

'fontawesome5' 1 और 4-दोनों मॉडल हैं, लेकिन उनकी क्षमताओं में कुछ अंतर

मैंने हाल ही में fontawesome5 पैकेज के साथ एक समस्या का सामना किया। 4-द्वारा प्रदान किया गया समाधान (fontawesome 2021 और tlmgr install fontawesome5 का उपयोग करके) काम नहीं आया, लेकिन 4 ने एक बेहतर तरीका सुझाया (fontawesome 2023 का उपयोग करके और tlmgr का उपयोग करके fontawesome5 इंस्टॉल करने के लिए), जिसे मैं अभी आजमा रहा हूँ। इससे समस्या पूरी तरह से ठीक नहीं हुई, लेकिन fontawesome 2021 के बजाय 2023 पर स्विच करने से काफी सुधार हुआ है।

4-का जवाब

आप जो त्रुटि देख रहे हैं, वह tlmgr (fontawesome) के यूजर मोड में ठीक से इनिशियलाइज़ नहीं होने से संबंधित है। विशेष रूप से, संदेश tlmgr: user mode not initialized, please read the documentation! इंगित करता है कि tlmgr यूजर एनवायरनमेंट में सही तरीके से सेट अप नहीं है।

इसे ठीक करने के लिए यहां एक विस्तृत विवरण दिया गया है:

tlmgr यूजर मोड त्रुटि को ठीक करना

1. **tlmgr को यूजर मोड में सही तरीके से सेट करें:** आपको tlmgr को यूजर मोड में चलाने के लिए स्पष्ट रूप से बताना होगा और यह सुनिश्चित करना होगा कि यह ठीक से इनिशियलाइज़ हो। अपने स्क्रिप्ट में इसे करने के लिए, निम्नलिखित चरणों का उपयोग करें:

- सुनिश्चित करें कि आप tlmgr को यूजर मोड में चला रहे हैं, सही एनवायरनमेंट वेरिएबल सेट करके।
- आप tlmgr का उपयोग करने से पहले यूजर-स्पेसिफिक fontawesome डायरेक्टरी को इनिशियलाइज़ भी कर सकते हैं।

2. **tlmgr को यूजर मोड में इनिशियलाइज़ करें:** tlmgr कमांड्स चलाने से पहले, यह सुनिश्चित करें कि tlmgr निम्नलिखित कमांड के साथ इनिशियलाइज़ हो:

```
tlmgr init-usertree
```

यह ~/texmf में यूजर-स्पेसिफिक fontawesome ट्री बनाएगा। एक बार इनिशियलाइज़ हो जाने के बाद, आप tlmgr update और install कमांड्स चला सकते हैं।

3. **अपने fontawesome वर्कप्लो को संशोधित करें:** अपने fontawesome वर्कप्लो फ़ाइल को इनिशियलाइज़ेशन स्टेप शामिल करने के लिए अपडेट करें। यहां एक उदाहरण दिया गया है कि आप वर्कप्लो को कैसे संशोधित कर सकते हैं:

```
- name: Set tlmgr to user mode
run: |
    # TeX Live
    tlmgr init-usertree
    tlmgr update --self
    tlmgr install fontawesome5
```

पूर्ण उदाहरण:

यदि आप इसे एक बड़े `docker-compose` वर्कप्लो के हिस्से के रूप में चला रहे हैं, तो आप इन परिवर्तनों को संबंधित जॉब सेक्शन में शामिल कर सकते हैं:

```
jobs:
  build:
    runs-on: ubuntu-latest
    steps:
      - name: Install TeX Live (xelatex)
        run: |
            sudo apt-get update
            sudo apt-get install -y texlive-xetex
            tlmgr init-usertree # TeX Live
            tlmgr update --self # tlmgr
            tlmgr install fontawesome5 #
```

कमांड्स की व्याख्या:

- `tlmgr init-usertree`: यह यूजर-स्पेसिफिक `texmf` ट्री (आमतौर पर `~/texmf` में स्थित) को इनिशियलाइज़ करता है।
- `tlmgr update --self`: `tlmgr` पैकेज को नवीनतम संस्करण में अपडेट करता है।
- `tlmgr install fontawesome5`: `fontawesome5` पैकेज को इंस्टॉल करता है।

अतिरिक्त नोट्स:

- **अनुमतियाँ**: सुनिश्चित करें कि `docker-compose` वर्कप्लो चलाने वाले यूजर के पास `texmf` यूजर डायरेक्टरी में फ़ाइलों को इंस्टॉल और संशोधित करने की अनुमति हो।
- **एनवायरनमेंट वेरिएबल्स**: यदि आप कस्टम पाथ के साथ काम कर रहे हैं, तो आपको `TEXMFSYSVAR` या `TEXMFHOME` को अपने विशिष्ट डायरेक्टरीज़ की ओर इंगित करने के लिए समायोजित करने की आवश्यकता हो सकती है।

tlmgr को यूजर मोड में इनिशियलाइज़ करके और इसे अपने `~/.local/share/texlive/texmf` वर्कप्लो में सही तरीके से सेट अप करके, त्रुटि को हल किया जाना चाहिए। मुझे बताएं कि क्या यह काम करता है या यदि आपको कोई अन्य समस्या आती है!

❶ का जवाब

मैं समझता हूँ कि आप अपने `~/.local/share/texlive/texmf` वर्कप्लो में tlmgr (`~/.local/share/texlive/texmf`) के साथ समस्याओं का सामना कर रहे हैं, विशेष रूप से fontawesome5 पैकेज को इंस्टॉल करने की कोशिश करते समय। त्रुटि संदेश:

```
TLPDB: not a directory, not loading: /home/runner/texmf
tlmgr: user mode not initialized, please read the documentation!
Error: Process completed with exit code 2.
```

इंगित करता है कि tlmgr “यूजर मोड” में काम नहीं कर पा रहा है क्योंकि आवश्यक डायरेक्टरीज़ और कॉन्फ़िगरेशन सही तरीके से सेट अप नहीं हैं। यह समस्या अक्सर तब उत्पन्न होती है जब apt-get जैसे पैकेज मैनेजर्स द्वारा प्रदान की गई `~/.local/share/texlive/texmf` इंस्टॉलेशन का उपयोग किया जाता है, जो `~/.local/share/texlive/texmf` से पूर्ण `~/.local/share/texlive/texmf` इंस्टॉलेशन की तुलना में सीमित कार्यक्षमता प्रदान कर सकते हैं।

इसे हल करने के लिए, मैं **2023 को मैनुअल रूप से इंस्टॉल करने** की सलाह देता हूँ, जो आपके `~/.local/share/texlive/texmf` वर्कप्लो के भीतर हो। यह दृष्टिकोण सुनिश्चित करता है कि tlmgr पूर्ण कार्यक्षमता के साथ काम करे और fontawesome5 जैसे अतिरिक्त पैकेज को बिना किसी समस्या के इंस्टॉल कर सके।

चरण-दर-चरण समाधान:

1. सिस्टम `~/.local/share/texlive/texmf` इंस्टॉलेशन को हटाएं:

- `apt-get install texlive-xetex` का उपयोग करने से बचें क्योंकि यह एक आंशिक `~/.local/share/texlive/texmf` इंस्टॉलेशन प्रदान करता है जिसमें tlmgr की कार्यक्षमता सीमित होती है।

2. `~/.local/share/texlive/texmf` 2023 को मैनुअल रूप से इंस्टॉल करें:

- `~/.local/share/texlive/texmf` से सीधे `~/.local/share/texlive/texmf` को डाउनलोड और इंस्टॉल करें।

3. `~/.local/share/texlive/texmf` एनवायरनमेंट वेरिएबल को कॉन्फ़िगर करें:

- यह सुनिश्चित करें कि `~/.local/share/texlive/texmf` बाइनरीज़ `~/.local/share/texlive/texmf` द्वारा एक्सेसिबल हैं।

4. आवश्यक पैकेज इंस्टॉल करने के लिए tlmgr का उपयोग करें:

- tlmgr को अपडेट करें और fontawesome5 इंस्टॉल करें।

5. कैशिंग के साथ वर्कप्लो को ऑप्टिमाइज़ करें (वैकल्पिक):

- भविष्य के वर्कप्लो रन को तेज़ करने के लिए `~/.local/share/texlive/texmf` इंस्टॉलेशन को कैश करें।

अपडेटेड ढाँचा वर्कफ्लो:

नीचे आपके ढाँचा वर्कफ्लो का एक अपडेटेड संस्करण दिया गया है जो 2023 के मैनुअल इंस्टॉलेशन को शामिल करता है और fontawesome5 पैकेज को सही तरीके से इंस्टॉल करने के लिए tlmgr को सेट अप करता है।

```
name: Deploy Jekyll with GitHub Pages dependencies preinstalled
```

```
on:
```

```
  push:
```

```
    branches: ["master"]
```

```
  workflow_dispatch:
```

```
permissions:
```

```
  contents: read
```

```
  pages: write
```

```
  id-token: write
```

```
concurrency:
```

```
  group: "pages"
```

```
  cancel-in-progress: false
```

```
jobs:
```

```
  awesome-cv-copy:
```

```
    runs-on: ubuntu-latest
```

```
    steps:
```

```
      - name: Checkout repository
```

```
        uses: actions/checkout@v4
```

```
      - name: Set up Make
```

```
        run: sudo apt-get update && sudo apt-get install -y make
```

```
      - name: Cache TeX Live
```

```
        uses: actions/cache@v3
```

```
        with:
```

```
          path: /usr/local/texlive/2023
```

```
          key: ${{ runner.os }}-texlive-2023
```

```
          restore-keys: |
```

```

    ${{ runner.os }}-texlive-

- name: Install TeX Live 2023
  if: steps.cache-texlive.outputs.cache-hit != 'true'
  run: |
    # TeX Live
    sudo apt-get update
    sudo apt-get install -y perl wget xz-utils

    # TeX Live
    wget http://mirror.ctan.org/systems/texlive/tlnet/install-tl-unx.tar.gz
    tar -xzf install-tl-unx.tar.gz
    cd install-tl-*/

    # texlive.profile
    cat <<EOF > texlive.profile
    selected_scheme scheme-full
    TEXDIR /usr/local/texlive/2023
    TEXMFCONFIG ~/.texlive2023/texmf-config
    TEXMFHOME ~/texmf
    TEXMFLOCAL /usr/local/texlive/2023/texmf-local
    TEXMFSYSCONFIG /usr/local/texlive/2023/texmf-config
    TEXMFSYSVAR /usr/local/texlive/2023/texmf-var
    binary_x86_64-linux 1
    collection-basic 1
    collection-latex 1
    collection-fontsrecommended 1
    EOF

    #
    sudo ./install-tl --profile=texlive.profile

- name: Add TeX Live to PATH
  run: echo "/usr/local/texlive/2023/bin/x86_64-linux" >> $GITHUB_PATH

- name: Update tlmgr and Install fontawesome5

```

```

    run: |
        tlmgr update --self
        tlmgr install fontawesome5

- name: Run make awesome-cv-copy
  run: make awesome-cv-copy

build:
  runs-on: ubuntu-latest
  needs: awesome-cv-copy
  steps:
    - name: Checkout repository
      uses: actions/checkout@v4

    - name: Setup Python 3
      uses: actions/setup-python@v4
      with:
        python-version: "3.x"

    - name: Install dependencies
      run: |
        python -m pip install --upgrade pip
        pip install python-frontmatter

    - name: Generate notes links
      run: |
        python scripts/generate_notes_link.py

    - name: Setup Pages
      uses: actions/configure-pages@v5

    - name: Build with Jekyll
      uses: actions/jekyll-build-pages@v1
      with:
        source: ./
        destination: ./_site

```

```

- name: Upload artifact
  uses: actions/upload-pages-artifact@v3

deploy:
  environment:
    name: github-pages
    url: ${ steps.deployment.outputs.page_url }
  runs-on: ubuntu-latest
  needs: build
  steps:
    - name: Deploy to GitHub Pages
      id: deployment
      uses: actions/deploy-pages@v4

```

विस्तृत व्याख्या:

1. रिपॉजिटरी को चेकआउट करें:

- actions/checkout@v4 एक्शन का उपयोग करके आपके रिपॉजिटरी को क्लोन करें।

2. डिपेंडेंसी सेट अप करें:

- apt-get का उपयोग करके make इंस्टॉल करें, यह सुनिश्चित करें कि यह आपके सिस्टम को बनाने के लिए उपलब्ध है।

3. डिपेंडेंसी को कैश करें:

- actions/cache@v3 का उपयोग करके /usr/local/texlive/2023 पर स्थित डिपेंडेंसी इंस्टॉलेशन को कैश करें। यह भविष्य के वर्कफ्लो रन के लिए इंस्टॉलेशन समय को काफी कम कर देता है।

4. डिपेंडेंसी 2023 इंस्टॉल करें:

- कंडीशनल एक्ज़ीक्यूशन:** if: steps.cache-texlive.outputs.cache-hit != 'true' कंडीशन यह सुनिश्चित करती है कि डिपेंडेंसी केवल तभी इंस्टॉल हो जब यह पहले से कैश न हो।
- डिपेंडेंसी इंस्टॉल करें:** apt-get, wget, और xz-utils जैसी आवश्यक डिपेंडेंसी इंस्टॉल करें।
- texlive इंस्टॉलर डाउनलोड करें**