Variables as Functions

In [1]:
```
x = 5
```

In [2]:
```
def x():
    return 5
```

Viewing function data with __code__

In [3]:
```
print(x.__code__.co_varnames)
print(x.__code__.co_code)
```

```
()
b'd\x01S\x00'
```

Text processing in Python

In [4]:
```
text = '''
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 [5]:
```
def lowercase(text):
    return text.lower()

def removePunctuation(text):
    punctuations = ['.', '-', ',', '*']
    for punctuation in punctuations:
        text = text.replace(punctuation, '')
    return text

def removeNewlines(text):
    text = text.replace('\n', ' ')
    return text

def removeShortWords(text):
    return ' '.join([word for word in text.split() if len(word) > 3])

def removeLongWords(text):
    return ' '.join([word for word in text.split() if len(word) < 6])
```

In [7]:
```
processingFunctions = [lowercase, removePunctuation, removeNewlines, removeLong

for func in processingFunctions:
    text = func(text)

print(text)
```

```
is than ugly is than is than is than flat is than is than dense cases to brea
k the rules beats never pass in the face of the to guess there be one and onl
y one way to do it that way may not be at first dutch now is than never never
is often than right now if the is hard to it's a bad idea if the is easy to i
t may be a good idea are one great idea let's do more of
```

Lambda functions

In [9]:
```
2 + 3
```

Out[9]:
```
5
```

In [10]:
```
(lambda x: x + 3)(5)
```

Out[10]:
```
8
```

In [11]:
```
myList = [5,4,3,2]
sorted(myList)
```

Out[11]:
```
[2, 3, 4, 5]
```

In [13]:
```
myList = [{'num': 3}, {'num': 2}, {'num': 1}]
sorted(myList, key=lambda x: x['num'])
```

Out[13]:
```
[{'num': 1}, {'num': 2}, {'num': 3}]
```

In [ ]: