

Разделяне на произволен текст на секции

Този документ описва **реалното** поведение на RIP Help System при разделяне на help-файл на секции. Източникът е `help_processor.py` след последните корекции (вкл. `merge_short_sections` / `merge_preamble_sections`). Не е маркетингово резюме.

Генерирането на ключови думи е **извън обхвата**, освен един ред в края.

За да прегенерираш PDF-а:

```
python docs/md_to_pdf.py
```

1. Място в обработката

За всеки файл `process_file()` прави:

- 1 Избира парсер по разширение (`.html` / `.htm`, `.docx`, `.doc`, `.pdf`, `.txt`).
- 2 Парсерът връща списък от `Section(title, text, level, images, html_text)`.
- 3 Върху списъка се вика `merge_short_sections()`, после `merge_preamble_sections()`.
- 4 Празните блокове се филтрират при запис; заглавие без тяло се запазва (виж §8).

ZIP не е формат за разделяне: разопакова се и всеки вътрешен файл минава по същия път.

Нива: 1 = H1, 2 = H2, 3 = H3 (по-дълбоки HTML/Word заглавия се режат до 3), 0 = без заглавие.

2. Какво е граница на секция

Граница = **открито заглавие ниво 1** (H1 / Heading 1 / Заглавие 1) в HTML и DOCX. Текущата секция се затваря (`flush`) и започва нова. Текстът на заглавието става `title` и **не** се копира в тялото на секцията.

В HTML/DOCX **не** отварят нова секция:

- **Word Title / Наименование** / CSS `Title` / `MsTitle` — корица; става заглавие на преамбюла (или ред в тялото, ако вече има съдържание);
- „**Съдържание**“ / `Contents` / `TOC` и еквиваленти — остават в тялото;
- H2–H6, `Subtitle` / Подзаглавие, изцяло **bold** параграфи — влизат като редове в **текущата** секция.

TXT и PDF запазват своите правила (номерирани редове / по-голям шрифт) — там ниво 2 е нормална граница.

Всичко, което не е граница — параграфи, списъци, таблици, картинки — се добавя към **текущата** секция.

Ако преди първото H1 има съдържание (корица, TOC), то образува преамбюл секция (с `title` от корицата, ако има).

Ако парсерът не открие нито една секция, файлът става **една** секция без заглавие, с целия извлечен текст.

3. Word (.docx)

Обхождат се децата на `w:body` в `document-order`: параграфи и таблици (`_iter_docx_blocks`). Стандартният `Document.paragraphs` пропуска таблиците, затова не се ползва за сегментиране.

3.1. Стилово заглавие

`_docx_heading_level()` гледа `style_id` и `style.name`, после `base_style` нагоре по веригата. Токенът се нормализира (без интервали, `_`, `-`, без значение на регистъра).

Разпознават се поне:

Токен (примери)	Ниво / роля
Heading 1 / heading1	1 — граница
Heading 2 / 3	2 / 3 — в тялото, не граница
Title / MsoTitle / Наименование	корица — не граница
Subtitle / Подзаглавие	2 — в тялото
Заглавие / Заглавие 1	1 — граница
Заглавие 2 / 3	2 / 3 — в тялото
Überschrift / Überschrift 1	1 — граница
MsoHeading1 / 2 / 3	1 / 2 / 3

Число над 3 се ограничава до 3. Subtitle / Подзаглавие без цифра → ниво 2; останалите именувани heading-стилове без цифра → ниво 1. **Title / Наименование** не връщат ниво за сегментиране.

Ако стилът не е heading, има fallback: Word outlineLvl 0–2 и текст под 120 символа → ниво = outlineLvl + 1.

3.2. Bold като подзаглавие (не граница)

Ако няма стилово ниво 1, параграфът се приема за **текст в тялото** (бивш „heading“ ниво 2), когато:

- видимият текст е непраzen и ≤ 120 символа;
- всички run-ове с текст са **bold**;
- стилът **не** започва с list;
- в параграфа **няма** картинки.

Такъв ред **не** отваря нова секция — остава под текущото H1.

3.3. Таблици и картинки

Таблица: всеки ред се сплесква до клетка | клетка | клетка и редовете се добавят в тялото на текущата секция.

Картинки в параграф: записват се като [IMG: img_NN] в същото тяло.

Празен параграф без картинка се пропуска.

3.4. Fallback

Ако след обхождането няма секции: една секция от всички непразни параграфи; ако и те са празни — от таблиците.

4. Стар .doc

Конвертира се към .docx с LibreOffice (soffice) или, на Windows, MS Word през COM. После се вика същият parse_docx. Ако и двете конверсии пропаднат, файлът се чете като обикновен текст (parse_txt).

5. HTML / HTM (вкл. Word „Запиши като уеб“)

Премахват се script, style, nav, footer, header, noscript.

Събират се **top-level** блокове: h1–h6, p, ul, ol, table, dl, pre, blockquote, figure, hr, самостоятелни img, и div **само** ако изглежда като заглавие. Вложен блок не се обработва втори път.

5.1. Заглавие

1 Тагове: h1→ граница (ниво 1); h2–h6→ в тялото (ниво 2/3).

- 2 CSS class с heading / заглавие / msoheading / überschrift (+ опционална цифра). Title / MsoTitle / наименование са **корица**, не граница. subtitle / подзаглавие → в тялото.
- 3 p или div с текст < 120 символа, изцяло обвит в / → в тялото (не граница).
- 4 Текст „Съдържание“ / Contents / TOC (дори като h1) → в тялото на преамбюла.

5.2. Съдържание

Списъци и таблици влизат в текущата секция. В plain text редовете им са с нов ред (не сплескани в един ред). Декоративни атрибути (class, style, on*, data-* ...) се махат от запазените HTML.

Картинки: локални и data: URI; HTTP(S) се пропускат. Дребни иконки под 50×50 px се изхвърлят.

Ако няма нито една секция: цялото body като една секция без заглавие.

6. Обикновен текст (.txt)

Кодирането се детектира с chardet.

Ред е граница, ако след strip():

- съвпада с Markdown заглавие #{1,3} текст — нивото е броят #, заглавието е текстът след тях; **или**
- съвпада с номерирано заглавие: 1. ..., 2) ..., I. ..., A. ... (1–2 цифри / римски / една главна буква, после . или)), дължина на реда < 120.

Всички останали редове отиват в тялото (десният край се реже, водещите интервали се пазят).

Ако няма такива редове: **една** секция с целия файл.

7. PDF

Няма стилове. Текстът се събира като визуални редове по Y (не по PDF stream), буквите от drop-cap се слепват, сплескани списъци от вида 1 Foo 2 Bar се разбиват на редове.

Граница: група редове с почти еднакъв размер на шрифта, ако размерът е **по-голям от предишния с повече от 1 pt** и текстът е **под 150 символа** → нова секция, ниво 2.

Това е по-ненадеждно от Word/HTML. Картинките се кропват от страницата и се редят в текущата секция по вертикална позиция.

8. Сливане на кратки секции (merge_short_sections / merge_preamble_sections)

Константа: MIN_SECTION_TOKENS = 60 (думи в **тялото**, str.split()).

merge_short_sections:

- Секция с **непразно заглавие не се слива**, дори тялото да е късо или празно.
- Секция **без заглавие** и с под 60 думи се залепва към предишната (текст, картинки, HTML).
- Няма предишна секция → късият untitled блок остава сам.

merge_preamble_sections (след горното):

- Ако първата секция е корица с късо/празно тяло, а втората е озаглавена „Съдържание“/ТОС → сливат се в един преамбюл (title = корицата; TOC заглавието влиза в тялото).

Следствия:

- Корица + съдържание + първа глава (H1) → **две** секции (не четири от Title / TOC / H1 / H2).
- Последователни H1 без тяло („Глава 1“ веднага следвано от друг H1) остават **две** секции.

- H2 / bold подзаглавия („Какво е ...?“) не режат секция — остават в тялото на текущото H1.
- Untitled текст в началото на файла остава отделна секция (докато не е под прага *и* няма към какво да се слее — първият елемент не се слива).

При запис в `process_file()`:

- Ако няма текст, картинки и HTML, но има заглавие → тялото става самото заглавие. Затова „голо“ заглавие оцелява като секция.
- Ако няма нито текст, нито заглавие, нито картинки → секцията се пропуска.

`clean_text()` скрива поредици от интервали/табове, но пази нови редове (списъци и абзаци).

9. Текст без заглавие

Типични случаи:

- Увод преди първия Heading 1.
- HTML/DOCX без нито едно разпознато заглавие → целият файл е една untitled секция.
- TXT без markdown/номерация.
- Къс untitled остатък след заглавие се слива с предишната секция (§8).

Празното `title` по-късно се попълва от класификатора (извън този документ).

10. Таблицы — обобщение

Формат	Поведение
DOCX	Редове `a \
HTML	Целият <code><table></code> е блок в текущата секция
TXT / PDF	Няма отделен table parser; таблицата е просто текст/редове

11. Ключови думи (един ред)

След като секциите са вече разделени, всяка получава заглавие и ключови думи чрез Claude или fallback; ръчните думи от таб „Сканиране“ се записват първи във всяка секция.