Gestion des réservoirs d'eau

Sujet n°19 — Corrigé

Question 1 *Réservoir en pénurie*

```
def est_en_penurie(reservoirs, nom_reservoir):
    for r in reservoirs:
        if r["nom"] == nom_reservoir:
            return r["volume"] / r["capacite"] < 0.2
```

Sur les données fournies : `"Motuavai"` → `True` (10 %) ; tous les autres → `False`.

Question 2 *Volume total par district*

```
def volume_par_district(reservoirs):
    volumes = {}
    for r in reservoirs:
        d = r["district"]
        if d not in volumes:
            volumes[d] = 0
        volumes[d] += r["volume"]
    return volumes
```

Données fournies : `{'Tepua': 90000, 'Fare': 160000, 'Hiva Oro': 90000, 'Avera': 130000}`.

Question 3 *Volume moyen : diagnostic et correction*

Cause : division par `len(reservoirs) - 1` au lieu de `len(reservoirs)` → `ZeroDivisionError` sur une liste à 1 élément, moyenne surestimée sinon.

```
# liste a 1 reservoir : ne doit pas lever une erreur
un = [{"nom": "A", "capacite": 1000, "volume": 500, "district": "X"}]
assert volume_moyen(un) == 500

# deux reservoirs de meme volume : la moyenne doit etre ce volume
deux = [{"nom": "A", "capacite": 1000, "volume": 500, "district": "X"},
        {"nom": "B", "capacite": 1000, "volume": 500, "district": "Y"}]
assert volume_moyen(deux) == 500
```

```
def volume_moyen(reservoirs):
    somme_totale = 0
    for r in reservoirs:
        somme_totale += r["volume"]
    return somme_totale / len(reservoirs)
```

Données fournies : 47 000,0 litres (version buguée : 52 222 litres, soit +11 %).

Question 4 *Districts vulnérables*

```
def districts_vulnerables(reservoirs):
    seuil = 0.8 * volume_moyen(reservoirs)
    rpd = reservoirs_par_district(reservoirs)
```

```
    vulnerables = []
    for district in liste_districts(reservoirs):
        if volume_moyen(rpd[district]) < seuil:
            vulnerables.append(district)
    return vulnerables
```

Données fournies (seuil 37 600 L) : `['Tepua']` (30 000 L de moyenne, seul district sous le seuil).