

Les classes et les objets

- Classes
- Méthodes
- Exceptions
- Exemples

Programmation Python Partie 1 : Les classes et les objets

Les classes et les objets

Classes

Méthodes

Exceptions

Exemples

Classes

- Les *classes* sont les éléments centraux de la *programmation orientée objet*
- Une classe déclare des propriétés communes à un ensemble d'objets.

Les classes et les objets

Classes

Méthodes

Exceptions

Exemples

- La classe déclare des **attributs** représentant l'état des objets et des **méthodes** représentant leur comportement.
- On parle d'un objet en tant qu'**instance** d'une classe.
- En Python une classe est définie de manière analogue aux fonctions mais en utilisant le mot clé `class`.

Les classes et les objets

Classes

Méthodes

Exceptions

Exemples

- Une méthode est une fonction dans la classe. Mot clé `def`
- Le premier argument d'un méthode doit être `self`: argument obligatoire. Cet objet `self` est une référence à l'instance.
- Certains noms de méthodes ont un sens particulier, par exemple :
 - `__init__`: nom de la méthode invoquée à la création de l'objet
 - `__str__`: méthode invoquée lorsque une représentation de la classe sous forme de chaîne de caractères est demandée, par exemple quand une instance est passée à la fonction `print`
 - voir <http://docs.python.org/2/reference/datamodel.html#special-method-names> pour les autres noms spéciaux

Exemple

```
In [130]: class Point(object):
            """
            Classe pour représenter un point dans le plan.
            """
            def __init__(self, x, y):
                """
                Creation d'un nouveau point en position x, y.
                """
                self.x = x
                self.y = y

            def translate(self, dx, dy):
                """
                Translate le point de dx and dy.
                """
                self.x += dx
                self.y += dy

            def __str__(self):
                return "Point: [%f, %f]" % (self.x, self.y)
```

Les classes et les objets

Classes

Méthodes

Exceptions

Exemples

Les classes et les objets

Classes

Méthodes

Exceptions

Exemples

Pour créer une nouvelle instance de la classe:

```
In [131]: p = Point(x=0, y=0) # appel à __init__  
          print(p.x)  
          print(p.y)  
          print("%s" % p)      # appel à la méthode __str__  
  
0  
0  
Point: [0.000000, 0.000000]
```

Les classes et les objets

Classes

Méthodes

Exceptions

Exemples

Pour invoquer une méthode de la classe sur une instance `p` de celle-ci:

```
In [132]: p = Point(1, 1)
          p.translate(0.25, 1.5)
          print(p)
```

```
Point: [1.250000, 2.500000]
```

Les classes et les objets

Classes

Méthodes

Exceptions

Exemples

Exceptions

- Dans Python les erreurs sont gérées à travers des "Exceptions"
- Une erreur provoque une *Exception* qui interrompt l'exécution normale du programme
- Des exemples d'exceptions sont: `SyntaxError`, `NameError`, `TypeError`, `KeyError`, `IndentationError`
- Une exécution à l'intérieur d'un bloc de code `try - except` peut être capturée

Les classes et les objets

Classes

Méthodes

Exceptions

Exemples

Exemple d'utilisation

- Une exception est levée à l'aide du mot clé `raise`
- Un cas d'utilisation typique est l'arrêt de l'exécution d'une fonction en cas d'erreur sur ses arguments

```
In [133]: def ma_fonction(arg):  
            if not isinstance(arg, list):  
                raise Exception("arg doit être de type list")  
            # et on continue  
            print(arg)
```

Les classes et les objets

Classes

Méthodes

Exceptions

Exemples

```
In [134]: ma_fonction([1, 2])  # list -> OK
```

```
[1, 2]
```

```
In [135]: ma_fonction({'a':1, 'b':2})  # dict -> Exception
```

```
-----  
-----  
Exception                                Traceback (most recent call last)  
<ipython-input-135-c1cdb6a4517f> in <module>()  
----> 1 ma_fonction({'a':1, 'b':2})  # dict -> Exception  
  
<ipython-input-133-5845d7d9e964> in ma_fonction(arg)  
      1 def ma_fonction(arg):  
      2     if not isinstance(arg, list):  
----> 3         raise Exception("arg doit être de type list")  
  
      4     # et on continue  
      5     print(arg)
```

```
Exception: arg doit être de type list
```

Les classes et les objets

Classes

Méthodes

Exceptions

Exemples

Pour capturer les erreurs on utilise `try` et `except`

```
In [136]: try:
            # code à exécuter si tout se passe bien
            pass
        except:
            # code à exécuter si une exception est générée
            pass
```

Par exemple:

```
In [137]: try:
            ma_fonction({'a':1, 'b':2})
            # Exception -> OK n'est pas affiché
            print('OK')
        except:
            print("Exception interceptée")
```

Exception interceptée

Les classes et les objets

Classes

Méthodes

Exceptions

Exemples

Pour obtenir de l'information sur l'erreur, on peut accéder à l'instance de la classe `Exception` concernée:

```
except Exception as e:
```

```
In [138]: try:
            # variable non définie
            print(var)
        except Exception as e:
            print("Exception interceptée: %s" % e)
```

Exception interceptée: name 'var' is not defined