Natural Questions 1

Instructions:

Quickly, without looking at your computer screen, circle the digits in the understanding columns to rate the following statement for each substantive line of code. Answer the two questions for as many lines as you can, starting at the top.

The questions are:

1) **I understand what this line does and its purpose in the larger program.**
(**1** for completely disagree, **5** for completely agree)

	Understand	Source Code of the Example From Your Task
1		`function print(string){`
2		`    document.getElementById("output_div").innerHTML += string;`
3		`    document.getElementById("output_div").innerHTML += "<br>";`
4		`}`
5		
6		
7	**1 2 3 4 5**	`var tokenizer = new natural.WordTokenizer();`
8	**1 2 3 4 5**	`print(tokenizer.tokenize("your dog has fleas."));`
9		`// output: [ 'your', 'dog', 'has', 'fleas' ]`
10		
11		
12	**1 2 3 4 5**	`print(natural.LancasterStemmer.stem("words"));`
13		`// output: 'word'`
14		
15	**1 2 3 4 5**	`natural.LancasterStemmer.attach();`
16	**1 2 3 4 5**	`print("i am waking up to the sounds of chainsaws".tokenizeAndStem());`
17		`// output: ["wak", "sound", "chainsaw"]`
18	**1 2 3 4 5**	`print("chainsaws".stem());`
19		`// output: chainsaw`
20		
21		
22	**1 2 3 4 5**	`var NGrams = natural.NGrams;`
23		
24	**1 2 3 4 5**	`print(NGrams.bigrams('some words here'));`
25		`// output: [ [ 'some', 'words' ], [ 'words', 'here' ] ]`
26	**1 2 3 4 5**	`print(NGrams.bigrams(['some',  'words',  'here']));`
27		`// output: [ [ 'some', 'words' ], [ 'words', 'here' ] ]`
28		
29	**1 2 3 4 5**	`print(NGrams.trigrams('some other words here'));`

30		// output: [['some', 'other', 'words'], ['other', 'words', 'here']]
31	**1 2 3 4 5**	print(NGrams.trigrams(['some', 'other', 'words', 'here']));
32		// output: [['some', 'other', 'words'], ['other', 'words', 'here']]
33		
34	**1 2 3 4 5**	print(NGrams.ngrams('some other words here for you', 4));
35		// output: [['some', 'other', 'words', 'here'], ['other', 'words', 'here', 'for'], ['words',
36		'here', 'for', 'you']]
37	**1 2 3 4 5**	print(NGrams.ngrams(['some', 'other', 'words',
38		'here', 'for', 'you'], 4));
39		// output: [['some', 'other', 'words', 'here'], ['other', 'words', 'here', 'for'], ['words',
40		'here', 'for', 'you']]
41		
42		
43	**1 2 3 4 5**	var TfIdf = natural.TfIdf;
44	**1 2 3 4 5**	var tfidf = new TfIdf();
45		
	1 2 3 4 5	tfidf.addDocument('this document is about node.');
46	**1 2 3 4 5**	tfidf.addDocument('this document is about ruby.');
47	**1 2 3 4 5**	tfidf.addDocument('this document is about ruby and node.');
48	**1 2 3 4 5**	tfidf.addDocument('this document is about node. it has node examples');
49		
		print('node --------------------------------');
50	**1 2 3 4 5**	tfidf.tfidfs('node', function(i, measure) {
51	**1 2 3 4 5**	print('document #' + i + ' is ' + measure);
52		});
53		// output: node --------------------------------
54		// output: document #0 is 1
55		// output: document #1 is 0
56		// output: document #2 is 1
57		// output: document #3 is 2
58		
59		print('ruby --------------------------------');
60	**1 2 3 4 5**	tfidf.tfidfs('ruby', function(i, measure) {
61	**1 2 3 4 5**	print('document #' + i + ' is ' + measure);
62		});
63		// output: ruby --------------------------------
64		// output: document #0 is 0
65		// output: document #1 is 1.2876820724517808
66		// output: document #2 is 1.2876820724517808
67		// output: document #3 is 0
68		
69	**1 2 3 4 5**	print(tfidf.tfidf('node', 0));
70		// output: 1
71	**1 2 3 4 5**	print(tfidf.tfidf('node', 1));
72		// output: 0

73		
74	**1 2 3 4 5**	tfidf.listTerms(0).forEach(function(item) {
75	**1 2 3 4 5**	print(item.term + ': ' + item.tfidf);
76		});
77		// output: node: 1
78		// output: document: 0.7768564486857903
79		
80		
81		
82	**1 2 3 4 5**	var lexicon = new natural.Lexicon("lib/lexicon_from_posjs.json", 'N');
83	**1 2 3 4 5**	var rules = new natural.RuleSet("lib/tr_from_posjs.txt");
84	**1 2 3 4 5**	var tagger = new natural.BrillPOSTagger(lexicon, rules);
85		
86	**1 2 3 4 5**	var sentence = ["I", "see", "the", "man", "with", "the", "telescope"];
87	**1 2 3 4 5**	print(JSON.stringify(tagger.tag(sentence)));
88		// output: [["I","NN"],["see","VB"],["the","DT"],["man","NN"],["with","IN"],["the","DT"],["telescope","NN"]]

Natural Questions 2

Instructions:

Write one or more sentences in answer to the following two questions. You may reference your code and the example code from this task.

1) What information or strategies did you find most helpful?

2) What additional information would you have wanted?