**TD 1**

**DEBUTER AVEC PYTHON**

**PARTIE I - ENVIRONNEMENT**

Nous allons utiliser l’environnement classique de Python pour apprendre à programmer. Cet environnement vous le trouverez sur le site officiel du langage <https://www.python.org/>. Vous pouvez aller sur « Downloads » et télécharger la version adaptée à votre système d’exploitation afin de l’installer sur votre ordinateur. Vous le ferez chez vous ! Pas en TD.

1. Connectez vous sur: https://jupyter.org/

2. Descendre et cliquer sur «Try it in your browser»

3. Puis cliquer sur «Try Classic Notebook» (temps d’attente moyen 20 secondes)

4. Dans la barre du haut, cliquer sur File, puis New Notebook, puis python 3

5. Taper votre code python dans le champ après Entrée[ ]

6. Une fois le code tapé, exécuter le (faite le marcher) en cliquant sur «Exécuter»

7. Une fois que votre programme (votre code) python est exacte (que l’execution du programme

donne les résultats attendus) il faut le sauvegarder. Pour cela cliquer sur File, puis download as, puis

Python et enfin sur Ok. Le fichier s’appellera «untitled.py» et il ser certainement sauvegarder dans

le dossier «téléchargement». Vous pouvez ensuite ouvrir ce fichier avec nimporte quel éditeur de

texte pour vérifier son contenu, puis modifier son nom (par exemple TD1Ex3.py).

Pour appréhender l’environnement, regardez la vidéo sur eCampus et suivez pas à pas le protocole. A la fin de la vidéo vous aurez : créé un répertoire de travail, ouvert l’environnement Python, créé et sauvegardé votre premier script Python dans votre répertoire de travail.

**PARTIE II - LES VARIABLES**

**Exercice 2.1 : Prise en main de l’environnement interactif.**

Ecrivez les lignes suivantes dans la fenêtre interactive **IDLE Shell**. Ecrivez le résultat qui s’affiche lorsque vous exécutez le programme :

>>> largeur = 12

>>> hauteur = 5.2 \* 9

>>> print(largeur \* hauteur)

*Résultat ? 46.8*

>>> a = 7

>>> b = 12.1

>>> y = 3 \* a + b / 5

>>> print(y)

*Résultat ? 23.42*

>>> y = 3 \* (a + b) /5

>>> print(y)

*Résultat ? Erreur de syntaxe*

>>> r = 7

>>> pi = 3.14159  
>>> s = pi \* r\*\*2   
>>> print(s)

*Résultat ? 153.93791*

>>> print(type(r), type(pi), type(s))

*Résultat ? int, float, float*

>>> print(27 + 5)

*Résultat ? 32*

>>> h, m, s = 12, 20, 47

>>> print("nombre de secondes écoulées depuis minuit = ", h\*3600 + m\*60 + s)

*Résultat ? nombre de secondes écoulées depuis minuit = 44447*

**Exercice 2.2 : affectation dynamique.**

Recopiez ce code dans la fenêtre de script **untitled**, enregistrez-le puis exécutez-le.

#mon premier script Python

nom = input('Quel est ton nom ? ')

print ('Bonjour', nom)

**Modifiez** le script pour que l’utilisateur entre son nom **et son prénom** et affiche **les deux**.

Exemple de fonctionnement du script :

Quel est ton nom ? Houlert

Quel est ton prenom ? Julie

Bonjour Julie Houlert

**Exercice 2.3 : affectation statique et opération sur variable.**

Recopiez ce code dans la fenêtre de script **untitled**, enregistrez-le puis exécutez-le.

#mon deuxieme script python

annee = 2022

nom = input('Quel est ton nom ?')

age = int(input('Quel age as-tu ?'))

date = annee - age

print('Salut ', nom, 'tu es ne(e) en ', date)

print('Tu as ',age,'ans')

Pourquoi utilise-t-on la fonction int() dans la ligne age = int(input('Quel age as-tu ?')) ? On l’utilise car la fonction input a pour toujours une valeur de type str, d’autant plus que l’âge est un entier.

Relevez les différentes variables du programme et donnez leur type :  
annee (int) / nom (str) / age (int) / date (int)

**Exercice 2.4 : écriture d’expressions**

* Affectez les variables temps et distance avec les valeurs 6.892 et 19.7 respectivement.
* Calculez et affichez la valeur de la vitesse.   
  2.8583865351131745

**Exercice 2.5 : écriture d’expressions**

* Affectez une valeur quelconque à une variable numérique dont vous choisirez le nom ;
* y ajouter 2;
* multiplier le résultat par deux;
* afficher la valeur finale;

**Exercice 2.6 : corriger les bugs de ces instructions**

* exécutez l’instruction suivante dans la fenêtre interactive **IDLE Shell** :

13 = v

print("v vaut", v)

* notez ici l’erreur indiquée par l’interpréteur Python : cannot assign to litteral
* expliquez l’erreur : on affecte 13 à v et non l’inverse. (13 ne peut pas être une variable)
* corrigez l’instruction dans la fenêtre interactive de manière à ce qu’il n’y ait plus d’erreur
* exécutez l’instruction suivante dans la fenêtre interactive **IDLE Shell** :

v = input("entrez un entier svp")

t = v + 12

print("t vaut", t)

* notez ici l’erreur indiquée par l’interpréteur Python : can only concatenate str (not "int") to str
* expliquez l’erreur : on ne peut pas additionner une variable de type str à une variable de type int.
* corrigez l’instruction dans la fenêtre interactive de manière à ce qu’il n’y ait plus d’erreur

**PARTIE III – Le choix simple**

**Exercice 3.1 :**

Recopiez ce script dans la fenêtre de script **untitled**, enregistrez-le et exécutez-le.

#mon troisieme script python

v = int(input("entrez un entier svp"))

if v < 0 :

print("v est négatif")

else :

print("v est positif")

1. Indiquez combien de cas différents, selon la valeur de v, détecte ce programme. 2
2. ~~Ajoutez un troisième cas qui détecte si v prend la valeur de zéro et affiche "v est nul".~~

**Exercice 3.2 :**

Vous allez créer votre quatrieme script qui demandera à l’utilisateur une valeur entière quelconque, affichera si cet entier est pair ou impair. Vous créerez ce script dans une fenêtre de script **untitled**.

1. Expliquez quel donnée(s) de départ vous devez avoir, et quel résultat d’arrivée on veut obtenir.
2. Cherchez la définition d’un nombre pair (vous pouvez vous aider d’internet).
3. Décrivez en langage naturel l’algorithme du programme.
4. Une fois l’algorithme décrit, implémentez cet algorithme en Python, puis écrivez-le dans un fichier dans l’environnement Python et Python et sauvegardez ce fichier. Testez-le avec plusieurs valeurs.

**Exercice 3.3 : Corrigez ces bugs.**

Le script suivant présente une erreur. Recopiez-le dans une fenêtre de script **untitled,** sauvegardez-le puis exécutez le script.

#mon cinquieme script python

choix = input("Voulez-vous continuer ? ")

if v = "oui" :

print("Ok, on continue alors.")

else :

print("Bien, on arrête...)

* notez ici l’erreur indiquée par l’interpréteur Python : invalid syntax
* expliquez l’erreur : pour vérifier une condition on utilise ‘==’ et non ‘=’ ! De plus v n’est pas spécifié, et les guillemets lors du print ne sont pas fermés.
* corrigez l’instruction dans le script de manière à ce qu’il n’y ait plus d’erreur

**Exercice 3.4 : Les tabulations.**

Recopiez-ce script dans une fenêtre de script **untitled,** sauvegardez-le puis exécutez le script. Testez le avec ‘oui’ puis avec ‘non’.

#mon sixieme script python

choix = input("Voulez-vous continuer ? ")

if choix == "oui" :

print("Ok, on continue alors.")

else :

print("Bien, on arrête...")

print("Et on est sorti du if...")

* qu’affiche-t-on si v vaut ‘oui’ : "Ok, on continue alors."
* qu’affiche-t-on si v vaut ‘non’ : "Bien, on arrête..." / "Et on est sorti du if..."

Enlevez la tabulation de la **dernière ligne** comme ci-dessous**,** sauvegardez le script puis exécutez-le. Testez le avec ‘oui’ puis avec ‘non’.

#mon sixieme script python

choix = input("Voulez-vous continuer ? ")

if choix == "oui" :

print("Ok, on continue alors.")

else :

print("Bien, on arrête...")

print("Et on est sorti du if...")

* qu’affiche-t-on si v vaut ‘oui’ : "Ok, on continue alors."
* qu’affiche-t-on si v vaut ‘non’ : "Bien, on arrête..."
* comment expliquez-vous la différence entre les deux versions ? La condition pour afficher « Et on est sorti du if.. » a été enlevée pour cet affichage.

**Exercice 3.5 :**

Ci-dessous vous avez un script python incomplet. Vous allez compléter les parties en jaunes pour que ce script calcule et affiche les solutions d’une équation du second degré. Pour cela, répondez aux 3 questions suivantes.

1. Décrivez l’équation du discriminant Δ en vous aidant d’internet si besoin.   
   Δ = b² -4ac
2. Ecrivez **en langage naturel** les 3 solutions selon les valeurs du discriminant Δ.  
   Si Δ est nul, alors S = -b / 2a. Sinon S1 = ( -b – sqrt(Δ) ) / 2a et S2 = ( -b + sqrt(Δ) ) / 2a
3. Complétez en Python les lignes en jaune dans le script ci-dessous, puis écrivez l’ensemble du programme dans la fenêtre de script **untitled** que vous sauvegarderez. Testez votre programme avec diverses valeurs de a, b et c.

#mon cinquieme script python

from math import \*

print("Calcul des solutions d'une equation du second degre")

a = int(input("Entrez la valeur de a "))

b = int(input("Entrez la valeur de b "))

c = int(input("Entrez la valeur de c "))

delta = ……………………………

#mon 7e script python

from math import \*

print("Calcul des solutions d'une equation du second degre")

a = int(input("Entrez la valeur de a "))

b = int(input("Entrez la valeur de b "))

c = int(input("Entrez la valeur de c "))

delta = b\*\*2 - 4\*a\*c

print("Delta est égal à" , delta

if delta == 0:

print("Il n'y a qu'une racine à notre polynôme qui est," , -b / (2\*a) )

elif delta > 0:

S1 = (-b - sqrt(delta)) / (2\*a)

S2 = (-b + sqrt(delta)) / (2\*a)

print("Il y a deux racines à notre polynôme qui sont," + "S1:", S1, "et S2:", S2 )

else:

print("Il n'y a aucune racine à notre polynôme.")

print("Merci de votre visite, à bientôt.")

1. Ecrivez l’algorithme en langage naturel correspondant au programme ci-dessus.  
     
   Affecter un entier à a  
   Affecter un entier à b  
   Affecter un entier à c  
   Affecter b² - 4ac  à delta  
     
   Début If  
   Si delta est nul :  
    Afficher -b / (2\*a)  
   Si delta est positif :  
    Afficher (–b + √delta) / (2\*a) et (–b - √delta) / (2\*a)  
   Sinon :  
    Afficher « Il n’y a aucune racine à ce polynôme »  
   Fin If  
   Afficher « Merci de votre visite, à bientôt. »

*Valeurs de a, b et c pour tester votre script :*

a ← 1, b ← 2, c ← 1 alors Δ ← 0 Test Validé

a ← 1, b ← 4, c ← 1 alors Δ ← 12 Test Validé

a ← 4, b ← 1, c ← 3 alors Δ ← -47 Test Validé