

# Python

## Les propriétés

Nicolas Delestre

# Contexte

## Getter, setter, pas très *pythonic*

- Les *setter* et *getter* permettent l'encapsulation :
  - séparer la façon dont est conçue la classe de la façon dont sont utilisés les objets (pouvoir modifier la conception tout en ayant la même façon de les utiliser)
  - vérifier que l'utilisation est bonne (et levée des *exceptions* si besoin)
- Mais pas très *pythonic* (faire confiance à l'utilisateur)

## Les propriétés (*properties*)

- faire croire à l'utilisateur qu'il accède directement aux attributs
- être *pythonic* tout en préservant l'encapsulation

```
>>> pt = Point2D(1,2)
>>> pt.get_x()
1
>>> pt.set_x(3)
```

⇒

```
>>> pt = Point2D(1,2)
>>> pt.x
1
>>> pt.x = 3
```

# Comment déclarer une propriété ?

## Première syntaxe

```
attr = property(getter[, setter[, deleter,[ doc]]])
```

## Mise à jour de la classe Point2D du module point.py

```
5 class Point2D:
6     def __init__(self, x, y, identifiant=None):
7         self._x = x
8         self._y = y
9         self._id = identifiant
10
11     def _get_x(self):
12         return self._x
13
14     def _set_x(self, x):
15         self._x = x
16
17     x = property(_get_x, _set_x, doc="l'abscisse du point 2D")
```

# Comment déclarer une propriété ?

## Seconde syntaxe (utilisation des décorateurs)

```
@property
def attr(self):
    ...

@attr.setter
def attr(self, valeur):
    ...

@attr.deleter
def attr(self):
    ...
```

### Attention

Ce n'est pas de la surcharge syntaxique  
(cf. cours sur les décorateurs)

## Mise à jour de la classe Point2D du module point.py (suite)

```
19 @property
20 def y(self):
21     return self._y
22
23 @y.setter
24 def y(self, y):
25     self._y = y
```

```
28 @property
29 def id(self):
30     return self._id
```

# Conclusion

## Nous avons dans ce cours

- L'objectif des propriétés en python ?
- Les deux syntaxes de déclaration des propriétés

## Principe général de python

Le concepteur/développeur d'une API écrit plus de code pour simplifier la vie de l'utilisateur