

```
# td1_arbres_binaires.py
```

```
01 | from graphviz import Digraph
02 |
03 | graphe = Digraph(strict = True)
04 |
05 |
06 | class Arbre :
07 |     nb_arbre = 0;
08 |     def __init__(self, info = None, fg = None,
09 | = None):
10 |         self.info = info
11 |         self.fg = fg
12 |         self.fd = fd
13 |         self.__class__.nb_arbre += 1
14 |         self.num = str(self.__class__.nb_arbre)
14 |         graphe.node(self.num, self.info) #trace
```

```
le noeud
```

```
15 |
16 |     def trace_graphique(self):
17 |         """
18 |         Trace les ponts de l'arbre sous le self
19 |         """
20 |
21 |         l = [self]
22 |         while l != [] :
23 |             nd = l.pop(0)
24 |             if nd.fg != None:
25 |                 graphe.edge(nd.num, nd.fg.num)
26 |                 l.append(nd.fg)
27 |             if nd.fd != None :
28 |                 graphe.edge(nd.num, nd.fd.num)
29 |                 l.append(nd.fd)
30 |
31 |
32 |     def inserer_droite(self, valeur):
33 |         if self.fd == None :
34 |             self.fd = Arbre(str(valeur))
35 |         else :
36 |             nouvel_arbre = Arbre(str(valeur))
```

```

37 |         nouvel_arbre.fd = self.fd
38 |         self.fd = nouvel_arbre
39 |
40 |     def inserer_gauche(self, valeur):
41 |         if self.fg == None:
42 |             self.fg = Arbre(str(valeur))
43 |         else :
44 |             nouvel_arbre = Arbre(str(valeur))
45 |             nouvel_arbre.fg = self.fg
46 |             self.fg = nouvel_arbre
47 |
48 | def taille(A) :
49 |     if A == None : return 0
50 |     else:
51 |         return 1 + taille (A.fg) + taille(A.fd)
52 |
53 |
54 | A = Arbre("racine B")
55 | A.fg = Arbre("fgB_1")
56 | A.fd = Arbre("fdB_1")
57 | A.fd.fd = Arbre("FdB_2")
58 | """
59 | B = Arbre('0')
60 | B.inserer_gauche(1)
61 | B.inserer_droite(2)
62 | print(B.taille())
63 | B.trace_graphique()
64 | """
65 | print(taille(A))
66 | A.trace_graphique()
67 | graphe.view()

```