UNIVERSITÉ JEAN MONNET SAINT-ÉTIENNE

Info. - TD : Feuille 3

CORRECTION CORRECTION CORRECTION

Exercice 1 – créations numpy

Dessiner et/ou expliquer le contenu (et la forme) du tableau créé par

Q1) shape (2,), (tableau 1 dimension avec 2 valeurs), qui contient 4 et 5.

Q2) shape (4,5), (tableau 2d avec 4 lignes 5 colonnes), contient 20 valeurs 0.

Q3) shape (5,4), (tableau 2d avec 5 lignes 4 colonnes), contient 20 valeurs 1.

Q4) shape (4,5), contient 20 valeurs (a priori différentes) tirées d'une loi uniforme de paramètres 2 et 3 (donc entre le réel 2 et le réel 3, de manière equiprobable).

Q5) shape (4,5), contient 20 valeurs (a priori différentes) tirées d'une loi normale de paramètres 2 et 3 (donc de moyenne 2 (réel) et d'écart type 3 (réel)).

Q6) shape (5,), (tableau 1d avec 5 valeurs), contenant exactement [3, 3.25, 3.5, 3.75, 4]

Q7) shape (90,), (tableau 1d avec 90 valeurs), contenant [100, 110, 120, 130, ..., 970, 980, 990]

Q8)

```
np.array([100, 125, 150, 175, 200])
```

```
np.arange(100, 201, 25)
```

```
np.linspace(100, 200, 5)
```

Exercice 2 – *shape* numpy

Q9) . **Q10)** (2,) **Q11)** (4,5) **Q12)** (5,4) **Q13)** (4,5) **Q14)** (4,5) **Q15)** (30,) **Q16)** (90,) . **Q17)** (100,)

Q18) le premier est 1d avec 100 valeurs (shape (100,)), le second est 2d avec 100 lignes et 1 colonne, le dernier est 2d avec 1 ligne et 100 colonnes.

. **Q19)** (5,20) **Q20)** (5,20) **Q21)** (5,20) **Q22)** (5,20) . **Q23)** (2,8) **Q24)** (4,4)

Exercice 3 – opérations

Q25) a = np.arange(100, 1000, 10)

Q26) b = a.reshape((9, 10))

Q27) c = np.ones((9, 10))

Q28) d = c*1000

Q29) e = a+b

Q30) np.ones((9, 10))*1000+ np.arange(100, 1000, 10).reshape((9, 10)) np.arange(110, 2000, 10).reshape((9, 10
. **Q31)**

10*np.arange(1, 11).reshape((2, 5))**2

Exercice 4 – indices numpy

On considère le tableau `e` de l'exercice précédent, mais vous devez donner des réponses les plus générales possibles. Par exemple, on évitera de supposer que `e` a 9 lignes et 10 colonnes.

Comment…

Q32) e[0, 0] **Q33)** e[0, 0] = 999 **Q34)** e[1, 0] **Q35)** e[1, 0] = 999 **Q36)** e[2, 0] **Q37)** e[6, 0]
Q38) e[3, -1] **Q39)** e[3, -1] = 999 **Q40)** e[3, -3] **Q41)** e[3, -3] = 999 **Q42)** e[-1, -2] **Q43)**
e[-1, -2] = 999

Exercice 5 – tranches numpy

Toujours à partir de `e` et en donnant des réponses générales, comment …

Q44) e[:, 0] **Q45)** e[:, 0] = 999 **Q46)** e[:, 1] **Q47)** e[:, 1] = 999 **Q48)** e[:, 1] = np.arange(1, 10)
Q49) e[:, 6] **Q50)** e[6, :]

Q51) e[1, 4:8] **Q52)** e[1, 4:8] = 999 **Q53)** e[1, 4:8] = np.arange(24, 28)

Q54) e[1, 4:8] = np.array([10, 100, 1000, 10000]) **Q55)** e[1, 4:8] = e[2, 4:8] **Q56)** e[1, 4:8] = e[1, 4:8]**2
Q57) e[1, :4] **Q58)** e[1, -4:] **Q59)** e[-1, -4:] **Q60)** e[-1, -4:] = e[-1, -4:]**2

Q61) e[::2, 3] **Q62)** e[::2, 3] = 999 **Q63)** e[::2, 3] = np.arange(np.shape(e)[0]//2) **Q64)** e[::-2, 3]
Attention : si on spécifie les bornes, dans l'autre sens, et du coup le 0 est exclu donc problème.
Q65) e[::2, 3:-2] **Q66)** e[::2, 3:-2] = 999

Q67)

```
e[::2, 3:-2] = np.arange(25).reshape((5,5))
# le premier 5 est e.shape[0]-3-2, le second est e.shape[1]//2
```

Q68) e[::2, 3:5] **Q69)** e[::2, 3:] **Q70)** e[::2, 3:-2] **Q71)** e[3::2, 3:-2]

Q72)

```
array([[ 1990.,  1970.,  1950.,  1930.,  1910.],
       [ 1790.,  1770.,  1750.,  1730.,  1710.],
       [ 1590.,  1570.,  1550.,  1530.,  1510.],
       [ 1390.,  1370.,  1350.,  1330.,  1310.],
       [ 1190.,  1170.,  1150.,  1130.,  1110.]])
```

Q73) ?} un peu compliqué pour l'affectation, dû à la parité d'une dimension mais pas de l'autre... schéma
Q74)

```
array([[ 1990.,  1960.,  1930.],
       [ 1690.,  1660.,  1630.]])
```

Q75)