

## Fiche de révision – Instructions Python du TP HTTP

### 1.1 Rappel – Ce que vous avez utilisé dans le TP

Notion	Ce qu'on a fait
<code>str.split(sep)</code>	Découper une chaîne → obtenir une <b>liste</b>
<code>str.split(sep, maxsplit)</code>	Découper en limitant le nombre de coupures
<code>liste[i]</code>	Accéder à un élément par son <b>indice</b>
<code>str.startswith(prefixe)</code>	Tester si une chaîne commence par un mot donné
<code>{}</code>	Créer un <b>dictionnaire</b>
<code>dico[clé] = valeur</code>	Ajouter une entrée dans un dictionnaire
<code>dico[clé]</code>	Lire une valeur dans un dictionnaire
<code>def f(param):</code>	Définir une <b>fonction</b> – cette première ligne s'appelle la <b>signature</b>
<code>assert</code>	Vérifier automatiquement qu'un résultat est correct

### 1.2 Exercices

Pour chaque question, écrire l'instruction Python demandée. Vous pouvez vous aider du tableau ci-dessus.

1. On dispose de la chaîne suivante :

```
1 s = "GET /index host=www.monsite.fr path=/accueil"
```

Écrire l'instruction qui découpe `s` en utilisant l'espace comme séparateur, et stocke le résultat dans une variable `mots`.

```
1 mots = ...
```

2. Après l'instruction précédente, `mots` contient :

```
1 ['GET', '/index', 'host=www.monsite.fr', 'path=/accueil']
```

Écrire l'instruction qui récupère la valeur `'/index'` dans une variable `chemin`.

```
1 chemin = ...
```

3. On dispose de la chaîne :

```
1 parametre = "host=www.monsite.fr"
```

Écrire l'instruction qui découpe `parametre` au niveau du `=`, en s'assurant d'obtenir exactement 2 morceaux, même si la valeur contenait elle-même un `=`.

```
1 morceaux = ...
```

---

4. Écrire l'instruction qui teste si la chaîne `"payload=id=alice"` commence par `"payload"`, et stocke le résultat dans `resultat`.

```
1 resultat = ...
```

Quelle valeur contient `resultat`? \_\_\_\_\_  
\_\_\_\_\_

5. Écrire les instructions qui créent un dictionnaire vide `infos`, puis y ajoutent les entrées suivantes :  
- clé `'nom'`, valeur `'Dupont'` - clé `'prenom'`, valeur `'Alice'`

```
1 infos = ...  
2 ...  
3 ...
```

---

6. Avec le dictionnaire `infos` de la question précédente, écrire l'instruction qui affiche uniquement la valeur associée à la clé `'prenom'`.

```
1 print(...)
```

---

7. Écrire la **signature** (première ligne) d'une fonction `analyser` qui prend en paramètre une chaîne `requete` et retourne un dictionnaire.

```
1 ...
```

---

8. On a écrit une fonction `double(n)` qui est censée retourner le double d'un nombre. Écrire une instruction `assert` qui vérifie que `double(5)` vaut bien 10.

```
1 ...
```

---

**9. Synthèse.** On dispose de la chaîne :

```
1 ligne = "utilisateur=pierre age=17 ville=Lyon"
```

Sans utiliser de boucle, écrire les instructions qui extraient les trois valeurs 'pierre', '17' et 'Lyon' dans trois variables nom, age, ville.

```
1 mots = ...
2 nom = ...
3 age = ...
4 ville = ...
```

---

**10. Fonction complète.** On dispose de la chaîne suivante, qui contient plusieurs couples clé-valeur séparés par des espaces :

```
1 donnees = "nom=Dupont prenom=Alice age=17 ville=Lyon"
```

Écrire une fonction `extraire_dict(chaine)` qui prend cette chaîne en paramètre et retourne un dictionnaire contenant tous les couples clé-valeur.

Résultat attendu :

```
1 extraire_dict(donnees)
2 # {'nom': 'Dupont', 'prenom': 'Alice', 'age': '17', 'ville': 'Lyon'}
```

```
1 def extraire_dict(chaine):
2     mots = ... # découper par les espaces
3     resultat = ... # créer un dictionnaire vide
4     for mot in mots:
5         ... # séparer la clé et la valeur
6         ... # ajouter au dictionnaire
7     return resultat
```