

# Exercices

Python

# Application : salutation (input)

- En utilisant la fonction input, écrire un programme qui propose de saisir le nom et renvoie « Bonjour *nom* ! »
- Ajouter au programme précédant la prise en compte du genre et renvoyer « Bonjour *genre nom* »
- Limiter le choix du genre en proposant de choisir un code (1) pour *madame* et (2) pour *monsieur*.
- Prendre en compte l'heure, pour afficher *bonjour* ou *bonsoir*.

# Application : nombres premiers

- Ecrire un programme qui renvoi la liste des nombres premiers inférieurs à  $n$ .
- On pourra utiliser:
  - 2 boucles for
  - Range
  - %
  - Break
  - Append pour remplir la liste
- Calculer le temps d'exécution pour  $n=10000$

# Application : recherche pdfs

- Soit une liste de fichiers d'extension différentes ( *.txt*, *.pdf*, *.py...* )
- Ecrire un programme qui retourne la liste des pdfs, en supprimant les extensions et en mettant en majuscule
- indications:
  - En créant une nouvelle liste à partir de la première (compréhension liste)

# Application – Compter mots

- Créer une fonction qui compte le nombre d'occurrences de chaque lettre d'une chaîne. `f('aaaeed')` -> `{'a': 3, 'e': 2, 'd': 1}`
- Refaire l'exercice avec une fonction qui retourne une liste `[('a', 3), ('e', 2), ('d', 1)]`
- Ordonner la liste selon le nombre d'apparition des éléments, en retournant une liste
- Indications:
  - `chaîne.count('c')` : retourne le nbre d'occurrence de 'c' dans chaîne
  - `set(L)` : retourne un ensemble à partir d'une liste, un set est un groupe d'éléments sans redondance
  - `L.sort()`

# Palindrome

- Ecrire une fonction qui vérifie si un mot est palindrome
- Exemple:
  - radar, ressasser

# Application calcul d'âge

- Définir une fonction qui prend en argument une date et retourne l'âge.
- Indices:
  - `from datetime import date`
  - `aujourd'hui = date.today()`
  - `naissance = date(1980, 2, 28)`
  - `anniversaire = naissance.replace(year=aujourd'hui.year)`
  - `anniversaire < aujourd'hui`

# Application : fonctions cap et iscap

Définir la fonction cap et iscap

- *cap(' ceci est UNE chaine')* -> ' Ceci est une chaine'
- *iscap('Chaine')*->*booléen*

Prendre en compte les chaines de caractères contenant un point

- *cap(' chaine. test, Ok')* -> ' Chaine. Test, ok'



# Application itérateur

- Créer un generator qui retourne une liste inversée
  - `yrange(10)` -> 10 ... 0
  - `yrange(10, 5)` -> 10 ... 5
  - `yrange(10, 5, 2)` -> 10 8 6

# Application – deux somme d'un

- retourner les indices des 2 nombres d'une liste dont la somme est égale à une cible
- exemple :
- $f((1, 2, 4), 5) \rightarrow 0, 2$

# Application – somme 0

- Retourner les éléments d'une liste dont les sommes sont nulles
- Exemple:
  - $f([1, 2, -3, 5, 7, -12, -5]) \rightarrow ([1, 2, -3], [5, 7, -12], [5, -5])$

Exercices classes

# Application : calcul d'âge

- Dans la classe, remplacer l'attribut `age` par date de naissance (year, month, day) de type `date`
- Définir la méthode `calcul_age` comme méthode de la classe `Personne`
- Définir l'attribut *age* en utilisant la fonction *property*

# Application héritage

- Définir la classe Paragraphe qui hérite de la classe str
- Définir les méthodes
  - `cap(self)` : qui prend en argument `self` et retourne la même chaîne avec une majuscule au début.
  - `iscap(self)` : vérifie si une chaîne de caractères est sous format `cap`.
  - `__add__(self, autre)` : concatène deux paragraphes en mettant le deuxième sous format `cap`.

# Application Exception

- Ré-implémenter la méthode calcul d'âge pour prendre en compte le 29 février
- Dans la class Paragraphe, interdire l'opération de multiplication par entier supérieur à 2
  - Avec assert
  - Avec TypeError
  - Avec MultiplicationError (à définir)