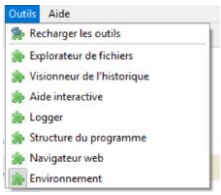
1. Code qui compte le nombre de caractères d'un string – Mode Débogage de Pyzo :

⇨ Enregistrer le code ci-contre dans un fichier nommé *ex_1b.py* :

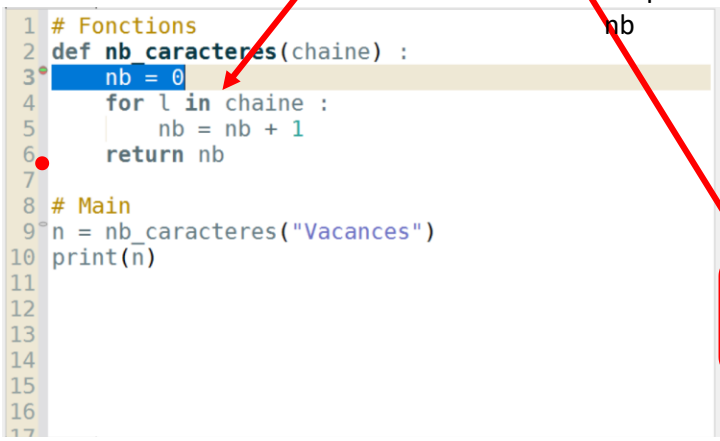
```
exercice_1.py
1  # Fonctions
2  def nb_caracteres(chaine) :
3      nb = 0
4      for l in chaine :
5          nb = nb + 1
6      return nb
7
8  # Main
9  n = nb_caracteres("Vacances")
10 print(n)
```

⇨ Que retourne la fonction *nb_caractere* après exécution ?

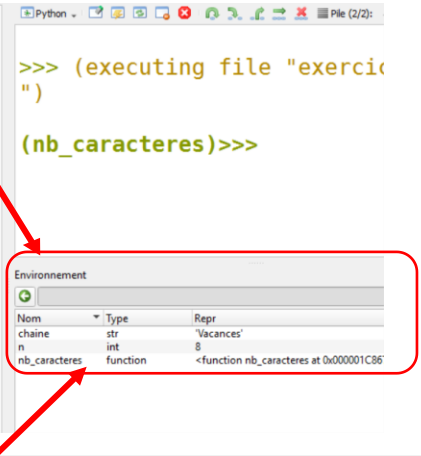
⇨ Afficher la fenêtre *Environnement* de pyzo :



⇨ Insérer un point d'arrêt en cliquant ici .



⇨ Exécuter alors le script (*F5 clavier*) et cliquer sur pour l'exécuter ligne par ligne et **compléter ci-contre le tableau** (*à reproduire*), **donnant les valeurs des variables** étape par étape :



nb

nb	chaine	l
0	Vacances	V
1	Vacances	aa
2	Vacances	c

AIDE : **La fenêtre *Environnement* affiche la valeur des différentes variables à chacune des étapes de l'exécution.**

2. Code qui inverse l'ordre des lettres dans une chaine de caractère :

⇨ Enregistrer le code ci-contre dans un fichier nommé *ex_2b.py* :

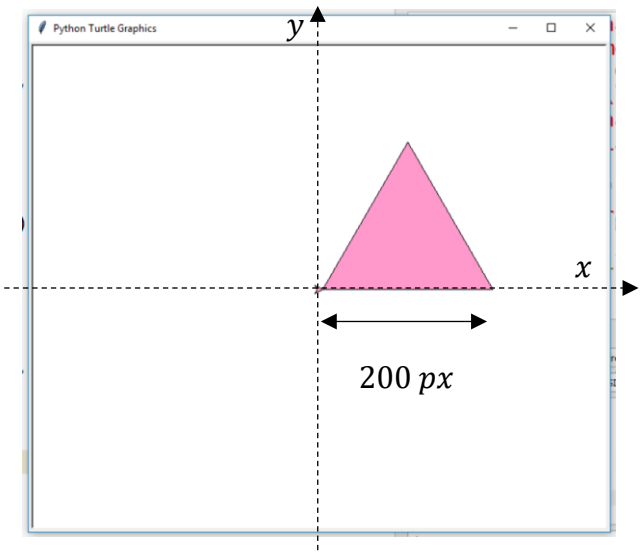
```
1  # Fonctions
2  def nb_caracteres(chaine) :
3      a_l_envers = ""
4      for l in chaine :
5          a_l_envers = l + a_l_envers
6      return a_l_envers
7
8  # Main
9  mot_1 = nb_caracteres("Nadal")
10 mot_2 = nb_caracteres(mot_1)
11 print(mot_1)
12 print(mot_2)
```

⇨ Réaliser une exécution normale (F5) ou étape par étape (avec point d'arrêt et ) et compléter le tableau ci-dessous, donnant la valeur des variables au cours de l'exécution : **s'aider ou pas de la fenêtre** *Environnement* **qui affiche le contenu des variables ….**

	mot_1	***mot_2***	***chaine***	***a_l_envers***	***l***
Etape 1			*Nadal*		*N*
Etape 2			*Nadal*	*N*	*a*
Etape 3			*Nadal*	*aN*	
Etape 4			*Nadal*		
Etape 5			*Nadal*		
Etape 6			*Nadal*		
Etape 7					
Etape 8					
Etape 9					
Etape 10					
Etape 11					
Etape 12					
Etape 13					

3 Code qui trace un triangle équilatéral en couleur :

L'exécution du code ci-contre donne le tracé ci-dessous **(le repère a été rajouté, il n'est pas tracé)** :




```
from turtle import *

# Fonctions
def triangle(a,x,y,r,g,b) :


    fillcolor(r,g,b)


# Main
triangle(200,0,0,1,0.6,0.8)

exitonclick()
```

⇨ Compléter ce code afin d'obtenir le même résultat.

(*INFO* : si **r** = 1 et g = b = 0, on a du **r**ouge

si r = b = 0 et **g** = 1, on a du **g**reen

si r = g = 0 et **b** = 1, on a du **b**lue)

4 Code qui trace plusieurs triangles roses

⇨ Faire un copier/coller de la fonction triangle dans un nouveau fichier *ex_4b.py* . Compléter alors ce code en utilisant cette fonction, afin d'obtenir le dessin ci-dessous à son exécution :