## Table des matières :

[EXERCICE 1](#_oqahslca0tq)

Méthode

[EXERCICE 2](#_j81mejfjyarr)

Méthode

Jeux d’essais

[EXERCICE 2B](#_kxjwbs3c3v4j)

Méthode (1)

Jeux d’essais (1)

Méthode (2)

Jeux d’essais (2)

## **EXERCICE 1 :**

### **Méthode :**



Fonctionnement normal du programme

Signalement d’une erreur si l’on retire ‘:’

 et lors de l'exécution.

Même chose pour “print” qui n’est pas reconnu si l’on écrit “prnt”



avec \_main\_, le programme tourne sans exécuter le contenu de \_\_main\_\_

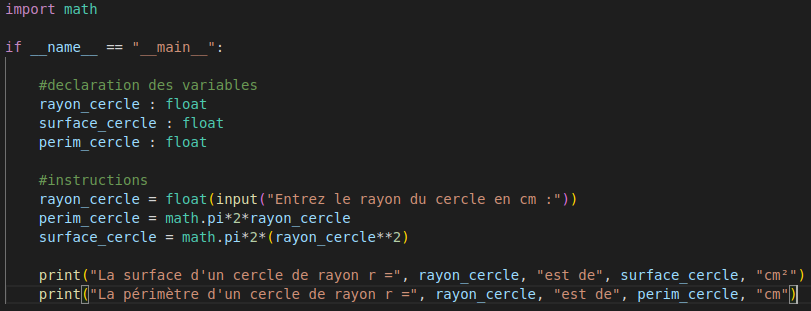


On obtient des erreurs en utilisant le mauvais type de variables 



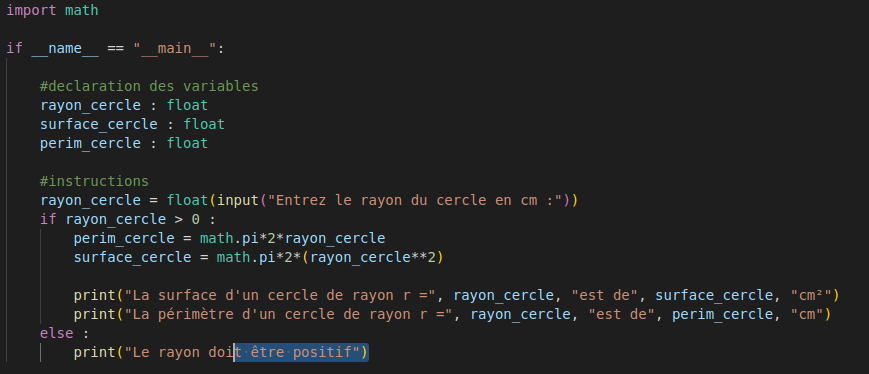
## **EXERCICE 2 :**

### **Méthode :**

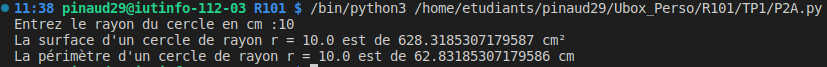


On utilise le type float (réel) sur “rayon\_cercle” pour permettre l’utilisation de nombre non-entiers dès la saisie. Le type est obligatoire par la suite (surface\_cercle et perim\_cercle) au vu des opérations utilisant PI.

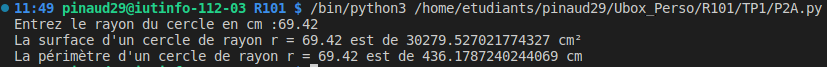
Mais le programme ci-dessus autorise un rayon négatif ou nul, ce qui n’est pas souhaitable dans ce cas. On le modifie donc tel que :



### **Jeux d’essais :**



Test concluant avec un entier positif



Test concluant avec un nombre non-entier

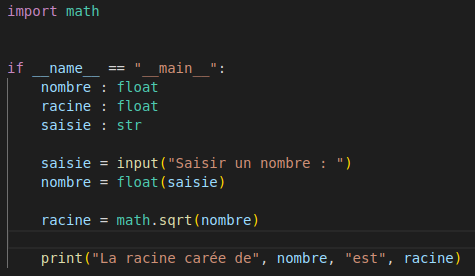


On gère la saisie d’un nombre négatif ou nul



## **EXERCICE 2B :**

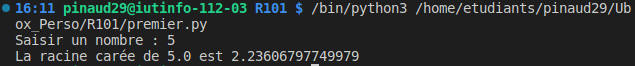
### **Méthode :**



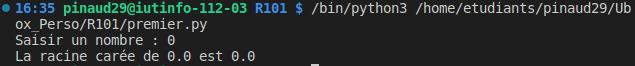
On utilise des flottants pour permettre l’opération racine et la saisie de nombre à virgule, et une string pour garder la saisie avant sa conversion sur la ligne suivante.

Cependant dans le cas actuel, si l’utilisateur indique une valeur négative, le programme renverra une erreur. On le modifie donc :

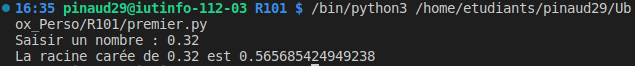
### **Jeux d’essais :**



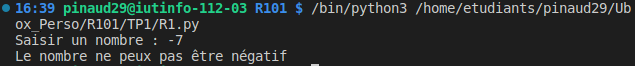
Fonctionne correctement avec un entier positif



Avec 0

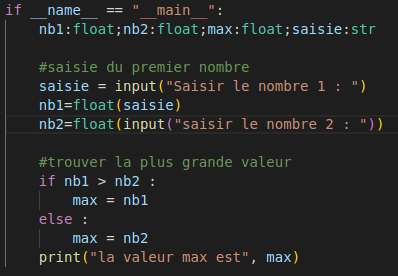


Avec un nombre à virgule



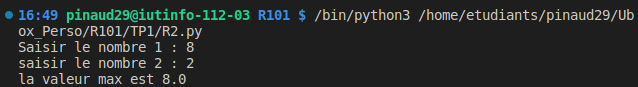
On anticipe l’erreur avec un nombre négatif

### **Méthode :**

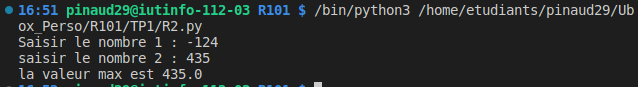


On utilise float pour permettre l’utilisation de nombre à virgule. Ici aucune opération de math ne restreint l’utilisation de certains nombres flottants.

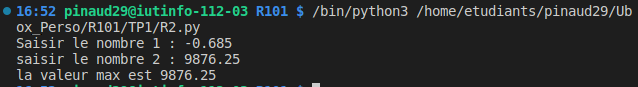
### **Jeux d’essais :**



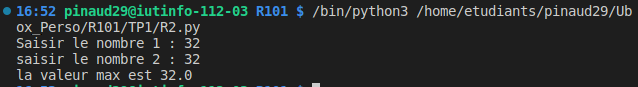
Le programme fonctionne avec deux entiers positifs



Avec des entiers négatifs



Avec des nombres positifs et négatifs à virgule



Dans le cas où nb1 = nb2