Найти 3 самых часто встречающихся слова в Zen of Python

In [1]:

```
import this
```

```
The Zen of Python, by Tim Peters

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!
```

In [2]:

```
zen = """Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never.
Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!"""
```

In [3]:

```
zen_map = dict()

for word in zen.split():
    cleaned_word = word.strip('.,!-*').lower()
    if cleaned_word not in zen_map:
        zen_map[cleaned_word] = 0

    zen_map[cleaned_word] += 1

print(zen_map)
```

```
{'beautiful': 1, 'is': 10, 'better': 8, 'than': 8, 'ugly': 1, 'explicit': 1, 'implicit': 1,
'simple': 1, 'complex': 2, 'complicated': 1, 'flat': 1, 'nested': 1, 'sparse': 1, 'dense':
1, 'readability': 1, 'counts': 1, 'special': 2, 'cases': 1, "aren't": 1, 'enough': 1, 'to':
5, 'break': 1, 'the': 5, 'rules': 1, 'although': 3, 'practicality': 1, 'beats': 1, 'purity':
1, 'errors': 1, 'should': 2, 'never': 3, 'pass': 1, 'silently': 1, 'unless': 2, 'explicitl
y': 1, 'silenced': 1, 'in': 1, 'face': 1, 'of': 2, 'ambiguity': 1, 'refuse': 1, 'temptatio
n': 1, 'guess': 1, 'there': 1, 'be': 3, 'one': 3, 'and': 1, 'preferably': 1, 'only': 1, 'obv
ious': 2, 'way': 2, 'do': 2, 'it': 2, 'that': 1, 'may': 2, 'not': 1, 'at': 1, 'first': 1, "y
ou're": 1, 'dutch': 1, 'now': 2, 'often': 1, 'right': 1, 'if': 2, 'implementation': 2, 'har
d': 1, 'explain': 2, "it's": 1, 'a': 2, 'bad': 1, 'idea': 3, 'easy': 1, 'good': 1, 'namespac
es': 1, 'are': 1, 'honking': 1, 'great': 1, '': 1, "let's": 1, 'more': 1, 'those': 1}
```

In [4]:

```
import operator

zen_items = zen_map.items()
word_count_items = sorted(
    zen_items, key=operator.itemgetter(1), reverse=True
)

print(word_count_items[:3])
```

```
[('is', 10), ('better', 8), ('than', 8)]
```

In [5]:

```
from collections import Counter



cleaned_list = []
for word in zen.split():
    cleaned_list.append(word.strip('.,-!').lower())



print(Counter(cleaned_list).most_common(3))
```

```
[('is', 10), ('better', 8), ('than', 8)]
```