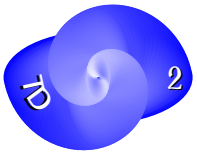
python powered

```
print("Hello, world!")
```

Université d'Orléans. Master LASTIC. Année 2013-2014

Programmation Python M1 TAL



Fonctions, listes, dictionnaires, etc

Les bases

Exercice 1 — *Palindromes*

1. Ecrire une fonction `affichePalindromes(txt,n)` qui recherche et affiche les palindromes de taille n dans txt. Ces palindromes devront être exclusiment composés de caractères alphabétiques et d'au plus un espace.
2. Utilisez la pour chercher la taille maximale des palindromes présents dans le texte des *3 Mousquetaires*

Exercice 2 — *listes simples*

On va construire et manipuler une liste de chaines comme par exemple la liste des prénoms de vos amis ou collègues.

- Créez une liste vide (2 méthodes)
- Ajoutez des éléments 1 par 1
- Ajoutez une autre liste à la première
- Triez la liste
- Affichez la taille de la liste
- Affichez le nombre d'occurences d'un prénom
- Supprimez un élément
- Insérez un élément à un index donné
- Essayez les opérateurs + et * sur les listes
- Visualiser des exemples sur les listes et votre code sur http://pythontutor.com

Exercice 3 — *listes par compréhension*

Quelles sont les listes suivantes :
```
>>> x = [a for a in [1, 2, 3]]
>>> x = [a for a in x if a == 2]
>>> x = [a for a in range(0, 21) if a % 2 == 0]
>>> x = [a * 2 for a in range(0, 11)]
```

- Ecrire une fonction de profil *impair(n)*
- En déduire à l'aide d'une compréhension la liste des nombres impairs inférieurs à n.
- Testez vos fonctions avec doctest

Exercice 4 — *Tri de 3 éléments*

Définir maintenant une fonction prenant 3 entiers en entrée et renvoyant le Tuple trié correspondant.

Exercice 5

Dictionnaires

1. Écrivez une fonction *decompte(L)* prenant en entrée une liste de mots L et renvoyant le dictionnaire associant à chaque mot son nombre d'occurences dans L. Par exemple :
   `decompte([ "blanc","bleu","blanc","noir","bleu","bleu","rouge", "rouge" ])` doit renvoyer le dictionnaire :
   `{'blanc': 2, 'bleu': 3, 'noir': 1, 'rouge': 2}`.
2. Écrivez une fonction *abrege(L,N)* prenant en entrée une liste de mots L et un entier N et renvoyant le dictionnaire associant à chaque mot de L sa version abbrégée à N lettres. Par exemple :
   abrege(["maison","immeuble","parking","hopital", "rue"] , 3) doit renvoyer :
   `{'hopital': 'hop.', 'immeuble': 'imm.', 'maison': 'mai.', 'parking': 'par.', 'rue': 'rue'}`