

23 NSI 10

```
In [ ]: class Pile:
        """
        Classe definissant une pile
        """
        def __init__(self):
            self.valeurs = []

        def est_vide(self):
            """
            Renvoie True si la pile est vide, False sinon
            """
            return self.valeurs == []

        def empiler(self, c):
            """
            Place l'element c au sommet de la pile
            """
            self.valeurs.append(c)

        def depiler(self):
            """
            Supprime l'element place au sommet de la pile, a condition qu'elle
            soit non vide
            """
            if self.est_vide() == False:
                self.valeurs.pop()
```

```
In [ ]: def parenthesage(ch):
        """
        Renvoie True si la chaine ch est bien parenthesee et False sinon
        """
        p = Pile()
        for c in ch:
            if c == '(':
                p.empiler(c)
            elif c == ')':
                if p.est_vide():
                    return False
                else:
                    p.depiler()
        return p.est_vide()
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js