

Uge 5

Tirsdag

# Dagens tekst

Kode stump fra sidste uges øvelser

Gen forudsigelse i bakterier - stor opgave i små skidt

Relevante øvelser

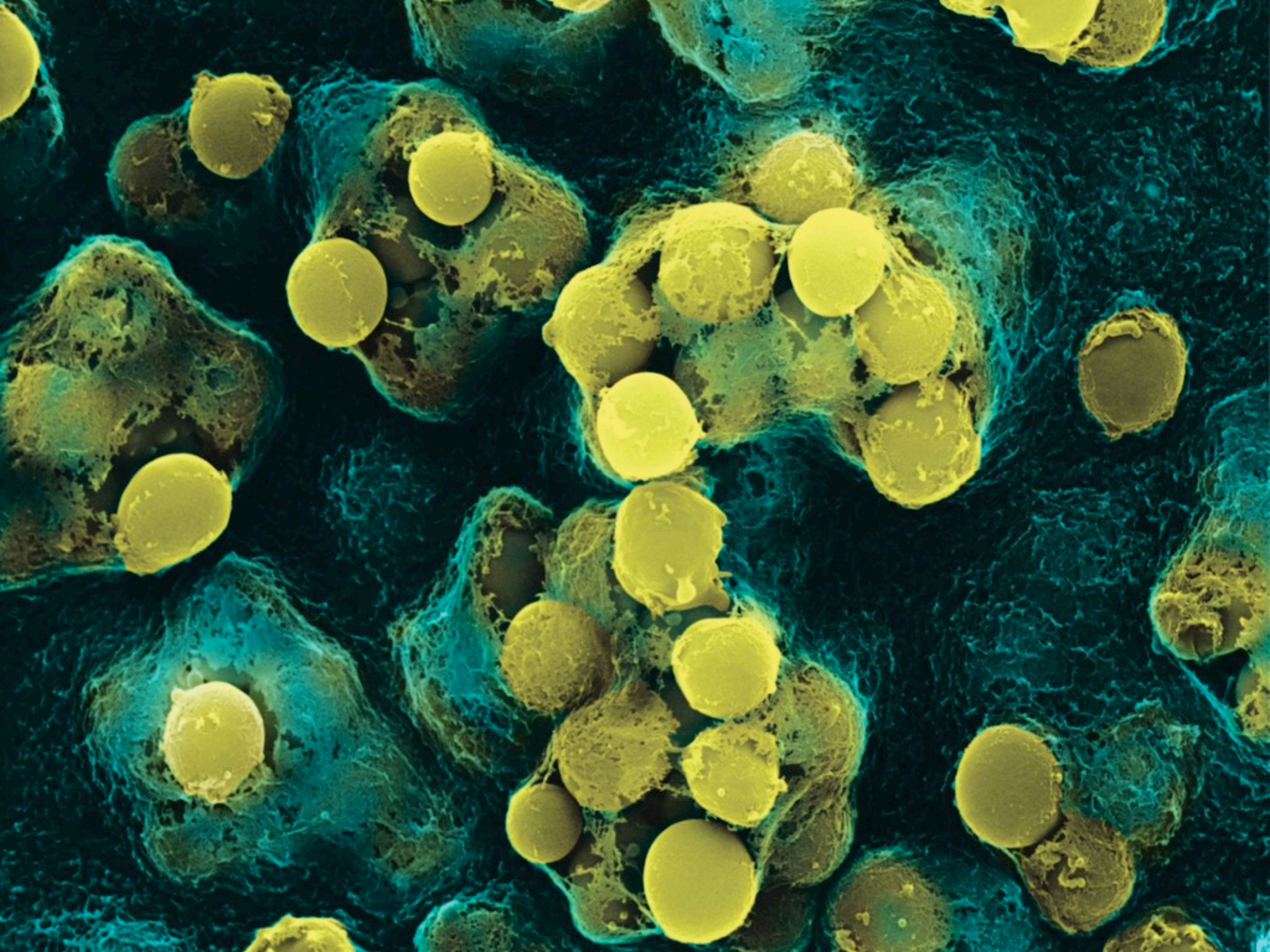
```
testSeqs = ["HIV-1.A ATGGGT", "HIV-1.B TGGAAG",
            "HIV-1.C GACTAA", "HIV-1.D GCGATT"]

def analyseHivSequences():
    hivFile = testSeqs
    statistics = {}
    for l in hivFile:
        name, seq = l.split()
        if name not in statistics:
            statistics[name] = {}
        for b in seq:
            if b not in statistics[name]:
                statistics[name][b] = 0
            statistics[name][b] += 1
    for name in statistics:
        print name
        total = sum(statistics[name].values())
        for b in statistics[name]:
            print "\t", b, statistics[name][b] / float(total)
```

Interactive version

<http://tinyurl.com/cs44e2t>







`findAndTranslateOpenReadingFrames(seq)`

findAndTranslateOpenReadingFrames(seq)



findOpenReadingFrames(seq)

findAndTranslateOpenReadingFrames(seq)



findOpenReadingFrames(seq)



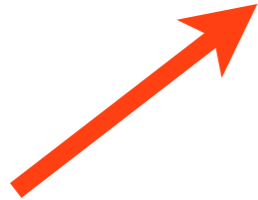
findStartPositions(seq)



findAndTranslateOpenReadingFrames(seq)



findOpenReadingFrames(seq)



findStartPositions(seq)



findNextStopCodon(seq, start)

findAndTranslateOpenReadingFrames(seq)



findOpenReadingFrames(seq)



findStartPositions(seq)



findNextStopCodon(seq, start)



findNextCodon(seq, start, codon)

findAndTranslateOpenReadingFrames(seq)

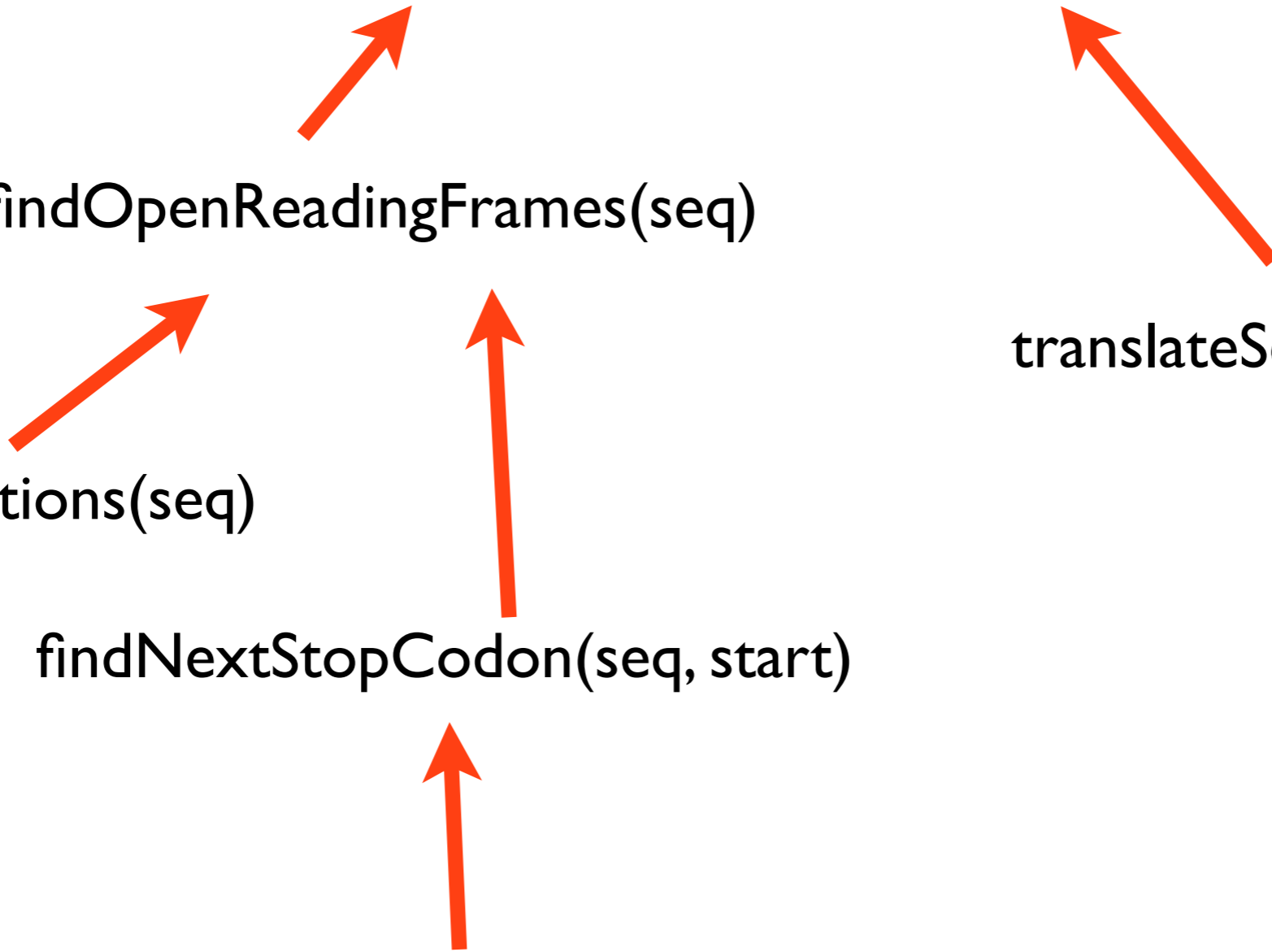
findOpenReadingFrames(seq)

translateSequence(orf)

findStartPositions(seq)

findNextStopCodon(seq, start)

findNextCodon(seq, start, codon)



findAndTranslateOpenReadingFrames(seq)

findOpenReadingFrames(seq)

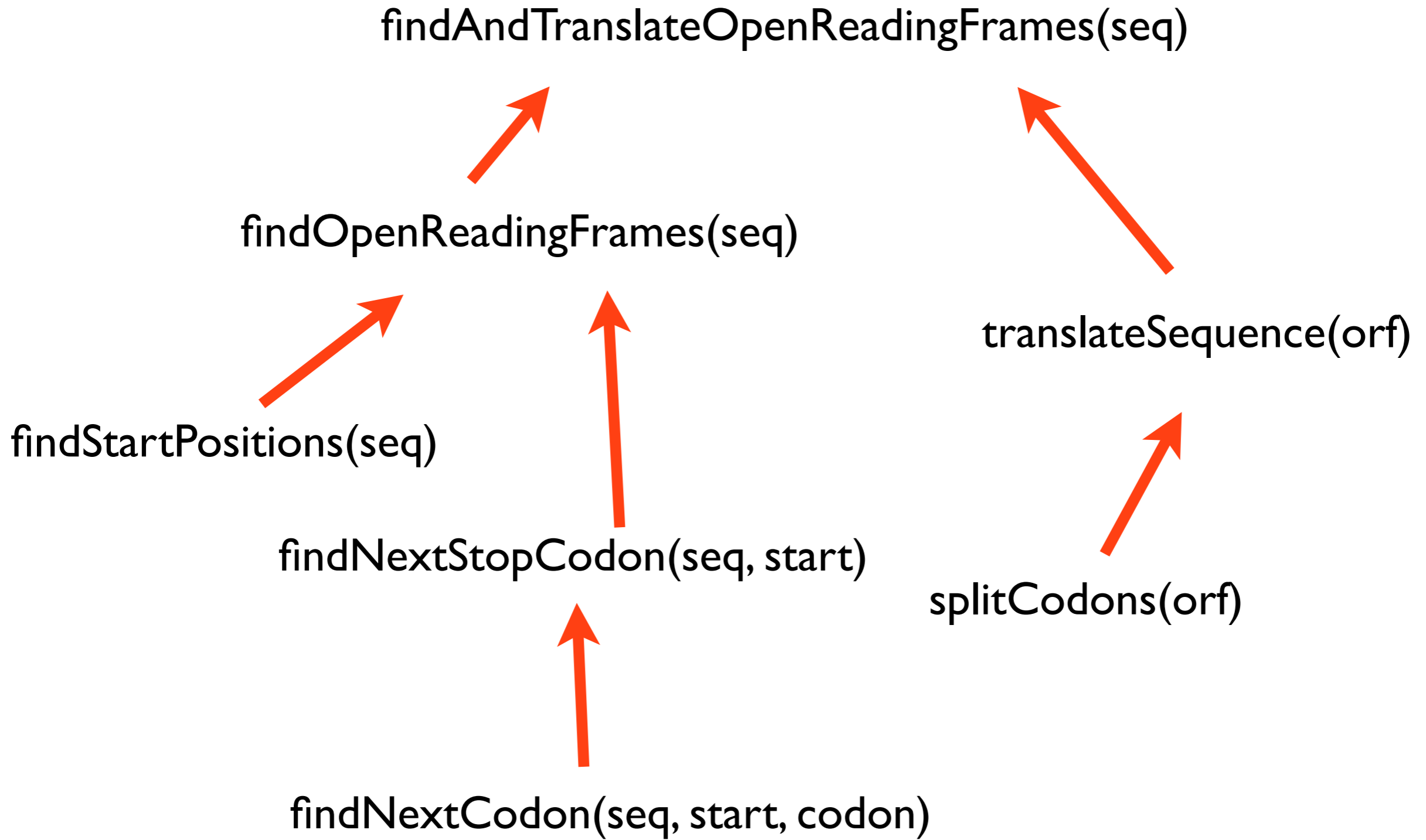
translateSequence(orf)

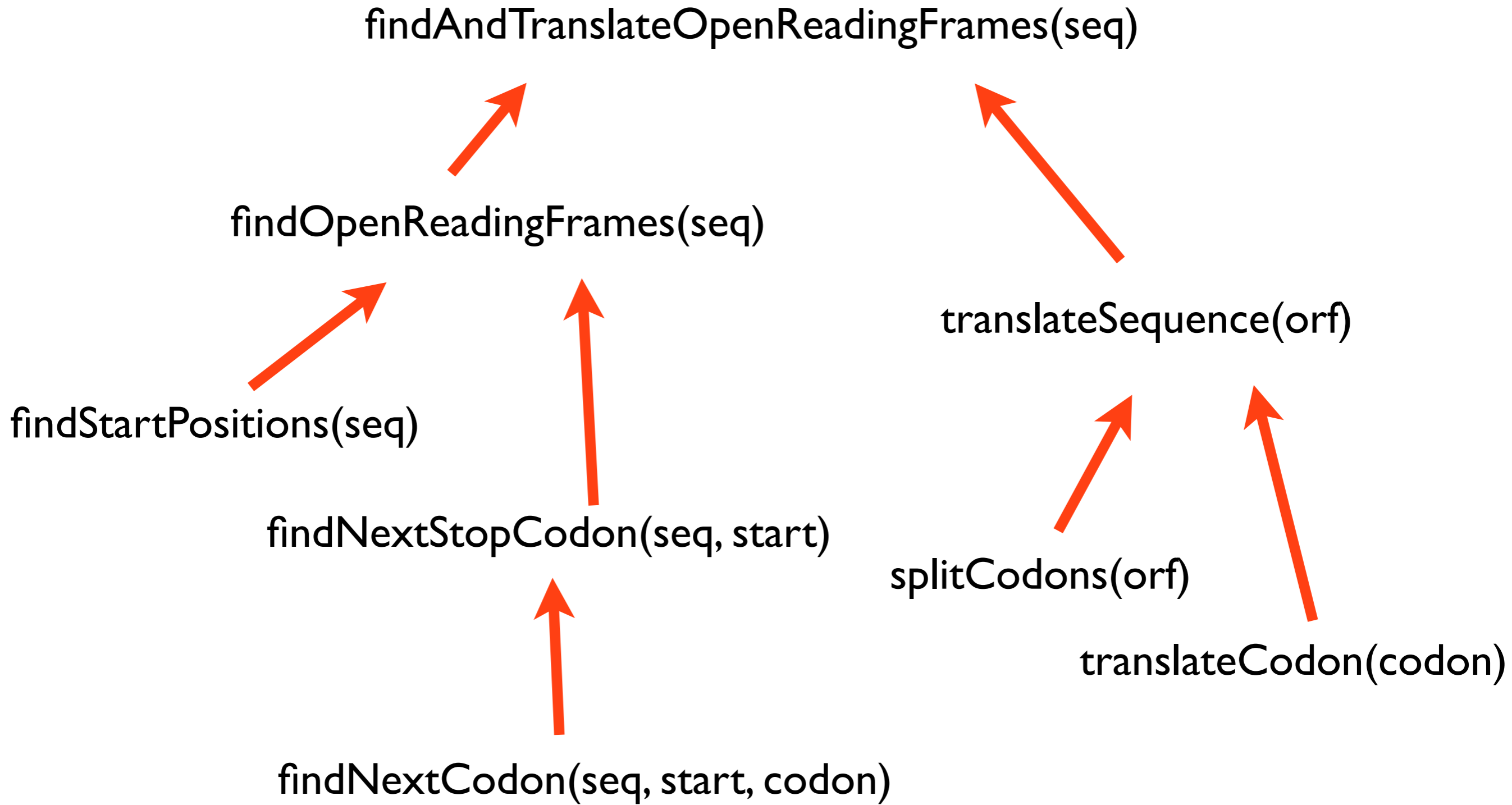
findStartPositions(seq)

findNextStopCodon(seq, start)

splitCodons(orf)

findNextCodon(seq, start, codon)







# Opgave

Lav et for loop der kan skrive alle overlappende tripletter i en streng ud.

F.eks. et loop over strengen "CCATGCTC" skal printe:

CCA

CAT

ATG

TGC

GCT

CTC

# Løsning

```
seq = "CCATGCTCTAGCC"  
for i in range(len(seq)-2):  
    print seq[i:i+3]
```

# Opgave

Lav et for loop der kan skrive alle på hinanden følgende tripletter i en streng ud.

F.eks. et loop over strengen "ATGCTCTAG" skal printe:

ATG

CTC

TAG

# Løsning

```
orf = "ATGCTCTAG"  
for i in range(0, len(orf), 3):  
    print orf[i:i+3]
```

# Opgave

Skriv en funktion der tager en liste af tal som argument og returnerer en liste af talpar hvor første element at et par første element er et tal fra liste argumentet og andet element er kvadratet på dette tal. Tag udgangspunkt i denne funktion:

```
def squaredNumbers(l):  
    result = []  
    for n in l:  
        result.append(n**2)  
    return result  
  
print squaredNumbers(range(4))
```



# Løsning

```
def squaredNumbers(l):  
    result = []  
    for n in l:  
        result.append((n, n**2))  
    return result
```

# Opgave

Skriv et for loop der løber over det talpar jeres funktion returnerer og skriver følgende ud:

```
the square of 0 is 0
the square of 1 is 1
the square of 2 is 4
the square of 3 is 9
osv...
```

# Løsning

```
def squaredNumbers(l):  
    result = []  
    for n in l:  
        result.append((n, n**2))  
    return result  
  
for number, squared in squaredNumbers(range(4)):  
    print "the square of", number, "is", squared
```

Uge 5

Torsdag

# Dagens tekst

Sidste uges aflevering (Fasta parsing)

List comprehensions

Streng formattering



# List comprehensions

```
lst = [1, 2, 3, 4]
```

```
squared = []  
for n in lst:  
    squared.append(n**2)
```

```
squared = [n**2 for n in lst]
```

# List comprehension eksempler

```
def even(lst):  
    res = []  
    for element in lst:  
        if element % 2 == 0:  
            res.append(element)  
    return res
```

```
lst = [1, 2, 3, 4]
```

```
evenLst = even(lst)
```

```
evenLst = [x for x in lst if x % 2 == 0]
```

# List comprehension eksempler

```
def differences(lst1, lst2):  
    res = []  
    for x, y in zip(lst1, lst2):  
        res.append(x-y)  
    return res
```

```
l1 = [8, 4, 5, 2]
```

```
l2 = [1, 2, 3, 4]
```

```
diffs = differences(l1, l2)
```

```
diffs = [x - y for x, y in zip(l1, l2)]
```

# Opgaver

```
l1 = [8, 4, 5, 2]
```

```
l2 = [1, 2, 3, 4]
```

Lav en liste af "squared differences" vha. en list comprehension.

Lav en liste med flg. strenge i upper case:

```
("make", "this", "uppercase")
```

# List comprehension eksempler

```
l1 = [8, 4, 5, 2]
l2 = [1, 2, 3, 4]
word_list = ("make", "this", "upper")

# differences
[x - y for x, y in zip(l1, l2)]

# squared differences
[(x - y)**2 for x, y in zip(l1, l2)]

# uppercase:
[s.upper() for s in word_list]
```

# List comprehension eksempler

```
# liste af amino syrer fra list af codons:
```

```
aaList = [translateCodon(cod) for cod in codonList]
```

```
# det samme, men her bruger vi splitCodons
```

```
# funktionen direkte:
```

```
aaList = [translateCodon(cod) for cod in splitCodons(orf)]
```

```
p = ''.join(translateCodon(cod) for cod in splitCodons(orf))
```

# String formatting

```
n = 4
```

```
print "the square of", n, "is", n**2
```

```
print "the square of %d is %d" % (n, n**2)
```

```
%s string
```

```
%d integer
```

```
%f float
```

# Streng formatting

```
"%s is %d years old" % ("He", 42)
```

```
"The result is: %f" % (8305.38)
```

```
"%e is as big as %.2e" % (100000)
```



# Opgave

$s = \text{"bananas"}$

$n = 7$

Lav følgende streng: "We have 7 bananas, 3.5 each"

# Løsning på opgave

```
s = "bananas"
```

```
n = 7
```

```
print "We have %d %s, %f each" % (n, s, n/2.0)
```