

Utilisation d'un débogueur

Les boutons à utiliser sont ici celles qui figurent en dessus de la console :



Les 5 derniers boutons de cette ligne nous concernent pour le débogueur, respectivement :

- Débogueur pas à pas principal (Next)
- Débogueur pas à pas entrant (Step)
- Débogueur pas à pas sortant
- Débogueur : exécuter jusqu'au prochain point d'arrêt
- Arrêter le débogueur

Ce qui revient plus généralement à considérer les instructions de débogueur :

Suivi de variables

1. Ouvrir l'explorateur de variables (outil **Workspace**)

2. Points d'arrêt

Se placer sur la ligne à laquelle on désire placer un point d'arrêt et en double-cliquant dans la marge gauche du programme.

```
18 def pg():
19     n=123456789
20     k=1
21     while n<=987654321:
22         if calcul(n):
23             print(n)
24             n=suivant(n)
25             k=k+1
26     print(k)
```

On peut alors exécuter normalement le programme jusqu'au prochain point d'arrêt.

Exercice 1

1. Taper le programme suivant dans l'éditeur et l'exécuter. Que se passe-t-il ?

```
i = 10
while i != 0:
    i = 1 - i
print ( i )
```

Il est possible qu'à ce stade il soit nécessaire de fermer complètement l'IDE et de le rouvrir (ou recharger un nouveau shell pour continuer à travailler).

2. Placer un point d'arrêt devant la ligne 3 et exécuter ce programme pas à pas ou jusqu'au prochain point d'arrêt et observer les valeurs successives prises par la variable *i*. Expliquer le comportement observé à la première question.

Exercice 2

Utiliser un débogueur pour identifier l'erreur commise dans le programme ci-après. Le comportement attendu est le suivant : on demande des nombres entiers au clavier tant que la suite des nombres fournis est strictement croissante. La variable *n* compte le nombre de valeurs entrées au clavier.

```
precedent = int ( input ("Veuillez entrer un nombre entier"))
nouveau = int ( input ("Veuillez entrer le nombre entier suivant"))
n = 2
while precedent < nouveau :
    nouveau = int ( input ("Veuillez entrer le nombre entier suivant"))
    precedent = nouveau
    n = n + 1
```

3. Entrer dans les fonctions (ou pas)

Lorsque le programme comporte des fonctions, considère chaque appel de fonction comme une instruction unique et passe donc directement à l'instruction suivante.

Pour entrer dans le corps d'une fonction au moment où elle est appelée, cliquer sur Pas à pas entrant (step) de la fonction.

Si le débogueur est entré dans une fonction et pour aller directement à la fin de l'exécution de cette fonction, cliquer sur pas à pas sortant jusqu'au retour de la fonction.

Exercice 3

On donne le programme suivant :

```
def f(x):  
    for i in range(100):  
        x = (13 * x + 1) % 256  
    return x  
def g():  
    s = 0.  
    for j in range(10):  
        a = f(j)  
        s = s + 1./ (a - 210)  
    return s  
print(g())
```

1. Que se passe-t-il lors de son exécution ?
2. Jusqu'à quel point la simple lecture du code permet-elle d'expliquer ce comportement ?
3. À quelle itération de la boucle de la fonction g se situe le problème ?

Exécution en débogage pas à pas d'une boucle

Dans un nouveau fichier module Python, saisissez le script suivant :

```
1. nb = int(input("Entrez un nb strictement positif:"))
2. cpt = 0
3. for i in range(nb+1):
4.     if i%2==0 and i%5==0:
5.         cpt = cpt + 1
6. print("j'ai compté", cpt)
```

Avec Pyzo, placez un point d'arrêt sur la ligne 1, en cliquant sur la colonne grise à droite de la colonne des n° de ligne (cela place un point rouge indiquant le point d'arrêt).



Lancez l'exécution (F5). Dans l'outil Shells, vous devriez voir s'afficher :

```
>>> (executing file "<tmp 1>")
(<module>)>>>
```

Vous êtes alors en mode débogage, le nouveau prompt `(<module>)>>>` est celui pour entrer au clavier les commandes du debugger (mais nous allons plutôt utiliser l'interface graphique pour piloter le debugger). L'exécution s'est mise en pause sur le point d'arrêt que vous avez placé.

Ouvrez l'outil *Workspace* s'il n'est pas déjà actif.

En utilisant la commande menu *Shell* → *Débogage pas à pas principal (next)* (ou dans l'outil *Shells*



l'icône), exécutez le script en pas à pas (chaque clic sur cete commande provoque l'exécution de la ligne suivante du script).

Les affichages et saisies se font toujours normalement dans la zone de l'outil *Shells*. À l'afchage de la question :

```
Entrez un nb strictement positif:
⇒ Saisissez la valeur : 10
```

Continuez en utilisant plusieurs fois la même commande de débogage pas à pas.

Remarquez le tiret vert afché en face de la prochaine ligne à exécuter.

Observez dans l'outil *Workspace* l'évolution des variables suivant les lignes qui sont exécutées dans la fenêtre du script.



Workspace		
Name	Type	Repr
cpt	int	1
i	int	6
nb	int	11

Une fois que la dernière ligne du programme a été exécutée et a afché :

```
j'ai compté 2
--Return--
```



Cliquez sur la commande *Shell* → *Arrêter le débogage* (ou icône) pour stopper (sinon vous allez continuer en débogant Pyzo...).

Plus généralement, on peut utiliser l'exécution en mode débogage pour constituer un tableau de suivi des programmes)