13.2 Plain Text

Function *TextPage.extractText()* (or *Page.get_text("text")*) extracts a page's plain **text in original order** as specified by the creator of the document (which may not equal a natural reading order).

An example output:

```
>>> print(page.get_text("text"))
Some text on first page.
```

13.3 BLOCKS

Function *TextPage.extractBLOCKS()* (or *Page.get_text("blocks")*) extracts a page's text blocks as a list of items like:

```
(x0, y0, x1, y1, "lines in block", block_type, block_no)
```

Where the first 4 items are the float coordinates of the block's bbox. The lines within each block are concatenated by a new-line character.

This is a high-speed method with enough information to re-arrange the page's text in natural reading order where required.

Example output:

```
>>> print(page.get_text("blocks"))
[(50.0, 88.17500305175781, 166.1709747314453, 103.28900146484375,
'Some text on first page.', 0, 0)]
```

13.4 WORDS

Function *TextPage.extractWORDS()* (or *Page.get_text("words")*) extracts a page's text **words** as a list of items like:

```
(x0, y0, x1, y1, "word", block_no, line_no, word_no)
```

Where the first 4 items are the float coordinates of the words's bbox. The last three integers provide some more information on the word's whereabouts.

This is a high-speed method with enough information to extract text contained in a given rectangle.

Example output:

```
>>> for word in page.get_text("words"):
        print(word)
(50.0, 88.17500305175781, 78.73200225830078, 103.28900146484375,
'Some', 0, 0, 0)
(81.79000091552734, 88.17500305175781, 99.5219955444336, 103.28900146484375,
'text', 0, 0, 1)
(102.57999420166016, 88.17500305175781, 114.8119888305664, 103.28900146484375,
'on', 0, 0, 2)
(117.86998748779297, 88.17500305175781, 135.5909881591797, 103.28900146484375,
'first', 0, 0, 3)
```

(continues on next page)