

# W3-S4-C4-instruction-del

December 15, 2014

## 1 L'instruction del

### 1.1 Complément - niveau basique

Voici un récapitulatif sur l'instruction `del` selon le contexte dans lequel elle est utilisée.

#### 1.1.1 Sur une variable

On peut annuler la définition d'une variable. Dans la suite on capture une exception `NameError` qui est produite lorsqu'une variable est utilisée alors qu'elle n'a aucune valeur. La capture de l'exception permet d'afficher un message plutôt qu'une exception. Nous discuterons du fonctionnement des exceptions en semaine 6.

```
In []: # la variable a n'est pas définie
      try:
          print 'a=', a
      except NameError as e:
          print "a n'est pas définie"
```

```
In []: # on la définit
      a = 10

      try:
          print 'a=', a
      except NameError as e:
          print "a n'est pas définie"
```

```
In []: # on peut effacer la variable
      del a

      try:
          print 'a=', a
      except NameError as e:
          print "a n'est pas définie"
```

#### 1.1.2 Sur une liste

On peut enlever d'une liste les éléments qui correspondent à une *slice*

```
In []: # on se donne une liste
      l = range(12)
      print l
```

```
In []: # on considère une slice dans cette liste
print 'slice=', l[2:10:3]

# voyons ce que ça donne si on efface cette slice
del l[2:10:3]
print "après del", l
```

### 1.1.3 Sur un dictionnaire

Avec `del` on peut enlever une clé, et donc la valeur correspondante, d'un dictionnaire

```
In []: # partons d'un dictionnaire simple
d = dict(foo='bar', spam='eggs', a='b')
print d
```

```
In []: # on peut enlever une clé avec del
del d['a']
print d
```

### 1.1.4 On peut passer à `del` une liste

```
In []: # Voyons où en sont nos données
print 'l', l
print 'd', d
```

```
In []: # on peut invoquer 'del' avec une liste - comme 'print'
del l[3:], d['spam']

print 'l', l
print 'd', d
```

### 1.1.5 Pour en savoir plus

La page sur [l'instruction `del`](#) dans la documentation python