

Définition de JSON

JSON est un format de données léger qui permet de stocker et d'échanger des données entre les applications. Il est facile à lire et à écrire pour les humains, et facile à analyser et à générer pour les machines.

Exemple de JSON

Voici un exemple de données JSON :

```
{
  "nom": "John",
  "prenom": "Doe",
  "age": 30
}
```

Comment utiliser JSON

En Python, vous pouvez utiliser le module `json` pour travailler avec des données JSON.

```
import json

# Conversion d'un dictionnaire Python en JSON
data = {"nom": "John", "prenom": "Doe", "age": 30}
json_data = json.dumps(data)

# Conversion de JSON en dictionnaire Python
json_str = '{"nom": "Alice", "prenom": "Smith", "age": 25}'
python_data = json.loads(json_str)
```

Définition de Sérialisation

La sérialisation est le processus de conversion de données en un format JSON.

Exemple de Sérialisation

```
import json

data = {"nom": "John", "prenom": "Doe", "age": 30}
json_data = json.dumps(data)
print(json_data)
```

Définition de Désérialisation

La désérialisation est le processus de conversion de données JSON en un format utilisable en Python.

Exemple de Désérialisation

```
import json

json_str = '{"nom": "Alice", "prenom": "Smith", "age": 25}'
python_data = json.loads(json_str)
print(python_data)
```

La transformation de types entre Python et JSON

Lors de la sérialisation et de la désérialisation, les types de données Python sont convertis en équivalents JSON et vice versa. Par exemple, les dictionnaires Python deviennent des objets JSON, les listes Python deviennent des tableaux JSON, etc.

Type Python	Exemple Python	Exemple JSON	Type JSON
int	42	42	Number
float	3.14159	3.14159	Number
str	"hello"	"hello"	String
bool	True	true	Boolean
None	None	null	Null
list	[1, 2, 3]	[1, 2, 3]	Array
tuple	(1, 2, 3)	[1, 2, 3]	Array
dict	{"a": 1}	{"a": 1}	Object

Type Python	Exemple Python	Exemple JSON	Type JSON
set	set([1, 2, 3])	Non pris en charge	Non pris en charge
datetime	datetime(2022, 1, 15)	"2022-01-15T00:00:00"	String (ISO 8601)
Decimal	Decimal("10.5")	"10.5"	String
complex	2+3j	Non pris en charge	Non pris en charge
bytes	b'hello'	"aGVsbG8="	String (encodé en base64)

Exercice