de votre code. La commande `from reportlab.pdfgen import canvas` importe le module `canvas` de la bibliothèque `reportlab`, qui permet de créer et manipuler des documents PDF.

### Sites intéressants :

- Photopea pour photoshop
- typst (comme overleaf mais mieux !)

## SaaS :

software as a service. Le logiciel en tant que service (SaaS) est traditionnellement considéré comme un modèle de logiciel basé sur le cloud qui fournit des applications aux utilisateurs finaux via un navigateur Internet. Les fournisseurs de SaaS hébergent des services et des applications auxquels les clients peuvent accéder à la demande.

## Spécial Photoshop :

W : pour object selection tool

L : pour définir par soi les contours

J : spot healing tool

then you cut it and paste it somewhere else

## Modules python :

Shutil (pour copier les fichiers).

flask (pour télécharger les fichiers).

os (operating system) (this module allows the use of all operating systems depending functionalities, it's mostly used to manipulate paths).

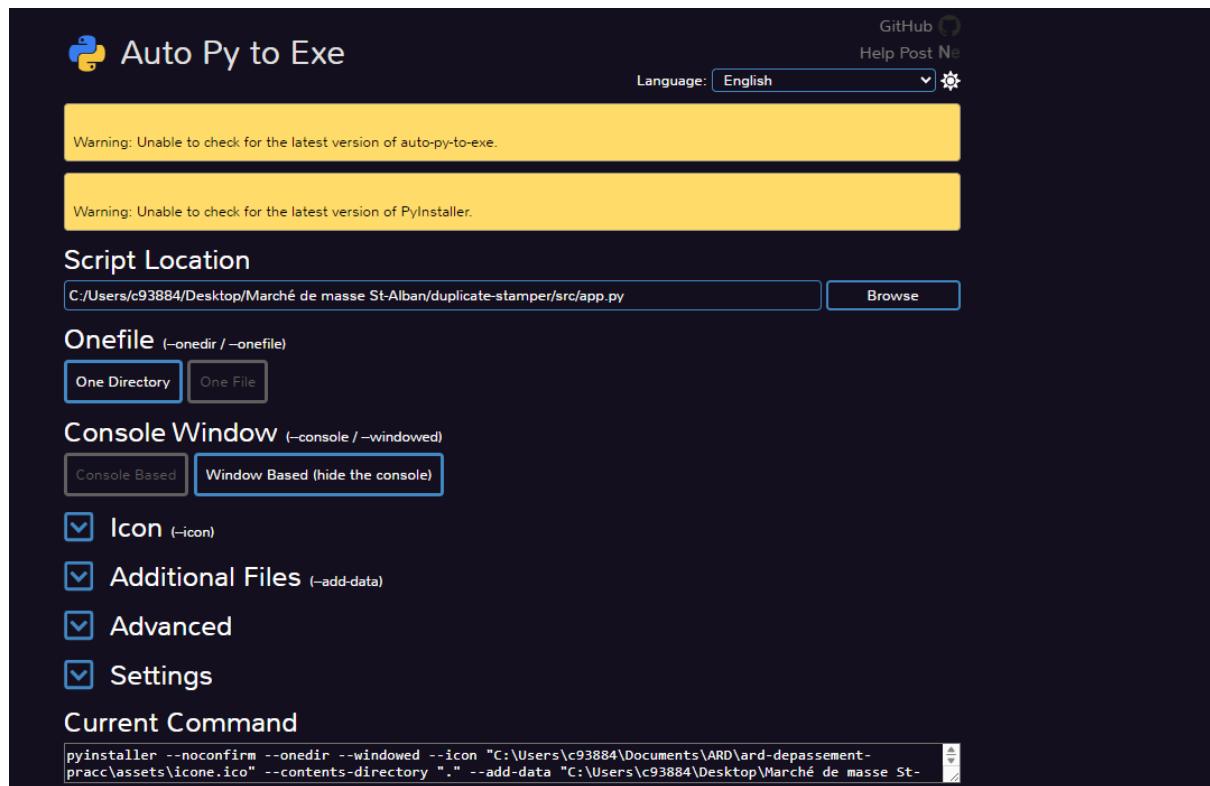
datetime (pour récupérer la date et le temps):

```
from datetime import datetime
```

```
    execution_time = datetime.now()  
    time = execution_time.strftime("%Y_%m_%d %H%M")
```

## Special packaging :

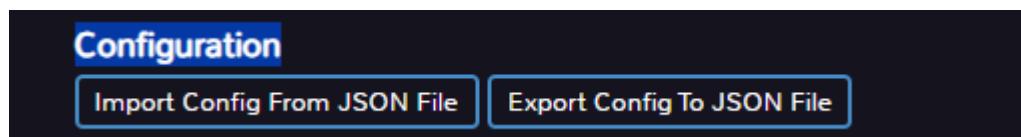
Pour packager il faut utiliser auto-to-py-exe. C'est un module à installer aussi. Après en lançant le packaging avec la commande "auto-to-py-exe" dans le terminal, on trouve l'interface suivante



Pour faire une appli sans console, il faut sélectionner window based(hide console) et pour le script location on sélectionne le main.py. Il faut aussi ajouter un fichier config dans settings pour bien set les choses. Le fichier config est un fichier json :

*[Dans le cadre de la conversion d'un script Python en un exécutable avec `auto-py-to-exe`, un fichier JSON de configuration peut être utilisé pour stocker et réutiliser les paramètres de configuration. Cela permet de sauvegarder la configuration de votre projet pour une utilisation future sans avoir à reconfigurer manuellement les options chaque fois que vous souhaitez générer l'exécutable. ]*

Si on le configure manuellement avec `auto-to-py-exe`, on peut toujours exporter la config utilisée en JSON file à l'aide du bouton "export..." :



## Storage spécial :

eMMC veut dire embedded MultiMediaCard. On l'appelle eMMC parce qu'elle est généralement soudée sur la carte mère de l'appareil. L'eMMC se trouve principalement dans les smartphones, mais elle peut également être utilisée dans les ordinateurs. Elle est populaire car, bien qu'elle offre une vitesse similaire à celle des disques SSD bas de gamme, elle est nettement moins chère. Les gens voulant construire des ordis à bas budget l'utilisent, mais le seul problème c'est qu'elle n'est disponible qu'en tailles de 32 Go à 128 Go.

Les disques SSD (solid state drive) sont nettement plus rapides que les disques durs, offrant des vitesses de 400 Mo/s à 3 Go/s, ils sont plus petits aussi que des disques durs. Les SSD ont des stockages allant de 128 Go à 1To max et sont plus chers que des disques durs. De ce fait, pour avoir un stockage max et avec un prix min, les gens optent plutôt pour des disques durs.

(cryptohack, CTF)

## Leet code :

1st problem :

## 1. Two Sum

Easy Topics Companies Hint

Given an array of integers `nums` and an integer `target`, return *indices of the two numbers such that they add up to target*.

You may assume that each input would have **exactly one solution**, and you may not use the *same* element twice.

You can return the answer in any order.

**Example 1:**

```
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].
```

**brilliant solution :**

```
class Solution:
    def twoSum(self, nums: List[int], target: int) -> List[int]:
        hashmap = {}
        for i in range(len(nums)):
            complement = target - nums[i]
            if complement in hashmap:
                return [i, hashmap[complement]]
            hashmap[nums[i]] = i
```

**Palindromes:**

## 5. Longest Palindromic Substring

Medium Topics Companies Hint

Given a string `s`, return *the longest palindromic substring* in `s`.

**Example 1:**

```
Input: s = "babad"
Output: "bab"
Explanation: "aba" is also a valid answer.
```

**Example 2:**

```
Input: s = "cbbd"
Output: "bb"
```

Palindromes are words that can be read the same from the two sides. To find them, we can use the core trick : noyau. An easy way to do it is to identify the cores (I only identified their indexes) composed of 2 or 3 letters, and then iterate on both sides of that core.

### vocab:

string/ascii to integer : atoi  
regex : regular expressions  
finite state machine python  
divide and conquer  
greedy approach :

## 11. Container With Most Water

Medium    Topics    Companies    Hint

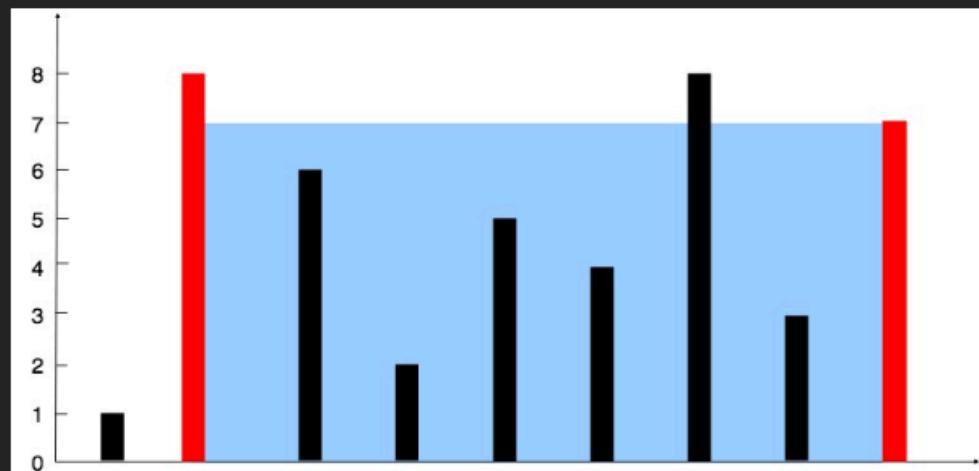
You are given an integer array `height` of length `n`. There are `n` vertical lines drawn such that the two endpoints of the `ith` line are `(i, 0)` and `(i, height[i])`.

Find two lines that together with the x-axis form a container, such that the container contains the most water.

Return *the maximum amount of water a container can store*.

**Notice** that you may not slant the container.

### Example 1:



**Input:** `height = [1,8,6,2,5,4,8,3,7]`

**Output:** 49

**Explanation:** The above vertical lines are represented by array `[1,8,6,2,5,4,8,3,7]`. In this case, the max area of water (blue section) the container can contain is 49.

### Example of a Greedy Algorithm:

Let's consider the **Fractional Knapsack Problem**:

- **Problem:** You have a knapsack with a weight capacity, and you need to fill it with items. Each item has a weight and a

value. The goal is to maximize the total value in the knapsack, and you can take fractions of items.

- **Greedy Algorithm :**

- Compute the value-to-weight ratio for each item.
- Sort items by this ratio in descending order.
- Start filling the knapsack with as much as possible from the item with the highest ratio, then move to the next item, and so on until the knapsack is full.

Python sorting lists specs :

- `L.sort()` : it sorts L in an ascending order, this action changes the list L.
- `sorted(L)` : it returns a sorted version of L without changing L itself.

```
>>> L
[-3, 1, 2, 5]
>>> L=[1,5,2,-3]
>>> sorted(L)
[-3, 1, 2, 5]
>>> L
>>> L=[1,5,2,-3]
>>> sorted(L)
[-3, 1, 2, 5]
>>> L
>>> L
[1, 5, 2, -3]
>>> L.sort()
>>> L
[-3, 1, 2, 5]
>>> 
```

HashMap 

A **hash map** is a data structure that stores key-value pairs, allowing for fast retrieval, insertion, and deletion of data. It's commonly used in programming because it provides average-case constant time complexity ( $O(1)$ ) for these operations. Exemple :

Le problème est le suivant :

## 15. 3Sum

Solved

Medium Topics Companies Hint

Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]] such that  $i \neq j$ ,  $i \neq k$ , and  $j \neq k$ , and  $nums[i] + nums[j] + nums[k] == 0$ .

Notice that the solution set must not contain duplicate triplets.

### Example 1:

**Input:** nums = [-1,0,1,2,-1,-4]

**Output:** [[-1,-1,2],[-1,0,1]]

**Explanation:**

$nums[0] + nums[1] + nums[2] = (-1) + 0 + 1 = 0$ .

$nums[1] + nums[2] + nums[4] = 0 + 1 + (-1) = 0$ .

$nums[0] + nums[3] + nums[4] = (-1) + 2 + (-1) = 0$ .

The distinct triplets are [-1,0,1] and [-1,-1,2].

Notice that the order of the output and the order of the triplets does not matter.

### Example 2:

**Input:** nums = [0,1,1]

**Output:** []

**Explanation:** The only possible triplet does not sum up to 0.

### Example 3:

**Input:** nums = [0,0,0]

**Output:** [[0,0,0]]

**Explanation:** The only possible triplet sums up to 0.

Une méthode est d'utiliser le two-pointers ou alors le hashmap. En gros le principe est de store toutes les valeurs de  $nums[i]$  comme key et leur indice comme value (on ne fait pas l'inverse car on souhaite éliminer les doublons du coup, si  $nums[i]$  et  $nums[j]$  ont la même valeur, on ne gardera que le plus grand entre les deux (car la list est sorted)). After that the lookup on the hashmap is so much more optimized because it's  $O(1)$ . En fait comme dans cette partie du code :

```
for(int j = i + 1 ; j < nums.size() - 1 ; ++j){      //Fixing another number  
after first number  
  
    int required = -1*(nums[i] + nums[j]);      //To make sum 0, we  
would require the -ve sum of both fixed numbers.  
  
    if(hashMap.count(required) && hashMap.find(required)->second  
> j){ //If it exists in hashmap and its last occurrence index > 2nd fixed  
index, we found our triplet.  
  
        answer.push_back({nums[i] , nums[j] , required});  
  
    }  
  
    j = hashMap.find(nums[j])->second; //Update j to last  
occurrence of 2nd fixed number to avoid duplicate triplets.
```

Il n'y a pas besoin d'itérer sur les indices pour retrouver si l'élément appartient ou pas. Avec une liste on peut aussi savoir si

un élément appartient à une liste ou pas mais on ne peut pas savoir son indice sans une boucle qui traverse toute la liste. Alors qu'ici avec une hashmap on récupère directement son indice car c'est la value.

Voici tout le code de hashmap pour ce problème , en C++, le `->second` désigne value et `->first` désigne le key.

```
class Solution {
public:
    vector<vector<int>> threeSum(vector<int>& nums) {
        sort(nums.begin() , nums.end());           //Sorted Array
        if(nums.size() < 3){                      // Base Case 1
            return {};
        }
        if(nums[0] > 0){                         // Base Case 2
            return {};
        }
        unordered_map<int , int> hashMap;
        for(int i = 0 ; i < nums.size() ; ++i){     //Hashing of Indices
            hashMap[nums[i]] = i;
        }
        vector<vector<int>> answer;
        for(int i = 0 ; i < nums.size() - 2 ; ++i){   //Traversing the array to fix the number.
            if(nums[i] > 0){             //If number fixed is +ve, stop there because we can't make it zero by searching aft
                break;
            }
            for(int j = i + 1 ; j < nums.size() - 1 ; ++j){ //Fixing another number after first number
                int required = -1*(nums[i] + nums[j]); //To make sum 0, we would require the -ve sum of both fixed numbers
                if(hashMap.count(required) && hashMap.find(required)->second > j){ //If it exists in hashmap and its index is greater than j
                    answer.push_back({nums[i] , nums[j] , required});
                }
                j = hashMap.find(nums[j])->second; //Update j to last occurrence of 2nd fixed number to avoid duplicates
            }
            i = hashMap.find(nums[i])->second;      //Update i to last occurrence of 1st fixed number to avoid duplicates
        }
        return answer; //Return answer vector.
    }
};
```

Pour générer toutes les combinaisons possibles de 3 chiffres (parmi 0-9), où chaque combinaison a seulement 2 éléments, vous pouvez utiliser la fonction `itertools.combinations_with_replacement` de Python. Cette fonction génère toutes les combinaisons possibles avec répétition.

### **Mutable ou non :**

Sur Python, si on applique une fonction sur une variable, la valeur de la variable ne change pas (non mutable). Par contre, si on

applique une fonction sur une liste, set ou dict, ils sont changés par la suite : mutables.

```
combinations.append(''.join(path))
```

- `''.join(path)` : Combine les éléments de `path` en une seule chaîne de caractères. Par exemple, si `path = ['a', 'b', 'c']`, alors `''.join(path)` produira la chaîne "abc".
- **Action** : Cette ligne ajoute la combinaison complète (`path` converti en chaîne) à la liste `combinations`.

`.join()` peut aussi être utilisé sur des entiers mais il faut les convertir d'abord. `''.join(blabla)` veut dire que les éléments de blabla se concatènent sans espace entre eux. On peut les concaténer en mettant un tiret par exemple : `'-'.join(blabla)`.

#### Pour utiliser `join()` avec d'autres types :

Si vous avez des éléments non `str` (comme des entiers) et que vous souhaitez les joindre, vous devez d'abord les convertir en chaînes de caractères.

#### Exemple avec conversion :

```
python
numbers = [1, 2, 3]
result = '-'.join(map(str, numbers))
print(result) # Sortie : "1-2-3"
```

Copier le code

```
>>> not "23"
False
>>> not ""
True
```

## Screens :

- Bleeding effect : When the screen is totally black, the edges are a little lighter. It is inevitable in LCD screens.



- IPS (In plane switching), VA(Vertical Alignment) and TN (Twisted Nematics) screens are all types of LED screens that are especially used by gamers. Pros and cons :

	TN	VA	IPS
Performance	Fastest: low response times, highest refresh rates, minimal motion blur; Low input lag	Longest response times typically; Higher refresh rates possible	Slower response times than TN, faster response times than VA; Gaming-quality refresh rates are rare
Display	Worst viewing angles; Worst color	Viewing angles typically better than TN, worse than IPS; Good color; Best contrast; Best image depth	Best viewing angles; Best color
Pricing	Cheapest	Pricier models can have performance comparable to TN	Most expensive
Best Use	Gaming	General Use	Professional

- LCD screens :

LCD stands for Liquid Crystal Display. So how do they work?

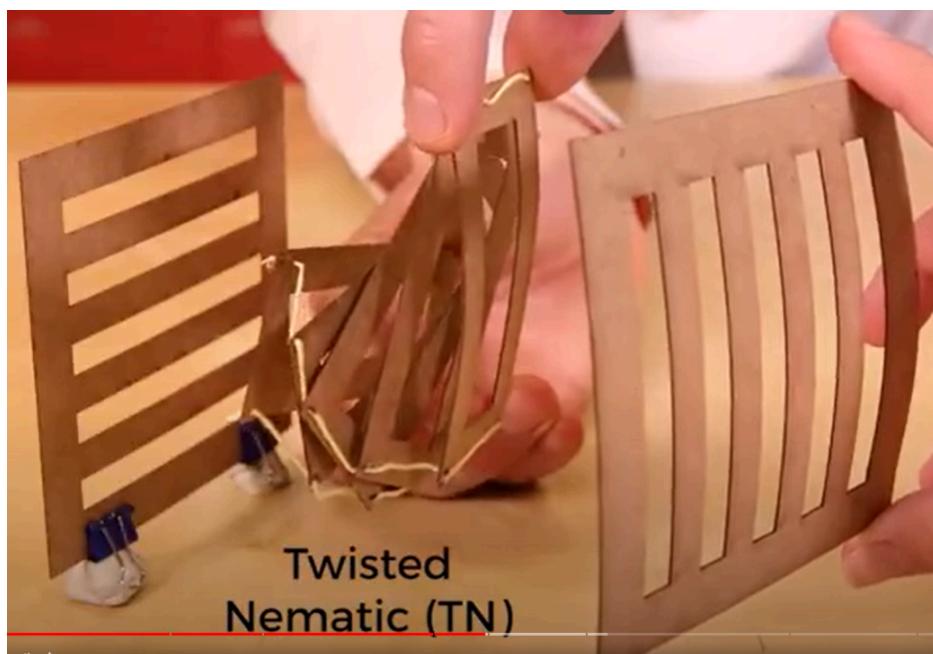
(Funny thing : two sunglasses put in perpendicular show nothing, cos both of them are initially oriented horizontally, so when we add the 90 degree flip, light is completely lost)

Liquid crystals flip the polarization of light . They have different properties at different temperatures. The important one for displays is nematic. the nematic phase is a phase in which crystals align with each other along the long axis :



Nematic  
phase

In the nematic crystal liquid phase, if light is polarized horizontally at the entry of the twisted crystal then it comes out polarized vertically if the end of the twisted crystal is vertical.



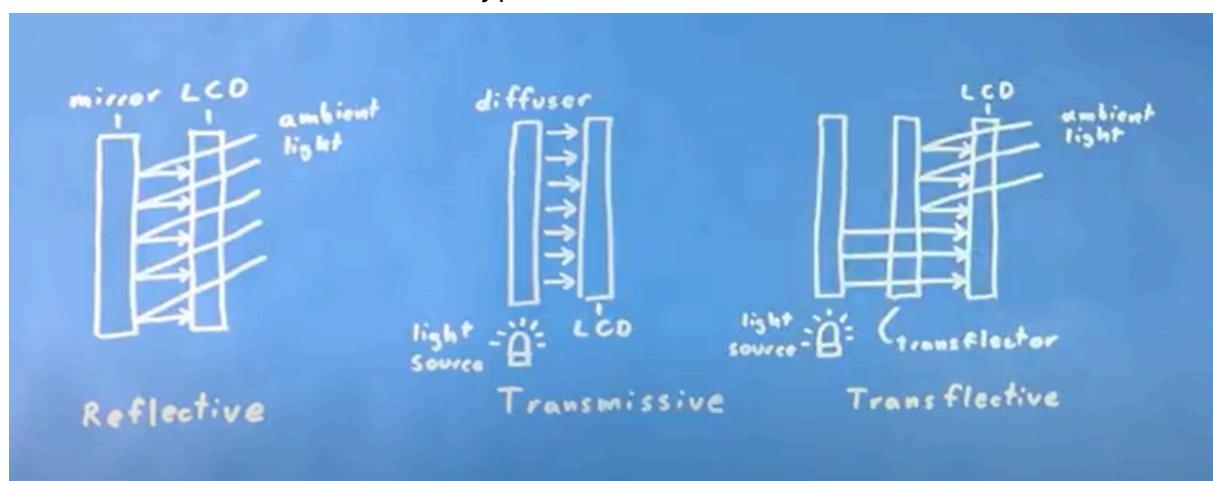
Though if a voltage is applied at the crystal then the twisted structure breaks up and untwists :

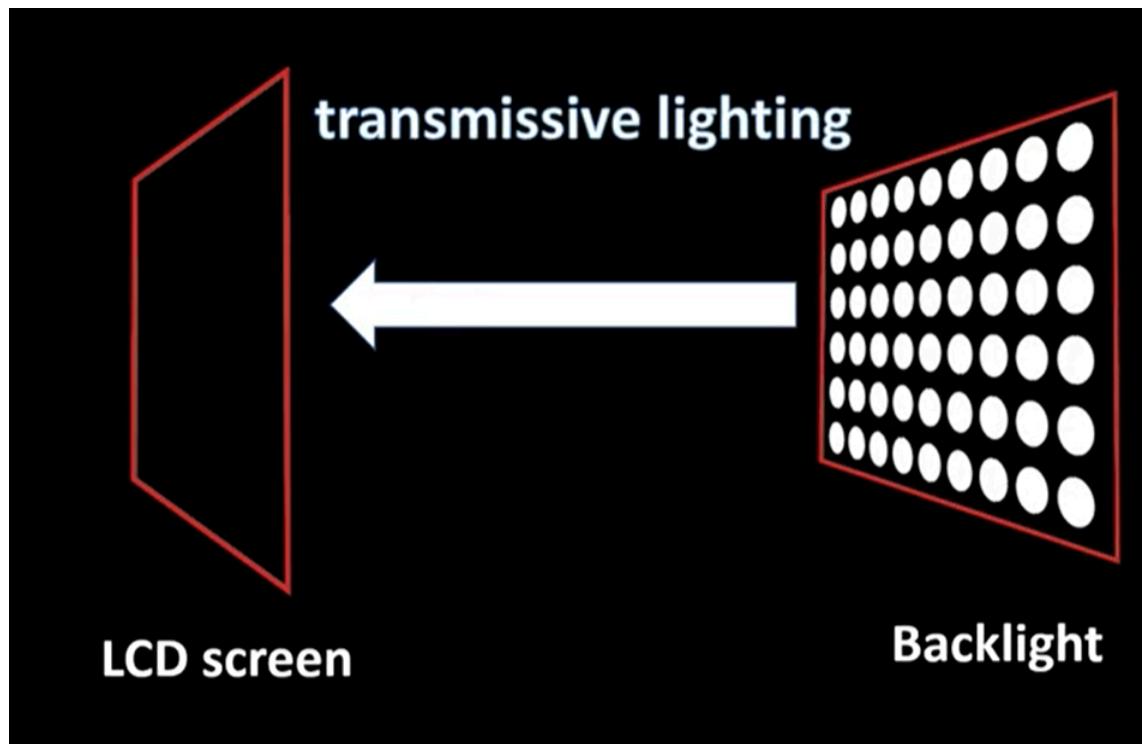


1. So in the first case (first image), if we have a screen emitting unpolarized light from the rear of the structure (before the horizontal polarizer) then it gets polarized vertically at the end.
2. In the second case (voltage applied), the light passes through the horizontal polarizer and through the untwisted crystal but gets blocked by the vertical polarizer.

One color LCD is a RGB unit. This means that a pixel contains 3 subpixels , each of which contains a different color and is an individually controlled segment (2 polarizers + twisted nematic). After the second filter we put in a Blue, Green or Red color filter over the subpixel. Their mix and combination with other pixels make the image.

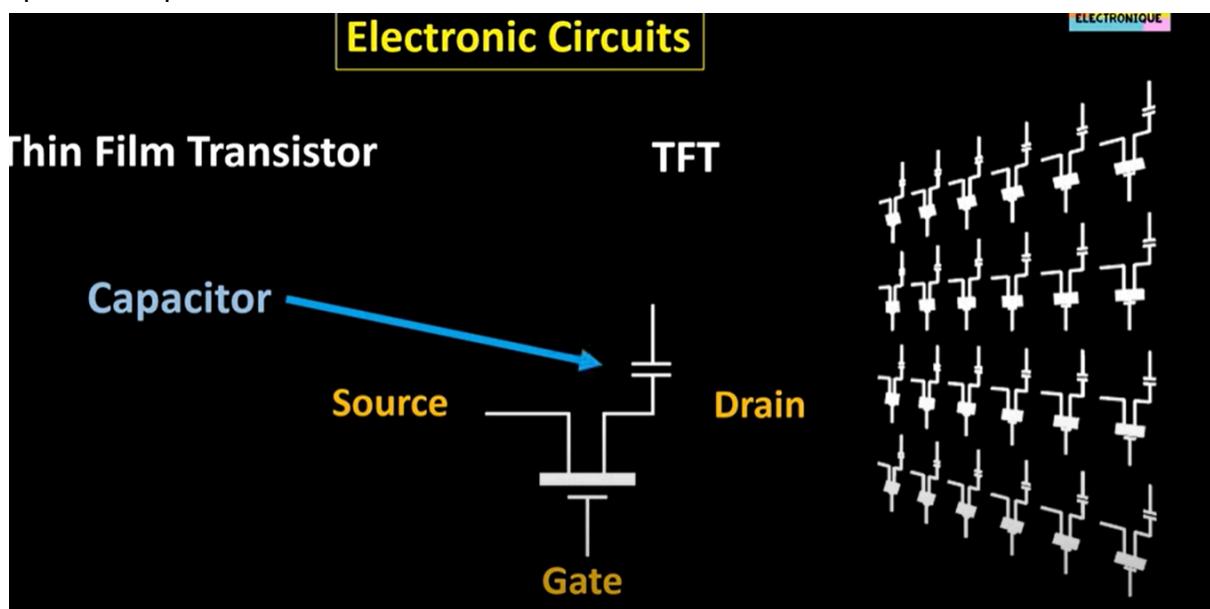
There are three different types of LCDs :

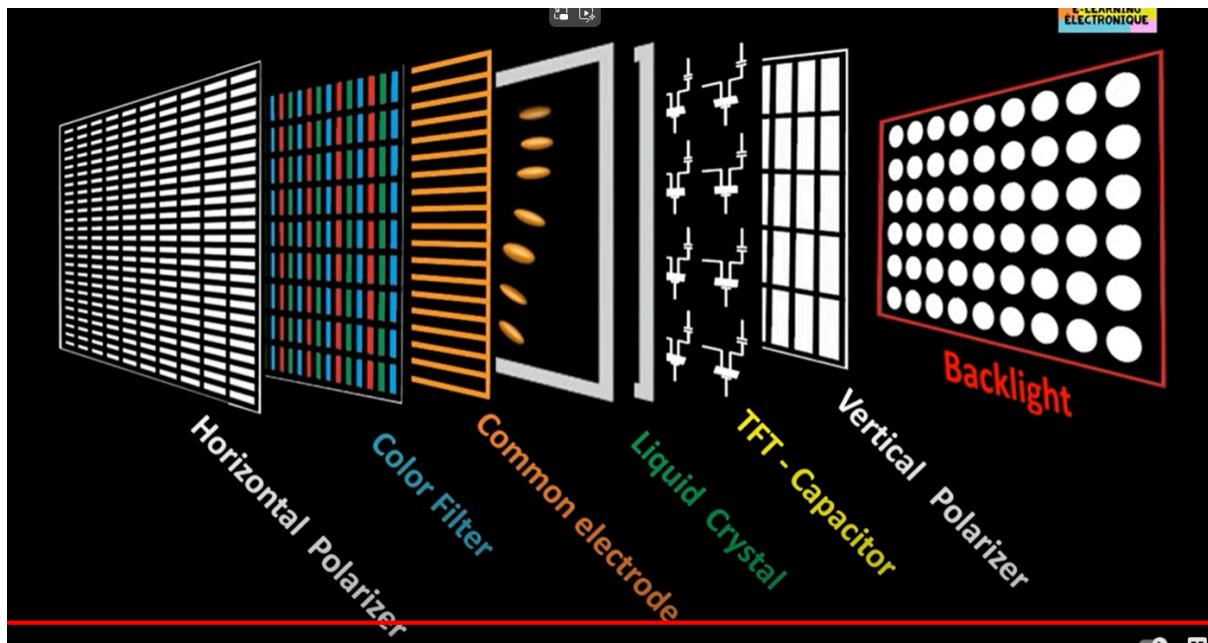




- Une calculatrice est réfléchissante (you need light to see it, in the dark it's impossible to see). Réflective LCD are usually cheaper because they don't require backlight.
- Un tel/pc est transmissif. They require a backlight, so in the dark they can still emit light.
- Les transflectives (the best of both worlds) sont trop chers.

Each subpixel is controlled by a type transistor called TFT (Thin Film Transistor), to allow or not the emission of the color in that specific pixel.





### **Appellation par référence et par copie :**

Par référence :

Supposons que je manipule des listes chaînées, construite comme suit :

```
# Definition for singly-linked list.
# class ListNode(object):
#     def __init__(self, val=0, next=None):
#         self.val = val
#         self.next = next
```

Alors pour construire une nouvelle liste l3 par exemple, il faudra en créer une copie :

```
l3 = ListNode()
current = l3
while condition is True :
    current.next = ListNode(valeur)
#Ceci est fait dans current est dans l3, car c'est par référence. On avance dans l3 et current simultanément
    current = current.next
#Ici on ditch le head de current mais il est toujours sur l3
    print(current)
    print(l3)
```

Si on print ça :

```
ListNode{val: 0, next: ListNode{val: 7, next: None}}
ListNode{val: 0, next: ListNode{val: 7, next: None}}
ListNode{val: 7, next: ListNode{val: 0, next: None}}
ListNode{val: 0, next: ListNode{val: 7, next: ListNode{val: 0,
next: None}}}
ListNode{val: 0, next: ListNode{val: 8, next: None}}
ListNode{val: 0, next: ListNode{val: 7, next: ListNode{val: 0,
next: ListNode{val: 8, next: None}}}}
```

On peut faire la même chose avec les listes mais en créant une deepcopy au lieu d'une affectation normale.

Les autres affectations( pour les entiers, les listes ...) sont faites par copie. exemple :

```
a= 10
b = a
a = 5
ici b est toujours égale à 10.
```

```
>>> a=10
>>> b=a
>>> b
10
>>> a
10
>>> a= 5
>>> b
10
>>> a
5
```

## Stack :

La structure Stack est une structure LIFO (Last in first out).

### Key Operations on a Stack

1. **Push:** Add an element to the top of the stack.
2. **Pop:** Remove the top element from the stack.

3. **Peek (or Top)**: Look at the top element of the stack without removing it.
4. **isEmpty**: Check if the stack is empty.

### Tri par comptage :

C'est une méthode de tri qui marche surtout quand tous les éléments qu'on a à trier sont dans un intervalle assez restreint.

1. Créer le tableau de comptage :

On parcourt toute la liste et on récupère son max et son min. Après on peut créer le tableau de comptage de taille  $\text{max} - \text{min} + 1$ .

2. Parcourir la liste et compter les occurrences:

On parcourt la liste et on incrémente la valeur dans le tableau de comptage à l'indice correspondant. c'est à dire pour un élément  $x$  on le met dans la case du tableau comptage d'indice : élément - min.

3. Reconstruire la liste :

On parcourt le tableau de comptage et on peut construire la liste car on sait chaque élément apparaît combien de fois d'après la liste comptage.

Complexité :  $O(n+k)$ , où  $n$  est le nombre d'éléments de la liste principale et  $k$  le décalage entre min et max.

### Exemple :

1. Exemple de la liste [4, 2, 2, 8, 3, 3, 1]
2. comptage = [0, 0, 0, 0, 0, 0, 0] (8 éléments)
3. Pour chaque élément de la liste :
  - 4 correspond à l'indice  $4-1=3$  dans le tableau de comptage. On incrémente `comptage[3]` :  
`comptage = [0, 0, 0, 1, 0, 0, 0, 0]`

4. Après avoir parcouru tous les éléments de la liste, notre tableau de comptage est le suivant : `comptage = [1, 2, 2, 1, 0, 0, 0, 1]`

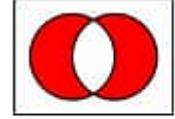
### 5. Interprétation du tableau de comptage

- L'indice 0 (qui correspond à la valeur 1) contient 1, ce qui signifie que la valeur 1 apparaît 1 fois dans la liste.
- L'indice 1 (qui correspond à la valeur 2) contient 2, ce qui signifie que la valeur 2 apparaît 2 fois.
- L'indice 2 (qui correspond à la valeur 3) contient 2, ce qui signifie que la valeur 3 apparaît 2 fois.
- L'indice 3 (qui correspond à la valeur 4) contient 1, ce qui signifie que la valeur 4 apparaît 1 fois.
- Les indices 4, 5 et 6 (qui correspondent aux valeurs 5, 6 et 7) contiennent tous 0, ce qui signifie que ces valeurs n'apparaissent pas dans la liste.
- L'indice 7 (qui correspond à la valeur 8) contient 1, ce qui signifie que la valeur 8 apparaît 1 fois.

## Bits and operators :

The XOR gate (^) :

a	b	$a \wedge b$
-	-	-
0	0	0
0	1	1
1	0	1
1	1	0



Fonction OU exclusif

S'ils sont différents on a 1, sinon on a 0. Souvent pour inverser un binaire (faire un NOT au binaire), on le XOR avec un nombre binaire composé de 111...1. Pour faire cela, on donne l'exemple suivant :

```

int findComplement(int num) {
    // Find the number of bits in num's binary representation
    int mask = 1;
    while (mask < num) {
        mask = (mask << 1) | 1; // Create a mask of all 1's in the same length as num, on fait "ou" 1
        à chaque fois pour ajouter un 1 à la fin du mask.
    }
    // Return the complement by XORing num with mask
    return num ^ mask; // l'opération XOR
}

```

### The bitwise NOT operator ~ :

C'est un complément à 2. L'opérateur fait ceci :

`~x = - x - 1`

du coup  $\sim 20 = -21$

A priori ce qu'il fait c'est qu'il inverse tous les bits même ceux qui sont tout à gauche et qui servent à s'étaler sur 64 bits :

The diagram illustrates the conversion of the decimal number 20 to its two's complement representation, -21. It shows the following steps:

- Bitwise Complement:** The value  $a = 20$  is given.
- Binary Representation:** The binary representation of 20 is shown as  $0\ 0001\ 0101$ .
- Bitwise NOT (~a):** The bitwise NOT of 20 is calculated. The result is  $1\ 1101011$ , which is circled in red.
- Two's Complement:** The two's complement of 20 is calculated by adding 1 to the bitwise NOT result. This is shown as  $1\ 1101011 + 1$ , resulting in  $1\ 1101100$ .
- Final Result:** The final result is circled in red and labeled as  $-21$ .

1 is the binary form of 21.

2 is the flip of that value.

and then we add 1, to make the two complementary.

Bitwise left shift :

`x<<n` : ça déplace les bits de deux vers la gauche. x doit être int et ça retourne aussi un int.

## Pointeurs et tableaux en C :

`int arr[5];` définit un tableau de 5 entiers.

Quand on a défini le tableau de cette manière, alors `arr` est en fait un pointeur vers le premier élément de ce tableau.

Ainsi, `arr` est équivalent à `&arr[0]`, et `*arr` est équivalent à `arr[0]`.

### **Pointeur simple (\*) :**

- Un pointeur est une variable qui stocke l'adresse d'une autre variable.
- Par exemple, `int *ptr;` est un pointeur vers un entier.
- Si `ptr` pointe vers `arr`, alors `*ptr` donne l'accès à la valeur de `arr[0]`

### **Pointeur vers un tableau (ou pointeur vers pointeur) :**

- Si vous avez un tableau à deux dimensions, comme `int arr[3][4];`, vous pouvez avoir un pointeur qui pointe vers ce tableau.
- `int (*ptr)[4];` est un pointeur vers un tableau de 4 entiers.
- `ptr = arr;` assigne à `ptr` l'adresse du premier sous-tableau (`arr[0]`).

### **Pointeur double (\*\*) :**

- Un pointeur double est un pointeur qui pointe vers un autre pointeur.

- Par exemple, `int **ptr2;` est un pointeur qui pointe vers un autre pointeur d'entier.
- C'est utile pour gérer des tableaux dynamiques ou des matrices (tableaux à deux dimensions dynamiques).
- Exemple : vous pourriez avoir `int **matrix;` où chaque élément de `matrix` est un pointeur vers une ligne du tableau.

A very good example of bit manipulation :

## 190. Reverse Bits

Solved 

[Easy](#) [Topics](#) [Companies](#)

Reverse bits of a given 32 bits unsigned integer.

**Note:**

- Note that in some languages, such as Java, there is no unsigned integer type. In this case, both input and output will be given as a signed integer type. They should not affect your implementation, as the integer's internal binary representation is the same, whether it is signed or unsigned.
- In Java, the compiler represents the signed integers using 2's complement notation. Therefore, in **Example 2** above, the input represents the signed integer `-3` and the output represents the signed integer `-1073741825`.

**Example 1:**

**Input:** `n = 00000010100101000001111010011100`  
**Output:** `964176192 (00111001011110000010100101000000)`  
**Explanation:** The input binary string `00000010100101000001111010011100` represents the unsigned integer 43261596, so return 964176192 which its binary representation is `00111001011110000010100101000000`.

**Example 2:**

**Input:** `n = 11111111111111111111111111111101`  
**Output:** `3221225471 (10111111111111111111111111111111)`  
**Explanation:** The input binary string `11111111111111111111111111111101` represents the unsigned integer 4294967293, so return 3221225471 which its binary representation is `10111111111111111111111111111111`.

## The idea and the code

```
Python3 ▾ 🔒 Auto

1 class Solution:
2     def reverseBits(self, n: int) -> int:
3         result = 0 #elle va contenir la lsite inversée
4         for i in range(32):
5             #Décalage de result et extraction d'un bit de n
6             result = (result << 1) | (n & 1) #le "et" sert a selectionner a chaque fois seulement le dernier bit
7             n >>= 1 #on al'avance de a à chaque fois pour recuperer le dernier bit et crop les autres
8             #result << 1 : On décale les bits de result d'une position vers la gauche pour faire de la place pour le prochain bit.
9             #n & 1 : Cela extrait le bit le moins significatif (le bit le plus à droite) de n. L'opération & (ET bit à bit) avec 1
10            #masque tous les bits de n sauf le dernier, ce qui permet de récupérer ce dernier bit.
11            #(result << 1) | (n & 1) : Ensuite, le bit extrait est ajouté à result avec l'opération | (OU bit à bit), qui place ce bit
12            #à la nouvelle position la plus à droite de result.
13
14     return result
```

Pour sélectionner le dernier bit à chaque fois d'un binaire :

```
dernier_bit = n&1
```

Pour insérer ce bit à chaque fois dans la position la plus à droite, il faut d'abord décaler : résultat\_décalé = result<<1 et après l'insérer en faisant un "ou" : résultat\_décalé | dernier\_bit.

Après avoir récupéré le dernier bit on en a plus besoin donc cette fois on redécale à droite pour crop notre nombre.

## Comment arranger les dossiers du Bureau :

```
import os

import shutil

bureau = os.path.join(os.path.expanduser("~/"), "Downloads")

# os.path.expanduser("~/") sert à trouver le chemin du
répertoire utilisateur de l'ordinateur

# pour moi sur windows ce serait par exemple :
C:\Users\NomUtilisateur
```

```
# os.path.join() sert à combiner les deux parties du chemin :  
# la premier de expanduser et celle de Desktop pour tomber  
# direct sur le bureau  
  
# Dictionnaire pour mapper les extensions aux dossiers  
  
extensions_dossiers = {  
  
    "Images": [".jpg", ".jpeg", ".png", ".gif", ".bmp",  
    ".svg"],  
  
    "Documents": [".pdf", ".doc", ".docx", ".txt", ".xls",  
    ".xlsx", ".ppt", ".pptx"],  
  
    "Vidéos": [".mp4", ".mov", ".avi", ".mkv", ".flv"],  
  
    "Musique": [".mp3", ".wav", ".aac", ".flac"],  
  
    "S": [".zip", ".rar", ".tar", ".gz", ".7z"],  
  
    "Scripts": [".py", ".js", ".html", ".css", ".sh", ".bat"],  
  
    "Dump": [],  
  
}  
  
# Créer les dossiers dans le bureau  
  
for dossier in extensions_dossiers:  
  
    chemin_dossier = os.path.join(  
  
        bureau, dossier  
  
    ) # précise l'le chemi du dossier qui doit etre dans le  
bureau  
  
    if not os.path.exists(chemin_dossier): # check si le  
dossier existe déjà  
  
        os.makedirs(chemin_dossier) # ceci crée les dossiers
```

```
for fichier in os.listdir():

    bureau

): # liste tous les fichiers présent dans le repertoire
bureau, et on boucle sur eux

    chemin_fichier = os.path.join(bureau, fichier)

    # Vérifier que c'est un fichier (pas un dossier)

    if os.path.isfile(chemin_fichier):

        # Obtenir l'extension du fichier
        _, extension = os.path.splitext(fichier)

        # Trouver le dossier correspondant à l'extension
        dossier_cible = "Dump" # Par défaut, mettre dans Dump

        for (
            dossier,
            extensions,
        ) in extensions_dossiers.items(): # récupère les
lignes dans un json

            if extension.lower() in extensions:

                dossier_cible = dossier
                break

        # Déplacer le fichier dans le dossier correspondant
        destination = os.path.join(
```

```
bureau, dossier_cible, fichier

    ) # après avoir récupéré le fichier, on construit le
path du dossier cible

    shutil.move(
        chemin_fichier, destination
    ) # on move après le fichier dans le dossier cible

    print(f"Movement de {fichier} vers {dossier_cible}")

print("Nettoyage du bureau terminé !")
```

## **Jaspersoft :**

Embedded BI sert à insérer les données BI dans l'application elle-même sans devoir passer par l'application Power-BI.

C'est quoi Jaspersoft du coup? C'est une plateforme intégrée d'analyse et de rapports pour les designers de logiciels. Les end users sont des gens non techniques qui auront accès à des visualisations de data.

## **Differences entre Jaspersoft et DENODO :**

→ Denodo est une plateforme de visualisation des données qui permet de gérer, d'accéder et combiner des données venant de différentes sources (API, services web et bases de données ...).

→ Jaspersoft est une plateforme de reporting et de business intelligence permettant de créer et distribuer des rapports interactifs, des tableaux de bord à partir de différentes sources de données.

Donc à priori, Jaspersoft peut s'appuyer sur du data provenant de DENODO. Comment exactement ?

- **DENODO peut être la source de données pour Jaspersoft :**

Denodo peut unifier plusieurs données venant de sources différentes en les présentant comme une seule source de données logique.

**Connexion à Jaspersoft :** Jaspersoft peut se connecter à Denodo de la même manière qu'il se connectera à toute autre source de données (comme une base de données SQL). Denodo expose ses données virtualisées via des interfaces standard telles que JDBC, ODBC, ou des services web. Jaspersoft peut utiliser ces interfaces pour interroger Denodo et récupérer les données nécessaires pour créer des rapports et des tableaux de bord.

**Accès en temps réel :** Les rapports générés dans Jaspersoft peuvent s'appuyer sur des données en temps réel, car Denodo interroge les sources de données sous-jacentes au moment de la requête.

Bien qu'on puisse exécuter des requêtes SQL pour extraire et filtrer des données, les traitements SQL complexes (comme des procédures stockées, des fonctions utilisateur avancées, ou des jointures très complexes) doivent être exécutés au niveau de la base de données elle-même avant d'être importés dans Jaspersoft.

Jaspersoft ne remplace pas un système de gestion de base de données (SGBD). Il est conçu pour l'extraction, la visualisation, et le reporting des données, pas pour le traitement lourd ou la transformation complexe des données.

**Petit interlude sur JDBC et ODBC:**

*Terminology :*

JDBC : Java DataBase Connectivity

ODBC : Open DataBase Connectivity

Ce sont deux interfaces permettant aux applications de se connecter à des bases de données et d'interagir avec elles.

### **Qu'est-ce que JDBC ?**

- JDBC est une API en Java qui permet aux applications Java de se connecter à une base de données, d'exécuter des requêtes SQL, et de traiter les résultats retournés.
- Elle fournit un ensemble de classes et d'interfaces pour accéder à une base de données relationnelle de manière standardisée.

Il existe plusieurs bases de données (MySQL, Oracle, PostgreSQL) et JDBC fonctionne en utilisant des pilotes spécifiques pour chaque base de données.

### **Qu'est-ce que ODBC ?**

- ODBC est une API standard pour accéder aux bases de données relationnelles, indépendamment du système d'exploitation, du langage de programmation, ou du SGBD (Système de Gestion de Base de Données) utilisé.
- Il permet à une application de se connecter à une base de données en utilisant un pilote ODBC qui traduit les commandes ODBC en commandes spécifiques à la base de données.

## **SELENIUM :**

### **from selenium import webdriver :**

- Cet import permet d'accéder à l'objet `webdriver`, qui est le point de départ pour lancer un navigateur automatisé (comme Chrome, Firefox, Edge, etc.). En utilisant `webdriver`, on peut créer une instance du navigateur souhaité.

### **from selenium.webdriver.edge.options import Options :**

- Cet import est utilisé pour configurer les options spécifiques du navigateur Microsoft Edge. Par exemple, on peut définir si le navigateur doit être lancé en mode sans tête (sans interface graphique), définir les arguments de ligne de commande, etc.

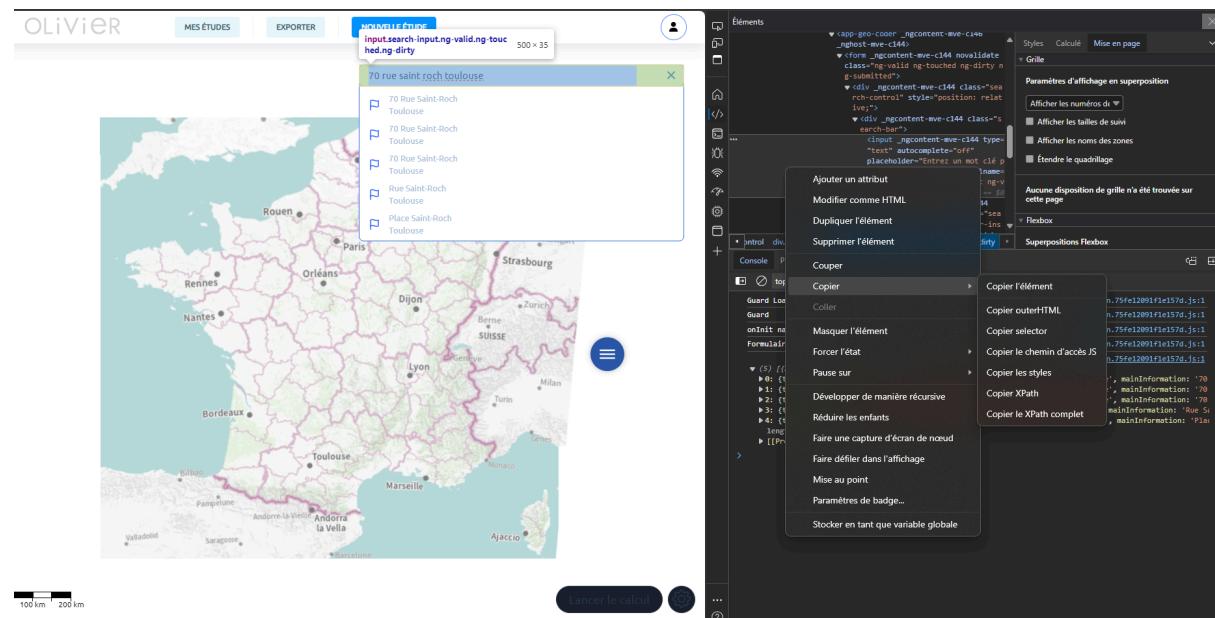
`from selenium.webdriver.edge.service import Service as EdgeService :`

- Cet import permet de gérer le service Edge pour contrôler le navigateur via Selenium. Cela peut être utile pour spécifier le chemin vers le pilote (driver) Edge ou pour configurer le service de manière spécifique.

`from selenium.webdriver.common.by import By :`

- Cet import fournit des méthodes pour localiser des éléments HTML sur une page web, comme `By.ID`, `By.XPATH`, `By.CSS_SELECTOR`, etc. Ces méthodes sont utilisées pour dire à Selenium où trouver un élément sur la page.

Pour récupérer le XPath sur le site internet 



Il suffit de cliquer sur la case où on veut remplir.

Méthodes avec selenium (.find\_element(attribut, "") et send\_keys("") :

```
driver = webdriver.Edge(service=service,
options=edge_options)

driver.get("https://olivier.cachalot.had.enedis.fr/")

time.sleep(30)

elem = driver.find_element(
    By.XPATH,
    "/html/body/app-root/nb-layout/div/div/div/div/nb-layout-column/app-map/div[4]/app-geo-coder/form/div/div/input",
)

elem.send_keys("Texte à insérer dans la case") #Pour
insérer

time.sleep(20)
```

### **dataframe special -Pandas :**

Si j'ai un dataframe et que je veux récupérer certaines colonnes, comment faire?

Alors déjà comment récupérer un dataframe? Il faut utiliser le module pandas de python et pip install openpyxl:

```
import pandas as pd
```

```
df = pd.read_excel(path)
```

après disons que je veux récupérer des colonnes exactes :

```
df = df[["colonne 1", "colonne 2", "colonne 3"]]
```

Pour retourner le résultat en liste de cette forme :

```
[[element1.colonne1,element1.colonne2,element1.colonne3],  
[element2.colonne1,element2.colonne2,element2.colonne3]].
```

Il suffit d'utiliser :

```
return df.values.tolist()
```

autre façon:

```
liste_de_listes = [  
    [int(row["colonne_1"]), float(row["colonne_2"]),  
     float(row["colonne_3"])]  
        for index, row in df.iterrows()  
    ]
```

if else sur une seule ligne :

```
t = 1 if element[9] == element[10] else 0
```

## Selenium et javascript

Nous pouvons exécuter direct des scripts javascript avec `driver.execute_script(...)`:

Cette méthode de Selenium permet d'exécuter du code JavaScript directement dans le contexte de la page Web chargée. Elle est utile pour réaliser des actions plus complexes qui ne sont pas directement accessibles via les commandes Selenium standard.

```
-> self.logo_label = ctk.CTkLabel( frame1,  
image=self.logo, text="", fg_color="transparent" ) ici du  
coup que veut dire frame 1 pour ce logo_label
```

### frame1 :

- `frame1` est un widget de type `Frame` (cadre) qui est utilisé comme conteneur pour organiser d'autres widgets à l'intérieur de lui.
- Le label `self.logo_label` est un enfant de `frame1`, ce qui signifie qu'il sera placé à l'intérieur de ce cadre spécifique.

## Pour retirer l'info formatée d'une certaine manière :

### 1. Utiliser `split()` pour séparer la date et l'heure

Si tu es sûr que le format est toujours `date heure`, tu peux simplement diviser la chaîne à l'aide d'un espace pour récupérer la partie vde la date.

```
date_heure = "07/01/2024 00:00:00"  
  
# Séparer la date de l'heure  
  
date_seule = date_heure.split()[0]  
  
print(date_seule) # Résultat : 07/01/2024
```

### 2. Utiliser `datetime` pour manipuler des dates de manière plus robuste

Si tu veux gérer différents formats de date ou faire d'autres opérations, tu peux utiliser la bibliothèque `datetime`.

```
from datetime import datetime
```

```
date_heure = "07/01/2024 00:00:00"

# Convertir la chaîne en objet datetime

date_obj = datetime.strptime(date_heure, "%d/%m/%Y %H:%M:%S")

# Récupérer uniquement la date dans le format souhaité

date_seule = date_obj.strftime("%d/%m/%Y")

print(date_seule) # Résultat : 07/01/2024
```

- Pour changer de pages en tkinter : tkraise
- Comment écrire un texte string comme ceci sans que les guillemets me mettent un problème : text= "qjgioqegqejn "hsh" hvi" :

### solution 1 :

```
text = """qjgioqegqejn "hsh" hvi"""

print(text)
```

### solution 2

```
text = 'qjgioqegqejn "hsh" hvi' print(text)
```

### solution 3

```
text = "qjgioqegqejn \"hsh\" hvi" print(text)
```

## Selenium :

```
driver = webdriver.Edge

driver.get("https://website")
```

```
search_bar = WebDriverWait(driver, 15).until(  
EC.visibility_of_element_located(( By.XPATH,"/htmlxpath", )))  
  
search_bar.send_keys(input)  
  
element =WebDriverWait(driver, 20).until(  
  
    EC.element_to_be_clickable( ( By.XPATH, "the  
xpath", )))  
  
element.click()
```