

I knew it! 'Chicken' comes first!

**CSE 111 Bio:
Program Design I
Lecture 21:
dictionaries**

Tanya Berger-Wolf (CS) & Boris Igić (Bio)
University of Illinois, Chicago
November 15, 2016

Generating DNA sequences

Probability of A = .4, prob of C = .2, prob of T = .3, prob of G = .1

```
def seqGen(probA, probC, probT, probG, len) :
    import random
    seq = ''
    for i in range(len) :
        num = random.uniform(0,1)
        if num < probA :
            seq = seq + 'a'
        elif num < probA + probC :
            seq = seq + 'c'
        elif num < probA + probC + probT :
            seq = seq + 't'
        else :
            seq = seq + 'g'
    return seq
```

Generating DNA sequences

Probability of A = .4, prob of C = .2, prob of T = .3, prob of G = .1

D = {'probA' : .4, 'probC' : .2, 'probT' : .3, 'probG' : .1}

```
def seqGen(D, len) :
    import random
    seq = ''
    for i in range(len) :
        num = random.uniform(0,1)
        if num < D['probA'] :
            seq = seq + 'a'
        elif num < D['probA'] + D['probC'] :
            seq = seq + 'c'
        elif num < D['probA'] + D['probC'] + D['probT'] :
            seq = seq + 't'
        else :
            seq = seq + 'g'
    return seq
```

Dictionaries!

```
>>> D = {}
```

Dictionaries!

the *key* the *value*
 >>> D = {}
 >>> D["spam"] = "a health food product"

Dictionaries!

the *key* the *value*
 >>> D = {}
 >>> D["spam"] = "a health food product"
 >>> D[42] = "an important number"
 >>> D["bart"] = ["bart@geemail.com", "springfield"]

Dictionaries!

the *key* the *value*
 >>> D = {}
 >>> D["spam"] = "a health food product"
 >>> D[42] = "an important number"
 >>> D["bart"] = ["bart@geemail.com", "springfield"]
 >>> D
 {42: 'an important number', 'bart': ['bart@geemail.com', 'springfield'], 'spam': 'a health food product'}

Dictionaries!

the *key* the *value*
 >>> D = {}
 >>> D["spam"] = "a health food product"
 >>> D[42] = "an important number"
 >>> D["bart"] = ["bart@geemail.com", "springfield"]
 >>> D
 {42: 'an important number', 'bart': ['bart@geemail.com', 'springfield'], 'spam': 'a health food product'}
 >>> D[42]
 'an important number'

Dictionaries!

```

the key      the value
>>> D = {}
>>> D["spam"] = "a health food product"
>>> D[42] = "an important number"
>>> D["bart"] = ["bart@geemail.com", "springfield"]
>>> D
{42: 'an important number', 'bart': ['bart@geemail.com', 'springfield'], 'spam': 'a health food product'}
>>> D[42]
'an important number'
>>> 42 in D
True
>>> 43 in D
False
>>> "ran" in D
False

```

Dictionaries!

```

the key      the value
>>> D = {}
>>> D["spam"] = "a health food product"
>>> D[42] = "an important number"
>>> D["bart"] = ["bart@geemail.com", "springfield"]
>>> D
{42: 'an important number', 'bart': ['bart@geemail.com', 'springfield'], 'spam': 'a health food product'}
>>> D[42]
'an important number'
>>> 42 in D
True
>>> 43 in D
False
>>> "ran" in D
False
>>> D["ran"]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
KeyError: 1

```

"an important number" in D
Python, you could be a bit more polite!



```

def translate():
    D = {"hello": "hola", "goodbye": "adios", "good": "bueno", "spam": "spam"}
    for word in D:
        print word, "is:", D[word]
    D["more"] = "mas"

```

>>> translate()

```

A. hello is: hola
   goodbye is: adios
   good is: bueno
   spam is: spam

B. good is: bueno
   hello is: hola
   goodbye is: adios
   spam is: spam

C. hello is: hola
   goodbye is: adios
   good is: bueno
   spam is: spam
   more is: mas

D. spam is: spam
   good is: bueno
   goodbye is: adios
   hello is: hola

```

```

def translate():
    D = {"hello": "hola", "goodbye": "adios", "good": "bueno", "spam": "spam"}
    for word in D:
        print word, "is:", D[word]
    D["more"] = "mas"

```

```

>>> translate()
good is: bueno
hello is: hola
goodbye is: adios
spam is: spam

```

The order is weird!



Try This...

```
scrabbleDict = {"a":1, "b":3, "c":3, "d":2, ...}
```

```
>>> score("spam", scrabbleDict)  
8
```

```
def score(myString, dict):
```



You can use
while or for
loops here...

Try This...

```
scrabbleDict = {"a":1, "b":3, "c":3, "d":2, ...}
```

```
>>> score("spam")  
8
```

```
def score(myString, dict):  
    total = 0  
    for letter in myString:  
        total += dict[letter]  
    return total
```