**ANS1. To get a PdfFileReader object that represents this PDF, call PyPDF2.PdfFileReader() that is returned from open()**

**ANS2. Read-binary ('rb') for PdfFileReader() and write-binary ('wb') for PdfFileWriter()**

**ANS3. Calling getPage(4) will return a Page object for**[**page 5**](https://automatetheboringstuff.com/2e/appendixc/#calibre_link-18)**, since page 0 is the first page.**

**ANS4. The numPages variable stores an integer of the number of pages in the PdfFileReader object.**

**ANS5. Call decrypt(“swordfish”)**

**ANS6. The pages of a PDF can also be rotated in 90-degree increments with the rotateClockwise() and rotateCounterClockwise() methods. Pass one of the integers 90, 180, or 270 to these methods.**

**ANS7. docx.Document('demo.docx') returns a Document object for a file named demo.docx**

**ANS8. The Document object contains a list of Paragraph objects for the paragraphs in the document.Each of these Paragraph objects contains a list of one or more Run objects. The text in a Word document is more than just a string. It has font, size, color, and other styling information associated with it. A style in Word is a collection of these attributes. A Run object is a contiguous run of text with the same style.**

**ANS9. doc.paragraphs obtain a list of Paragraph objects for a Document object that’s stored in a variable named doc**

**ANS10. Run object has bold, underline, italic, strike, and outline variables**

**ANS11. True always makes the Run object bolded and False makes it always not bolded, no matter what the style’s bold setting is. None will make the Run object just use the style’s bold setting.**

**ANS12. docx.Document() function, creates a Document object for a new Word document**

**ANS13. doc.add\_paragraph('Hello there!')**

**ANS14. Only integers 0, 1, 2, 3, and 4**