PYTHON

Présenté par :

Xavier TABUTEAU

Surcharges

- **Définition de la surcharge**

La surcharge (overloading) en programmation désigne la capacité d’une fonction ou d’un opérateur à fonctionner différemment selon le type ou le nombre d’arguments.

En Python, contrairement à d'autres langages comme Java ou C++, la surcharge des fonctions n'est pas directement prise en charge, car Python ne permet pas d’avoir plusieurs fonctions portant le même nom avec des signatures différentes. Cependant, il existe des moyens d’implémenter un comportement similaire.

- **La surcharge des fonctions**

Python ne permet pas plusieurs définitions d’une fonction avec le même nom. Si on définit plusieurs fois une fonction avec le même nom, seule la dernière définition est conservée. Pour contourner cette limitation, on peut utiliser des valeurs par défaut ou *args pour gérer un nombre variable d'arguments.

- **La surcharge d’opérateur**

Python permet la surcharge des opérateurs en redéfinissant des méthodes spéciales (dunder methods ou méthodes magiques).

Surcharges

- **Méthodes mathématiques :**

__add__(self, other)	: +	(addition)
__sub__(self, other)	: -	(soustraction)
__mul__(self, other)	: *	(multiplication)
__truediv__(self, other)	: /	(division)
__floordiv__(self, other)	: //	(division entière)
__mod__(self, other)	: %	(modulo)
__pow__(self, other)	: **	(puissance)
__matmul__(self, other)	: @	(multiplication matricielle)
__and__(self, other)	: &	(Et logique)
__or__(self, other)	: |	(Ou logique)

- **Méthodes de comparaison :**

__eq__(self, other)	: ==	(égalité)
__ne__(self, other)	: !=	(différent)
__lt__(self, other)	: <	(inférieur)
__le__(self, other)	: <=	(inférieur ou égal)
__gt__(self, other)	: >	(supérieur)
__ge__(self, other)	: >=	(supérieur ou égal)

Surcharges

- **Méthodes in-place :**

__iadd__(self, other)	: +=	(addition)
__isub__(self, other)	: -=	(soustraction)
__imul__(self, other)	: *=	(multiplication)
__itruediv__(self, other)	: /=	(division)
__ifloordiv__(self, other)	: //=	(division entière)
__imod__(self, other)	: %=	(modulo)
__ipow__(self, other)	: **=	(puissance)
__imatmul__(self, other)	: @=	(multiplication matricielle)
__iand__(self, other)	: &=	(Et logique)
__ior__(self, other)	: |=	(Ou logique)

- **Méthodes d’affichage :**

__str__(self)	: str(objet), print(objet)	(affichage)
__repr__(self)	: repr(objet)	(autre affichage)
__format__(self, format_spec)	: format(objet, format_spec)	(format)
__byte(self)	: bytes(objet)	(bytes code)

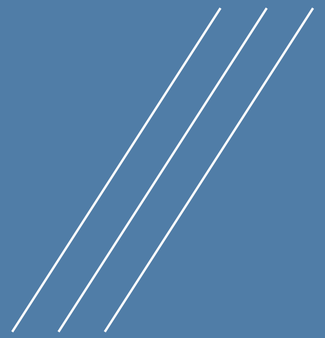
Surcharges

- **Fonctions sur les attributs d'une classe**

__getattribute__(self, attr)	: objet.attr
__gettattr__(self, attr)	: objet.attr
__setattr__(self, attr, val)	: objet.attr = val
__delattr__(self, attr)	: del objet.attr
__dir__(self)	: dir(objet)

surcharge_op.py
ex_surcharge_op

PYTHON



Présenté par

Xavier TABUTEAU