In this assignment, you will write four functions that use regular expressions. For Problem 4, you need to first download the Dodds et al happiness dictionary, `happiness_dictionary.py`.

Instructions: Name your file `hw4.py` and submit on CCLE. Add comments to each function.

- **Problem 1**:
  Write a function `mytype(v)` that performs the same action as `type()`, and can recognize integers, floats, strings, and lists. Do this by first using `str(v)`, and then reading the string. Assume that lists can only contain numbers (not strings, other lists, etc...), and assume that strings can be anything that is not an integer, float or list. Note that an empty list `[]` should also be recognized as a list.

  Test cases:

  ```
  mytype(10) and mytype(-10) should return "int";
  mytype(-1.25) and mytype(10.0) should return "float";
  mytype([1, 2, 3]) and mytype([]) should return "list";
  mytype("abc") and mytype({1,2}) should return "string";
  ```

- **Problem 2**:
  Write a function `findpdfs(L)` that takes as input a list L of filenames (such as “IMG2309.jpg”, “lecture1.pdf”, “homework.py”), and returns a list of the names of all PDF files, without extension (“lecture1”). Assume that filenames may contain only letters and numbers.

  Test case:

  ```
  L = ["IMG2309.jpg", "lecture1.pdf", "homework.py", "homework2.pdf"]
  findpdfs(L) should return ["lecture1", "homework2"].
  ```

- **Problem 3**:
  Write a function `findemail(url)` that takes as input a URL, and outputs any email addresses that look like “xxx@xxx.xxx.xxx” with any number of dots after the @-sign on this page. The order of the email addresses in the output doesn’t matter. Your function should also get around tricks people use to hide their email addresses, such as

  ```
  hangjie@math.ucla.edu
  hangjie AT math DOT ucla DOT edu
  hangjie at math dot ucla dot edu
  hangjie[AT]ucla[DOT]edu
  hangjie[at]ucla[dot]edu
  ```

  Test cases:

  ```
  url1 = "https://www.math.ucla.edu/~hangjie/contact/"
  url2 = "https://www.math.ucla.edu/~hangjie/teaching/Winter2019PIC16/regexTest"

  findemail(url1) should return ["hangjie@math.ucla.edu"];
  findemail(url2) should return ["hangjie1@math.ucla.edu",
   "hangjie2@math.ucla.edu", "hangjie3@math.ucla.edu",
    "hangjie4@ucla.edu","hangjie5@ucla.edu","xxx@xxx.xxx.xxx"].
  ```

- **Problem 4**:
  Write a function `happiness(text)` that uses the Dodds et al [1] happiness dictionary to rate the happiness of a piece of english text (input as a single string). The happiness score is the average score over all words in the text that appear in the dictionary. For simplicity, you may neglect the words with special characters in the dictionary.

  Test cases:

  ```
  s1 = "Mary had a little lamb."
  s2 = "Mary had a little lamb. Mary had a little lamb!"
  s3 = "A quick brown fox jumps over a lazy dog."

  happiness(s1) and happiness(s2) should return 5.368;
  happiness(s3) should return 5.275.
  ```

References

[1] Peter Sheridan Dodds, Kameron Decker Harris, Isabel M Kloumann, Catherine A Bliss, and Christopher M Danforth. Temporal patterns of happiness and information in a global social network: Hedonometrics and twitter. *PloS one*, 6(12):e26752, 2011.