



ACADÉMIE  
DE GRENOBLE

*Liberté  
Égalité  
Fraternité*



Réalisation en cours de formation

## Développement applicatif & Programmation logiciel



0 - Sommaire :

## Sommaire

- 1 – Langage de programmation vu en classe (*Page 3 → Page 4*)
  - 1.1 Langage C# (C Sharp) avec Microsoft Learn
  - 1.2 Langage Python
  - 1.3 Utilisation de WinDev avec langage WLangage par PC Soft
  - 1.4 Langage de requête structuré SQL avec SSMS par Microsoft
  
- 2 – Gestion de Projet de 1<sup>ère</sup> année (*Page 4 → Page 7*)
  - 2.1 Projet choisi
  - 2.2 Utilisation de Gantt Project & Project Libre
  - 2.3 Résultat
  
- 3 – Développement du Portfolio (*Page 8 → Page 9*)
  - 3.1 Objectifs de la réalisation
  - 3.2 Langage utilisé
  - 3.3 Résultat

## 1 – Langage de programmation vu en classe :

### - 1.1 Langage C# (C Sharp) avec Microsoft Learn

Le premier langage de programmation vu en classe a été le langage C#, un langage très puissant qui est compilé directement dans l'IDE (Environnement de développement). Pour ma part, celui que j'ai toujours utilisé est VSC (Visual Studio Code) très pratique pour tout type de langage. On a suivi les cours en ligne de Microsoft Learn et j'ai en parallèle fait la certification MOOC C# de SoloLearn :



### - 1.2 Langage Python

On a également vu Python en mathématiques algorithmiques appliqué à l'informatique. Personnellement, j'en avais déjà fait précédemment pour des projets persos tels que des Bots Discord ou des scripts. J'ai aussi passé la certification MOOC Python de SoloLearn :



### - 1.3 Utilisation de WinDev avec langage WLangage par PC Soft

Lors de la première année on a brièvement abordé le WLangage de WinDev de manière à avoir le choix pour les projets **avenir**. Je n'ai pas tellement accroché dessus, malgré tout ça reste assez simple d'utilisation car c'est un langage qui peut s'écrire par des manipulations visuelles comme avec la création d'un bouton par exemple où des macros existent déjà.

### - 1.4 Langage de requête structuré SQL avec SSMS par Microsoft

Enfin, le SQL, ce fameux langage de requêtes dont j'ai eu l'occasion d'utiliser régulièrement en entreprise lors de mon alternance. On a vu les bases en classe mais j'avais commencé à l'apprendre car c'est le langage le plus important lorsqu'on utilise une base de données. Il existe des variétés de ce langage selon l'environnement choisi entre MySQL et SQL Server par exemple. Personnellement j'ai utilisé Microsoft SQL Server Management Studio car c'est celui dont j'ai l'habitude en **entreprise, j'ai** aussi passé la certification en ligne SoloLearn :



## 2 – Gestion de projet de 1<sup>ère</sup> année :

### - 2.1 Projet choisi

Il a fallu réaliser un projet, j'ai donc fait le choix de réaliser une calculatrice scientifique graphique en Python en utilisant la bibliothèque TKinter pour la partie graphique. L'objectif était surtout de gérer le projet, donc j'ai utilisé Gantt Project & Project Libre.



## - 2.2 Utilisation de Gantt Project & Project Libre

Mon projet à réaliser en 16 heures est une calculatrice graphique/scientifique, la partie scientifique était un objectif secondaire (bonus). Mon but principal était de réaliser l'interface graphique en premier temps (frontend), puis ensuite de passer sur les calculs logiques qui sont derrières, en backend. Les tâches sont écrites dans la réalisation de projet établi en avance par le chef de projet, moi-même :

		Nom	Durée	Début	Fin	Pré...	Noms des ressources
1		<b>Calculatrice graphique(python)</b>	81 jours	24/11/23 13:00	19/01/24 17:00		Anthony PETRICCA
2		Réserver du temps pour tester et faire des ajustements	0,25 jours	19/01/24 16:00	19/01/24 17:00	16;9	Anthony PETRICCA
3		<b>Faire l'interface graphique</b>	31 jours	24/11/23 13:00	15/12/23 17:00		Anthony PETRICCA;Visual Studio Code
4		Faire le design du support	0,5 jours	24/11/23 13:00	24/11/23 15:00		Anthony PETRICCA;Visual Studio Code
5		Créer la fenêtre principal	0,25 jours	24/11/23 15:00	24/11/23 16:00	4	Anthony PETRICCA;Visual Studio Code
6		1. Programmer l'affichage des résultats de la fenêtre principal	0,25 jours	24/11/23 16:00	24/11/23 17:00	5	Anthony PETRICCA;Visual Studio Code
7		2. Programmer l'affichage des résultats de la fenêtre principal	0,25 jours	15/12/23 13:00	15/12/23 14:00	6	Anthony PETRICCA;Visual Studio Code
8		créer les boutons (numérique + symbol)	0,25 jours	15/12/23 14:00	15/12/23 15:00	7	Anthony PETRICCA;Visual Studio Code
9		Optimisez le code de l'interface graphique et polissez l'apparence	0,5 jours	15/12/23 15:00	15/12/23 17:00	8	Anthony PETRICCA;Visual Studio Code
10		<b>Faire les calculs logiques</b>	10,75 jours	12/01/24 13:00	19/01/24 16:00		Anthony PETRICCA;Visual Studio Code
11		Etablir une logique de construction pour l'écriture des calculs	0,25 jours	12/01/24 13:00	12/01/24 14:00		Anthony PETRICCA;Visual Studio Code
12		Ecrire la base des calculs(addition, soustraction, multiplication, division)	0,25 jours	12/01/24 14:00	12/01/24 15:00	11	Anthony PETRICCA;Visual Studio Code
13		Gérer l'appel des fonction python par l'utilisation des boutons de l'interface graphique	0,5 jours	12/01/24 15:00	12/01/24 17:00	12	Anthony PETRICCA;Visual Studio Code
14		Tester le code et déboguer	0,25 jours	19/01/24 13:00	19/01/24 14:00	13	Anthony PETRICCA;Visual Studio Code
15		Améliorer le code en rajoutant d'autres calculs possibles	0,25 jours	19/01/24 14:00	19/01/24 15:00	14	Anthony PETRICCA;Visual Studio Code
16		Optimiser le script des calculs	0,25 jours	19/01/24 15:00	19/01/24 16:00	15	Anthony PETRICCA;Visual Studio Code

La première colonne des lignes ci-dessus représente : les ressources, la date et certains post-it qui sont des remarques supplémentaires sur la tâche, ici elles correspondent à un changement de date, par exemple :

Informations sur la tâche - 7

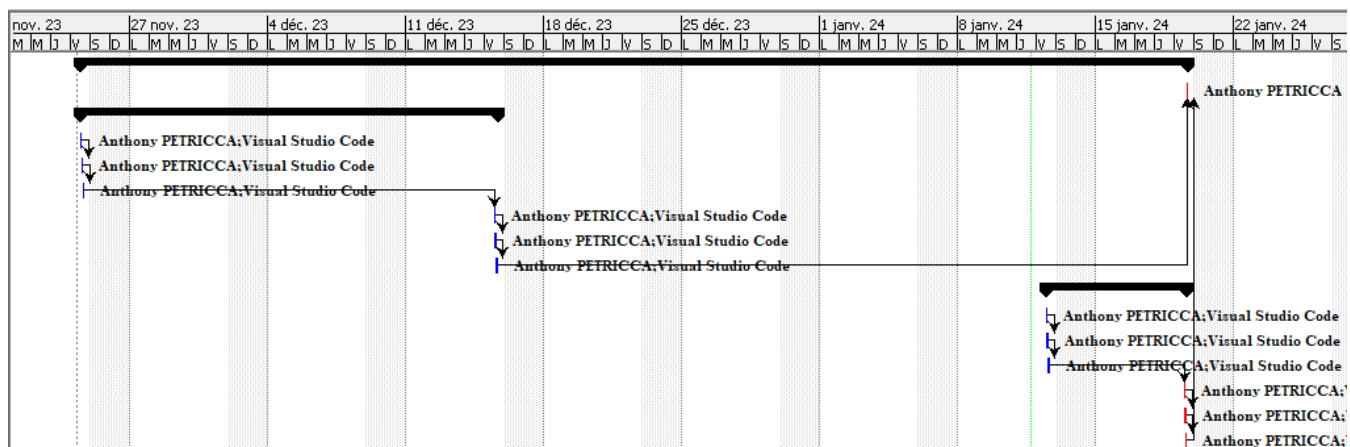
Général Prédécesseurs Successeurs Ressources Avancé Remarques

Nom: 2. Programmer l'affichage des résultats de la fenêtre principal

Remarques:

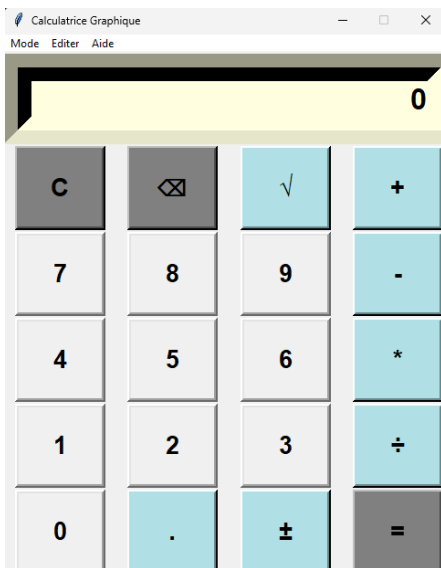
Tâche avancé au 12/12/23

Chaque sous parties regroupent en elles les tâches correspondantes à leurs énoncées dans l'ordre de réalisation. Une fois ces tâches établies par ordres de réalisation, on obtient alors un schéma représentant chaque tâche sur une échelle de temps :



## - 2.3 Résultat

Une fois le programme python écrit de A à Z, on obtient à son lancement une première page qui ressemble à cela :

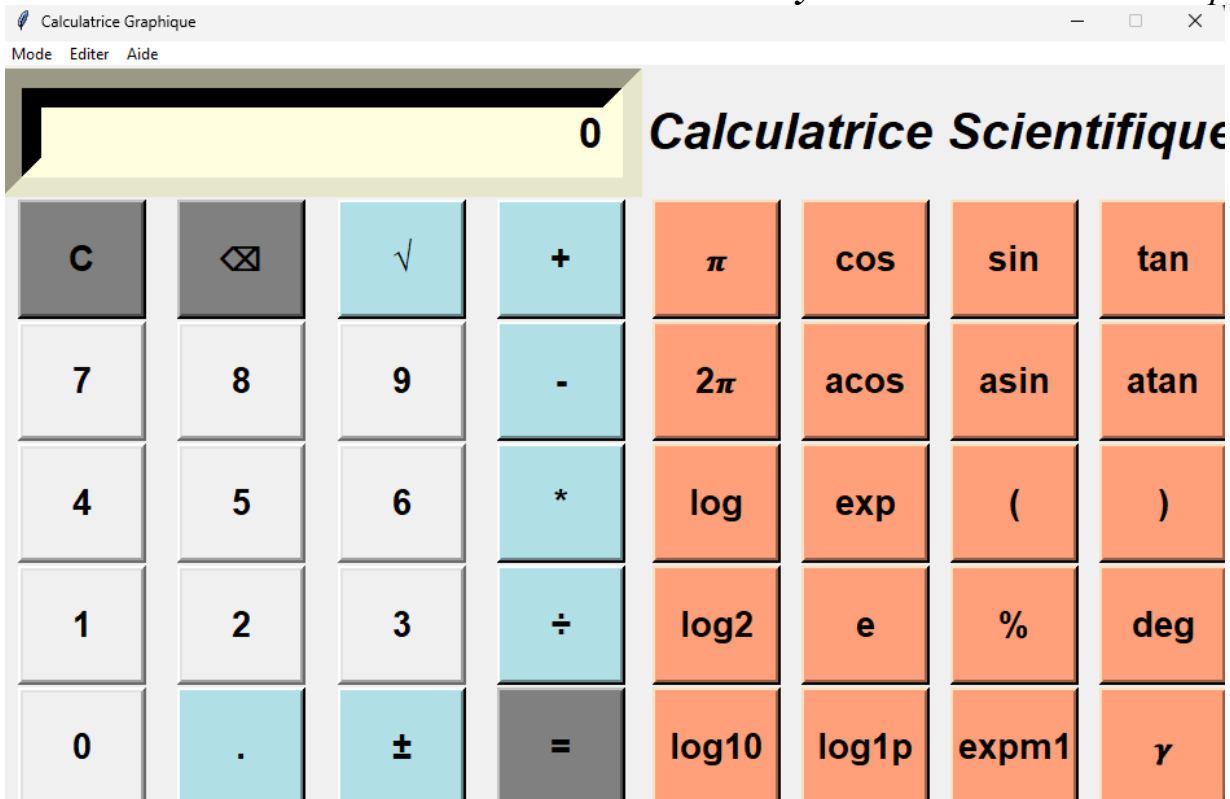


On y voit les calculs de bases tels que l'addition, la soustraction, la multiplication, la division + la racine carrée et les boutons de bases d'une calculatrice comme la suppression par caractères, la suppression totale et le bouton égal, la virgule des nombres décimaux et le +/-, tous fonctionnant parfaitement.

On remarque au-dessus 3 menus déroulant ayant chacun des tâches spécifiques et modifiables à volonté selon l'utilisation voulue. Exemple :



La calculatrice scientifique peut s'ouvrir grâce au menu déroulant « Mode » :



Extrait d'une partie (calcul logique) du code python :

```
def resultat(self):
    self.result = True
    if self.current:
        self.current = float(self.current)
        if self.check_sum:
            self.valid_function()
        else:
            self.total = eval(self.pending_operator + str(self.current))
            self.pending_operator = ''
            self.display(self.total)
    else:
        self.display("Erreur, opération invalide!")

def valid_function(self):
    print(f"Avant operation: total={self.total}, valeur={self.current}, oper={self.oper}")

    if self.oper == 'add':
        self.total += self.current
    elif self.oper == 'sous':
        self.total -= self.current
    elif self.oper == 'mul':
        self.total *= self.current
    elif self.oper == 'div':
        if self.current == 0:
            self.display("Erreur, 0 ne peut diviser !")
            return #Permet de sortir de l'instance de calcul
        else:
            self.total /= self.current
    elif self.oper == 'mod':
        self.total %= self.current

    print(f"Après operation: total={self.total}, valeur={self.current}, oper={self.oper}")

    self.input_value = True
    self.check_sum = False
    self.display(self.total)
```

Le total de lignes de ce script s'élève à 357.

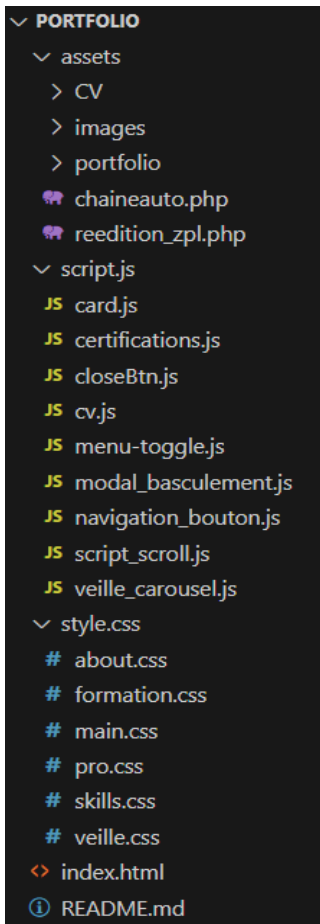
## 3 – Développement du Portfolio :

### - 3.1 Objectifs de la réalisation

L'épreuve E5 du BTS SIO correspond à la réalisation d'un support en ligne présentant les réalisations en cours de formation et dans la période en entreprise. Pour ma part j'ai décidé de développer mon Portfolio à la place d'utiliser des solutions en lignes tel que Wix ou Ionos auquel je n'avais aucune idée de leur fonctionnement tandis que j'avais déjà eu l'occasion de développer des petites pages Web en HTML (HyperText Markup Language) et en CSS (Cascading Style Sheets) pour le style.

### - 3.2 Langage utilisé

J'ai donc continué sur cette voie en ayant pour partie principal de mon Portfolio le fichier Index.html développer en HTTP et plusieurs fichiers CSS correspondant à chaque partie de mon Portfolio. J'ai ajouté du script.js (JavaScript) pour l'interaction et le visuel animé grâce à des ChatBot tel que Mirage, ChatGPT... et des tutos en lignes car pour le coup je ne suis pas familier avec ce langage. Mon architecture ressemble à ceci :





### - 3.3 Résultat

Mon portfolio est disponible au lien suivant : <https://antho-p.github.io/Portfolio-E5/>

