

1. In what modes should the PdfFileReader() and PdfFileWriter() File objects will be opened?
2. From a PdfFileReader object, how do you get a Page object for page 5?
3. What PdfFileReader variable stores the number of pages in the PDF document?
4. If a PdfFileReader object's PDF is encrypted with the password swordfish, what must you do before you can obtain Page objects from it?
5. What methods do you use to rotate a page?
6. What is the difference between a Run object and a Paragraph object?
7. How do you obtain a list of Paragraph objects for a Document object that's stored in a variable named doc?
8. What type of object has bold, underline, italic, strike, and outline variables?
9. What is the difference between False, True, and None for the bold variable?
10. How do you create a Document object for a new Word document?
11. How do you add a paragraph with the text '&#39;Hello, there!&#39;' to a Document object stored in a variable named doc?
12. What integers represent the levels of headings available in Word documents?

1. PdfFileReader() and PdfFileWriter() File objects should be opened in binary mode using the "rb" (read binary) and "wb" (write binary) modes respectively.
2. To get a Page object for page 5 from a PdfFileReader object, you can use the following code: `page_five = pdf_reader.getPage(4)` since pages are zero-indexed.
3. The `numPages` variable stores the number of pages in the PDF document for a PdfFileReader object. You can access it using `pdf_reader.numPages`.
4. If a PdfFileReader object's PDF is encrypted with the password swordfish, you must call the `decrypt('swordfish')` method on the PdfFileReader object before you can obtain Page objects from it.
5. To rotate a page in PyPDF2, you can use the `rotateClockwise(degrees)` or `rotateCounterClockwise(degrees)` method on a Page object, where `degrees` is the number of degrees to rotate the page.
6. A Run object represents a contiguous run of text with the same set of character properties, while a Paragraph object represents a block of text with its own formatting properties.
7. To obtain a list of Paragraph objects for a Document object stored in a variable named `doc`, you can use the `doc.paragraphs` attribute.
8. A Run object has bold, underline, italic, strike, and outline variables.
9. The `bold` variable for a Run object can be set to `True`, `False`, or `None`. `True` indicates that the text should be bold, `False` indicates that it should not be bold, and `None` indicates that the text should inherit its bold property from the style of the paragraph it belongs to.
10. To create a Document object for a new Word document, you can use the following code: `doc = docx.Document()`.

11. To add a paragraph with the text "Hello, there!" to a Document object stored in a variable named `doc`, you can use the following code:  
`doc.add_paragraph('Hello, there!')`.
12. The integers 0 to 9 represent the levels of headings available in Word documents, with 0 being the highest level (main heading) and 9 being the lowest level (sub-sub-sub-sub-sub-sub-sub-sub-subheading).