

```
#!/usr/bin/env python
# coding: utf-8

# In[1]:

import numpy as np

# In[2]:

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

# In[6]:

#Q26
b = a.reshape( (9, -1) )
b = a.reshape( (9, 10) )
b = a.reshape( (-1, 10) )
print(b)

# In[10]:

#Q27
c = np.ones((9, 10))
c = np.ones(b.shape)
print(c)

# In[17]:

#Q28
#d = np.full(c.shape, 1000.)
d = c * 1000
d = c + 999
print(d)

# In[19]:

#Q29
e = b + d
print(e)

# In[33]:

#Q30
e = np.arange(1100, 2000, 10).reshape((9, 10))
e = np.arange(1100, 2000, 10).reshape((9, -1))
e = np.arange(1100, 2000, 10).reshape((-1, 10))
e = np.linspace(1100, 1990, 90).reshape((9, 10))
e = np.linspace(1100, 1990, 90).reshape((9, -1))
e = np.linspace(1100, 1990, 90).reshape((-1, 10))
e = (np.arange(90)*10. + 1100).reshape((9, 10))
e = (np.arange(90)*10. + 1100).reshape((9, -1))
e = (np.arange(90)*10. + 1100).reshape((-1, 10))
print(e)

# In[37]:

#Q31
(10 * np.arange(1, 11)**2).reshape((2, -1))

# # exercice 4

# In[38]:
```

```
e[0,0]

# In[40]:

e[0,0] = 999
print(e)

# In[41]:

e[1,0]

# In[42]:

e[1,0] = 999

# In[45]:

#q36
e[2,0]

# In[46]:

e[6,0]

# In[47]:

e[3,-1]

# In[48]:

e[3,-1] = 999

# In[49]:

#q40
e[3,-3]

# In[50]:

e[3,-3] = 999

# In[51]:

e[-1,-2]

# In[52]:

e[-1,-2] = 999

# In[53]:

print(e)

# # exercice 5

# In[54]:
```

```
e[:,0]
```

```
# In[55]:
```

```
e[:,0] = 888  
print(e)
```

```
# In[56]:
```

```
#q46  
e[:,1]
```

```
# In[57]:
```

```
e[:,1] = 777  
print(e)
```

```
# In[62]:
```

```
e[:,1] = np.arange(1, 10)  
print(e[:, :-2]) # afficher une partie du tableau pour éviter les retours à la ligne
```

```
# In[64]:
```

```
#q49  
e[:,6]
```

```
# In[65]:
```

```
e[6,:]
```

```
# In[66]:
```

```
e[1,4:8]
```

```
# In[68]:
```

```
#q52  
e[1,4:8] = 999
```

```
# In[69]:
```

```
e[1,4:8] = np.arange(24, 28)
```

```
# In[75]:
```

```
#q54  
e[1,4:8] = [10, 100, 1000, 1000]  
e[1,4:8] = np.array([10, 100, 1000, 1000])  
# si on voulait 10 100 1000 10000 (10k à la fin)  
e[1,4:8] = 10 ** np.arange(1, 5)  
print(e[1,4:8])
```

```
# In[76]:
```

```
e[1,4:8] = e[2,4:8]  
print(e[:, :-2]) # afficher une partie du tableau pour éviter les retours à la ligne
```

```
# In[77]:
```

```
#q56
e[1,4:8] = e[1,4:8]**2
```

```
# In[78]:
```

```
#q57
e[1,0:4]
e[1,:4]
```

```
# In[80]:
```

```
e[1, -4: ]
e[1,-4:]
```

```
# In[81]:
```

```
e[-1,-4:]
```

```
# In[82]:
```

```
#q60
e[-1,-4:] = e[-1,-4:]**2
```

```
# In[83]:
```

```
#q61
e[ : : 2 , 3 ]
e[:,2,3]
```

```
# In[84]:
```

```
e[:,2,3] = 999
```

```
# In[87]:
```

```
#q63
e[:,2,3] = np.arange(5)
#
nb_lignes = e.shape[0]
N = (nb_lignes + 1) // 2
e[:,2,3] = np.arange(N)
#
N = e[:,2, 3].shape[0]
e[:,2,3] = np.arange(N)
```

```
# In[96]:
```

```
e[ : :-2 , 3 ]
```

```
# In[95]:
```

```
# pas ok, on commence à l'indice 0 puis on s'arrête car on va dans le négatif
# ... e[0::-2,3]
```

```
# In[97]:
```

```
#q65
e[ : :2 , 3:-2 ]
e[:,2,3:-2]
```

```
# In[98]:
```

```
e[:,2,3:-2] = 999
```

```
# In[101]:
```

```
e[:,2,3:-2] = np.arange(e[:,2,3:-2].shape[0])  
#  
N = e[:,2,3:-2].shape[0]  
e[:,2,3:-2] = np.arange(N)
```

```
# In[102]:
```

```
#q68  
e[::2, 3:6]  
e[:,2,3:6]
```

```
# In[103]:
```

```
e[:,2,3:]
```

```
# In[104]:
```

```
e[:,2,3:-2]
```

```
# In[105]:
```

```
#q71  
e[3::2, 3:]  
e[3::2,3:]
```

```
# In[ ]:
```

```
# In[ ]:
```

```
# In[ ]:
```