Loading, Scraping, Cleaning, and Storing the Data

The scraping procedure for `parse_salaries_single_year()` is as follows:

1. read the entire PDF
   (a) `pdftools::pdf_text()` reads the contents as a *vector*, each element being a character string of one page in the file
   (b) combine the elements of this vector into a *single* character string
      - this is necessary because some entries span 2 pages
2. clean out the junk
   (a) remove the header text
   (b) remove the footer text
   (c) remove excess whitespace
3. separate the different faculty members
   (a) split the string using the delimiter given in the PDF file (the dashed lines)
   (b) `readr::str_split()` will return a `list()` instead of a vector of character strings; so, we extract element `[[1]]` of the list
   (c) this `[[1]]`st element is the vector we want; but, its first element is an empty string. So, remove it.
4. separate the variables *within* each faculty member
   (a) split each string in the vector by the delimiters given by `employee.dlms` variable created above. Setting `simplify=TRUE` will ensure the result is returned as a *matrix* instead of a list. The columns of this matrix will contain the variables we want.
   (b) remove excess whitespace from each element of the matrix
   (c) using `reader::str_split()` will create more empty strings, so we remove them (as in step 3(c))
5. store the result
   (a) turn the matrix into a `data.frame` object
   (b) set the column names to those stored in the `employee.vars` variable defined above
6. alter the format of the data
   - some faculty have (or have had) multiple assignments. What we have done so far will create a `data.frame` with enough columns to store the variables for *all* positions a person has / has had
   - this is an okay way to store the data; but, there will be MANY `NA` values since most people only have 1 assignment, so

   (a) create *new* observations for each position
   (b) throwing away the extra columns
      - that is, there will be duplicate *people* within the dataset, but each observation for that person will be a different position
7. clean the data
   (a) split `AnnSalary` into 2 columns: 1 for the actual salary and 1 for the appointment type
      - otherwise, the elements of the `AnnSalary` variable look like `45000.00 9 mo` to denote a person with a 9-month appointment making $45k salary (not usable)
   (b) split the `Name` column into first- and last-name columns
   (c) delete observations with tentative values (marked with a `*`)
      - in the `View()` pane, just search for `*` and you'll see what I mean
   (d) set the intended variable types