PDF extraction practice pack

Answer key & original experiment results | 12 November 2026

Practice challenges

A. Consecutive pages: PDF positions **2–3**. This gives you the budget summary and detailed cost breakdown. The resulting file has two pages, even though their printed page numbers are 1 and 2.

B. Non-adjacent pages: PDF positions **1,3,5–7**. You should get five pages, in this order: Cover, Cost breakdown, Milestone diagram, Intake form, Terms.

C. Single page: PDF position **6**. This is the editable intake form. Try changing the contact field, save, reopen, and check whether the change persisted in your PDF editor.

Why the numbers differ

The source PDF contains eight physical pages, but its printed numbering starts at 1 after the cover, then restarts at 1 on PDF position 7. Use **page positions in the viewer** for extraction, not the printed numerals.

Measured local extraction results

We generated the source PDF programmatically and extracted selections with **pypdf 5.9.0**. We then compared the source and extracted renderings with **PyMuPDF 1.26.7** at a 1.3× rasterization scale. **These are not Soda PDF Desktop or Online test results.**

Selection	Pages	Size	% of 5.38 MB source
Budget (positions 2–3)	2	5,807 B	0.11%
Scanned receipt (position 4)	1	124,312 B	2.31%
High-detail image (position 8)	1	5,247,149 B	97.45%
Scattered set (1,3,5–7)	5	13,869 B	0.26%

What this shows: page count alone is a poor predictor of output file size when one page contains most of the document's embedded data. We intentionally designed the last page to be very image-heavy; these percentages are not representative averages.

Content and behavior checks

Appearance: Across eight selections covering 13 output pages, every tested rendering matched its original page pixel-for-pixel at the chosen resolution. Text extracted through pypdf matched the source pages for those selections.

Links: In this pypdf run, a link to a page outside the chosen selection was removed. Links whose destination pages were included, plus an external web link, remained present. This is a software-specific result; other PDF programs may differ.

Forms: When PDF position 6 was extracted alone, its text field and checkbox remained present. We programmatically edited the contact name, saved the result, reopened it and confirmed the changed value. Try it yourself in Soda PDF.

Scans: The scanned receipt remained a bitmap. Extraction did not add searchable text to the receipt image.

Digital signatures: This pack deliberately contains no signed PDF. Never infer signature validity from a visually unchanged extracted page.

Reproducibility

The package includes the original source PDF, extracted-example PDFs, machine-readable JSON and CSV results, and the Python generator. We rendered each output using the same local PDF engine for the comparison.

Scope limitation: No controlled extraction was executed in Soda PDF Desktop or Soda PDF Online in this environment. All numeric and behavior claims in this report refer specifically to the documented local pypdf experiment.