

✓ AIDE MEMOIRE, GESTION DES ERREURS:

NSI | 04

QUELQUES TYPES D'ERREURS

Type d'erreur :	exemples:
ValueError	<pre>>>> int("NSI") >>> from math import * >>> sqrt(-4)</pre>
ZeroDivisionError	<pre>>>> 10/0</pre>
TypeError	<pre>>>> "3" + 3</pre>

SYNTAXE DES BLOCS TRY ET EXCEPT

Pour:	on tape en Python:
exécuter des instructions en cas d'erreur	<pre>try: instructions except TypeErreur: instructions en cas d'erreur dans le bloc try</pre>
forcer le programme à recommencer tant que l'erreur n'est pas levée (raised)	<pre>while True: try: instructions break except TypeErreur: instructions en cas d'erreur dans le bloc try</pre>

✓ AIDE MEMOIRE, FONCTIONS:

SYNTAXE DES DECLARATIONS ET APPELS DE FONCTIONS

Déclaration et appel d'une fonction sans argument en entrée ni valeur retournée en sortie (on parle de procédure).	<pre>def ma_fonction(): instructions # appel: ma_fonction()</pre>
Déclaration et appel d'une fonction sans argument en entrée avec valeur retournée en sortie. <i>ma_fonction() est maintenant une variable de type int, float, str, bool...</i>	<pre>def ma_fonction(): instructions return variable # appel: ma_fonction()</pre>
Déclaration d'une autre fonction avec arguments (arg) en entrée puis appel avec paramètres (param) correspondants. <i>Les paramètres et arguments peuvent être de type int, float, str, bool..</i>	<pre>def ma_fonction(arg1, arg2, arg3, ..): instructions return variable # appel: ma_fonction(param1, param2, param3, ..)</pre>
Appeler une fonction définie dans un autre programme (module): <i>fichier.py</i>	<pre>from fichier import ma_fonction # appel: ma_fonction(param1, param2, param3, ..)</pre>

QUELQUES REGLES A CONNAITRE

- + Le programme ignore la déclaration d'une fonction (définie avec le mot clé **def**) tant qu'elle n'est pas appelée.
- + En PYTHON, les fonctions se déclarent avant leur appel.
- + L'instruction **return** permet à la fonction d'être utilisée comme variable de type, *int, float, str, bool...*
- + Dans le corps d'une fonction, on peut appeler une autre fonction.
- + Une fonction peut être passée en paramètre d'une autre fonction.
- + Une variable définie à l'intérieur du corps d'une fonction est **locale**, elle est invisible du corps du programme.