08

**Automne**

 **AIde memoire, GESTION DES ERREURS:**

**NSI**

04

|  |
| --- |
| **Quelques types d´erreurs** |
| Type d´erreur : | exemples: |
|  ValueError | >>> **int("NSI")**>>> from math **import** \*>>> **sqrt(-4)** |
|  ZeroDivisionError | >>> **10/0** |
|   TypeError | >>> **"3" + 3** |
| **Syntaxe des blocs try et except** |
| Pour: | on tape en Python: |
|   exécuter des instructions en cas d´erreur | 1. **try**:
2. instructions
3. **except** TypeErreur:
4. instructions en cas d´erreur  dans le bloc **try**
 |
|   forcer le programme à recommencer **tant que** l´erreur n´est pas levée (raised) | 1. **while** True:
2. **try**:
3. instructions
4. **break**
5. **except** TypeErreur:
6. instructions en cas d´erreur  dans le bloc **try**
 |

 **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). | 1. **def ma\_fonction**():
2. instructions
3. # appel:**ma\_fonction**()
 |
| Déclaration et appel d´une fonction sans argument en entrée avec valeur retournée en sortie. *ma\_fonction() est maintenant une variable de type int, float, str, bool...* | 1. **def ma\_fonction**():
2. instructions **return** variable
3. # appel:**ma\_fonction**()
 |
| Déclaration d´une autre fonction avec arguments (arg) en entrée puis appel avec paramètres (param) correspondants. *Les paramètres et arguments peuvent être de type int, float, str, bool..* | 1. **def ma\_fonction**(arg1,arg2,arg3,..):
2. instructions **return** variable
3. # appel:**ma\_fonction**(param1,param2,param3,..)
 |
| Appeler une fonction définie dans un autre programme (module): ***fichier.py*** | 1. **from** fichier **import** ma\_fonction
2. # appel:**ma\_fonction**(param1,param2,param3,..)
 |
| **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. |