

Using Function Calling For Structure

Simple Example 1: Address Extraction

```
text = \
"""
John Doe lives at 123 Elm Street, Springfield. Next to him is Jane Smith, residing at 456 Oak Avenue, Lakeview. Not far away, we find
"""
print (text)
```

John Doe lives at 123 Elm Street, Springfield. Next to him is Jane Smith, residing at 456 Oak Avenue, Lakeview. Not far away, we find

```
raven_prompt = \
f'''
Function:
def address_name_pairs(names : list[str], addresses : list[str]):
"""
Give names and associated addresses.
"""

{text}<human_end>
'''
```

```
from utils import query_raven

def address_name_pairs(names : list[str], addresses : list[str]):
"""
Give names and associated addresses.
"""
for name, addr in zip(names, addresses):
    print (name, ": ", addr)

result = query_raven(raven_prompt)
eval(result)
```

```
John Doe : 123 Elm Street, Springfield
Jane Smith : 456 Oak Avenue, Lakeview
Dr. Emily Ryan : 789 Pine Road, Westwood
Mr. Alan Turing : 101 Binary Blvd, Computerville
Ms. Olivia Newton : 202 Music Lane, Harmony
Prof. Charles Xavier : 505 Mutant Circle, X-Town
```

Alternative Way of Doing Extraction

```
unbalanced_text = \
"""
Dr. Susan Hill has a practice at 120 Green Road, Evergreen City, and also consults at 450 Riverdale Drive, Brookside. Mark Twain,
"""
print (unbalanced_text)
```

Dr. Susan Hill has a practice at 120 Green Road, Evergreen City, and also consults at 450 Riverdale Drive, Brookside. Mark Twain, t

```
raven_prompt = \
f'''

@dataclass
class Record:
    name : str
    addresses : List[str]

Function:
def insert_into_database(names : List[Record]):
'''
```

```

    Inserts the records into the database.
    """

    {unbalanced_text}<human_end>

    """
Insert code cell below (Ctrl+M B)
result = query_raven(raven_prompt)
print (result)

```

```

insert_into_database(names=[Record(name='Dr. Susan Hill', addresses=['120 Green Road', '450 Riverdale Drive']), Record(name='Mark T

```

Generating Valid JSONs

```

{
  "city_name" : "London"
  "location" : {
    "country" : "United Kingdom",
    "continent" : {
      "simple_name" : "Europe",
      "other_name" : "Afro-Eur-Asia"
    }
  }
}

```

```

def city_info(city_name : str, location : dict):
    """
    Gets the city info
    """
    return locals()
def construct_location_dict(country : str, continent : dict):
    """
    Provides the location dictionary
    """
    return locals()
def construct_continent_dict(simple_name : str, other_name : str):
    """
    Provides the continent dict
    """
    return locals()

```

```

print (city_info("London", {}))

```

```

{'city_name': 'London', 'location': {}}

```

```

raven_prompt = \
    """
    Function:
    def city_info(city_name : str, location : dict):
        """
        Gets the city info
        """

    Function:
    def construct_location_dict(country : str, continent : dict):
        """
        Provides the location dictionary
        """

    def construct_continent_dict(simple_name : str, other_name : str):
        """
        Provides the continent dict
        """

    User Query: {question}<human_end>
    """

```

```
question = "I want the city info for London, "\n\nwhich is in the United Kingdom, which is in Europe or Afro-Eur-Asia."
```

```
output = query_raven(raven_prompt.format(question = question))\njson0 = eval(output)\nprint (json0)
```

Insert code cell below (Ctrl+M B)

```
{'city_name': 'London', 'location': {'country': 'United Kingdom', 'continent': {'simple_name': 'Europe', 'other_name': 'Afro-Eur-As
```

```
import json\njson.dumps(json0)
```

```
'{"city_name": "London", "location": {"country": "United Kingdom", "continent": {"simple_name": "Europe", "other_name": "Afro-Eur-Asia"}}}'
```

Try These yourself!

```
question = "I need details for the city of Tokyo, "\n\nsituated in Japan, a part of the Asian continent, "\n\nwhich is sometimes referred to as Eurasia."
```

```
output = query_raven(raven_prompt.format(question = question))\njson1 = eval(output)\nprint (json1)
```

```
{'city_name': 'Tokyo', 'location': {'country': 'Japan', 'continent': {'simple_name': 'Asian', 'other_name': 'Eurasia'}}}
```

```
import json\njson.dumps(json0)
```

```
'{"city_name": "London", "location": {"country": "United Kingdom", "continent": {"simple_name": "Europe", "other_name": "Afro-Eur-Asia"}}}'
```

Start coding or generate with AI.