

Asynchronous 7

<https://github.com/hanattaw/class7-asyncio.git>

Async download

1-1-aioimage > 1-1-async-download-answer.py > ...

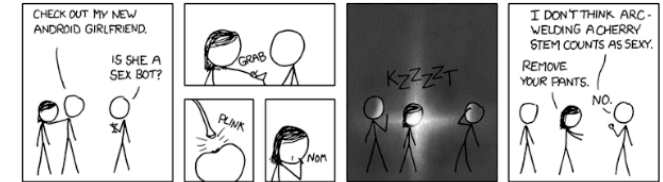
```
1  import io
2  import time
3  import asyncio
4  import aiohttp, aiofiles
5  from PIL import Image
6
7  unsplash_search_url = "https://source.unsplash.com/random/300x300"
8
9  async def download_image_async(session: aiohttp.ClientSession, num: int = 1):
10     print(f"{time.ctime()} - Start download image {num}")
11     async with session.get(unsplash_search_url) as response:
12         if response.status == 200:
13             image_buffer = await response.read()
14             img = Image.open(io.BytesIO(image_buffer))
15             print(f"{time.ctime()} - Image {num} shape: {img.size}")
16
17     async with aiofiles.open(f'./filename{num}.jpg', "wb") as new_file:
18         print(f"{time.ctime()} - Writing to 'filename{num}.jpg'...")
19         await new_file.write(image_buffer)
20
21  async def download_images_async(n: int = 10):
22     async with aiohttp.ClientSession() as session:
23         tasks = [download_image_async(session, i + 1) for i in range(n)]
24         _ = await asyncio.gather(*tasks)
25     return
26
27  if __name__ == "__main__":
28     tick = time.perf_counter()
29     asyncio.run(download_images_async())
30     tock = time.perf_counter()
31     print(f"{time.ctime()} - elapsed: {tock-tick:.2f} seconds")
```

Flask 1 comic

Time taken: 0.5905414580000006

1-2-comic-flask > 1-xkcd_commic_flask_answer.py > ...

```
1  import time
2  from random import randint
3  import requests as requests
4  from flask import Flask, render_template
5
6  app = Flask(__name__)
7
8  def get_xkcd_image():
9      comicid = randint(0, 1000)
10     response = requests.get(f'https://xkcd.com/{comicid}/info.0.json')
11     return response.json()['img']
12
13 @app.get('/comic')
14 def hello():
15     start = time.perf_counter()
16     url = get_xkcd_image()
17     end = time.perf_counter()
18     return render_template('index.html', end=end, start=start, url=url)
19
20 if __name__ == '__main__':
21     app.run(debug=True, port=5555)
```



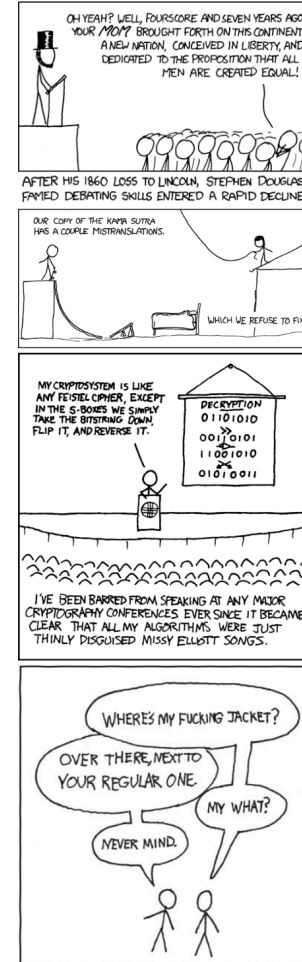
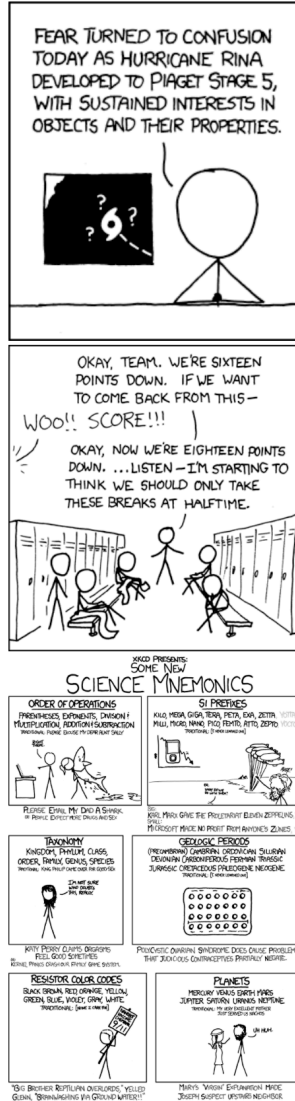
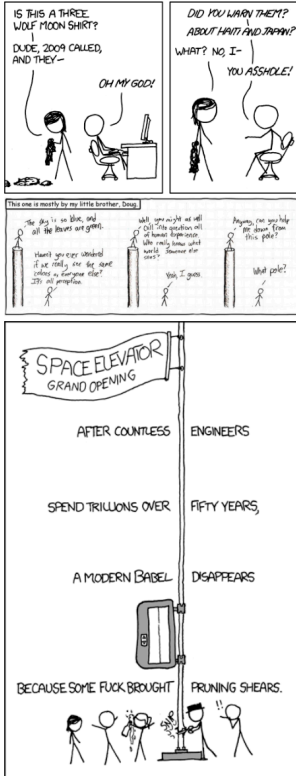
Flask 10 comics

1-2-comic-flask > 2-multiple_xkcd_comic_flask_answer.py > ...

```
1  import time
2  from random import randint
3  import requests as requests
4  from flask import Flask, render_template
5
6  app = Flask(__name__)
7
8  def get_xkcd_image():
9      comid = randint(0, 1000)
10     response = requests.get(f'https://xkcd.com/{comid}/info.0.json')
11     return response.json()['img']
12
13 def get_multiple_images(number):
14     return [get_xkcd_image() for _ in range(number)]
15
16 @app.get('/comic')
17 def hello():
18     start = time.perf_counter()
19     urls = get_multiple_images(10)
20     end = time.perf_counter()
21     return render_template('index.html', end=end, start=start, urls=urls)
22
23 if __name__ == '__main__':
24     app.run(debug=True, port=5555)
```

Flask 10 comics

Time taken: 4.04618675



OH YEAH? WELL, FOURSORE AND SEVEN YEARS AGO
YOUR MOM BROUGHT FORTH ON THIS CONTINENT
A NEW NATION, CONCEIVED IN LIBERTY, AND
DEDICATED TO THE PROPOSITION THAT ALL
MEN ARE CREATED EQUAL!

AFTER HIS 1860 LOSS TO LINCOLN, STEPHEN DOUGLAS' FAMED DEBATING SKILLS ENTERED A RAPID DECLINE

OUR COPY OF THE KAMA SUTRA
HAS A COUPLE MISTRANSLATIONS.

WHICH WE REFUSE TO FIX

MY CRYPTOSYSTEM IS LIKE ANY FEISTEL CIPHER, EXCEPT IN THE S-BLOCKS WE SIMPLY TAKE THE BITSTRING DOWN, FLIP IT, AND REVERSE IT.

[illegible]

I'VE BEEN BARRED FROM SPEAKING AT ANY MAJOR CRYPTOGRAPHY CONFERENCES EVER SINCE IT BECAME CLEAR THAT ALL MY ALGORITHMS WERE JUST THINLY DISGUISED MISSY ELLIOTT SONGS.

WHERE'S MY FUCKING JACKET?

OVER THERE, NEXT TO
YOUR REGULAR ONE.

MY WHAT?

NEVER MIND.



\wedge

Flask async 10 comics

1-2-comic-flask > 3-async_flask_api_with_httpx.py > ...

You, 6 minutes ago | 1 author (You)

```
1  # https://flask.palletsprojects.com/en/2.3.x/
2  # https://www.python-httpx.org/async/
3  # https://flask.palletsprojects.com/en/2.3.x/quickstart/#rendering-templates
4
5  ∨ import asyncio
6  import time
7  from random import randint
8  import httpx
9  from flask import Flask, render_template
10
11  app = Flask(__name__)
12
13  # function converted to coroutine
14  async def get_xkcd_image(session): # dont wait for the response of API
15
16
17  # function converted to coroutine
18  ∨ async def get_multiple_images(number):
19  ∨     async with
20
21  @app.get('/comic')
22  async def hello():
23
24
25  ∨ if __name__ == '__main__':
26  ∨     ⚡ app.run(debug=True, port=5555)
```

You, 6 minutes ago • add assignment

Flask async 10 comics

Time taken: 1.347276333

