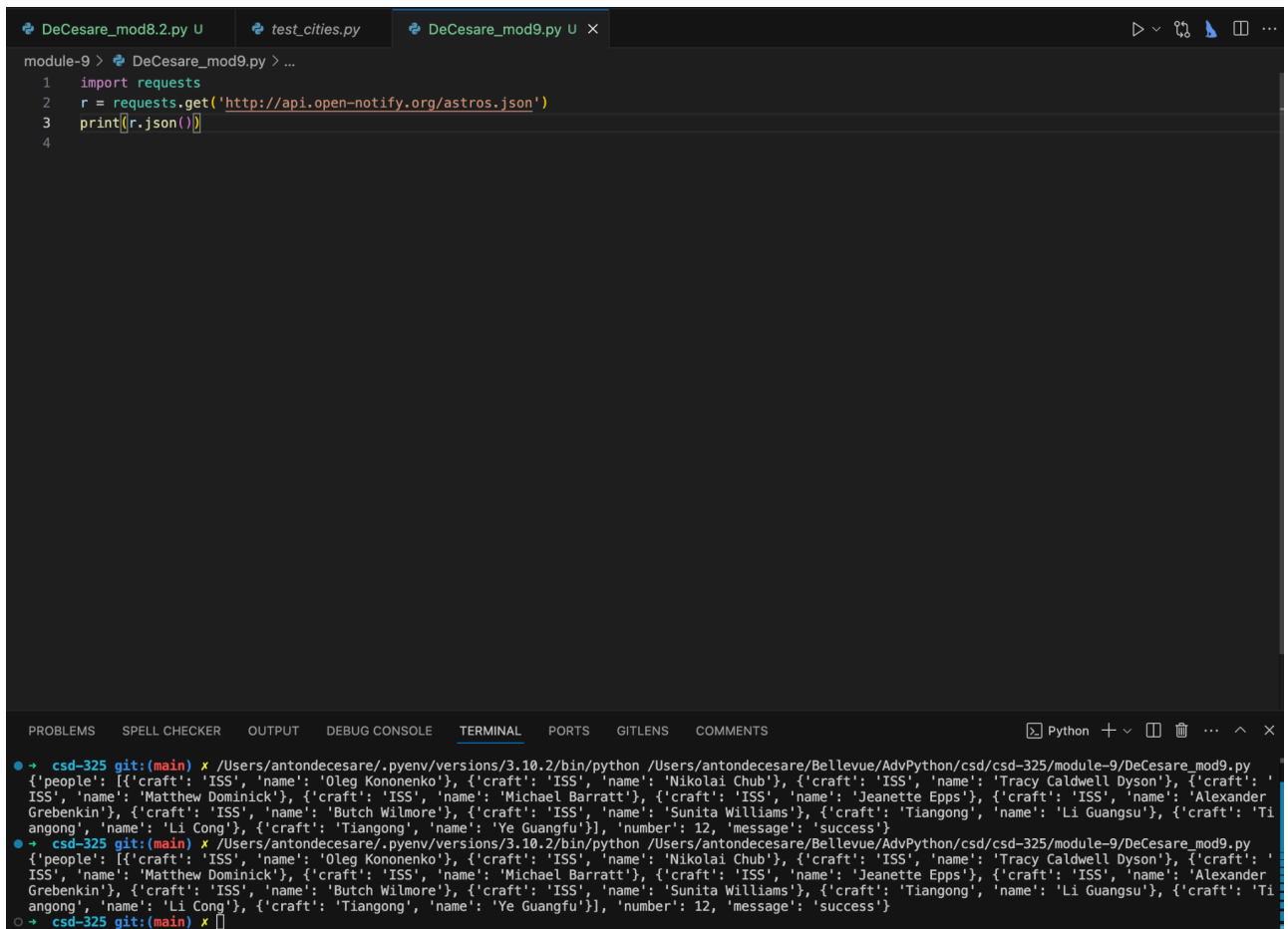


Anton DeCesare 325 Mod 9.1



The image shows a Visual Studio Code editor window with three tabs: `DeCesare_mod8.2.py U`, `test_cities.py`, and `DeCesare_mod9.py U X`. The active tab is `DeCesare_mod9.py`, which contains the following Python code:

```
1 import requests
2 r = requests.get('http://api.open-notify.org/astros.json')
3 print(r.json())
4
```

Below the editor, the **TERMINAL** panel is open, showing the output of the script. The first command executed is:

```
csd-325 git:(main) x /Users/antondecesare/.pyenv/versions/3.10.2/bin/python /Users/antondecesare/Bellevue/AdvPython/csd/csd-325/module-9/DeCesare_mod9.py
```

The output is a JSON array of objects, each representing an astronaut on the ISS:

```
{
  "people": [
    {
      "craft": "ISS",
      "name": "Oleg Kononenko"
    },
    {
      "craft": "ISS",
      "name": "Nikolai Chub"
    },
    {
      "craft": "ISS",
      "name": "Tracy Caldwell Dyson"
    },
    {
      "craft": "ISS",
      "name": "Matthew Dominick"
    },
    {
      "craft": "ISS",
      "name": "Michael Barratt"
    },
    {
      "craft": "ISS",
      "name": "Jeanette Epps"
    },
    {
      "craft": "ISS",
      "name": "Alexander Grebenkin"
    },
    {
      "craft": "ISS",
      "name": "Butch Wilmore"
    },
    {
      "craft": "ISS",
      "name": "Sunita Williams"
    },
    {
      "craft": "Tiangong",
      "name": "Li Guangsu"
    },
    {
      "craft": "Tiangong",
      "name": "Li Cong"
    },
    {
      "craft": "Tiangong",
      "name": "Ye Guangfu"
    }
  ]
}
```

The second command executed is:

```
csd-325 git:(main) x /Users/antondecesare/.pyenv/versions/3.10.2/bin/python /Users/antondecesare/Bellevue/AdvPython/csd/csd-325/module-9/DeCesare_mod9.py
```

The output is the same JSON array as above.

The third command executed is:

```
csd-325 git:(main) x
```

The output is an empty line.

```
DeCesare_mod8.2.py U  test_cities.py  DeCesare_mod9.py U X
module-9 > DeCesare_mod9.py > ...
4  def astros(obj):
5      obj = json.dumps(obj, sort_keys=True, indent=4)
6      print(obj)
7
8
9  Tabnine | Edit | Test | Explain | Document | Ask
10 def main():
11     r = requests.get('http://api.open-notify.org/astros.json')
12     obj = r.json()
13     print(astros(obj))

PROBLEMS  SPELL CHECKER  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

{
  "message": "success",
  "number": 12,
  "people": [
    {
      "craft": "ISS",
      "name": "Oleg Kononenko"
    },
    {
      "craft": "ISS",
      "name": "Nikolai Chub"
    },
    {
      "craft": "ISS",
      "name": "Tracy Caldwell Dyson"
    },
    {
      "craft": "ISS",
      "name": "Matthew Dominick"
    },
    {
      "craft": "ISS",
      "name": "Michael Barratt"
    },
    {
      "craft": "ISS",
      "name": "Jeanette Epps"
    },
    {
      "craft": "ISS",
      "name": "Alexander Grebenkin"
    },
    {
      "craft": "ISS",
      "name": "Butch Wilmore"
    },
    {
```

```
DeCesare_mod8.2.py U test_cities.py DeCesare_mod9.2.1.py U DeCesare_mod9.2.2.py U X
module-9 > DeCesare_mod9.2.2.py > ...

2
3 BASE_URL = "https://jsonplaceholder.typicode.com"
4
5 response = requests.get(f"{BASE_URL}/posts")
6
7 if response.status_code == 200: # Make sure the request was successful
8     posts = response.json() # Parse the JSON response
9     for post in posts[:5]: # Loop through and display the first 5 posts
10         print(f"Post ID: {post['id']}")
11         print(f"Title: {post['title']}")
12         print(f"Body: {post['body']}\n")
13 else:
14     print(f"Failed to retrieve posts. Status code: {response.status_code}") # Get status code from response for error handling purposes

PROBLEMS SPELL CHECKER OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS Python + - X
• - csd-325 /Users/antondecasare/.pyenv/versions/3.10.2/bin/python /Users/antondecasare/Bellevue/AdvPython/csd/csd-325/module-9/DeCesare_mod9.2.2.py
Post ID: 1
Title: sunt aut facere repellat provident occaecati excepturi optio reprehenderit
Body: quia et suscipit
suscipit recusandae consequuntur expedita et cum
reprehenderit molestiae ut ut quas totam
nostrum rerum est autem sunt rem eveniet architecto

Post ID: 2
Title: qui est esse
Body: est rerum tempore vitae
sequi sint nihil reprehenderit dolor beatae ea dolores neque
fugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis
qui aperiam non debitis possimus qui neque nisi nulla

Post ID: 3
Title: ea molestias quasi exercitationem repellat qui ipsa sit aut
Body: et iusto sed quo iure
voluptatem occaecati omnis eligendi aut ad
voluptatem doloribus vel accusantium quis pariatur
molestiae porro eius odio et labore et velit aut

Post ID: 4
Title: eum et est occaecati
Body: ullam et saepe reiciendis voluptatem adipisci
sit amet autem assumenda provident rerum culpa
quis hic commodi nesciunt rem tenetur doloremque ipsam iure
quis sunt voluptatem rerum illo velit

Post ID: 5
Title: nesciunt quas odio
Body: repudiandae veniam quaerat sunt sed
alias aut fugiat sit autem sed est
voluptatem omnis possimus esse voluptatibus quis
est aut tenetur dolor neque
```

```
DeCesare_mod8.2.py U    test_cities.py    DeCesare_mod9.2.1.py U    DeCesare_mod9.2.2.py U
Edit with Tabnine
module-9 > DeCesare_mod9.2.2.py > main

Tabnine | Edit | Test | Explain | Document | Ask
18 def responses(obj):
19     formatted_obj = json.dumps(obj, sort_keys=True, indent=4) # Formatting
20     return formatted_obj # Return the formatted JSON string.
21
Tabnine | Edit | Test | Explain | Document | Ask
22 def main():
23     url = 'https://jsonplaceholder.typicode.com/posts' # Use a valid endpoint that returns JSON data
24     r = requests.get(url)
25
26     if r.status_code == 200: # Make sure the request is successful
27         obj = r.json()
28         print(responses(obj)) # Call the function and print the formatted JSON.

PROBLEMS  SPELL CHECKER  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  GITLENS  COMMENTS
Python + - [] ... ^ x
csd-325 git:(main) x /Users/antondecesare/.pyenv/versions/3.10.2/bin/python /Users/antondecesare/Bellevue/AdvPython/csd/csd-325/module-9/DeCesare_mod9.2.2.py
{
  "body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto",
  "id": 1,
  "title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",
  "userId": 1
},
{
  "body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla",
  "id": 2,
  "title": "qui est esse",
  "userId": 1
},
{
  "body": "et iusto sed quo iure\nvoluptatem occaecati omnis eligendi aut ad\nvoluptatem doloribus vel accusantium quis pariatur\nmolestiae porro eius odio et labore et velit aut",
  "id": 3,
  "title": "ea molestias quasi exercitationem repellat qui ipsa sit aut",
  "userId": 1
},
{
  "body": "ullam et saepe reiciendis voluptatem adipisci\nsit amet autem assumenda provident rerum culpa\nquis hic commodi nesciunt rem tenetur doloremque ipsam iure\nquis sunt voluptatem rerum illo velit",
  "id": 4,
  "title": "eum et est occaecati",
  "userId": 1
},
{
  "body": "repudiandae veniam quaerat sunt sed\nnihil aut fugiat sit autem sed est\nvoluptatem omnis possimus esse voluptatibus quis\nest aut tenetur dol
```

