

# Correction

NSI - 2021 Étranger Jour 1 (21-NSIJ1G11)

## Exercice 2 - Gestion d'un parc de vélo

1.

L'instruction `flotte[26]` renvoie le dictionnaire `{'type': 'classique', 'etat': 1, 'station': 'Coliseum'}`.

L'instruction `print(flotte[80]["etat"])` renvoie l'entier `0`.

L'instruction `print(flotte[99]["etat"])` génère une erreur : **KeyError** car la clé `99` n'existe pas dans le dictionnaire `flotte`.

2. a. Les valeurs possibles de la variable `choix` sont les chaînes de caractères `"electrique"` et `"classique"`.

2. b. Cette fonction renvoie la station du premier vélo disponible et correspondant au choix du type de l'utilisateur. Les dictionnaires en python étant non-ordonnés, on ne peut pas savoir quel sera ce premier vélo.

3. a.

# Affiche les identifiants des vélos disponibles à la station Citadelle

```
for id_velo in flotte: # équivalent à for id_velo in flotte.keys() :
```

```
    if flotte[id_velo]["station"] == "citadelle" and flotte[id_velo]["etat"] == 1:
        print(id_velo)
```

3. b.

# Renvoie les identifiants et les stations des vélos électrique

# qui ne sont pas en panne.

```
for id_velo in flotte: # équivalent à for id_velo in flotte.keys() :
```

```
    if flotte[id_velo]["type"] == 'electrique' and flotte[id_velo]["etat"] != -1:
        print(id_velo, flotte[id_velo]["station"])
```

4.

```
def dispo_proche(coord_u):
```

```
    """ Renvoie le nom, la distance et les identifiants des vélos disponibles
        pour chaque station située à moins de 800 mètres
        et ayant au moins un vélo disponible.
    """
```

```
    """
```

```
    rep = []
```

```
    for nom_s, coord_s in stations.items():
```

```
        velos = dispo(nom_s)
```

```
        d = distance(coord_u, coord_s)
```

```
        if d < 800 and velos:
```

```
            rep.append((nom_s, d, velos))
```

```
    return rep
```