

```

1 # .\scripts\part2\onction1.py
2
3 def carree(a) :
4     """ La fonction carree calcule la valeur du carre du parametre """
5     valeur = a**2
6     return valeur # renvoie l'image de a par la fonction carree
7
8
9
10 # .\scripts\part2\tp1.py
11 a = 6 # a Int entier
12 b = 4.5 # b float nombre à virgule
13 c = "hello world !" # c str chaîne de caractères
14 d = True # d bool booléen (vrai/faux)
15 e = (1,4,5,6) # e tuple p-uplet liste immutable / immuable (non
    modifiable)
16 f = [3,5,7] # f liste (mutable) valeurs modifiables
17 g = {"Paul":7,"Marc":17,"Marie":8} # g dictionnaire
18 h = {2,6,7,9} # h set ensemble
19 # déterminer le type de la variable => executer type(a)
20
21
22 # .\scripts\part2\fonctionCube.py
23 def cube(a):
24     return (a**3)
25
26
27 # .\scripts\part2\fonctionCubeCommentee.py
28 def cube(a):
29     """la fonction cube :
30         calcule la valeur du cube du paramètre"""
31     return (a**3)
32 # avoir de l'aide sur la fonction => help(cube)
33 # calculer a**6 à partir de la fonction cube() => cube(3)*cube(3) ou cube(3)**2
34
35
36 # .\scripts\part2\fonction_f.py
37 def f(a):
38     return(a*2)
39
40 # .\scripts\part2\fonction_fg.py
41 def f(a):
42     return(a*2)
43
44 def g(a):
45     return(a+2)
46
47
48 # .\scripts\part2\fonction_fg_cascade.py
49 def f(x):
50     return(x*2)
51
52 def g(x):
53     return(x+2)
54
55 def cascade(x,f,g):
56     return(x,f(x),g(x))
57 # appel de la fonction cascade(3,f,g) renvoie (3,6,5)
58 # appel de la fonction cascade(3,g,f) renvoie (3,5,6)
59
60
61 # .\scripts\part2\tp2.py
62 def division(a,b):
63     assert (b != 0), "Le denominateur doit etre non nul !"
64     return(a//b)
65
66
67 # .\scripts\part2\tp2_asset_entier.py
68 def division(a,b):
69     assert (b != 0), "Le denominateur doit etre non nul !"
70     assert (type(a)==int), "la variable a doit être un entier !"
71     return(a//b)

```

```

72
73
74 # .\scripts\part2\tp2_asset_sup0.py
75 def division(a,b):
76     assert (b != 0), """Le denominateur doit etre non nul !"""
77     assert (a>=0), """la variable a doit être supérieure ou égale à 0 !"""
78     return(a//b)
79
80
81 # .\scripts\part2\tp3_assert_str.py
82 def affichage(nom,prenom):
83     assert type(nom) ==str and type(prenom) ==str, """les variables nom et prénom
84     doivent être de type chaine de caractère (str)"""
85     print("Bonjour ",prenom,end=" ")
86     print(nom)
87
88 # .\scripts\part2\moyenne.py
89 def moyenne(a,b,c,d):
90     return((a+b+c+d)/4)
91
92
93 # .\scripts\part2\pente.py
94 def pente(x1,y1,x2,y2):
95     return((y2-y1)/(x2-x1))
96
97
98 # .\scripts\part2\ordonneeOrigine.py
99 def pente(x1,y1,x2,y2):
100     return((y2-y1)/(x2-x1))
101
102 def ordonneeOrigine(x1,y1,x2,y2):
103     return(y1-pente(x1,y1,x2,y2)*x1)
104
105
106 # .\scripts\part2\ordonneeOrigine_assert.py
107 def pente(x1,y1,x2,y2):
108     return((y2-y1)/(x2-x1))
109
110 def ordonneeOrigine(x1,y1,x2,y2):
111     assert x1!=x2, """les deux points ont la même abscisse !"""
112     return(y1-pente(x1,y1,x2,y2)*x1)
113
114
115
116 # .\scripts\part2\CirconferenceCercle.py
117 from math import *
118
119 def CirconferenceCercle(r):
120     return pi*2*r
121
122 #ou
123 import math
124
125 def CirconferenceCercle(r):
126     return math.pi*2*r
127
128 #ou
129 from math import pi
130
131 def CirconferenceCercle(r):
132     return pi*2*r
133
134
135 # .\scripts\part2\norme.py
136 from math import *
137 def norme(x1,y1,x2,y2):
138     xab=x2-x1
139     yab=y2-y1
140     ab=sqrt(xab**2+yab**2)
141     return ab
142

```

