# Exercice2

Comportement de l’interpréteur python à la lecture de chacune des instructions suivant :

* >>> str (2021) \* int ("3")



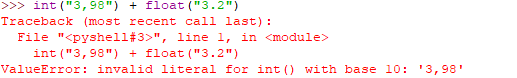
**Commentaire :** dans un premier temps il convertit 2021 en chaine de caractère et ‘3’ en entier puis répète la chaine ‘2021’ 3 fois.

* >>> int ("3") + float ("3.2")



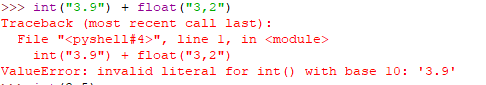
**Commentaire :** dans un premier temps il convertir la chaine ‘3’ en entier et ‘3.2’ en réel ensuite retourne la somme des deux nombres (le résultat est un réel).

* >>> int ("3,98") + float ("3.2")



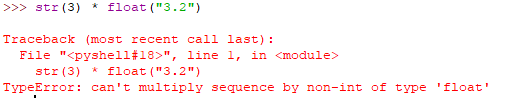
**Commentaire :** retourne une erreur car on essaye de convertir en entier nombre non reconnu par la classe « int »

* >>> int ("3.9") + float ("3,2")



**Commentaire :** deux problèmes se posent à ce niveau. Premièrement, on essaye de convertir en entier une chaine qui n’est pas reconnu par la classe « int ». Deuxièmement, on essaye de convertir en flottant une chaine dont la valeur n’est pas reconnue par la classe « float » (car la virgule en python est représentée par le **point(.)).**

* >>> str (3) \* float ("3.2")



**Commentaire :** erreur car on essaye de copier une chaine de caractère un nombre de fois qui n’est pas un entier

* >>> str (3/4) \* 2



**Commentaire :** répète la chaine ‘0.75’ 2 fois, ce qui se passe c’est que l’interpréteur effectue d’abord l’opération de division avant la convention.

* >>> "SEED" + [“innovation”, “hub”]



**Commentaire :** erreur car les délimiteurs sont incorrects, on délimite une chaine de caractère avec soit des **guillemets**, soit **des simples cotes** soit **des triples cotes**. Ensuite, on essaye de concaténer une chaine et une liste ce qui est impossible.

* >>> "SEED" + str (["innovation", "hub"])



**Commentaire**: l’interpréteur dans un premier temps convertir une liste en chaine ensuite il concatène avec une autre.