1. PdfFileReader() needs to be opened in read-binary mode by passing 'rb' as the second argument to open(). Likewise, the File object passed to PyPDF2. PdfFileWriter() needs to be opened in write-binary mode with 'wb'.

2. Calling getPage(4) will return a page object for page 5, since page 0 is the first page.

3. The total number of pages in the document is stored in the numPages variable of a PdfFileReader object.

4. To read an encrypted PDF, call the decrypt() function and pass the password as a string, in our case it is swordfish. After we call decrypt() with the correct password, you’ll see that calling getPage() no longer causes an error. Call decrypt('swordfish').

5. The pages of a PDF can be rotated in 90-degree increments with the rotateClockwise() and rotateCounterClockwise() methods.

6. A document contains multiple paragraphs. A paragraph begins on a new line and contains multiple runs. Runs are contiguous groups of characters within a paragraph.

7. len(doc.paragraphs).

8. Run Object.

9. True always makes the run object bolded and False makes it always not bolded, no matter what the style's bold settings is. None will make the Run object just use the style's bold settings.

10. mydoc = docx.Document().

11. doc.add\_paragraph('Hello, there!').

12. The arguments to add\_heading() are a string of the heading text and an integer from 0 to 4. The integer 0 makes the heading the Title style, which is used for the top of the document. Integers 1 to 4 are for various heading levels, with 1 being the main heading and 4 the lowest subheading. The add\_heading() function returns a Paragraph object to save you the step of extracting it from the Document object as a separate step.