

मार्कडेन समस्याएँ: `fontspec` और `fontspec`

अपने `fontspec` ब्लॉग के लिए `fontspec` का उपयोग करके `fontspec` जनरेट करने के लिए, मैं निम्नलिखित `fontspec` कमांड का उपयोग करता हूँ:

```
command = [  
  'pandoc',  
  input_markdown_path,  
  '-o', output_pdf_path,  
  '-f', 'markdown',  
  '--pdf-engine', 'xelatex',  
  '--resource-path=.:assets',  
  '-V', f'CJKmainfont={CJK_FONT}',  
  '-V', f'CJKsansfont={CJK_FONT}',  
  '-V', f'CJKmonofont={CJK_FONT}',  
  '-V', f'geometry:{GEOMETRY}',  
  '-V', 'classoption=16pt',  
  '-V', 'CJKoptions=Scale=1.1',  
  '-V', 'linestretch=1.5'  
]
```

`fontspec` और `fontspec` दोनों का समर्थन करना

जब आप `fontspec` लिख रहे होते हैं जो `fontspec` (`fontspec` आउटपुट के लिए) और `fontspec` (`fontspec` के माध्यम से `fontspec` आउटपुट के लिए) के साथ काम करने की आवश्यकता होती है, तो कुछ बातों का ध्यान रखना चाहिए:

- छवि पथ संगतता `fontspec` (`fontspec`): संपत्तियों को संदर्भित करने के लिए / से शुरू होने वाले पथों को प्राथमिकता देता है। `fontspec` (`fontspec`): अग्रणी / के बिना सापेक्ष पथों को प्राथमिकता देता है।

समाधान: दोनों के लिए काम करने वाले सापेक्ष पथों का उपयोग करें:

```
![] (assets/images/chatgpt/block.jpg)
```

- `fontspec` एट्रिब्यूट्स को संभालना `{:.fontspec}` `fontspec` के लिए विशिष्ट है और `fontspec` आउटपुट को स्टाइल करने के लिए उपयोग किया जाता है। `fontspec` इन एट्रिब्यूट्स को सपोर्ट नहीं करता है और इससे त्रुटि उत्पन्न होगी।

समाधान: `fontspec` जनरेशन के लिए इरादा किए गए `fontspec` में `fontspec`-वशिष्ट विशेषताओं को हटा दें। उदाहरण के लिए:

```
<!-- Kramdown-specific -->
```

ChatGPT

Sign in failed

Something went wrong. You may be connected to a disallowed ISP. If you are using VPN, try disabling it. Otherwise try a different Wi-Fi network or data connection. (2) (400, 833e777fddf11a98-GRU)

OK

 Continue with Apple

 Continue with Google

 Sign up with email

Log in

.□□□□□□□□□□ }

! [] (assets/images/chatgpt/block.jpg)

यदि {:.□□□□□□□□□□} आपके □□□□□□ □□□□ लेआउट के लिए महत्वपूर्ण है, तो इसे वेब आउटपुट के लिए चुनिंदा रूप से जोड़ने पर विचार करें, जबकि □□□□ जनरेशन प्रक्रिया में इसे छोड़ दें।

द्वैध संगतता के लिए वर्कफ़्लो

1. □□□□□□□□-वशिष्ट सुविधाओं पर न्यूनतम निर्भरता के साथ □□□□□□□□ सामग्री लिखें।
2. □□□□ में उन्नत स्टाइलिंग के लिए, □□□□□□□□ में इनलाइन के बजाय अपने □□□□□□ टेम्पलेट्स में सीधे □□□ क्लासेस लागू करें।
3. □□□□□□□□ पोर्टेबिलिटी को बनाए रखते हुए □□□ फ़ॉर्मेटिंग को नियंत्रित करने के लिए □□□□□□ विकल्पों का उपयोग करें।

इन प्रथाओं का पालन करके, □□□□□□□□ सामग्री □□□□□□ □□□□ रेंडरिंग और □□□□□□□□ □□□□ जनरेशन में संगत रहती है, जिससे बहु-प्रारूप प्रकाशन के लिए एक सहज वर्कफ़्लो सुनिश्चित होता है।