

# JSON (JavaScript Object Notation)

{ j s o n }

# Table des matières

- JSON, CSV et XML
- Qu'est-ce-que le JSON ?
- Structure du JSON
- Avantages et inconvénients
- Évaluation

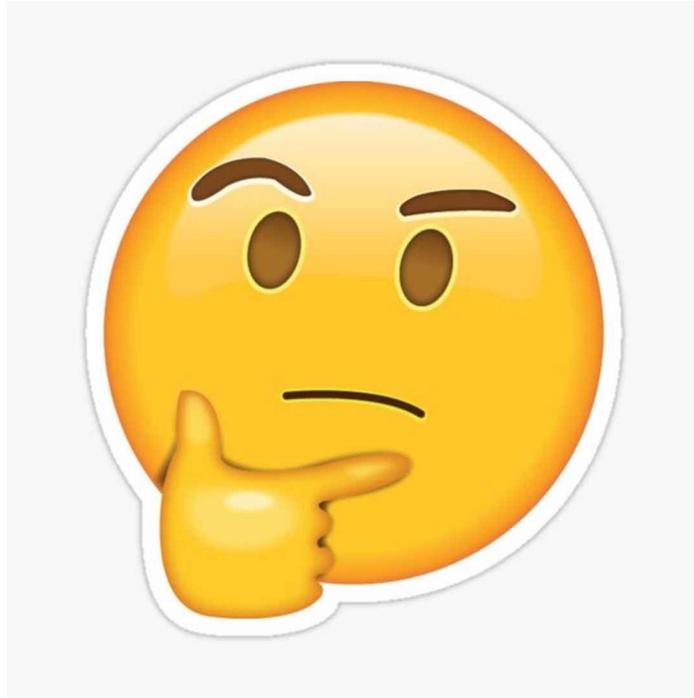
# JSON, CSV et XML



# JSON, CSV et XML

- Trois formats pour représenter un tableau de données
- Fichiers composés d'une suite de caractères
- Deux types d'informations : les données et les caractères permettant de structurer ces données

# Qu'est-ce-que le JSON ?



# Qu'est-ce-que le JSON ?

- JavaScript Object Notation
- Format léger d'échange de données dérivé de la notation des objets du langage JavaScript
- Représenter de l'information structurée (comme XML)
- Douglas Crockford entre 2002 et 2005

# Qu'est-ce-que le JSON ?

- Facile à lire ou à écrire
- Aisément analysable
- Sous-ensemble du langage de programmation JavaScript
- Format texte complètement indépendant de tout langage
- Langage d'échange de données idéal

# Structure du JSON

```
"header": {  
  "title": "The JSON example",  
  "descriptionText": "This is some title text."  
},  
"content": {  
  "title": "The content example text",  
  "elements": [  
    {  
      "title": "The first element",  
      "mainText": "First element main text",  
      "additionalText": "First element additional te  
    },  
    {  
      "title": "The second element",  
      "mainText": "Second element main text",  
      "additionalText": "Second element additional
```

# Structure du JSON

- Format plus récent
- Représenter des objets
- Ensemble de paires constituées d'une étiquette et d'une valeur ou d'une liste de valeurs
- Paires : entre accolades et séparées par des virgules
- Valeurs : listes placées entre crochets et séparées par des virgules

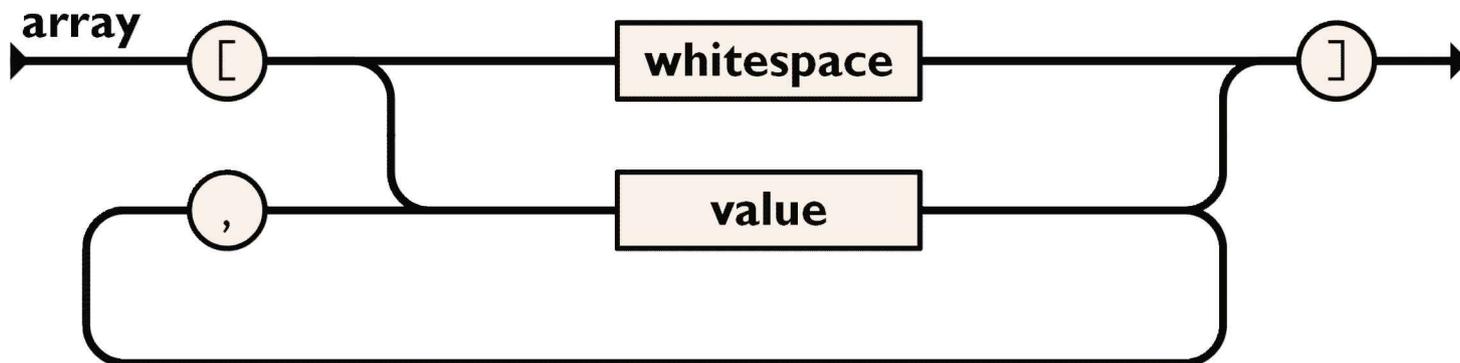
# Structure du JSON

- Collection de couples nom/valeur
- Liste de valeurs ordonnées
- Structures de données universelles

```
{
  "code_insee": 63113,
  "code_postal": 63110,
  "nb_habitants": 145041,
  "commune": 1,
  "voie": "ABERDEEN (Rue d') / entre Rue de GOMEL et Rue Alphonse DAUDET",
  "ordures_menagers_jours": "Mardi : 05h00 / 12h00",
  "ordures_menagers_periode": "",
  "ordures_menagers_observations": "",
  "dechets_recyclables_jours": "Mercredi : 14h00 / 21h00",
  "dechets_recyclables_periode": "",
  "dechets_recyclables_observations": "",
  "dechets_vegeteaux_periodes": "",
  "dechets_vegeteaux_jours": "Vendredi : 14h00 / 21h00",
  "dechets_vegeteaux_observations": "",
  "emcombrants_periode": "",
  "emcombrants_jours": "",
  "emcombrants_observations": "",
  "verre_periode": "",
  "verre_jours": "",
  "verre_observations": "",
  "deee_periode": "",
  "deee_jours": "",
  "deee_observations": "",
  "gravats_periode": "",
  "gravats_jours": "",
  "gravats_observations": "",
  "dechets_toxique_periode": "",
  "dechets_toxique_jours": "",
  "dechets_toxique_observations": "",
  "nom": "CLERMONT-FERRAND"
},
```

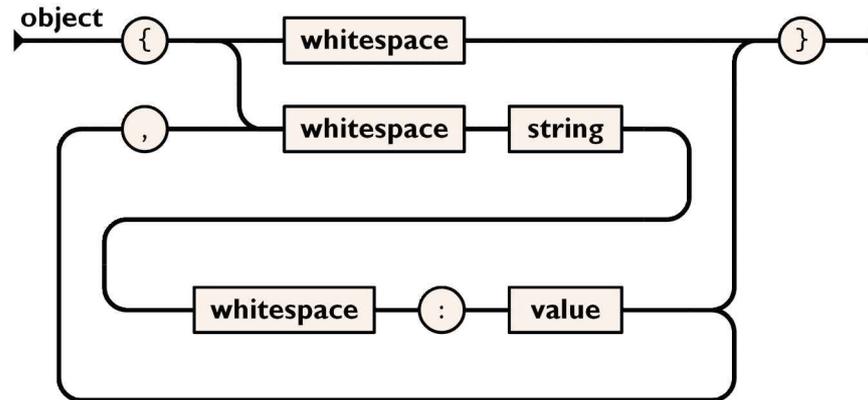
# Les crochets

- Crochets : un tableau (collection de valeurs ordonnées), séparé par des virgules



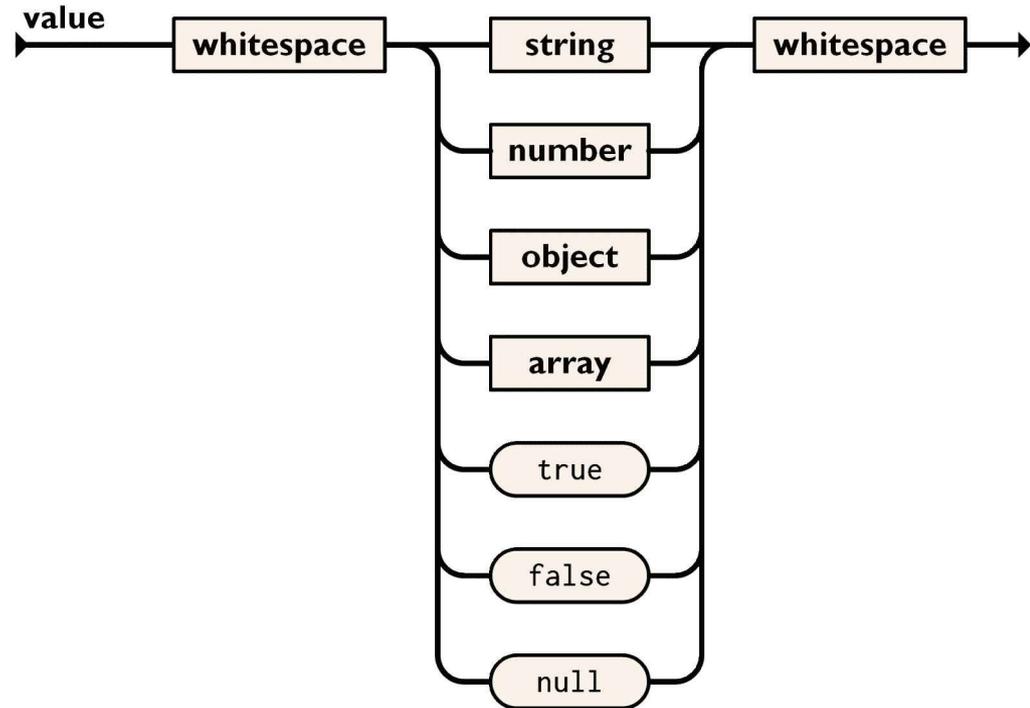
# Les accolades

- Accolades : un objet (ensemble de couples nom/valeur non ordonnés)
- Nom suivi de deux points
- Couples nom/valeur séparées par des virgules



# Les valeurs

- String (Chaîne de caractères)
- Number (Nombre)
- Object (Objet JSON)
- Array (Tableau)
- True ou False (Boolean)
- Null

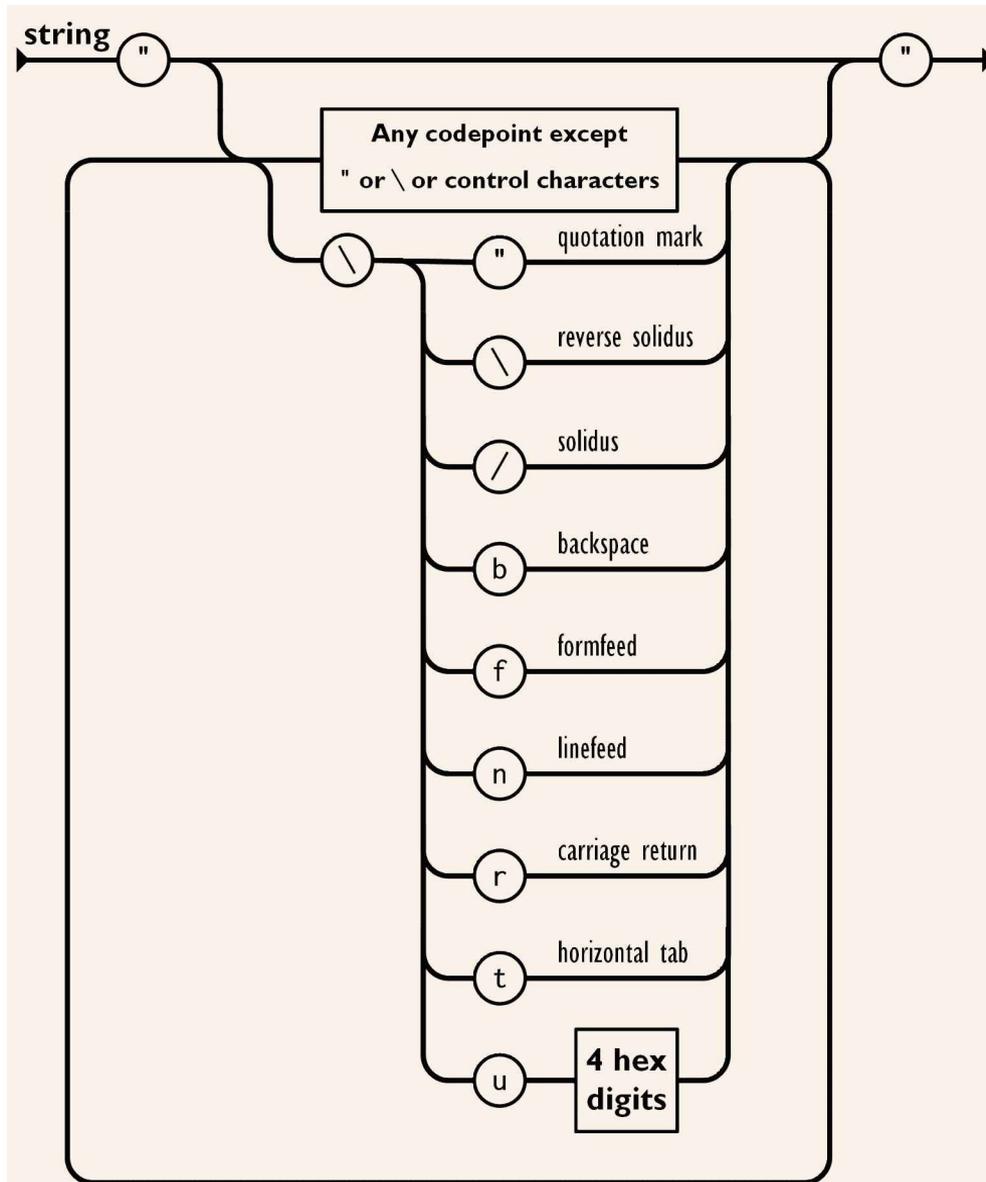


# Les valeurs

- Chaînes : entre guillemets doubles
- Nombres : un entier ou une virgule flottante

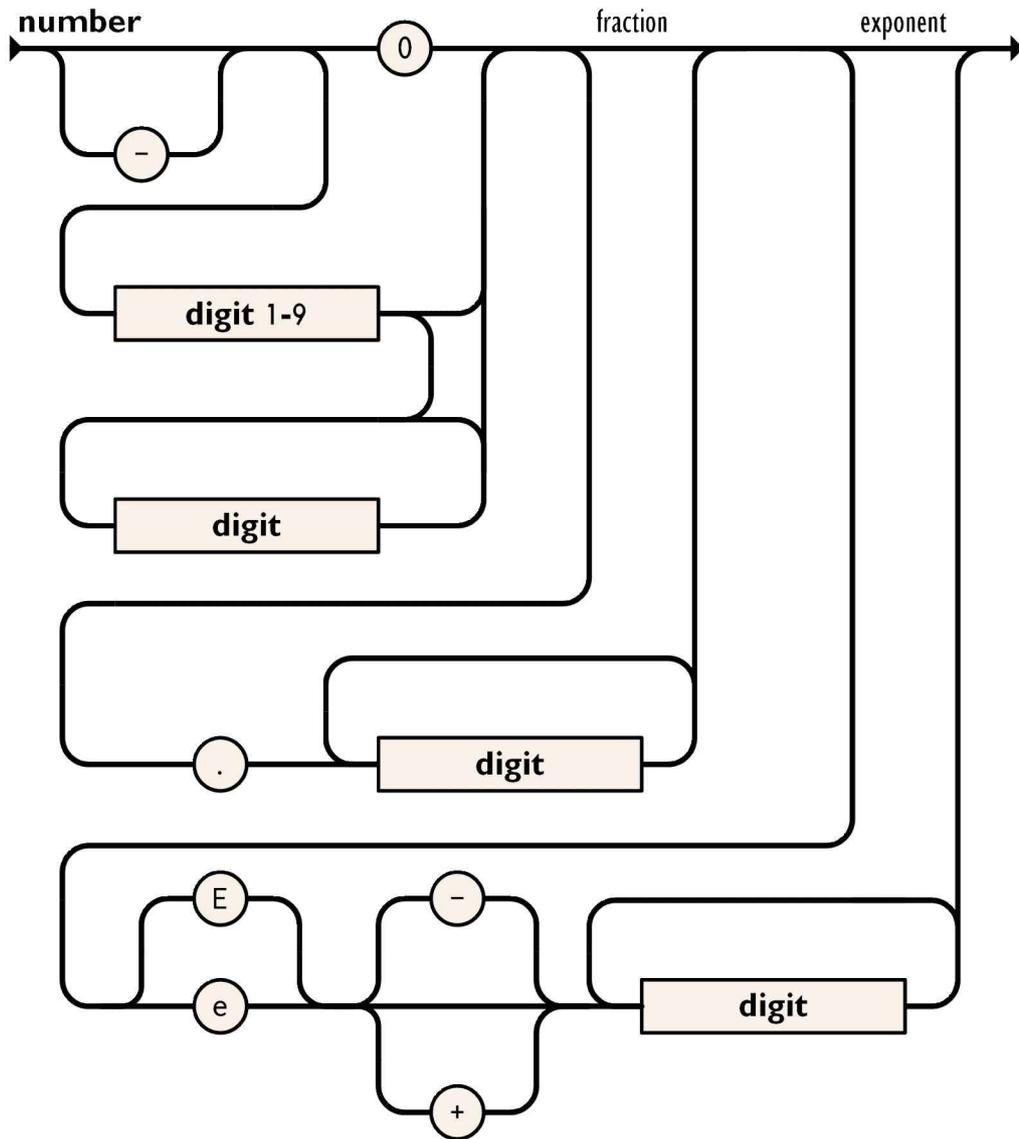
# Une chaîne de caractères

- Une chaîne de caractères : une suite de zéro ou plus
- Caractères Unicode
- Entre guillemets
- Échappements avec antislash
- Un caractère : une chaîne d'un seul caractère

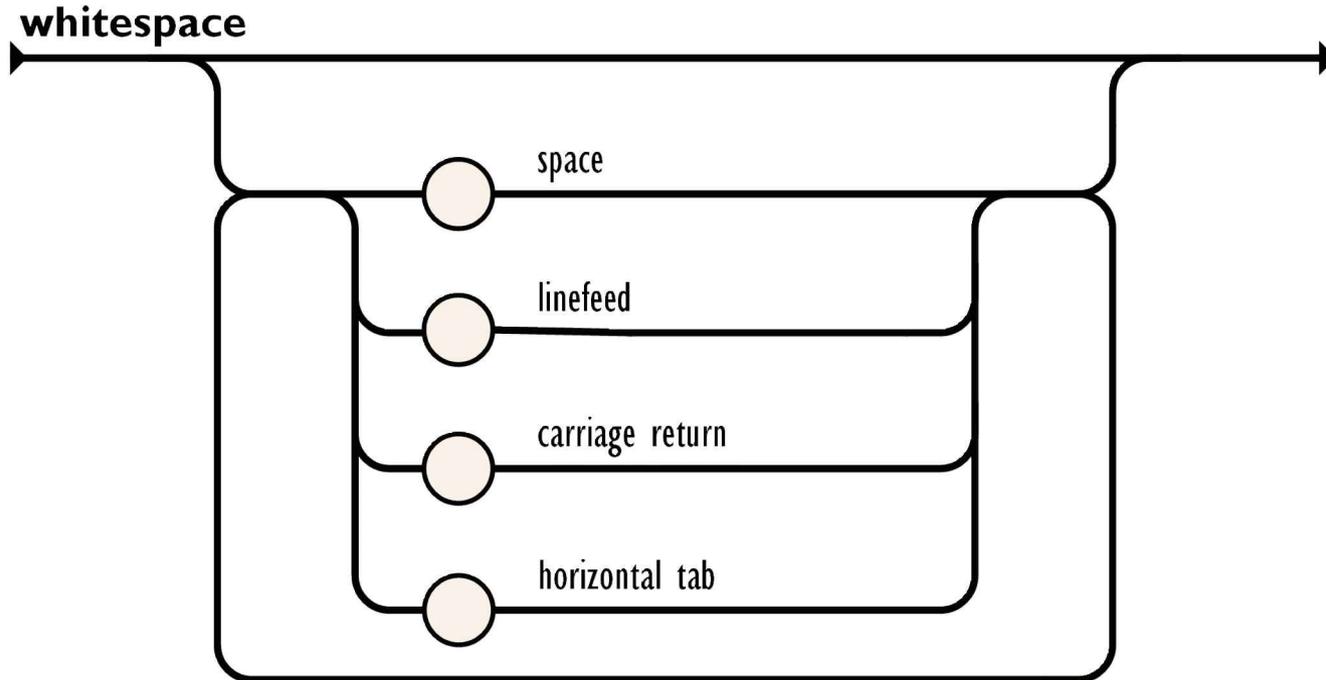


# Un nombre

- Un nombre : très proche de C ou de Java
- Les formats octal et hexadécimal ne sont pas utilisés



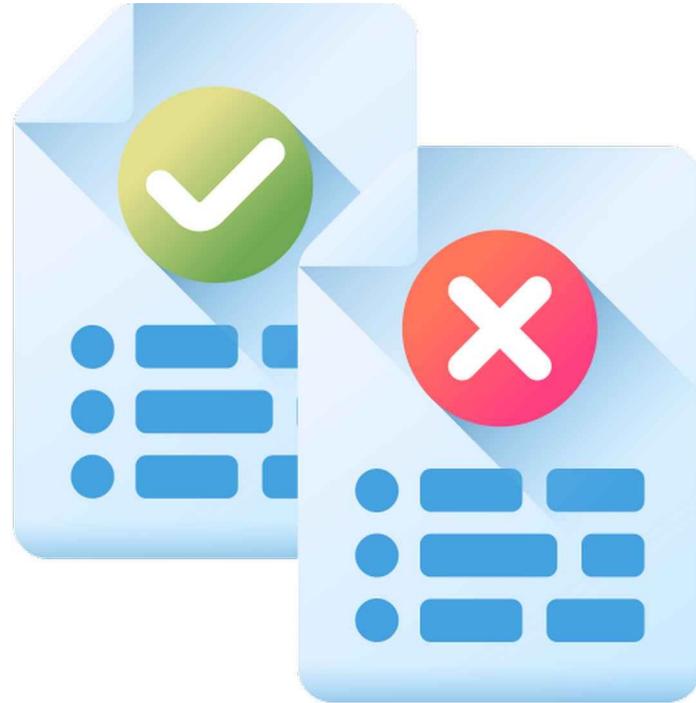
# Espace blanc



# Syntaxe JSON

- Les données sont dans des paires nom/valeur
- Les données sont séparées par des virgules
- Les accolades maintiennent des objets
- Les crochets contiennent des tableaux

# Avantages et inconvénients



# Avantages

- Langage plus concis que le langage XML
- Lecture plus simple
- Contient plus d'information (objet ou tableau, type de donnée, etc.)
- Uniquement du texte
- Données facilement envoyées et utilisées

# Avantages

- Format de plus en plus reconnu
- Similaire à JavaScript
- Fonctions intégrées
- `JSON.parse()` pour convertir les chaînes JSON en objets JavaScript
- `JSON.stringify()` pour convertir un objet en une chaîne JSON

# Inconvénients

- Absence de schéma : accroît le risque de « déformer » les données
- Pas de type date : problèmes de divergences de format
- Absence de commentaires : augmente les risques d'erreurs de compréhension

# Fin du cours

Avez-vous des questions ?



*Avant de passer à l'évaluation ...*

# Évaluation



# Déroulement

- Vous allez devoir réaliser un passage de fichiers JSON en CSV à l'aide d'un programme réalisé avec le langage PHP
- L'objectif étant d'approfondir vos connaissances ainsi que vos compétences en PHP mais également au niveau de la manipulation des données au sein d'un fichier JSON/CSV

# Conditions de mise en œuvre

Avant de commencer l'évaluation, vous devez :

- Télécharger le fichier JSON concernant la collecte des déchets de la ville de Clermont-Ferrand
- Télécharger le programme PHP (incomplet)
- public\_html : 192.168.197.22/~ant

# Objectif

- Compléter le programme suivant (en vous aidant de la documentation PHP) en respectant la structure du CSV :

Code Insee / Code postal / Commune / Nombre habitants / Rue / Ordures / Récurrence / Tri / Récurrence

# Bonus

- Séparez les jours et horaires des ordures ménagères et du tri sélectif en deux colonnes dans le fichier CSV
- Utiliser les ressources/aides recommandées (Documentation PHP, Regex101, etc.)