Click to prove you're human

Freecad manual

The FreeCAD Manual Converter is a Python script that creates PDF and EPUB versions of the FreeCAD manual from the FreeCAD Wiki. This script generates a complete, formatted offline manual with all chapters, images, and formatting from the online manual. The manual is constantly updated by community members, but this script allows users to access it offline or on e-readers and tablets. The script can be used to generate an up-to-date offline copy of the manual at any time, create both PDF and EPUB formats for different reading preferences, include all latest community contributions and updates, maintain proper formatting and image quality, and have a complete reference even without internet access. An error occurs when converting an SVG to PNG, it prints the error message and returns None. A session is created with a User-Agent header set to mimic a Chrome browser on Windows 10. Two empty lists are initialized to store chapter titles and HTML content for EPUB creation. The extract_manual_links function fetches manual links from a specified base URL, defaulting to the FreeCAD wiki manual page, and extracts all links and subchapters. It excludes certain languages and returns a dictionary of unique manual links with their subchapters. The fetch_page function retrieves the HTML content of a given URL, handling exceptions and returning None if the request fails. The extract_main_content function processes the HTML content, removing unwanted elements, fixing links, and processing image paths. It extracts the main content within the 'mw-parser-output' div and adds a chapter title as a heading if not present. Images are processed and stored for EPUB creation, with SVG images converted to PNG and raster images optimized. The function returns None if the main content is not found. for link in main_content.find_all('a'): href = link.get('href') if href and href.startswith('/'): link['href'] = f"{base_url}{href}" self.chapters_html.append({ 'title': chapter_title, 'id': chapter_id, 'content': main_content, 'number': len(self.chapters_html) + 1 }) return str(main_content) def convert_to_pdf(self, url, chapter_number, chapter_id, subchapters, output_dir='pdfs'): print(f"Processing {url}...") os.makedirs(output_dir, exist_ok=True) html_content = self.fetch_page(url) if not html_content: return None soup = BeautifulSoup(html_content, 'html.parser') chapter_title = soup.title.string.split(" - ")[0] if soup.title else "Chapter" chapter_title = chapter_title.replace("Manual:", "").strip() base_url = " " main_content = self.extract_main_content(html_content, base_url, chapter_title, chapter_id, subchapters) if not main_content: return None output_file = os.path.join(output_dir, f"{chapter_id}.pdf") self.toc_entries.append((chapter_number, chapter_title, chapter_id, subchapters)) styled_content = f""" body {{ margin: 20px; font-family: Arial, sans-serif; }} h1 {{ text-align: center; font-size: 24px; margin-bottom: 10px; }} h2 {{ font-size: 18px; margin-top: 10px; }} img {{ max-width: 100%; height: auto; }} code, pre {{ font-family: Consolas, "Courier New", monospace; font-size: 14px; background-color: #f4f4f4; padding: 10px; display: block; white-space: pre-wrap; word-wrap: break-word; overflow-x: auto; border: 1px solid #ddd; border-radius: 5px; }} .mw-highlight {{ padding: 10px; margin: 10px 0; border: 1px solid #ddd; background-color: #f9f9f9; border-radius: 5px; }} .wikitable {{ width: 100%; border-collapse: collapse; margin: 20px 0; font-size: 14px; text-align: left; page-break-inside: avoid; }} .wikitable th, .wikitable td {{ border: 1px solid #ddd; padding: 10px; page-break-inside: avoid; }} .wikitable tr {{ page-break-inside: avoid; }} .wikitable tr:nth-child(even) {{ background-color: #f9f9f9; }} .wikitable tr:hover {{ background-color: #f1f1f1; }} .wikitable th {{ background-color: #f2f2f2; text-align: center; font-weight: bold; }} {main_content} """ try: HTML(string=styled_content).write_pdf(output_file) print(f"Created {output_file}") return output_file except Exception as e: print(f"Error creating PDF for {url}: {e}") return None def generate_toc(self, output_file='pdfs/Table_of_Contents.pdf'): """Generate a Table of Contents PDF with indented subchapters.""" print("Generating Table of Contents...") toc_html = """ body {{ margin: 20px; font-family: Arial, sans-serif; }} h1 {{ text-align: center; font-size: 28px; }} ul {{ list-style: none; padding: 0; }} li {{ margin: 5px 0; }} .chapter {{ font-weight: bold; font-size: 16px; }} .subchapter {{ font-size: 14px; margin-left: 20px; }} a {{ color: blue; text-decoration: none; }} Table of Contents """ for chapter_number, title, chapter_id, subchapters in self.toc_entries: toc_html += f'{chapter_number}. {title}' for i, sub in enumerate(subchapters): toc_html += f' Subchapter {i+1}: {sub["title"]}' def create_epub(self, output_file='FreeCAD_User_Manual.epub'): book = epub.EpubBook() book.set_identifier(str(uuid.uuid4())) book.set_title('FreeCAD User Manual') book.set_language('en') book.add_author('FreeCAD Community') style = '''body { margin: 20px; font-family: Arial, sans-serif; } h1 { text-align: center; font-size: 2em; margin-bottom: 1em; } h2 { font-size: 1.5em; margin-top: 1em; } img { max-width: 100%; height: auto; } code, pre { font-family: monospace; background-color: #f4f4f4; padding: 0.5em; display: block; white-space: pre-wrap; border: 1px solid #ddd; border-radius: 5px; } table { width: 100%; border-collapse: collapse; margin: 1em 0; } th, td { border: 1px solid #ddd; padding: 0.5em; }''' nav_css = epub.EpubItem(uid="style_nav", file_name="style/nav.css", media_type="text/css", content=style) book.add_item(nav_css) chapters = [] for chapter_data in self.chapters_html: chapter = epub.EpubHtml(title=chapter_data['title'], file_name=f'chapter_{chapter_data["number"]}.xhtml', lang='en') book.add_item(chapter) chapters.append(chapter) From the provided text, it appears that a Python class named `FreeCADManualConverter` is designed to convert a manual into various formats such as PDF and EPUB. The conversion process involves extracting links from the manual, creating separate PDF files for each chapter, generating a Table of Contents, merging the PDF files, and finally converting them to an EPUB format if required. The class also includes methods to extract manual links, batch convert the manual to multiple formats, and generate the necessary metadata such as the title and date. Additionally, the code provides information about the FreeCAD software, which is an open-source 3D CAD modeling tool. FreeCAD is suitable for both personal and commercial use, with its source code openly published under the LGPL license, making it an attractive option for hobbyists and experienced users alike. Initially released in 2002, FreeCAD has undergone significant growth thanks to contributions from its community of developers and users. The software's capabilities have expanded through the addition of various workbenches, including the FEM workbench for Finite Element Analysis and the BIM workbench for Building Information Modeling. This user manual, written for version 1.0 of FreeCAD, aims to guide individuals in using the software for their projects, requiring no prior CAD knowledge. The manual is organized into application areas, covering installation, prominent workbenches, and advanced topics like Python scripting, all presented in a step-by-step manner with images. Published under the Creative Commons 4.0 license, the manual can be freely used and modified. Continuous improvement is ensured through user contributions, and additional resources include a Python script for converting the manual to offline formats and the comprehensive FreeCAD wiki, although it is recommended to start with the user manual for a smoother introduction. The FreeCAD Forums also provide valuable support and insights, making them a helpful resource for users.