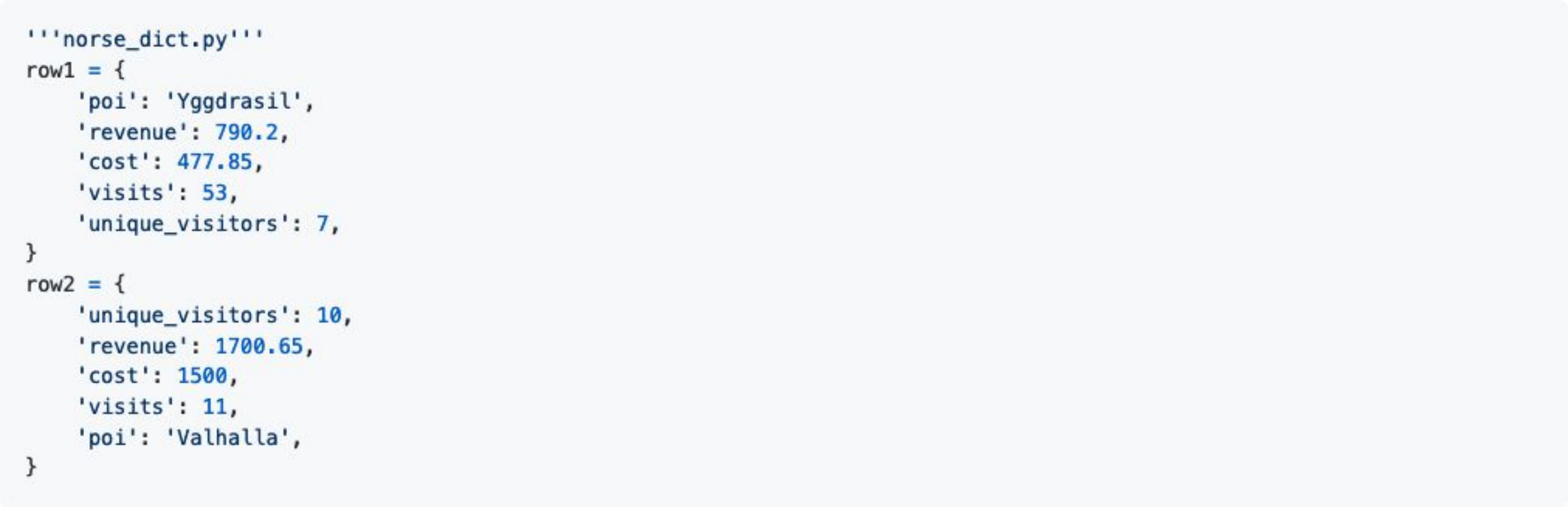
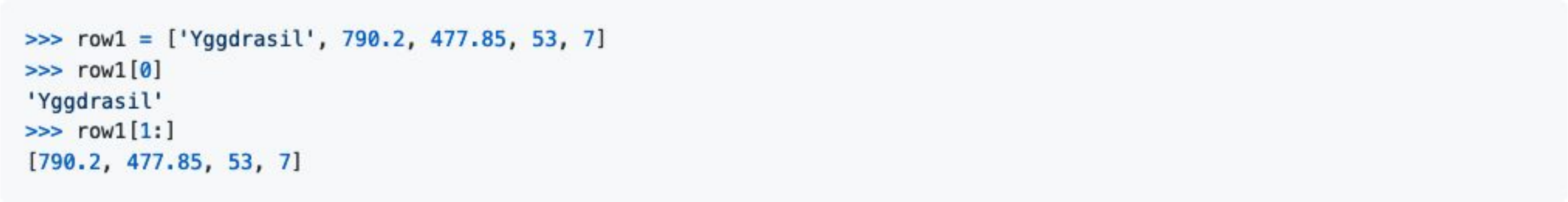
Python Curriculum

Part 03 - Data Containers and Repetitions (2/3)

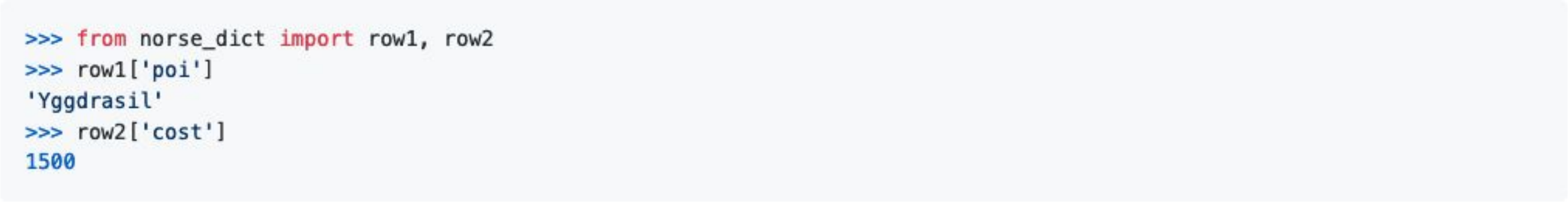
Dictionaries



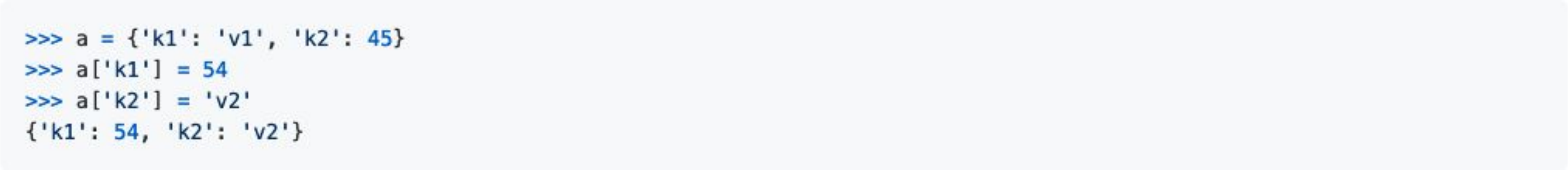
```
'''norse_dict.py'''
row1 = {
    'poi': 'Yggdrasil',
    'revenue': 790.2,
    'cost': 477.85,
    'visits': 53,
    'unique_visitors': 7,
}
row2 = {
    'unique_visitors': 10,
    'revenue': 1700.65,
    'cost': 1500,
    'visits': 11,
    'poi': 'Valhalla',
}
```



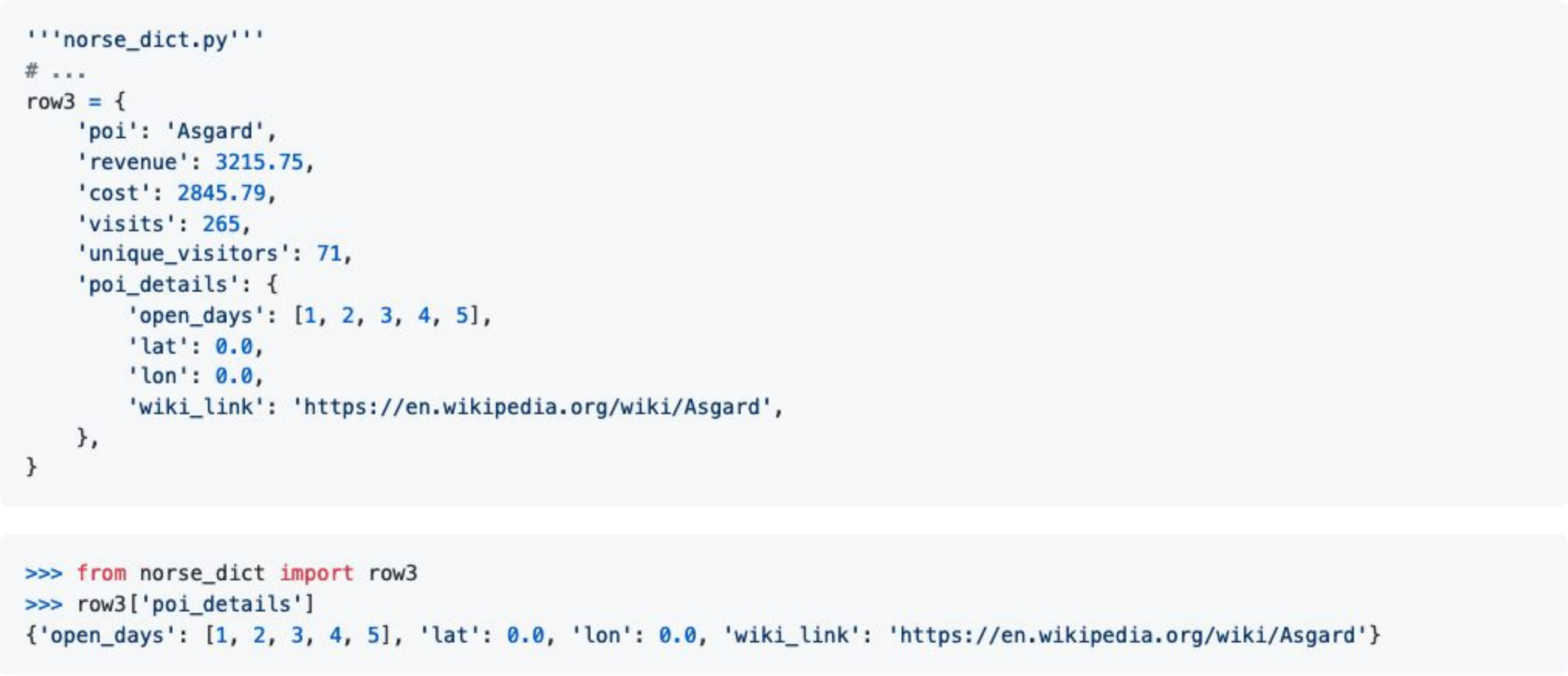
```
>>> row1 = ['Yggdrasil', 790.2, 477.85, 53, 7]
>>> row1[0]
'Yggdrasil'
>>> row1[1:]
[790.2, 477.85, 53, 7]
```



```
>>> from norse_dict import row1, row2
>>> row1['poi']
'Yggdrasil'
>>> row2['cost']
1500
```

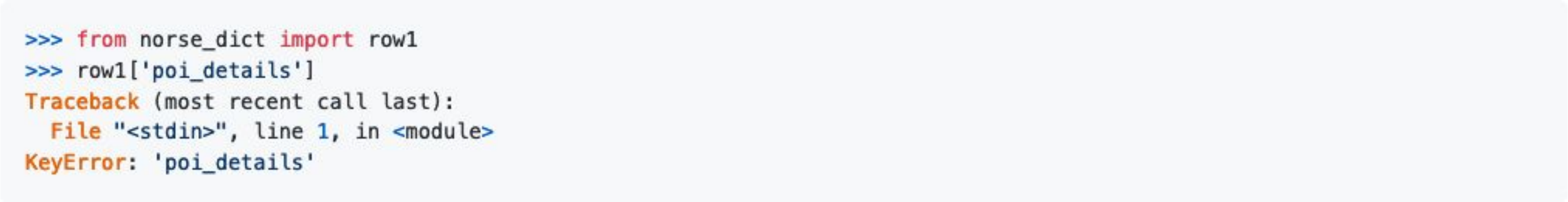


```
>>> a = {'k1': 'v1', 'k2': 45}
>>> a['k1'] = 54
>>> a['k2'] = 'v2'
{'k1': 54, 'k2': 'v2'}
```

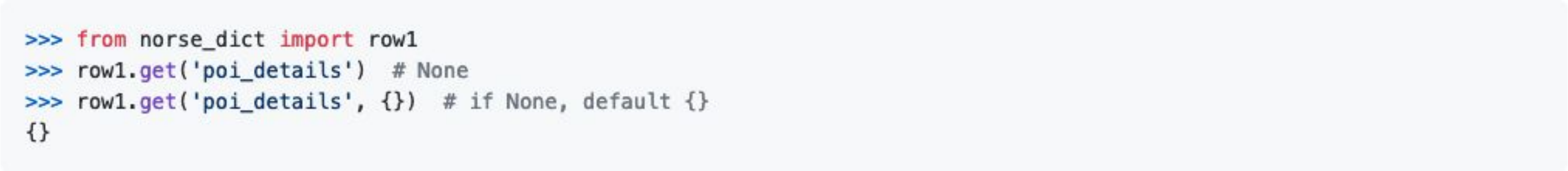


```python
'''norse_dict.py'''
# ...
row3 = {
    'poi': 'Asgard',
    'revenue': 3215.75,
    'cost': 2845.79,
    'visits': 265,
    'unique_visitors': 71,
    'poi_details': {
        'open_days': [1, 2, 3, 4, 5],
        'lat': 0.0,
        'lon': 0.0,
        'wiki_link': 'https://en.wikipedia.org/wiki/Asgard',
    },
}
```

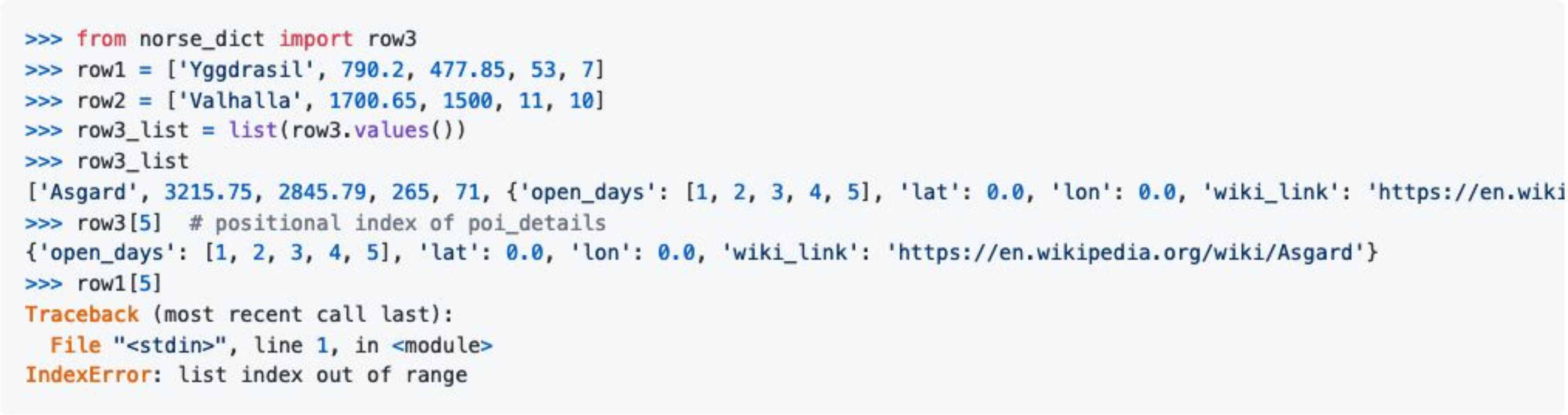
```
>>> from norse_dict import row3
>>> row3['poi_details']
{'open_days': [1, 2, 3, 4, 5], 'lat': 0.0, 'lon': 0.0, 'wiki_link': 'https://en.wikipedia.org/wiki/Asgard'}
```



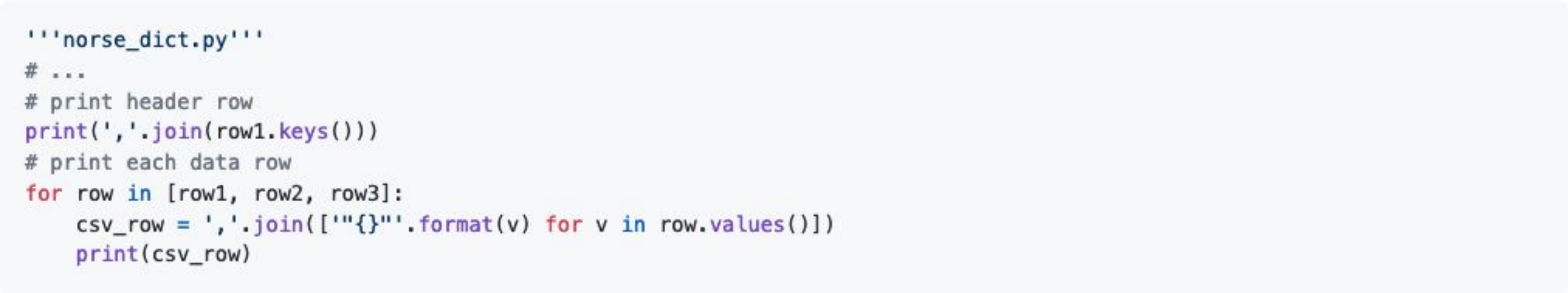
```
>>> from norse_dict import row1
>>> row1['poi_details']
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 'poi_details'
```



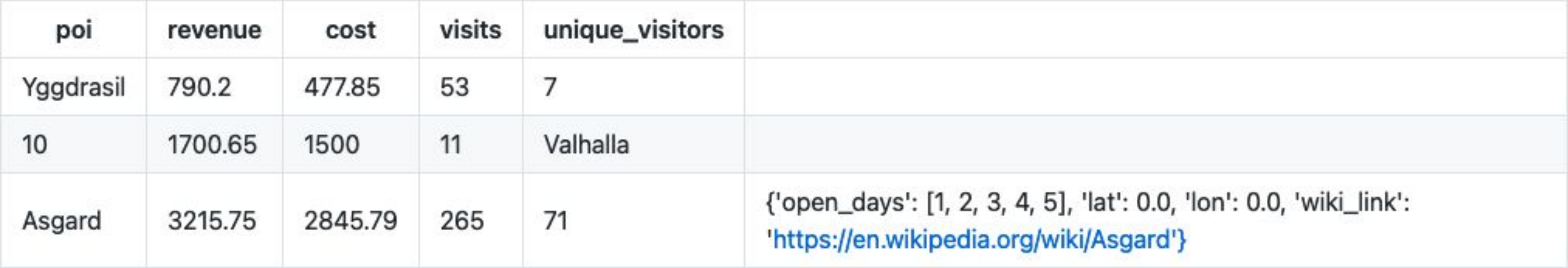
```
>>> from norse_dict import row1
>>> row1.get('poi_details')  # None
>>> row1.get('poi_details', {})  # if None, default {}
{}
```



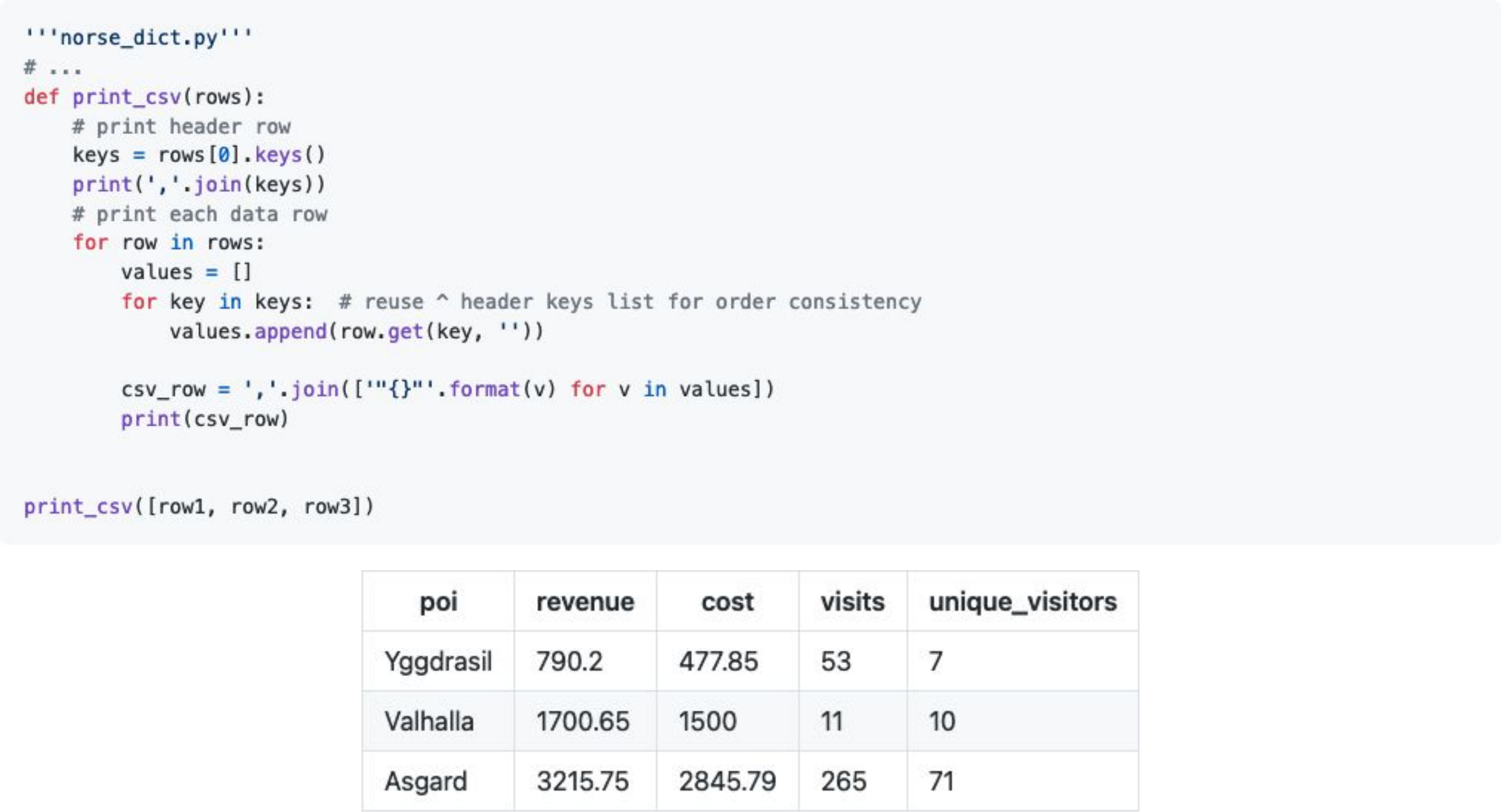
```
>>> from norse_dict import row3
>>> row1 = ['Yggdrasil', 790.2, 477.85, 53, 7]
>>> row2 = ['Valhalla', 1700.65, 1500, 11, 10]
>>> row3_list = list(row3.values())
>>> row3_list
['Asgard', 3215.75, 2845.79, 265, 71, {'open_days': [1, 2, 3, 4, 5], 'lat': 0.0, 'lon': 0.0, 'wiki_link': 'https://en.wiki
>>> row3[5]  # positional index of poi_details
{'open_days': [1, 2, 3, 4, 5], 'lat': 0.0, 'lon': 0.0, 'wiki_link': 'https://en.wikipedia.org/wiki/Asgard'}
>>> row1[5]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
IndexError: list index out of range
```



```
'''norse_dict.py'''
# ...
# print header row
print(','.join(row1.keys()))
# print each data row
for row in [row1, row2, row3]:
    csv_row = ','.join(['"{}"'.format(v) for v in row.values()])
    print(csv_row)
```



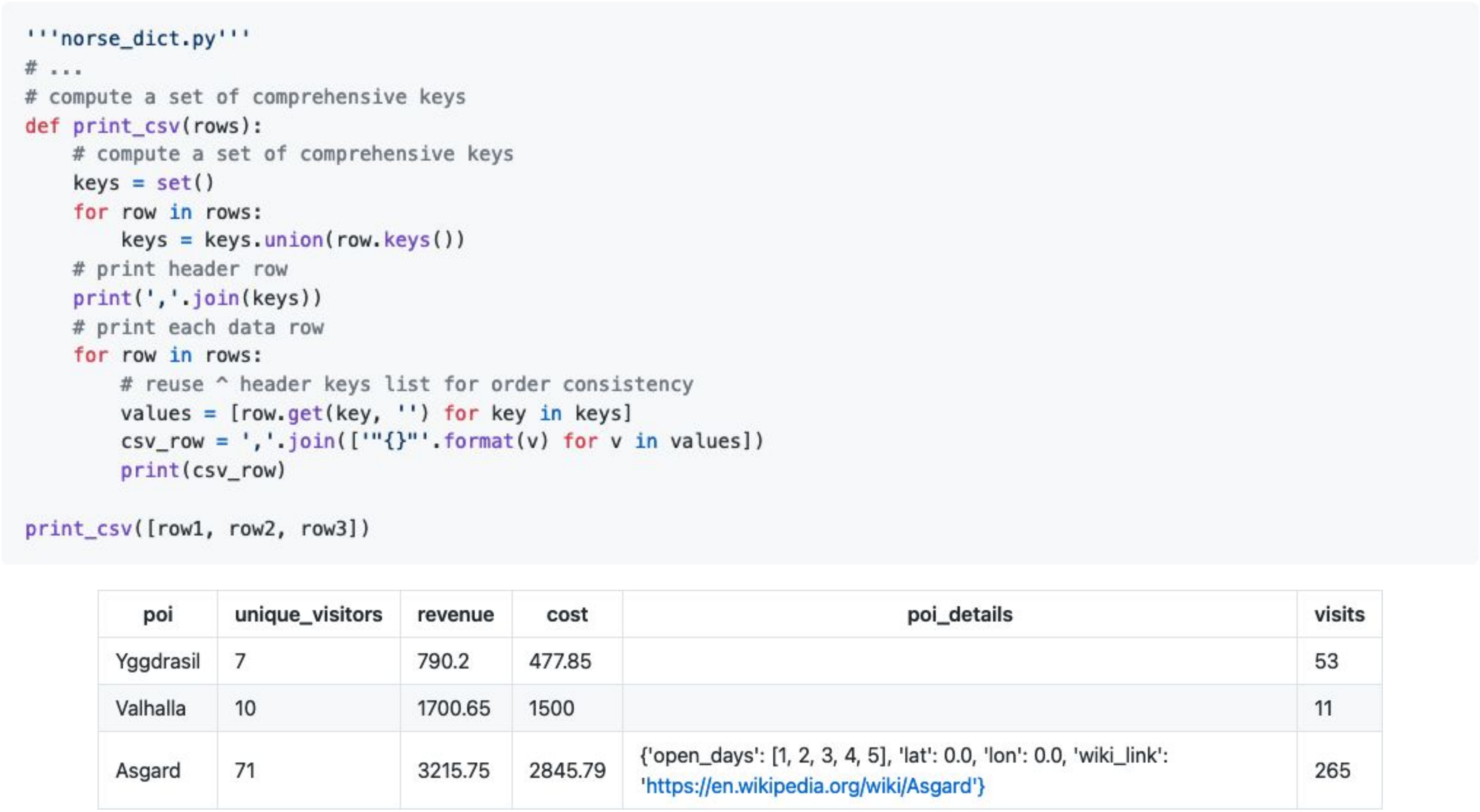
poi	revenue	cost	visits	unique_visitors	
Yggdrasil	790.2	477.85	53	7	
10	1700.65	1500	11	Valhalla	
Asgard	3215.75	2845.79	265	71	{'open_days': [1, 2, 3, 4, 5], 'lat': 0.0, 'lon': 0.0, 'wiki_link': 'https://en.wikipedia.org/wiki/Asgard'}



```
'''norse_dict.py'''
# ...
def print_csv(rows):
    # print header row
    keys = rows[0].keys()
    print(','.join(keys))
    # print each data row
    for row in rows:
        values = []
        for key in keys:  # reuse ^ header keys list for order consistency
            values.append(row.get(key, ''))

        csv_row = ','.join(['"{}"'.format(v) for v in values])
        print(csv_row)


print_csv([row1, row2, row3])
```

poi	revenue	cost	visits	unique_visitors
Yggdrasil	790.2	477.85	53	7
Valhalla	1700.65	1500	11	10
Asgard	3215.75	2845.79	265	71

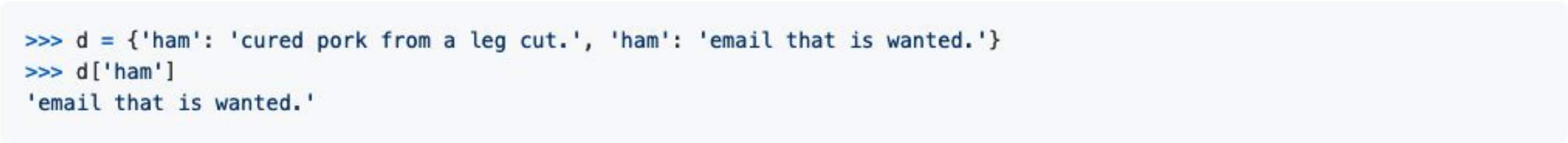
Sets



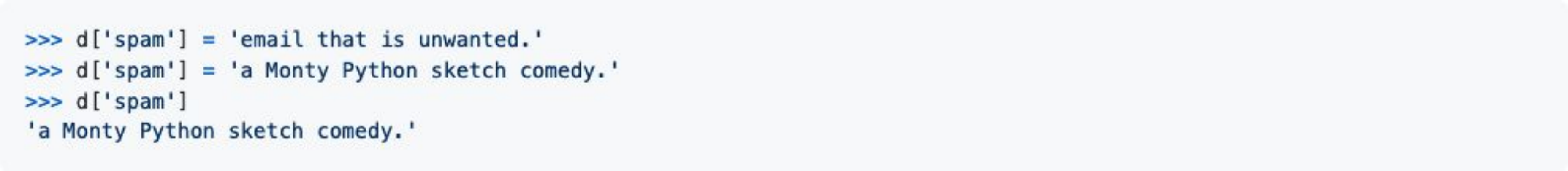
```
'''norse_dict.py'''
# ...
# compute a set of comprehensive keys
def print_csv(rows):
    # compute a set of comprehensive keys
    keys = set()
    for row in rows:
        keys = keys.union(row.keys())
    # print header row
    print(','.join(keys))
    # print each data row
    for row in rows:
        # reuse ^ header keys list for order consistency
        values = [row.get(key, '') for key in keys]
        csv_row = ','.join(['"{}"'.format(v) for v in values])
        print(csv_row)

print_csv([row1, row2, row3])
```

poi	unique_visitors	revenue	cost	poi_details	visits
Yggdrasil	7	790.2	477.85		53
Valhalla	10	1700.65	1500		11
Asgard	71	3215.75	2845.79	{'open_days': [1, 2, 3, 4, 5], 'lat': 0.0, 'lon': 0.0, 'wiki_link': 'https://en.wikipedia.org/wiki/Asgard'}	265

Dictionary Keys are a Set

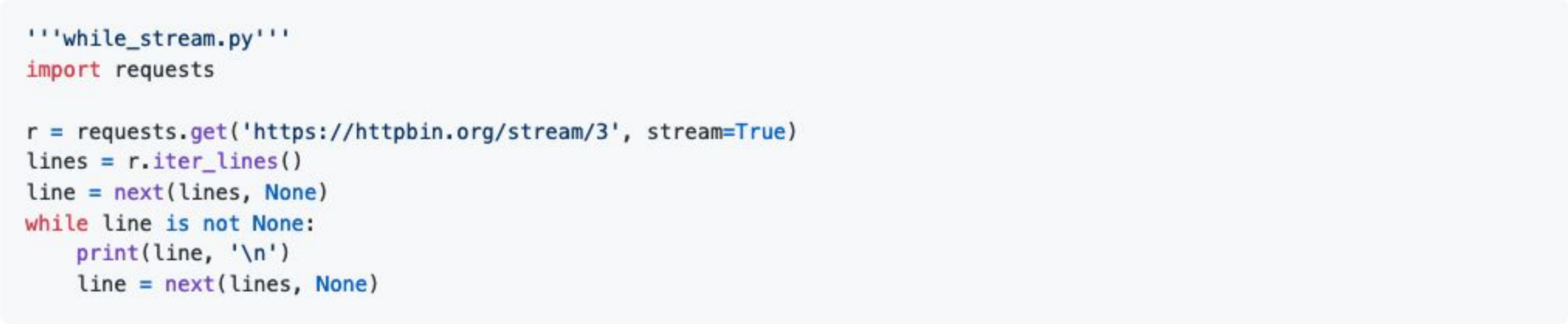


```
>>> d = {'ham': 'cured pork from a leg cut.', 'ham': 'email that is wanted.'}
>>> d['ham']
'email that is wanted.'
```

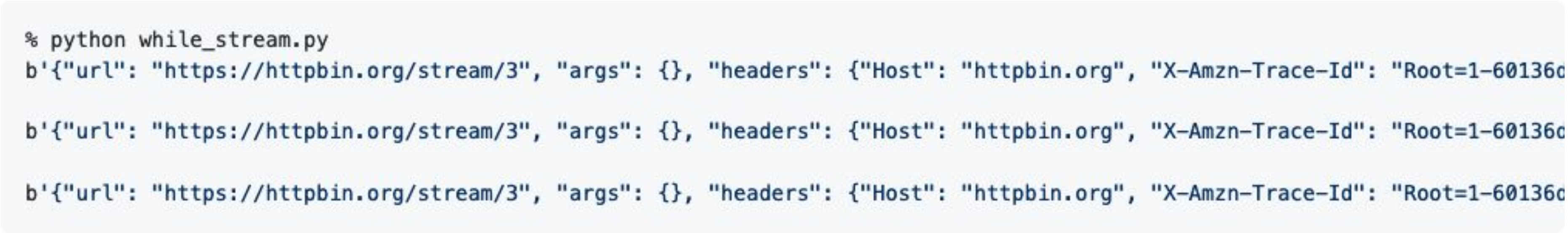


```
>>> d['spam'] = 'email that is unwanted.'
>>> d['spam'] = 'a Monty Python sketch comedy.'
>>> d['spam']
'a Monty Python sketch comedy.'
```

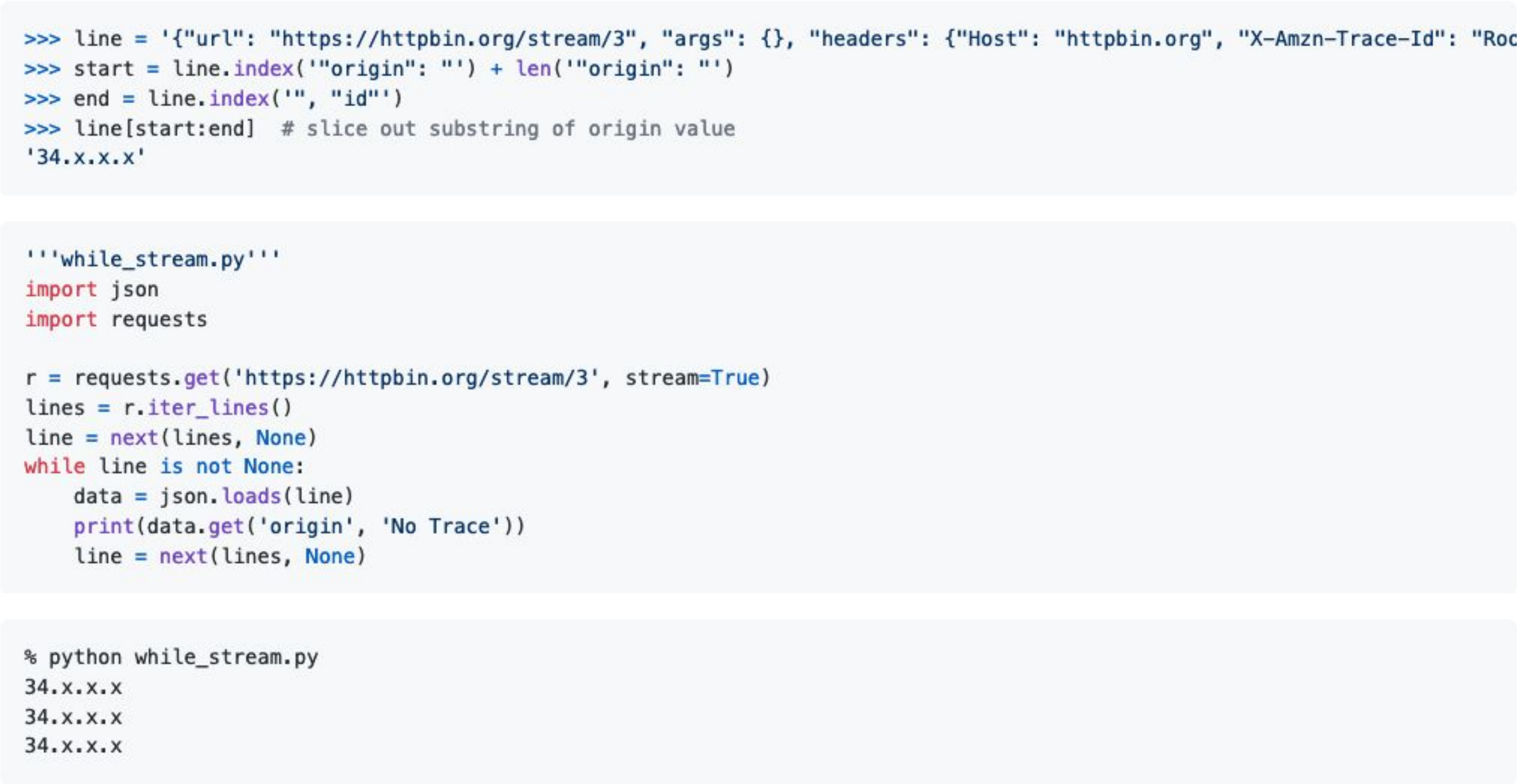
While Loop



```python
'''while_stream.py'''
import requests

r = requests.get('https://httpbin.org/stream/3', stream=True)
lines = r.iter_lines()
line = next(lines, None)
while line is not None:
    print(line, '\n')
    line = next(lines, None)
```



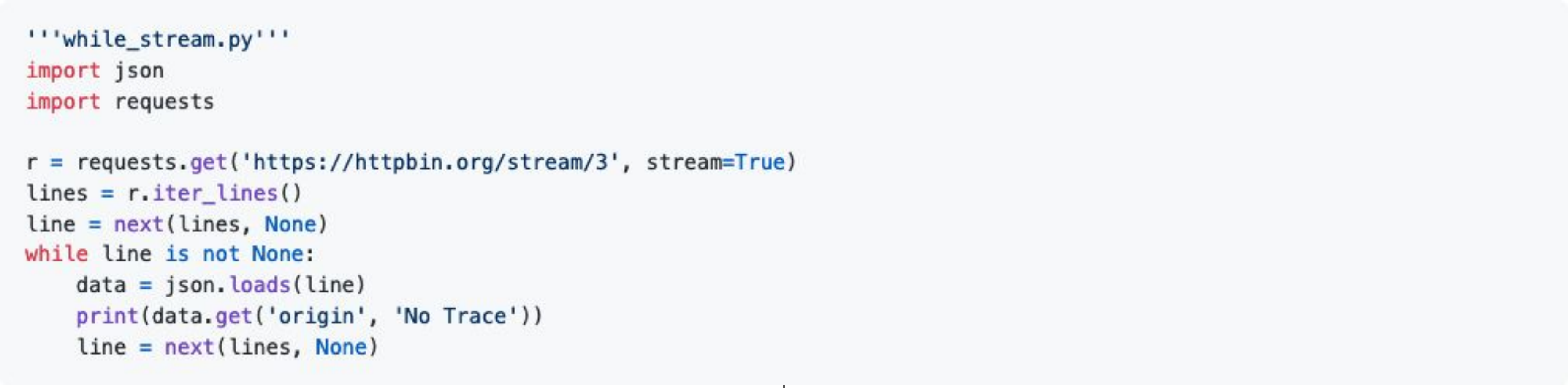
```
% python while_stream.py
b'{"url": "https://httpbin.org/stream/3", "args": {}, "headers": {"Host": "httpbin.org", "X-Amzn-Trace-Id": "Root=1-60136d

b'{"url": "https://httpbin.org/stream/3", "args": {}, "headers": {"Host": "httpbin.org", "X-Amzn-Trace-Id": "Root=1-60136d

b'{"url": "https://httpbin.org/stream/3", "args": {}, "headers": {"Host": "httpbin.org", "X-Amzn-Trace-Id": "Root=1-60136d
```



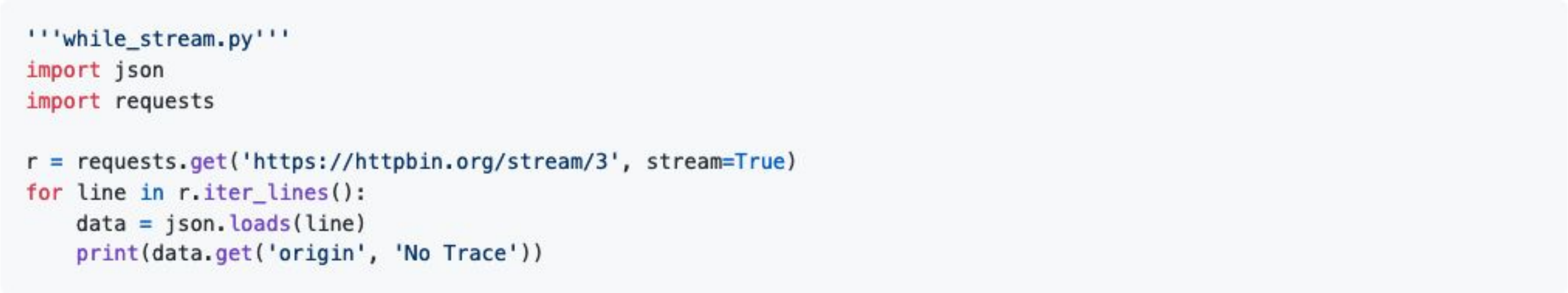
```
>>> line = '{"url": "https://httpbin.org/stream/3", "args": {}, "headers": {"Host": "httpbin.org", "X-Amzn-Trace-Id": "Roc
>>> start = line.index('"origin": "') + len('"origin": "')
>>> end = line.index('", "id"')
>>> line[start:end]  # slice out substring of origin value
'34.x.x.x'
```

```
'''while_stream.py'''
import json
import requests

r = requests.get('https://httpbin.org/stream/3', stream=True)
lines = r.iter_lines()
line = next(lines, None)
while line is not None:
    data = json.loads(line)
    print(data.get('origin', 'No Trace'))
    line = next(lines, None)
```

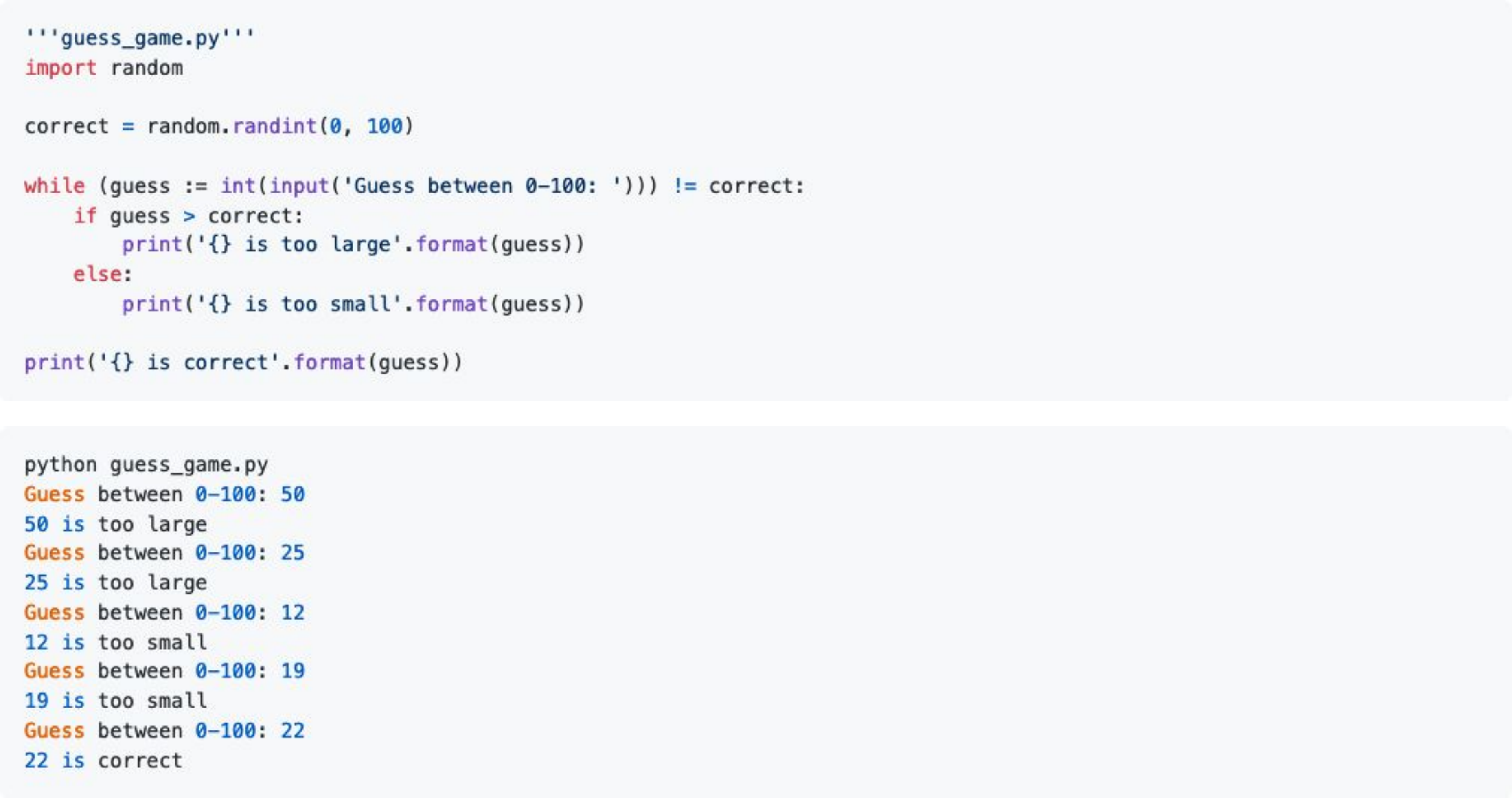
```
% python while_stream.py
34.x.x.x
34.x.x.x
34.x.x.x
```



```python
'''while_stream.py'''
import json
import requests

r = requests.get('https://httpbin.org/stream/3', stream=True)
lines = r.iter_lines()
line = next(lines, None)
while line is not None:
    data = json.loads(line)
    print(data.get('origin', 'No Trace'))
    line = next(lines, None)
```



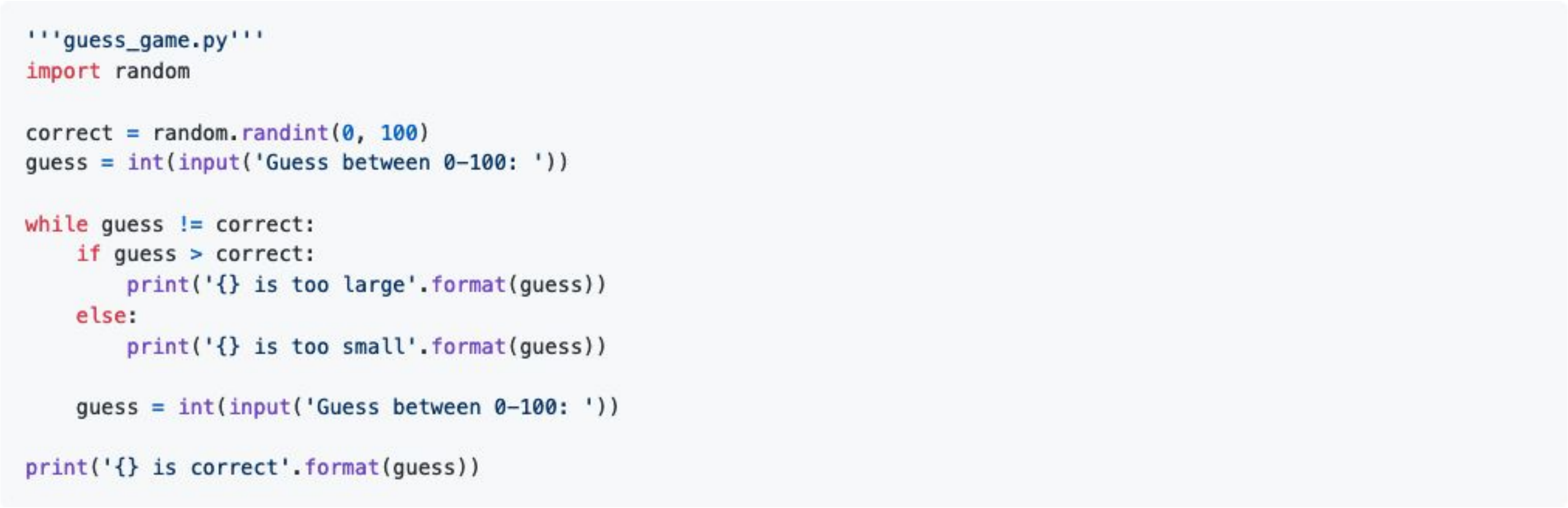
```python
'''while_stream.py'''
import json
import requests

r = requests.get('https://httpbin.org/stream/3', stream=True)
for line in r.iter_lines():
    data = json.loads(line)
    print(data.get('origin', 'No Trace'))
```



```python
'''guess_game.py'''
import random

correct = random.randint(0, 100)

while (guess := int(input('Guess between 0-100: '))) != correct:
    if guess > correct:
        print('{} is too large'.format(guess))
    else:
        print('{} is too small'.format(guess))

print('{} is correct'.format(guess))
```

```
python guess_game.py
Guess between 0-100: 50
50 is too large
Guess between 0-100: 25
25 is too large
Guess between 0-100: 12
12 is too small
Guess between 0-100: 19
19 is too small
Guess between 0-100: 22
22 is correct
```

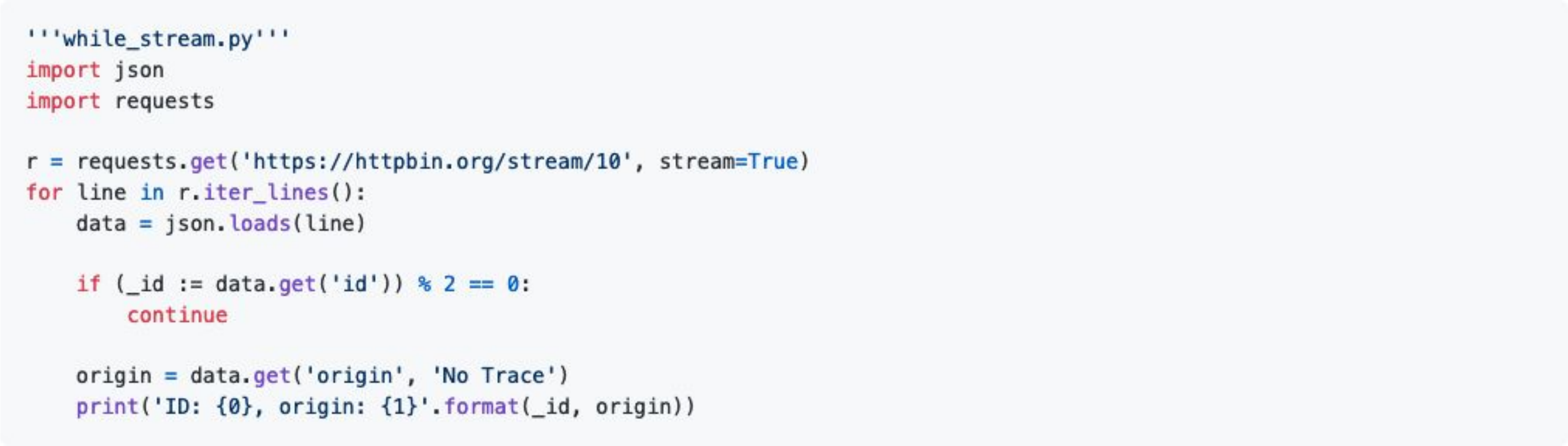
Before Python 3.8



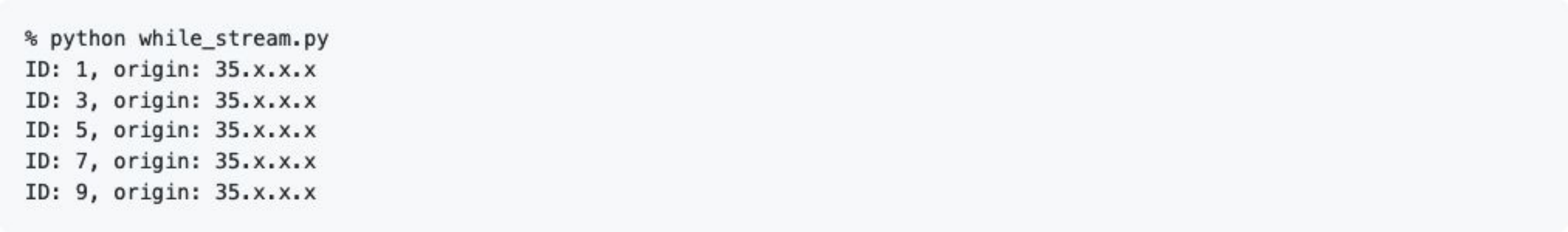
```
'''guess_game.py'''
import random

correct = random.randint(0, 100)
guess = int(input('Guess between 0-100: '))

while guess != correct:
    if guess > correct:
        print('{} is too large'.format(guess))
    else:
        print('{} is too small'.format(guess))

    guess = int(input('Guess between 0-100: '))

print('{} is correct'.format(guess))
```

Loop Controls

Continue

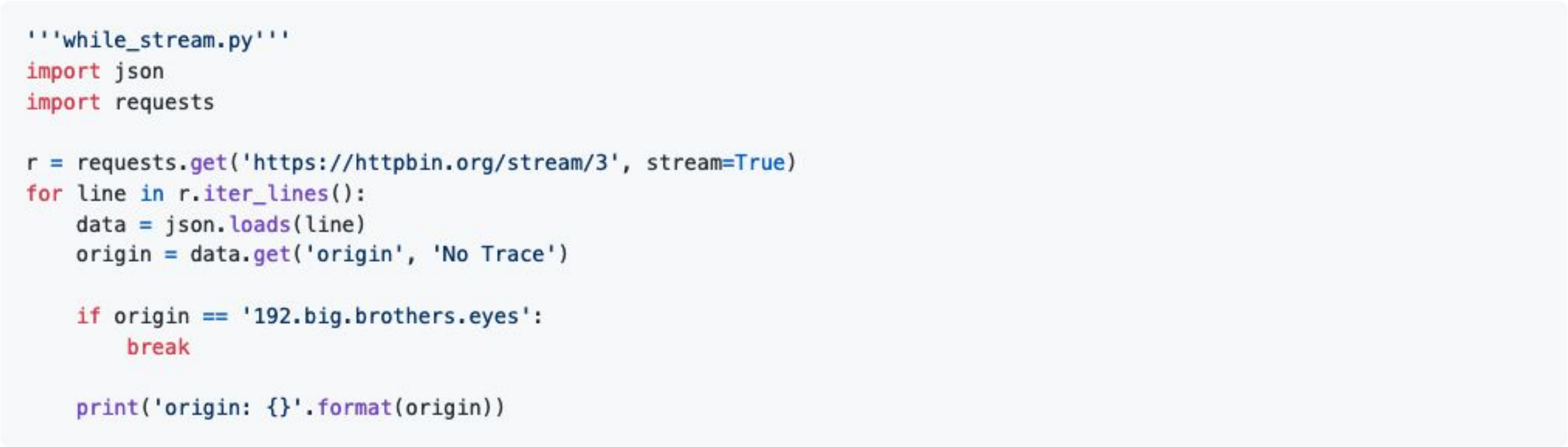


```python
'''while_stream.py'''
import json
import requests

r = requests.get('https://httpbin.org/stream/10', stream=True)
for line in r.iter_lines():
    data = json.loads(line)

    if (_id := data.get('id')) % 2 == 0:
        continue

    origin = data.get('origin', 'No Trace')
    print('ID: {0}, origin: {1}'.format(_id, origin))
```



```
% python while_stream.py
ID: 1, origin: 35.x.x.x
ID: 3, origin: 35.x.x.x
ID: 5, origin: 35.x.x.x
ID: 7, origin: 35.x.x.x
ID: 9, origin: 35.x.x.x
```

Break

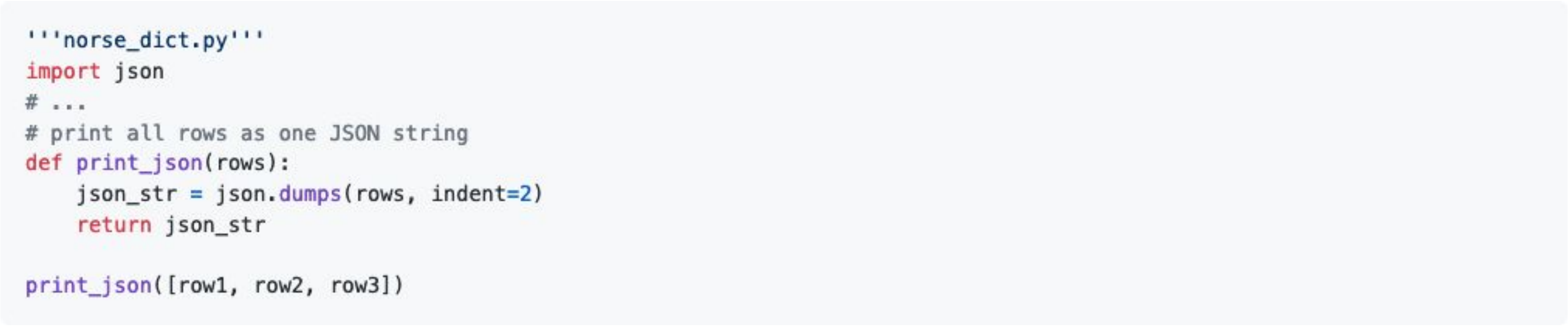


```
'''while_stream.py'''
import json
import requests

r = requests.get('https://httpbin.org/stream/3', stream=True)
for line in r.iter_lines():
    data = json.loads(line)
    origin = data.get('origin', 'No Trace')

    if origin == '192.big.brothers.eyes':
        break

    print('origin: {}'.format(origin))
```

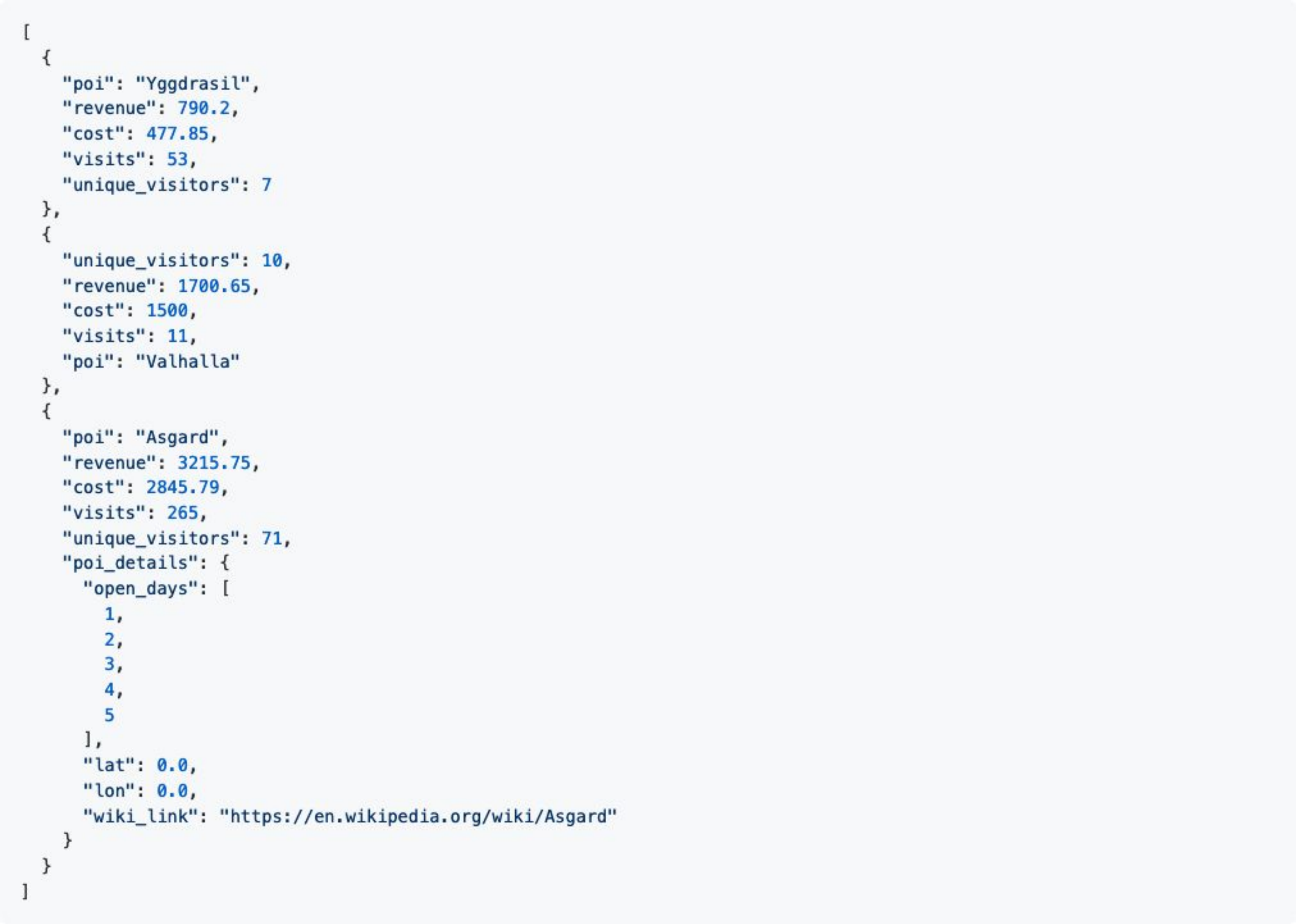
Questions?

Extra

JSON Format

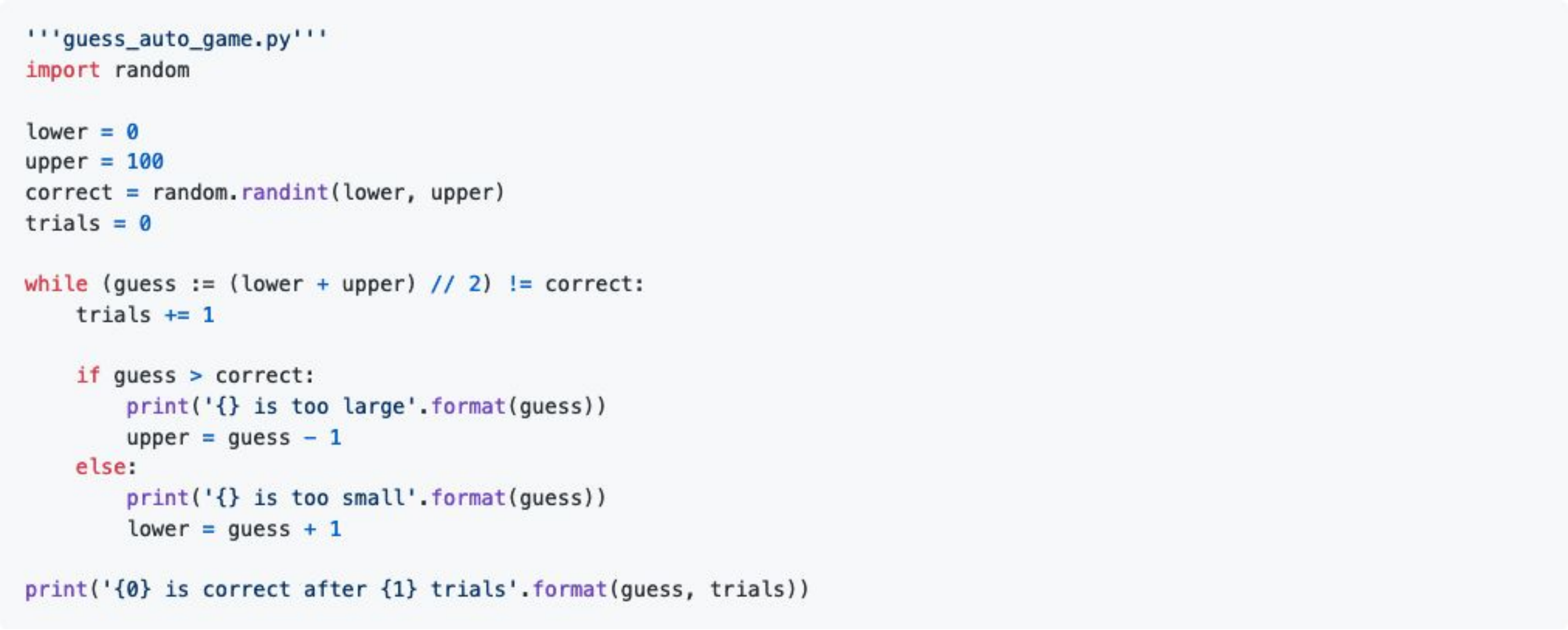


```python
'''norse_dict.py'''
import json
# ...
# print all rows as one JSON string
def print_json(rows):
    json_str = json.dumps(rows, indent=2)
    return json_str

print_json([row1, row2, row3])
```



```
[
  {
    "poi": "Yggdrasil",
    "revenue": 790.2,
    "cost": 477.85,
    "visits": 53,
    "unique_visitors": 7
  },
  {
    "unique_visitors": 10,
    "revenue": 1700.65,
    "cost": 1500,
    "visits": 11,
    "poi": "Valhalla"
  },
  {
    "poi": "Asgard",
    "revenue": 3215.75,
    "cost": 2845.79,
    "visits": 265,
    "unique_visitors": 71,
    "poi_details": {
      "open_days": [
        1,
        2,
        3,
        4,
        5
      ],
      "lat": 0.0,
      "lon": 0.0,
      "wiki_link": "https://en.wikipedia.org/wiki/Asgard"
    }
  }
]
```

Guess AUTO Game



```python
'''guess_auto_game.py'''
import random

lower = 0
upper = 100
correct = random.randint(lower, upper)
trials = 0

while (guess := (lower + upper) // 2) != correct:
    trials += 1

    if guess > correct:
        print('{} is too large'.format(guess))
        upper = guess - 1
    else:
        print('{} is too small'.format(guess))
        lower = guess + 1

print('{0} is correct after {1} trials'.format(guess, trials))
```