

Программирование на Python

Занятие 3. Цикл for. Строки.

Павел Федотов
fedotovp@gmail.com

Цикл for

```
for i in 2, 3, 5:  
    print(i * i)
```

Вывод:

4 9 25

(с переводами строк
между числами)

```
for i in range(10):  
    print(i * i)
```

Вывод:

0 1 4 9 16 25 36 49
64 81

(с переводами строк
между числами)

range

- range — функция, возвращающая последовательность
- Примеры
 - range(5): 0, 1, 2, 3, 4 — все числа от 0 до 5 (не включая 5) с шагом 1
 - range(2, 5): 2, 3, 4 — все числа от 2 до 5 (не включая 5) с шагом 1
 - range(2, 15, 4): 2, 6, 10, 14 — все числа от 2 до 15 (не включая 15) с шагом 4
- range(start=0, to, step=1)

Задача 1

- Вывести квадрат:

Задача 2

- Вывести треугольник

*

**

Задача 3

- <http://rosalind.info/problems/ini4/>
- **Conditions and Loops**
- Given: Two positive integers a and b ($a < b < 10000$).
- Return: The sum of all odd integers from a through b , inclusively.
- Sample Dataset
100 200
- Sample Output
7500

СИМВОЛЫ СТРОКИ

- genome = 'ATGG'
 - genome[0] A
 - genome[1] T
 - genome[2] G
 - genome[3] G
 - genome[-1] G
 - genome[-2] G
 - genome[-3] T
 - genome[-4] A
-
- i = 1
 - print(genome[i])
 - T

Строки неизменяемы

- `genome = 'ATGG'`
- `genome[1] = 'C'` – ошибка!

Перечисление символов строки с ПОМОЩЬЮ ИНДЕКСОВ

- ```
for i in range(4):
 print(genome[i])
```

- Вывод

A

T

G

G

# Перечисление символов строки

- Цикл `for` можно использовать для поочередного перечисления всех символов строки

- Пример:

```
contig = 'ATTG'
```

```
for c in contig:
```

```
 print(c)
```

- Вывод:

A

T

T

G

# Задача — цитозин

- Дан контиг. Вывести, сколько раз в нем встречается нуклеотид цитозин.
- Входные данные  
CACCTGGAC
- Выходные данные  
4
- Входные данные  
GATTACA
- Выходные данные  
1

# Решение — ЦИТОЗИН

```
contig = input()
```

```
cnt = 0
```

```
for nucl in contig:
```

```
 if nucl == 'C':
```

```
 cnt += 1
```

```
print(cnt)
```

```
contig = input()
```

```
print(contig.count('C'))
```

- Строки имеют методы
  - функция, которая применяется к данной строке
  - `s.count(p)` — сколько раз строка `p` встречается в строке `s`

# Задача — Counting DNA

- <http://rosalind.info/problems/dna/>
- Given: A DNA string  $s$  of length at most 1000 nt.
- Return: Four integers (separated by spaces) counting the respective number of times that the symbols 'A', 'C', 'G', and 'T' occur in  $s$ .
- Sample Dataset
- AGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAAAAGAGTGTCTGATAGCAGC
- Sample Output
- 20 12 17 21

# Задача — GC-контент

- Дан геном. Вычислите его GC-контент.
  - GC-контент — процент нуклеотидов G, C от общего числа нуклеотидов в последовательности.
- <http://rosalind.info/problems/gc/>

# Некоторые методы у строк

- `s = 'aTGcc' p = 'cc'`
  - `s.upper()` → 'ATGCC'
  - `s.lower()` → 'atgcc'
  - `s.count(p)` → 1 # сколько раз p встречается в s
  - `s.find(p)` → 3 # первое вхождение (индекс) p в s
  - `s.find('A')` → -1 # строка 'A' не входит в s
    - Проверка вхождения в строку: `if 'TG' in s: ...`
  - `s.replace('c', 'C')` → 'aTGCC' # заменяем все вхождения 'c' на 'C'

# Последовательные вызовы методов

- `s = 'agttcagtc'`
- `s.upper().count('gt'.upper())`
  - 2



# Slicing

- `dna = 'ATTCGGAGCT'`
- `dna[1] → 'T'`
- `dna[1:4] → 'TTC'`
- `dna[:4] → 'ATTC'`
- `dna[4:] → 'GGAGCT'`
- `dna[-4:] → 'AGCT'`
- `dna[1:-1] → 'TTCGGAGC'`
- `dna[1:-1:2] → 'TCGG'`
- `dna[::-1] → 'TCGAGGCTTA'` (символы в обратном порядке)

# Задача — палиндром

- Дан контиг. Проверить, является ли он палиндромом.
  - Строка является палиндромом, если читается в обоих направления одинаково
- Входные данные  
CAGGTGGAC
- Выходные данные  
YES
- Входные данные  
GATTACA
- Выходные данные  
NO

# Решение — палиндром

```
s = input()
i = 0
j = len(s) - 1
is_palindrom = True
while i < j:
 if s[i] != s[j]:
 is_palindrom = False
 i += 1
 j -= 1
if is_palindrom:
 print('YES')
else:
 print('NO')
```

```
s = input()
r = s[::-1]
if s == r:
 print('YES')
else:
 print('NO')
```

# Задача — Transcribing DNA into RNA

- <http://rosalind.info/problems/rna/>
- Given a DNA string  $t$  corresponding to a coding strand, its transcribed RNA string  $u$  is formed by replacing all occurrences of 'T' in  $t$  with 'U' in  $u$ .
- Given: A DNA string  $t$  having length at most 1000 nt.
- Return: The transcribed RNA string of  $t$ .
- Sample Dataset
- GATGGAACTTGACTACGTAAATT
- Sample Output
- GAUGGAACUUGACUACGUAAAUU

# Задача — Complementing a Strand of DNA

- <http://rosalind.info/problems/revc/>
- The reverse complement of a DNA string  $s$  is the string  $sc$  formed by reversing the symbols of  $s$ , then taking the complement of each symbol (e.g., the reverse complement of "GTCA" is "TGAC").
- Given: A DNA string  $s$  of length at most 1000 bp.
- Return: The reverse complement  $sc$  of  $s$ .
- Sample Dataset
- AAAACCCGGT
- Sample Output
- ACCGGGTTTT

# Задача — Restriction fragment lengths

- Here's a short DNA sequence:

**ACTGATCGATTACGTATAGTAGAATTCTATCAT  
ACATATATATCGATGCGTTCAT**

- The sequence contains a recognition site for the EcoRI restriction enzyme, which cuts at the motif **G\*AATTC** (the position of the cut is indicated by an asterisk). Write a program which will calculate the size of the two fragments that will be produced when the DNA sequence is digested with EcoRI.

# Задача - Strings and Lists

- **Given:** A string `s` of length at most 200 letters and four integers `a`, `b`, `c` and `d`.
- **Return:** The slice of this string from indices `a` through `b` and `c` through `d` (with space in between), inclusively.
- **Sample Dataset**
- HumptyDumptysatonaWallHumptyDumptyhadagreatfallAll  
theKingshorsesandalltheKingsmenCouldntputHumptyDu  
mptyinhisplaceagain.
- 22 27 97 102
- **Sample Output**
- Humpty Dumpty

# ССЫЛКИ

- Официальная документация:
  - <http://docs.python.org>
  - <https://docs.python.org/3/tutorial>
- Презентации, примеры
  - <http://pythonforbiologists.com/index.php/introduction-to-python-for-biologists/2-printing-and-manipulating-text/>
  - <http://software-carpentry.org/v4/python/>
  - [http://nbviewer.ipython.org/github/ehmatthes/intro\\_programming/blob/master/notebooks/lists\\_tuples.ipynb](http://nbviewer.ipython.org/github/ehmatthes/intro_programming/blob/master/notebooks/lists_tuples.ipynb)
- Online-курсы, задачи:
  - <http://www.codecademy.com/tracks/python>
  - <http://informatics.mccme.ru>