

Exercice 1 : Récupérer des saisies claviers

Ecrivez un script permettant d'obtenir le résultat suivant

```
Saisissez une chaîne
une chaîne
La chaîne inversée est:
eniahc enu
```

Astuces : la fonction `input()` permet de saisir une chaîne de caractères au clavier. La concaténation entre deux chaînes `s1` et `s2` se fait par l'instruction `s1 + s2`. Une chaîne vide se définit par `s=""`.

Exercice 2 : Manipulation des listes

Tapez chacune des instructions suivantes (en ligne de commande) et observez le résultat.

```
>>> list=['lundi', 2, 'janvier']
>>> print list
>>> list[0]
>>> print list[2]
>>> len(list)
>>> list.append(2010) # list est un objet - nous verrons cela plus tard
>>> list
>>> list[3]=list[3]+1
>>> del list[0]
>>> list
>>> list.insert(0,'mardi')
>>> list
>>> 'mardi' in list
>>> 'lundi' in list
>>> list.index(2)
>>> list2=list[1:3]
>>> list2
>>> list3=list[:2]
>>> list3
>>> list4=list[1:]
>>> list4
>>> list3=list3 + [2011]
>>> list3
>>> list5=3*list
>>> list5
>>> list.extend([3,4])
>>> list
>>> list6=list.pop(0)
>>> list6
>>> list
# Utilisation d'une liste pour échanger deux variables
>>> x = 5
>>> y = 12
>>> [x,y] = [y,x]
>>> x
12
>>> y
5
```

Exercice 3 :

Ecrivez un programme qui demande à l'utilisateur d'entrer des notes d'élèves. Si l'utilisateur entre une valeur négative, le programme s'arrête. En revanche, pour chaque note saisie, le programme construit progressivement une liste. Après chaque entrée d'une nouvelle note (et donc à chaque itération de la boucle), il affiche le nombre de notes entrées, la note la plus élevée, la note la plus basse, la moyenne de toutes les notes.

Astuces : Pour créer une liste vide, il suffit de taper la commande `list = []`.

Exercice 4 : Manipulation des dictionnaires

Tapez le code suivant :

```
>>> dico = {}
>>> dico['computer'] = 'ordinateur'
>>> dico['mouse'] = 'souris'
>>> dico['keyboard'] = 'clavier'
>>> print dico
>>> print dico.keys()
>>> print dico.values()
>>> del dico['mouse']
>>> print dico
>>> print len(dico)
>>> print dico.has_key('computer')
>>> print dico.items()
>>> for clef in dico:
...     print clef # Attention ne pas oublier la tabulation!!
>>> for clef in dico:
...     print clef, dico[clef] # Attention ne pas oublier la tabulation!!
>>> for clef, value in dico.items():
...     print clef, value # Attention ne pas oublier la tabulation!!
>>> dico2={'ordinateur': 'computer', 'souris' : 'mouse'}
>>> print dico2
>>> dico2=dico
>>> print dico
>>> del dico2['computer']
>>> print dico
>>> print dico2
>>> dico3=dico.copy()
>>> del dico3['keyboard']
>>> print dico
>>> print dico3
```

Exercice 5 : Définir une fonction sans paramètre

La liste des fonctions Python existantes est disponible en ligne à l'adresse :
<http://docs.python.org/library/functions.html>

Tapez le code suivant et observez le résultat.

```
>>> def fonction(): # Définition de fonction sans paramètre - Ne pas oublier les :
...     n=10 # Faire une tabulation en début de ligne
...     while n>0: # Faire une tabulation en début de ligne et ne pas oublier les :
...         print n/2, n%2 # Faire 2 tabulations en début de ligne
...         n=n-1 # Faire 2 tabulations en début de ligne
# Tapez encore une fois <Enter> si vous êtes en ligne de commande
>>> fonction() # Appel de la fonction
```

Exercice 6 : Fonction à 2 paramètres et valeurs par défaut

Tapez le code suivant et observez le résultat.

```
>>> def fonction(entree,diviseur=2): # fonction avec valeur par défaut pour diviseur
...     while(entree>0):
...         print entree/diviseur, entree%diviseur
...         entree=entree-1
# Tapez encore une fois <Enter> si vous êtes en ligne de commande
>>> fonction(10,2)
>>> fonction(10)
```

Exercice 7 :

Tapez le code suivant et observez le résultat.

```
>>> def fonction(entree=10,diviseur=2):
...     while(entree>0):
...         print entree/diviseur, entree%diviseur
...         entree=entree-1
# Tapez encore une fois <Enter> si vous êtes en ligne de commande
>>> fonction()
>>> fonction(4,2)
>>> fonction(diviseur=3,entree=9)
```

Exercice 8 : Affecter une instance de fonction à une variable

Tapez le code suivant et observez le résultat.

```
>>>def multiplication(n,p):
...     return n*p # pour retourner une valeur
# Tapez encore une fois <Enter> si vous êtes en ligne de commande
>>> a=multiplication(3,6)
>>> a
```

Exercice 9 : Fonction lambda

Python permet la création de fonctions anonymes (i.e. sans nom et donc non définie par `def`) à l'aide du mot-clé `lambda`. Une fonction anonyme ne peut pas avoir d'instruction `return` et doit forcément retourner une expression. De telles fonctions permettent de représenter tout sous forme de fonctions sans réellement en définir explicitement.

```
>>> def f (x): return x**2
>>> print f(8)
>>> g = lambda x: x**2
>>> print g(8)
>>> (lambda x: x*2)(3) //2
>>> def make_incrementor (n): return lambda x: x + n
>>> f = make_incrementor(2)
>>> g = make_incrementor(6)
>>> print f(42), g(42)
>>> print make_incrementor(22)(33)
>>> sentence = 'It is raining cats and dogs'
>>> words = sentence.split()
>>> print words
>>> lengths = map(lambda word: len(word), words)
>>> print lengths
>>> print map(lambda w: len(w), 'It is raining cats and dogs'.split())
```