

Sujet 1: niveau moyen

Consignes

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format NOM1 NOM2 Sujet1.py
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 2 points pour l'activité de recherche en classe (différencié par binôme), 1 point pour la qualité du code, 1 point pour le respect des consignes, 1 point pour les tests, 2 points pour la qualité du code, 1 point pour la correction du code, 2 points pour la prestation orale (différencié par binôme).

Exercice 1 Fonction et conditionnelles

1. Écrire une fonction max2(a, b) qui renvoie le maximum des arguments entiers a et b sans utiliser la fonction built-in max.

Mettre entre triples guillemets, les tests effectués pour vérifier la correction de cette fonction.

```
>>> \max 2(10, 8)
10
```

2. Écrire une fonction max3(a, b) qui renvoie le maximum des arguments entiers a et b sans utiliser la fonction built-in max.

Mettre entre triples guillemets, les tests effectués pour vérifier la correction de cette fonction.

```
>>> max2(10, 8, 10)
10
```

Exercice 2 Boucle interactive

Écrire un programme qui demande d'abord à l'utilisateur un nombre entier de notes tant que le nombre saisi n'est pas strictement positif.

Ensuite le programme demande à l'utilisateur de saisir chaque note et affiche en sortie la note maximale et le nombre de fois où elle est atteinte.

> Site Web Page 1/12



Interdiction d'utiliser les listes/tableaux Python. Voici une trace d'exécution du programme :

```
Nombre de notes ? -2

Saisir un nombre > 0, nombre de notes ? -1

Saisir un nombre > 0, nombre de notes ? 3

Nouvelle note ? 10

Nouvelle note ? 8

Nouvelle note ? 10

Note maximale : 10.0 atteinte 1 fois
```

Page 2/12 Site Web



Sujet 2: niveau moyen

Consignes

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format NOM1_NOM2_Sujet2.py
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 2 points pour l'activité de recherche en classe (différencié par binôme), 1 point pour la qualité du code, 1 point pour le respect des consignes, 1 point pour les tests, 2 points pour la qualité du code, 1 point pour la correction du code, 2 points pour la prestation orale (différencié par binôme).

Exercice 1 Fonction et conditionnelles

- 1. Écrire une fonction bac (moyenne) qui prend en argument une moyenne de type float entre 0 et 20, vérifie si la moyenne est entre 0 et 20 avec une instruction assert et renvoie :
 - True si moyenne >= 10
 - False si moyenne < 10
- **2.** Écrire une fonction mention (moyenne) qui prend en argument une moyenne de type float entre 0 et 20 et qui renvoie une chaîne de caractère :
 - "recalé" si 0 ≤ note < 8
 - "second groupe" $si 8 \le note < 10$
 - "reçu" si 10 ≤ note < 12
 - "assez bien" $si 12 \leqslant note < 14$
 - "bien" $si\ 14 \leqslant note < 16$
 - "très bien" $\sin 16 \leqslant \text{note} \leqslant 20$
 - "valeur incohérente" sinon

On donne quelques tests unitaires qui doivent être vérifiés. Compléter ces tests en essayant de couvrir tous les cas possibles. Mettre entre triples guillemets, les tests effectués.

Page 3/12 Site Web



```
>>> assert mention(9) == "second groupe"
>>> assert mention(-1) == "valeur incohérente"
```

Exercice 2 Boucle interactive

Écrire un programme qui demande d'abord à l'utilisateur un nombre entier de notes tant que le nombre saisi n'est pas strictement positif.

Ensuite le programme demande à l'utilisateur de saisir chaque note et affiche en sortie la moyenne des notes saisies, arrondie au centième. On utilisera la fonction round, pour afficher sa documentation, saisir help(round) dans une console interactive.

Attention! Interdiction d'utiliser les listes/tableaux Python.

Voici une trace d'exécution du programme :

```
Nombre de notes ? -2
Saisir un nombre > 0, nombre de notes ? 3
Nouvelle note ? 10
Nouvelle note ? 8
Nouvelle note ? 5
Moyenne: 7.67
```

Site Web Page 4/12



Sujet 3: niveau moyen

Consignes

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format NOM1_NOM2_Sujet3.py
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 2 points pour l'activité de recherche en classe (différencié par binôme), 1 point pour la qualité du code, 1 point pour le respect des consignes, 1 point pour les tests, 2 points pour la qualité du code, 1 point pour la correction du code, 2 points pour la prestation orale (différencié par binôme).

Exercice 1 Fonction et conditionnelles

1. Écrire une fonction prenant en paramètres les bornes de deux intervalles fermés [a;b] et [c;d] de $\mathbb R$ et qui renvoie True si leur intersection est non vide et False sinon. On donne ci-dessous la spécification de cette fonction.

```
def intersection_intervalle(a,b,c,d):
    """"
    Signature : intersection_intervalle(a:float,b:float,c:float,d:
        float)->bool
    Précondition : vérifier si les intervalles [a,b] et [c,d] sont
        valides avec un assert
    Postcondition : Détermine si les intervalles [a,b] et [c,d] ont
        une intersection non vide
    """"
```

Si tout se passe bien, on aura ça:

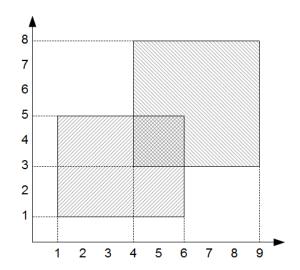
```
>>> intersection_intervalle(843, 844, 841, 842) False
```

Mettre entre triples guillemets, les tests effectués en essayant de couvrir tous les cas possibles.

2. Écrire une fonction prenant en paramètres les abscisses minimale et maximale et les ordonnées minimale et maximale de deux rectangles du plan et qui renvoie True si leur intersection est non vide et False sinon.

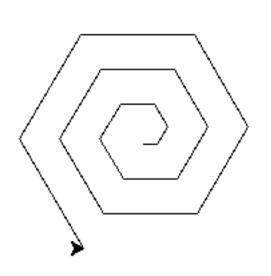
Page 5/12 Site Web

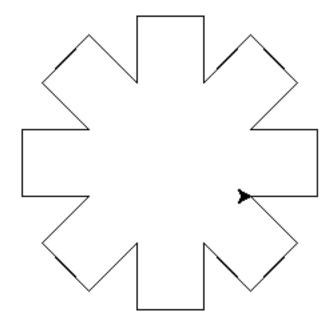




Exercice 2 Boucles imbriquées et tortue

Écrire avec le module turtle deux programmes permettant de réaliser les figures ci-dessous en au plus huit lignes (from turtle import *) compris.





Page 6/12 Site Web



Sujet 4: niveau moyen

Consignes

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format NOM1_NOM2_Sujet4.py
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 2 points pour l'activité de recherche en classe (différencié par binôme), 1 point pour la qualité du code, 1 point pour le respect des consignes, 1 point pour les tests, 2 points pour la qualité du code, 1 point pour la correction du code, 2 points pour la prestation orale (différencié par binôme).

Méthode Concaténation de chaînes de caractères

On peut concaténer deux caractères avec l'opérateur + :

```
>>> '(' + ':'
'(:'
>>> '(' + '(' + '(' + '('
,((((,
```

Si on concatène plusieurs fois le même caractère on peut utiliser l'opérateur de duplication *. On peut bien sûr combiner concaténation et duplication :

```
>>> '(' * 4
,((((,
>>> '(' * 4 + ')' * 4
,(((()))),
```

Exercice 1 Fonction et boucles

Fournir des tests d'exécutions entre triple guillemets.

1. Écrire une fonction carre (cote, caractere) qui affiche un carré de côté passé en argument et rempli du caractère passé dans le second argument. La fonction doit vérifier que cote est un entier positif et provoquer une erreur sinon.

> Page 7/12 Site Web



```
>>> carre(5, '0')
00000
00000
00000
00000
00000
00000
```

2. Écrire une fonction triangle_rectangle_gauche(cote, caractere) qui affiche un triangle rectangle avec l'angle droit à gauche, rempli du caractère passé en argument et avec cote caractères sur chaque côté adjacent à l'angle droit.

```
>>> triangle_rectangle_gauche(4, '@')

@

@@

@@

@@

@@@

@@@
```

3. Écrire une fonction triangle_rectangle_droite(cote, caractere) qui affiche un triangle rectangle avec l'angle droit à droite, rempli du caractère passé en argument et avec cote caractères sur chaque côté adjacent à l'angle droit.

4. Écrire une fonction triangle_isocele(base, caractere) qui affiche un triangle isocèle rempli du caractère passé en argument et avec base caractères sur sa base.

Page 8/12 Site Web



Exercice 2

Fournir des tests d'exécutions entre triple guillemets.

1. Écrire une fonction sapin_noel(base, tronc) qui affiche un sapin dont le feuillage est un triangle isocèle rempli du caractère '*' avec base caractères sur sa base et son tronc est constitué de tronc caractères '|'.

```
>>> sapin_noel(8,3)
  ******
 ******
******
******
```

2. Écrire une fonction losange (diagonale, caractere) qui affiche un losange rempli du caractère passé en argument. La fonction doit vérifier que diagonale est un entier positif impair et provoquer une erreur sinon.

```
>>> losange(11, '@')
   000
  @@@@@
 0000000
000000000
0000000000000
000000000
 0000000
  @@@@@
   @@@
    0
```

Page 9/12 Site Web



Sujet 5: niveau difficile

Consignes

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format NOM1_NOM2_Sujet5.py
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 2 points pour l'activité de recherche en classe (différencié par binôme), 1 point pour la qualité du code, 1 point pour le respect des consignes, 1 point pour les tests, 2 points pour la qualité du code, 1 point pour la correction du code, 2 points pour la prestation orale (différencié par binôme).

Exercice 1 Exercice 34 p. 46 du manuel Indice

Page 10/12 Site Web



Sujet 6: niveau moyen

Mini-projet n°1

Consignes

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format NOM1_NOM2_Sujet6.py
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 2 points pour l'activité de recherche en classe (différencié par binôme), 1 point pour la qualité du code, 1 point pour le respect des consignes, 1 point pour les tests, 2 points pour la qualité du code, 1 point pour la correction du code, 2 points pour la prestation orale (différencié par binôme).

📝 Exercice 1 Exercice 28 p. 44 du manuel Indice

Page 11/12 Site Web



Sujet 7: niveau difficile

Consignes

- Déposer dans l'espace indiqué par M.Junier un script Python contenant les réponses aux deux exercices en le nommant selon le format NOM1_NOM2_Sujet7.py
- Insérer un commentaire avec les noms des deux membres du binôme au début du script.
- Respecter les consignes données dans chaque exercice.
- Commenter les parties du code qui ne seraient pas immédiatement compréhensibles à la lecture.
- Fournir pour chaque question des tests permettant de vérifier la validité du code en les plaçant entre triple guillemets.
- Préparer une présentation orale de son travail devant le groupe (5 minutes).
- Barème sur 10 points : 2 points pour l'activité de recherche en classe (différencié par binôme), 1 point pour la qualité du code, 1 point pour le respect des consignes, 1 point pour les tests, 2 points pour la qualité du code, 1 point pour la correction du code, 2 points pour la prestation orale (différencié par binôme).

Exercice 1 Exercice 41 p. 47 du manuel Indice

Page 12/12 Site Web