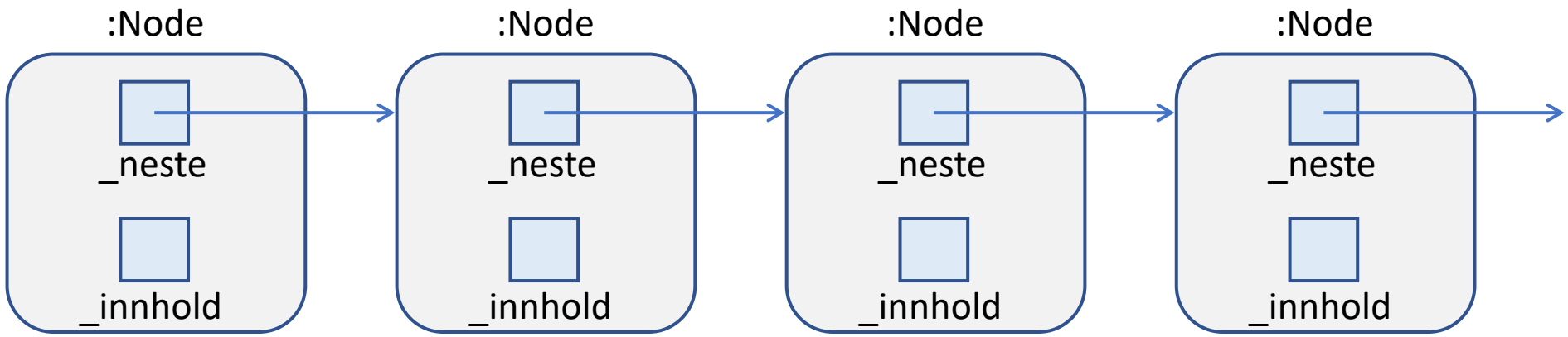
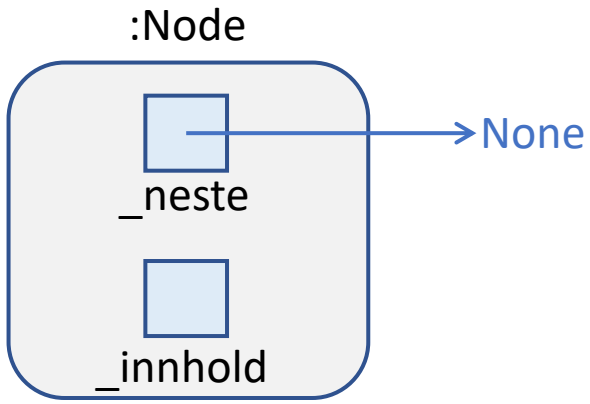
Datastrukturtegninger

Tilstandsdiagrammer

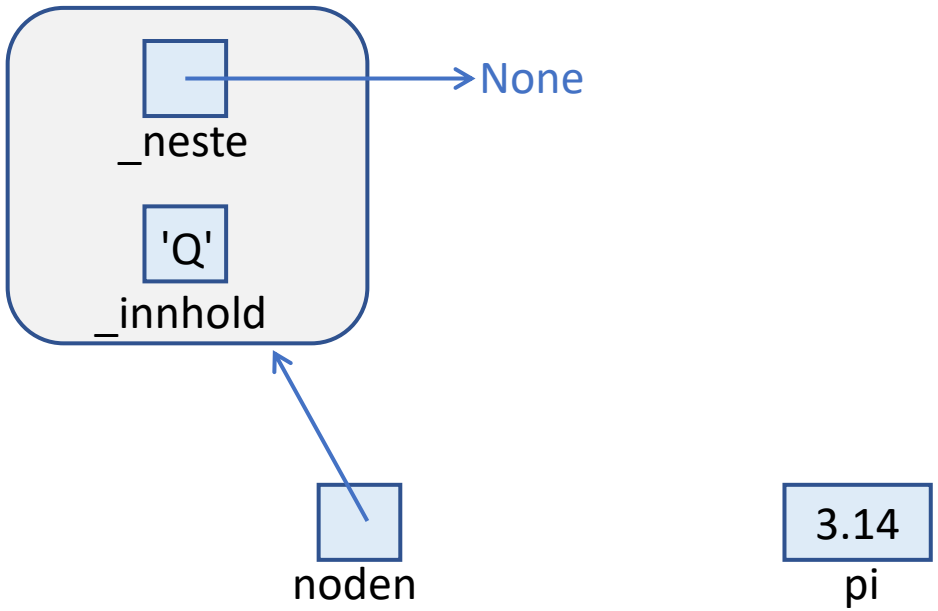
Illustrert på et eksempel med lenkelister


:Node
_neste
_innhold
:Node
_neste
_innhold
:Node
_neste
_innhold
:Node
_neste
_innhold



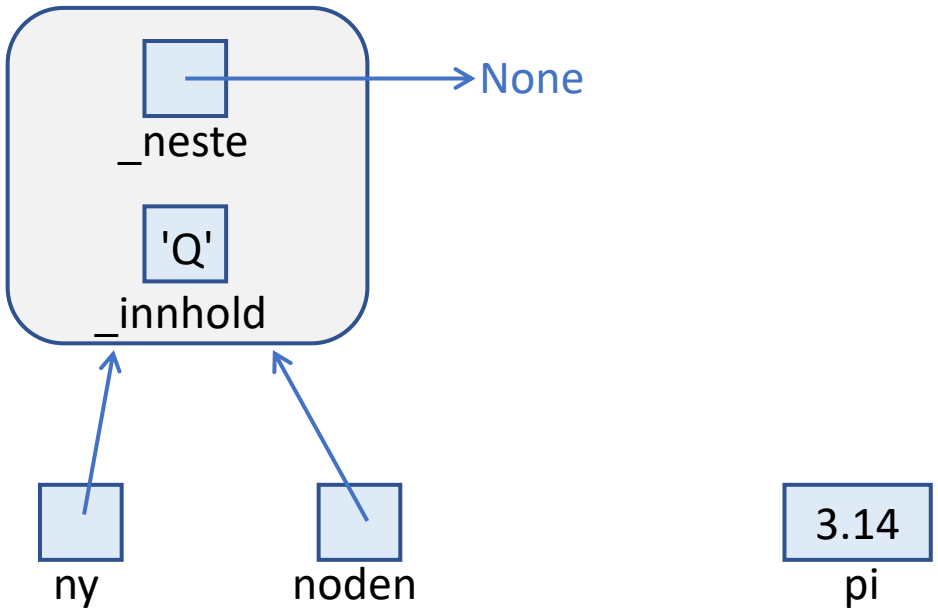

```
class Node :
    def __init__(self, nytt) :
        self._innhold = nytt
        self._neste = None
```

Tegning av muterbare objekter/verdier og immuterbare objekter/verdier



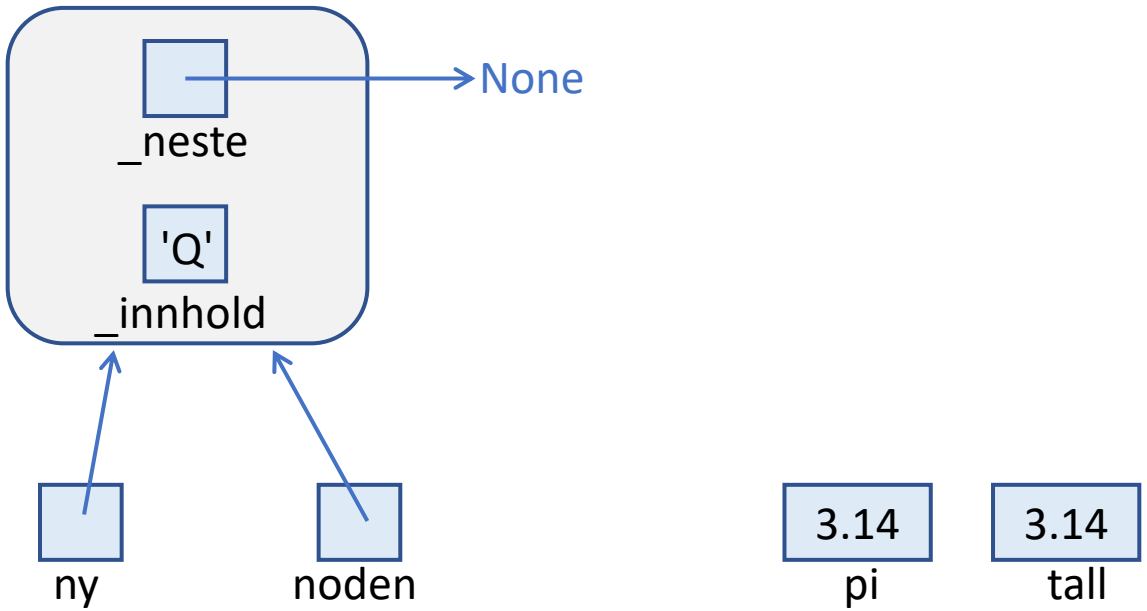

```
ny = noden
tall = pi
noden._neste = Node('A')
noden = noden._neste
t = ny._innhold
```

Tegning av muterbare objekter/verdier og immuterbare objekter/verdier



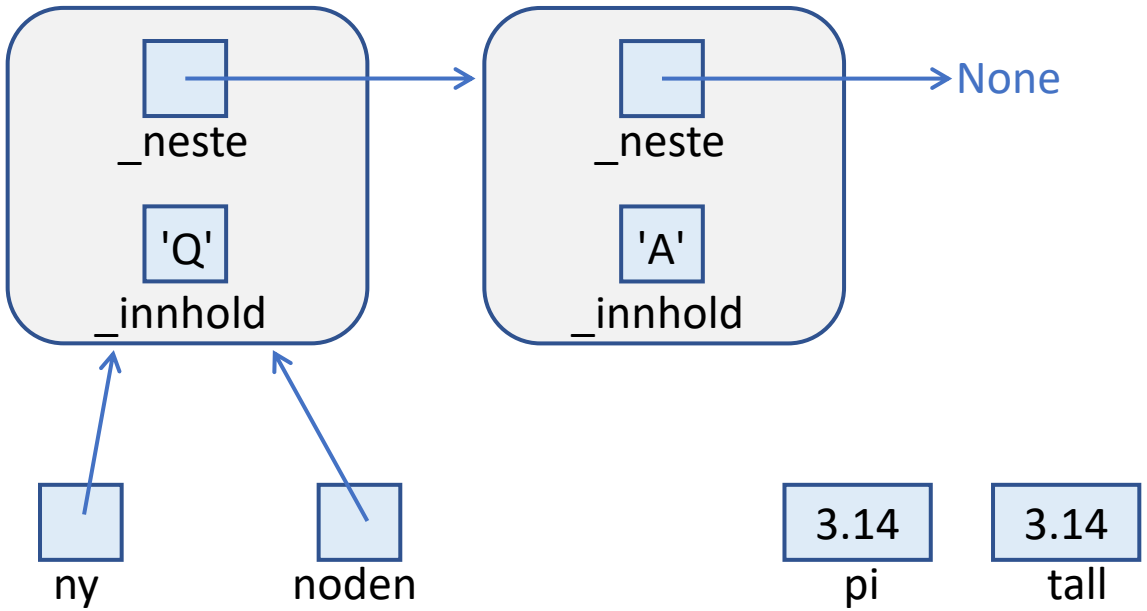

```
ny = noden
tall = pi
noden._neste = Node('A')
noden = noden._neste
t = ny._innhold
```

Tegning av muterbare objekter/verdier og immuterbare objekter/verdier



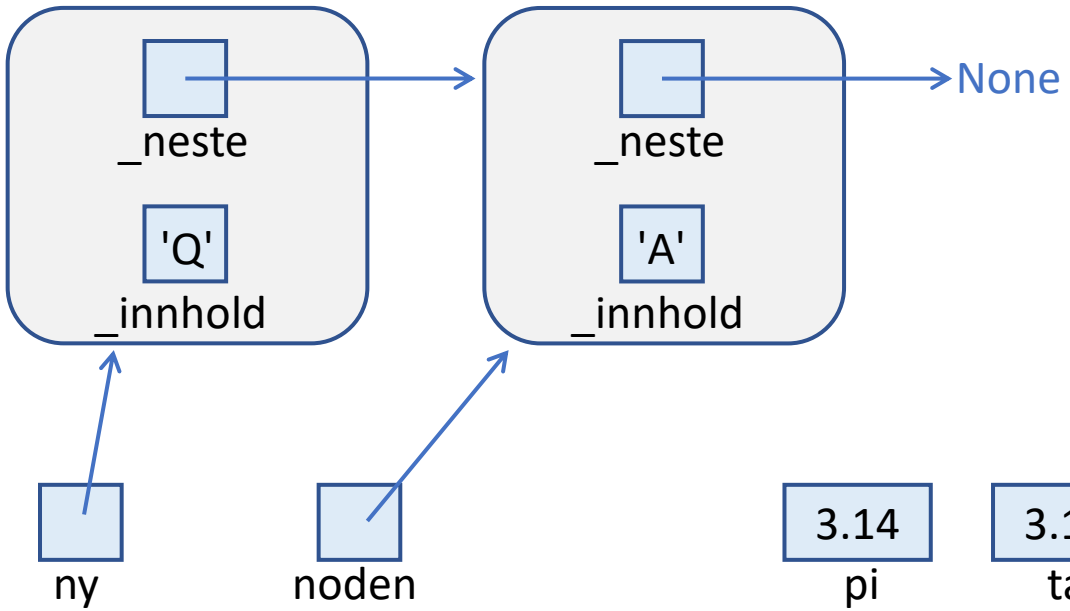

```
ny = noden
tall = pi
noden._neste = Node('A')
noden = noden._neste
t = ny._innhold
```

Tegning av muterbare objekter/verdier og immuterbare objekter/verdier



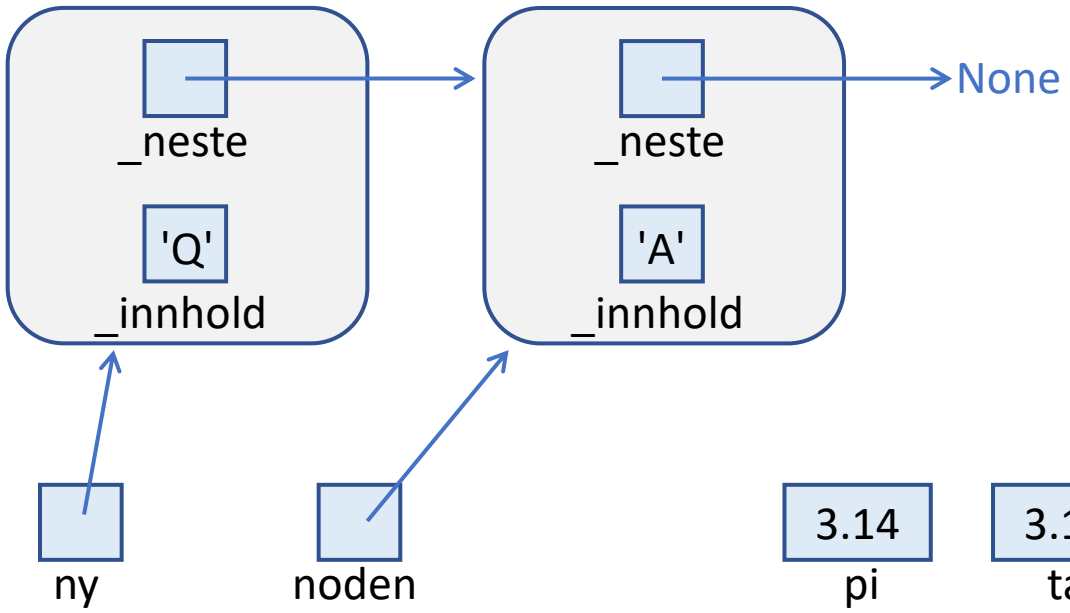

```
ny = noden
tall = pi
noden._neste = Node('A')
noden = noden._neste
t = ny._innhold
```

Tegning av muterbare objekter/verdier og immuterbare objekter/verdier

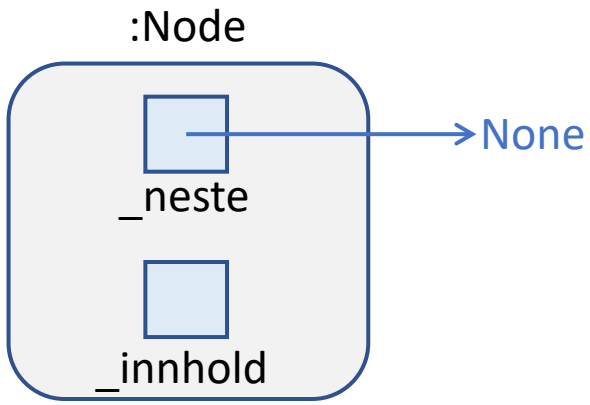



```
ny = noden
tall = pi
noden._neste = Node('A')
noden = noden._neste
t = ny._innhold
```

Tegning av muterbare objekter/verdier og immuterbare objekter/verdier

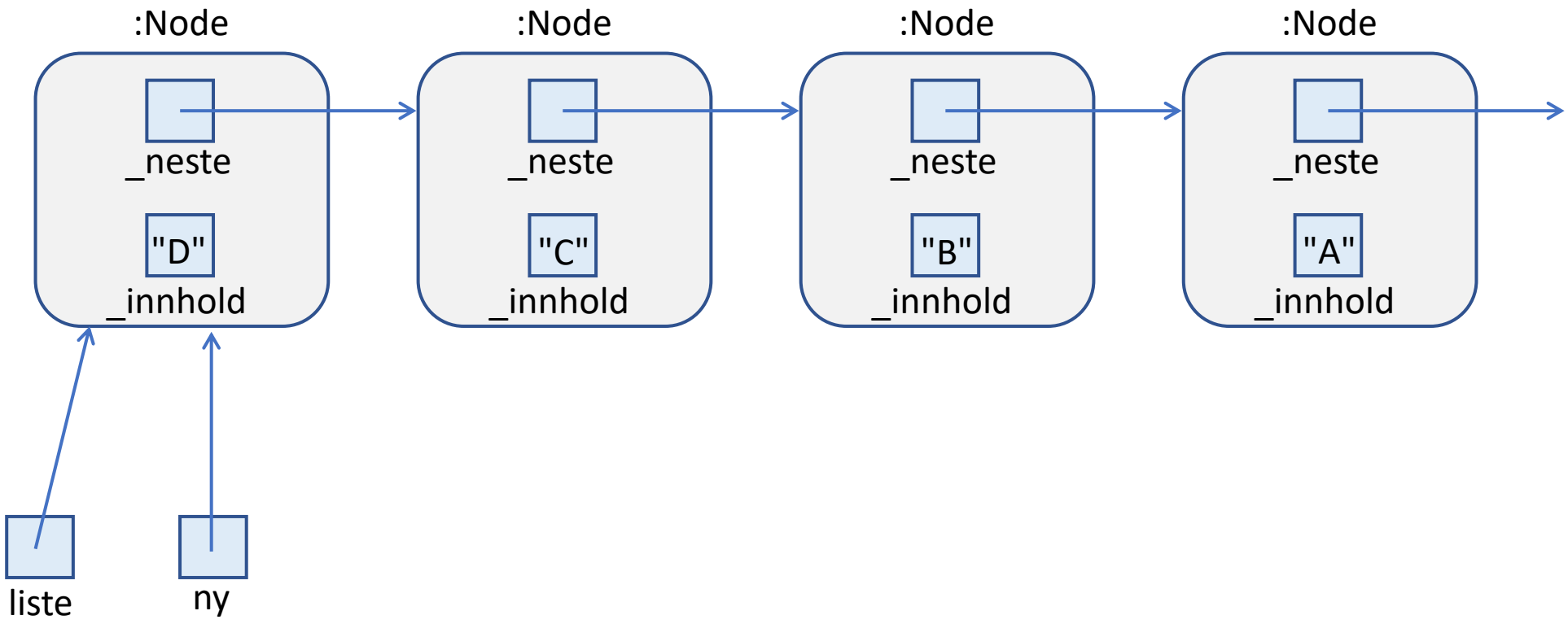



```
ny = noden
tall = pi
noden._neste = Node('A')
noden = noden._neste
t = ny._innhold
```


:Node
None
_neste
_innhold

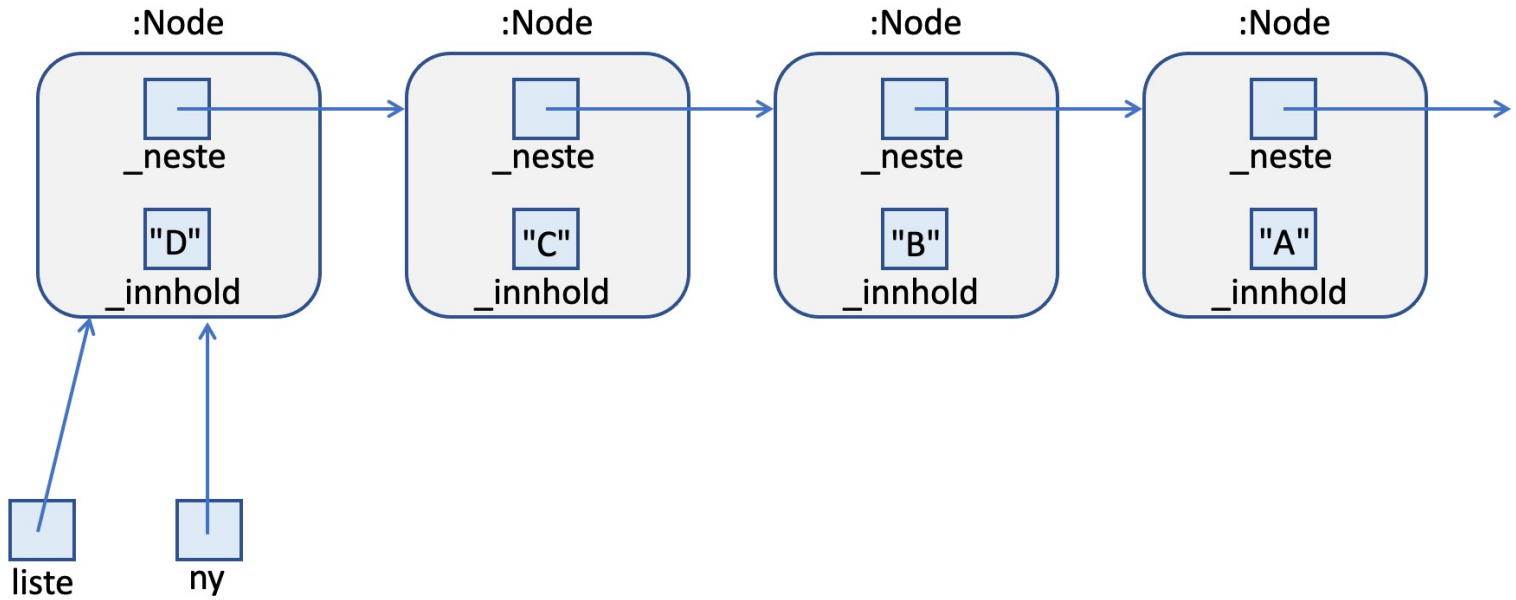
```python
class Node :
    def __init__(self, nytt) :
        self._innhold = nytt
        self._neste = None

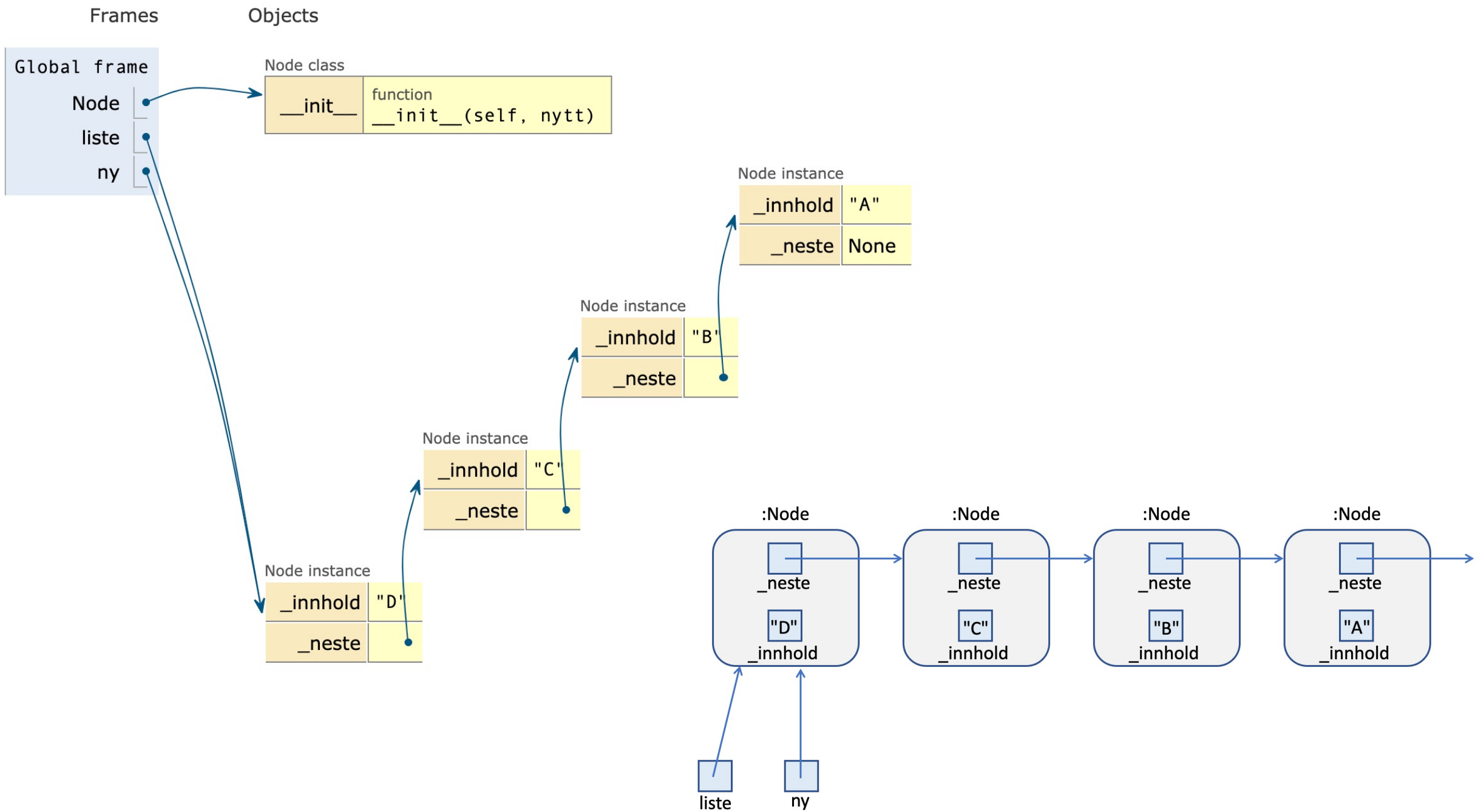
liste = None
ny = Node("A")
ny._neste = liste
liste = ny
ny = Node("B")
ny._neste = liste
liste = ny
ny = Node("C")
ny._neste = liste
liste = ny
ny = Node("D")
ny._neste = liste
liste = ny
```

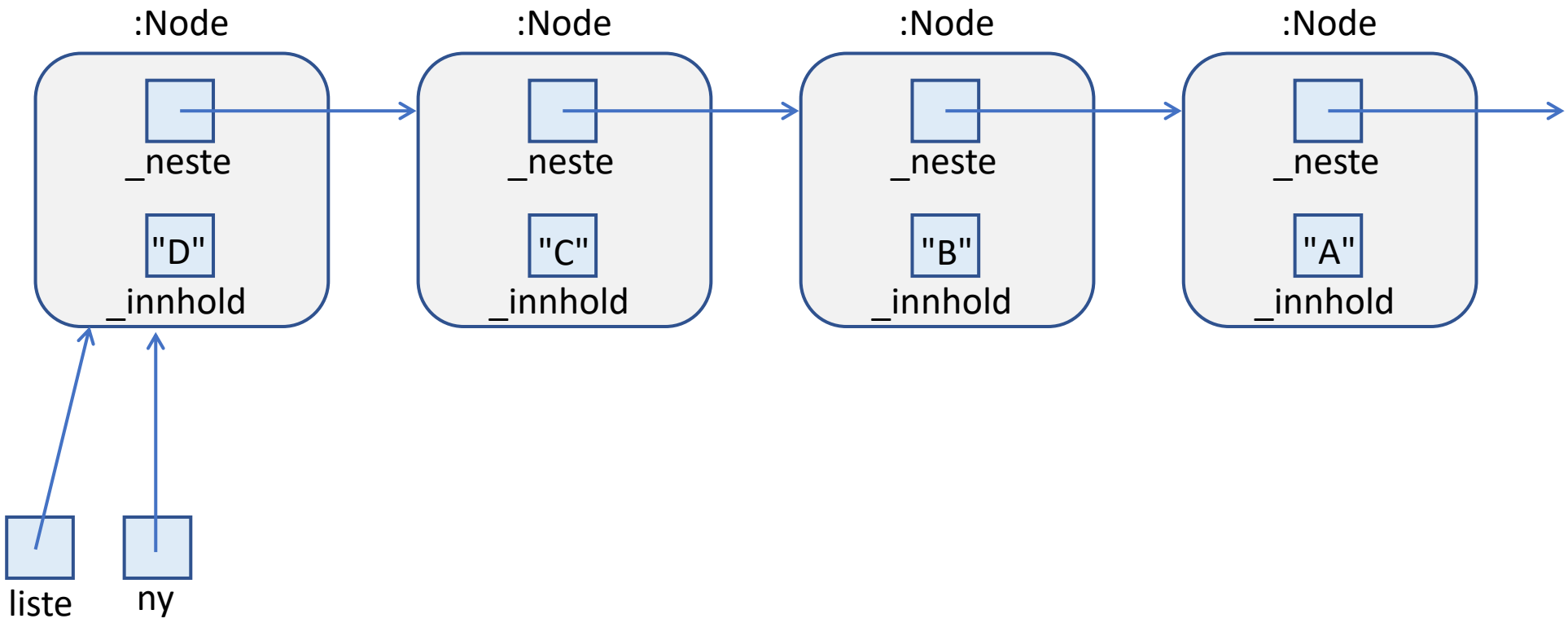

:Node
:Node
:Node
:Node
_neste
_neste
_neste
_neste
"D"
_innhold
"C"
_innhold
"B"
_innhold
"A"
_innhold
liste
ny

```python
class Node :
    def __init__(self, nytt) :
        self._innhold = nytt
        self._neste = None

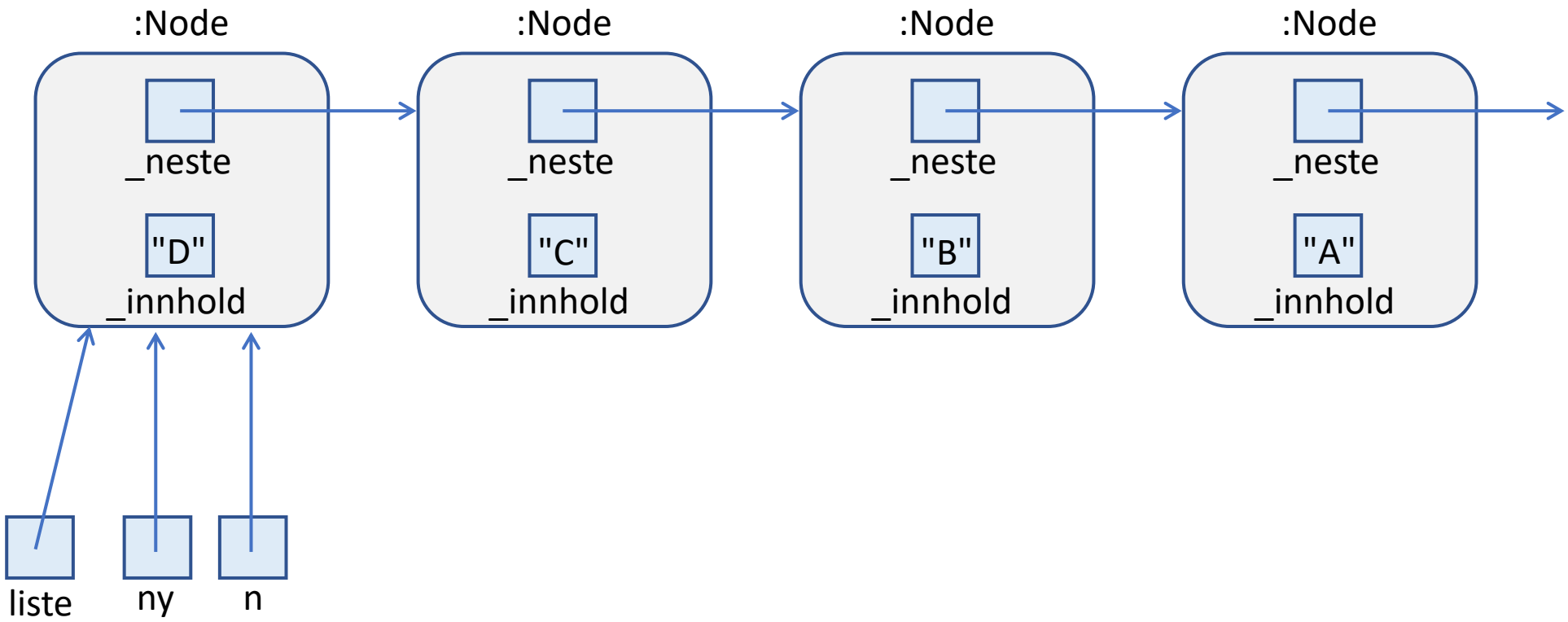
liste = None
ny = Node("A")
ny._neste = liste
liste = ny
ny = Node("B")
ny._neste = liste
liste = ny
ny = Node("C")
ny._neste = liste
liste = ny
ny = Node("D")
ny._neste = liste
liste = ny
```



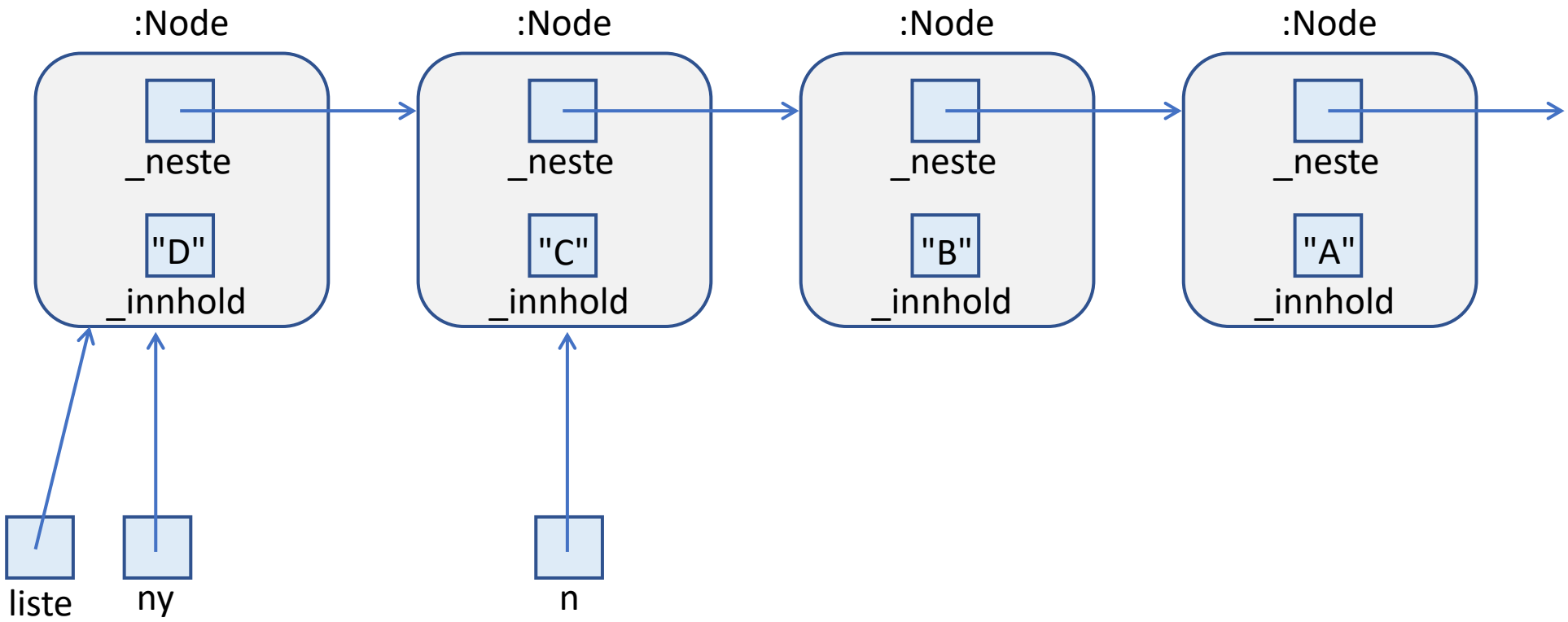

Frames
Objects
Global frame
Node
liste
ny
Node class
__init__
function
__init__(self, nytt)
Node instance
_innhold
"A"
_neste
None
Node instance
_innhold
"B"
_neste
Node instance
_innhold
"C"
_neste
Node instance
_innhold
"D"
_neste
:Node
_neste
"D"
_innhold
:Node
_neste
"C"
_innhold
:Node
_neste
"B"
_innhold
:Node
_neste
"A"
_innhold
liste
ny



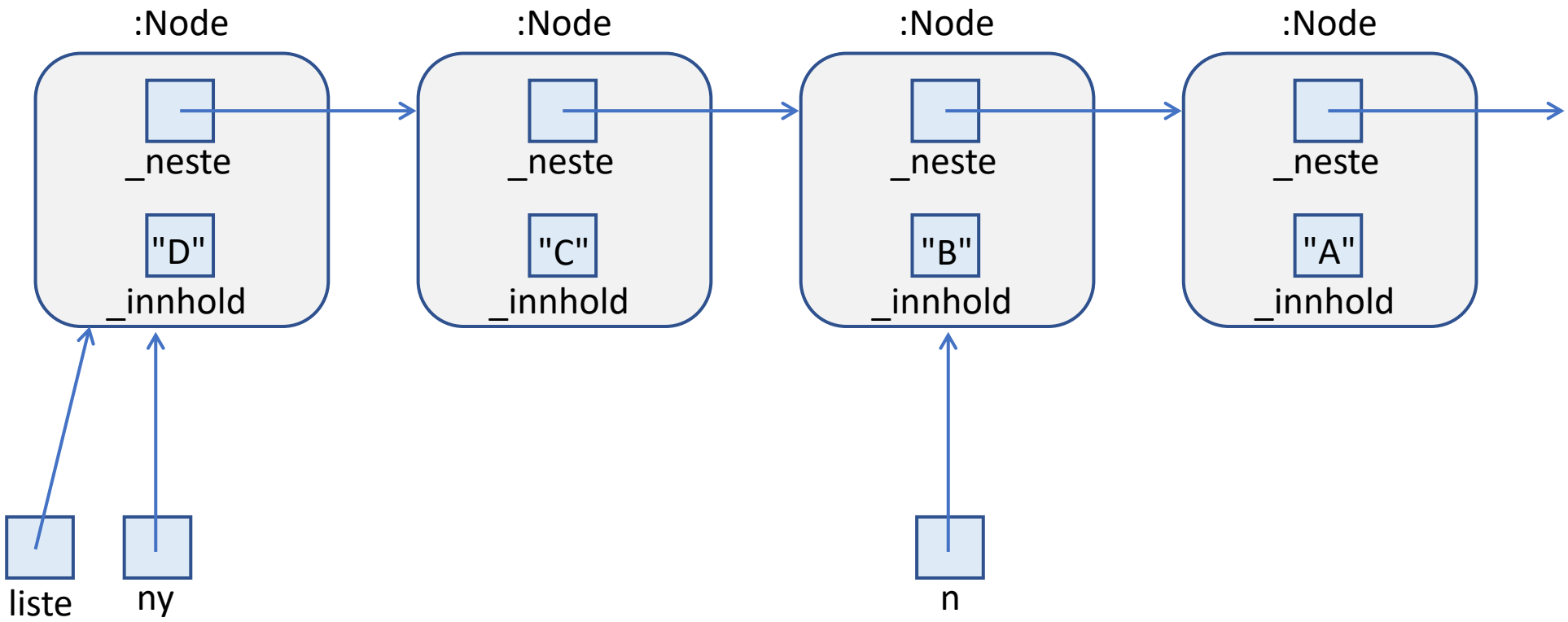

```
20  n = liste
21  while n != None:
22      print(n._innhold)
23      n = n._neste
```



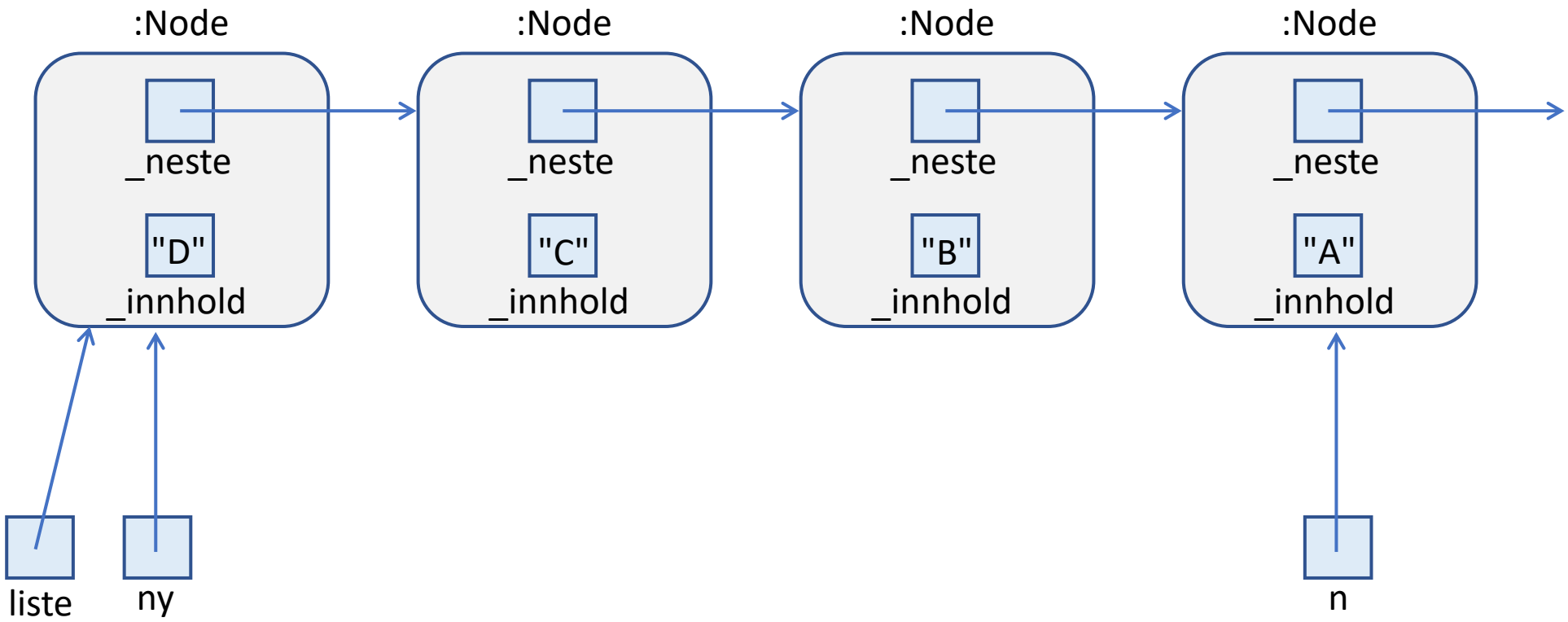

```
20  n = liste
21  while n != None:
22      print(n._innhold)
23      n = n._neste
```



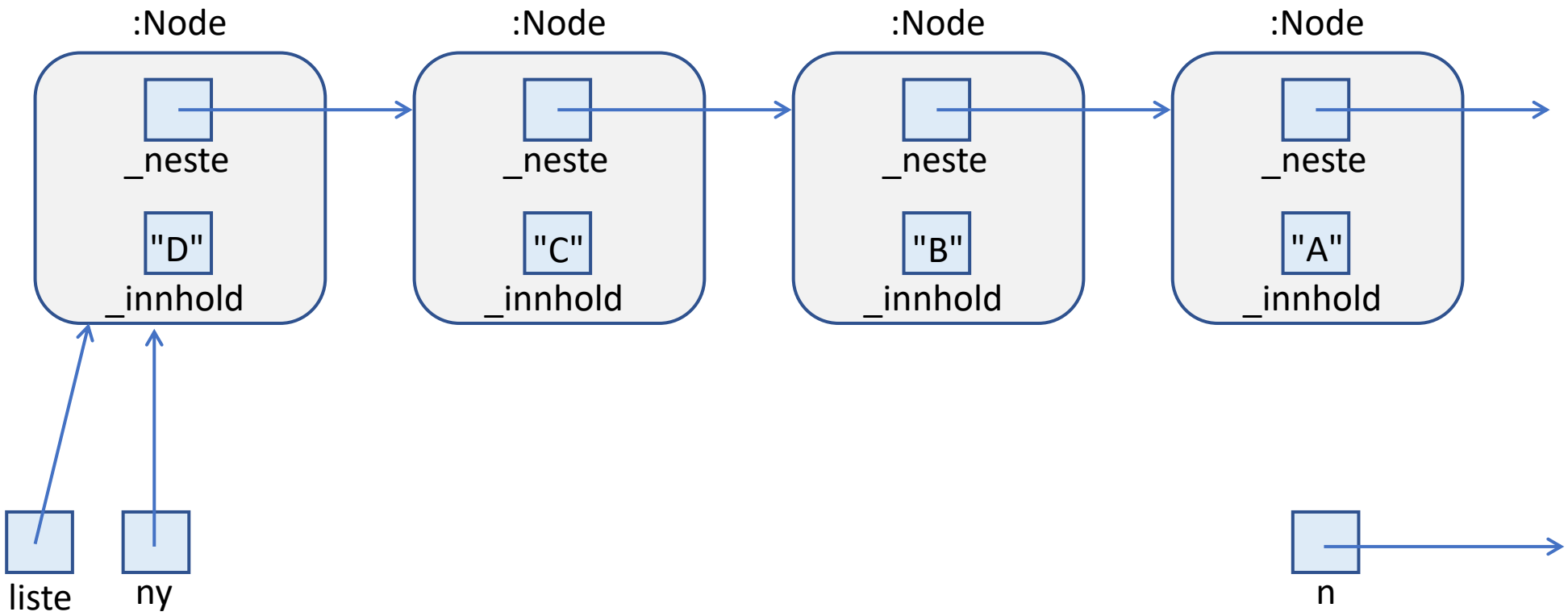

```
20  n = liste
21  while n != None:
22      print(n._innhold)
23      n = n._neste
```



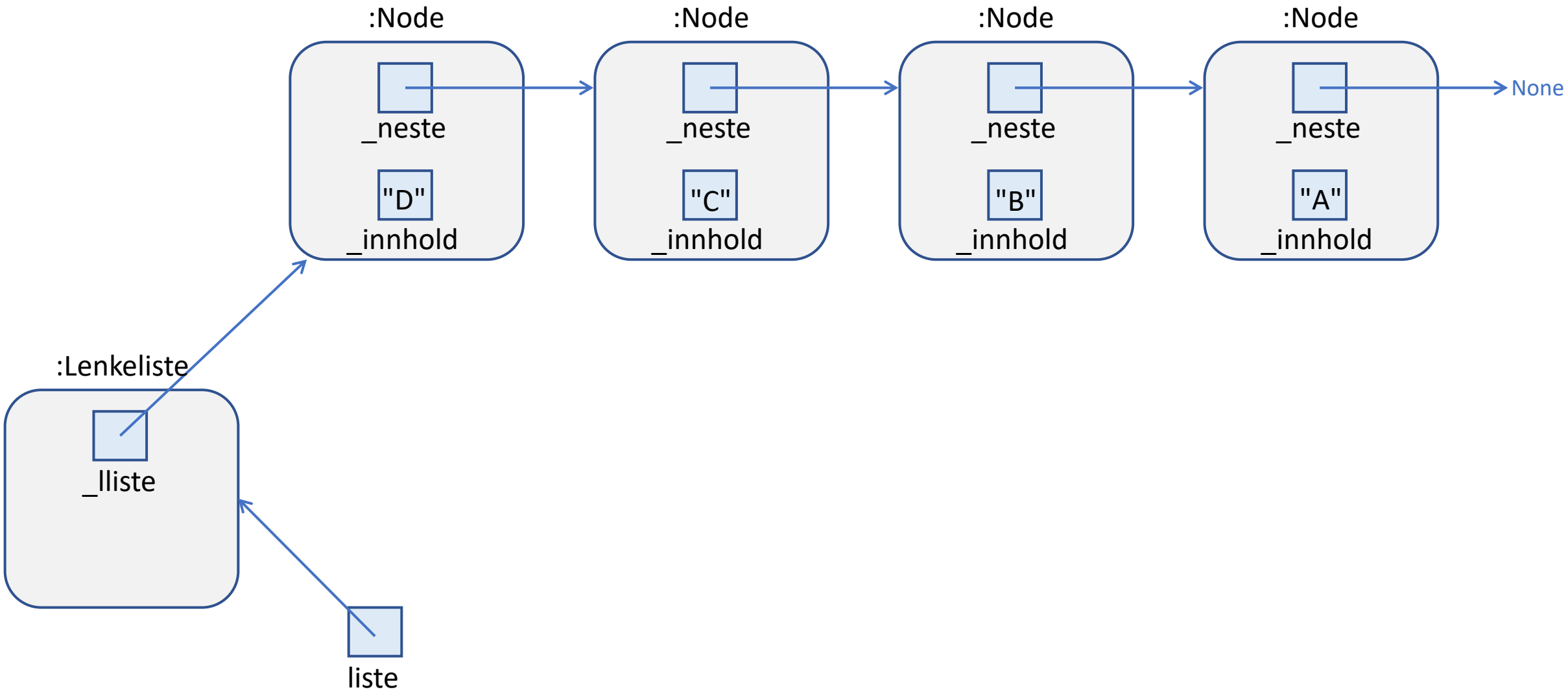

```
20  n = liste
21  while n != None:
22      print(n._innhold)
23      n = n._neste
```

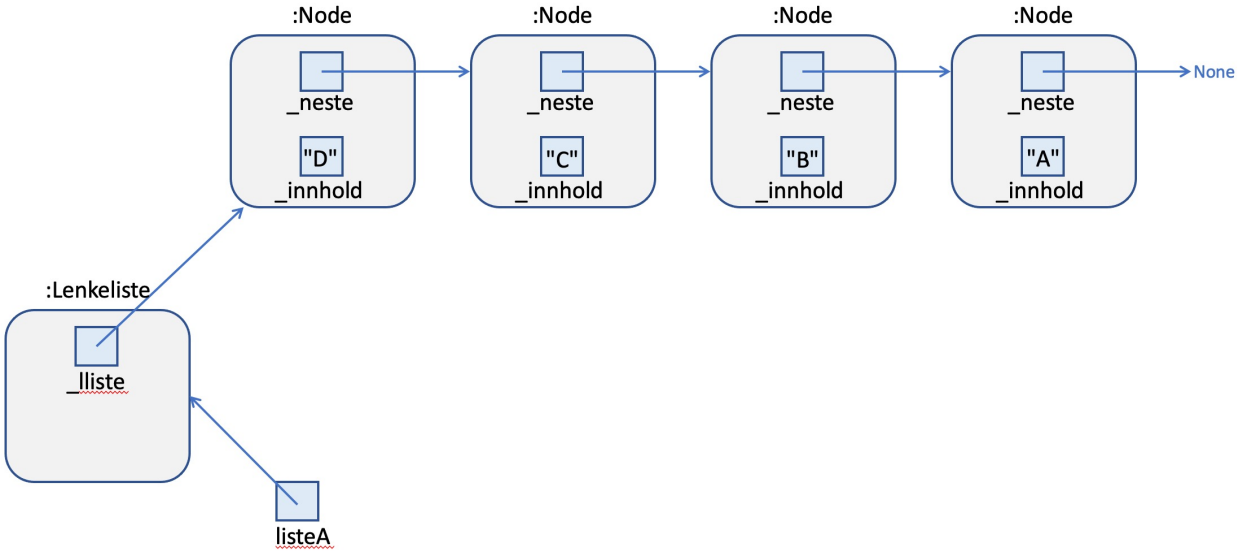



```
20  n = liste
21  while n != None:
22      print(n._innhold)
23      n = n._neste
```




```
20  n = liste
21  while n != None:
22      print(n._innhold)
23      n = n._neste
```


:Node
:Node
:Node
:Node
_neste
_neste
_neste
_neste
"D"
"C"
"B"
"A"
_innhold
_innhold
_innhold
_innhold
None
:Lenkeliste
_lliste
liste




```python
class Node :
    def __init__(self, nytt) :
        self._innhold = nytt
        self._neste = None

    def settNeste (self, ny) :
        self._neste = ny

    def neste (self) :
        return self._neste

    def hentData (self) :
        return self._innhold

```

```python
class Lenkeliste :
    def __init__(self) :
        self._lliste = None

    def push(self, innhold):
        ny = Node(innhold)
        ny.settNeste(self._lliste)
        self._lliste = ny

    def pop(self):
        ut = self._lliste.hentData()
        self._lliste = self._lliste.neste()
        return ut

    def ikkeTom(self):
        return self._lliste != None

listeA = Lenkeliste()
for tegn in 'ABCD':
    listeA.push(tegn)

while listeA.ikkeTom():
    print(listeA.pop())
```