

```

#!/usr/bin/env python
# -*- coding: utf-8 -*-
"""
Parses the OSM file and counts the tags by type.
"""

import xml.etree.ElementTree as ET
from pprint import pprint
import operator

OSMFILE = 'data/austin.osm'

def count_tags(filename, limit=-1, verbose=False):
    """
    Parses the OSM file and counts the tags by type.
    """
    # initialize dict objects and counter
    tag_count = {}
    tag_keys = {}
    counter = 0

    # iterate through elements
    for _, element in ET.iterparse(filename, events=("start",)):
        # add to tag count
        add_tag(element.tag, tag_count)

        # if tag and has key, add the tag key to tag_keys dict
        if element.tag == 'tag' and 'k' in element.attrib:
            add_tag(element.get('k'), tag_keys)

        # print if verbose output enabled
        if verbose:
            print "{0}: {1}".format(counter, element.tag)

        # break if exceed limit
        if limit > 0 and counter >= limit:
            break
        counter += 1

    # produces a sorted-by-decreasing list of tag key-count pairs
    tag_keys = sorted(tag_keys.items(), key=operator.itemgetter(1))[::-1]

    # return values
    return tag_count, tag_keys

def add_tag(tag, tag_count):
    """ adds a tag to tag_count, or initializes at 0 if does not yet exist """
    if tag in tag_count:
        tag_count[tag] += 1

```

```
else:
    tag_count[tag] = 1

def main(limit=-1, verbose=False):
    """ main function """
    tags, tag_keys = count_tags(OSMFILE, limit, verbose)
    pprint(tag_keys)
    pprint(tags)
    return tags, tag_keys

if __name__ == "__main__":
    main()
```