



Programming for Data Science: Nested loop

Instructor: Jakramate Bootkrajang



Outlines

- Loop refresher
- Nested loop
 - Nested for loop
 - Nested while loop
 - Mix of for loop and while loop

Loop refresher

- To execute a block of codes multiple times we use iteration control structure
- For loop

```
for [iterating variable] in [sequence]:  
    [do something].
```

- While loop

```
while [boolean expression == True]:  
    [do something].
```



When to use for / while

- For loop get used when we know the number of repetitions beforehand
- While loop get used when we do not know how many times we should repeat the block of codes



Nested loop

- It is possible for a loop to contain loops
- We call loop(s) in loop a nested loop

DNA sequence

- We've learn from previous lab session that DNA sequence is usually represented by string
 - ATCGGATA
- To process a sequence, for example to count the number of A's, we could loop over the string

```
seq = "ATCGGATC"  
cnt = 0  
for base in seq:  
    if base == "A":  
        cnt += 1
```

DNA sequence [2]

- This works for one input sequence
- But if we want to count A's in multiple sequences we need another repetition

```
with open("sequence.txt") as fin
for seq in fin:
    cnt = 0
    for base in seq:
        if base == "A":
            cnt += 1
```

Nested for loop

```
for [loop variable] in [sequence]:  
    [some statements]  
    for [another loop variable] in [another sequence]:  
        [some statements.]
```

- Be careful about the indentation
- Nested loop is not limited to two levels
- Deeper nested loop is possible

Example

- Total number of times loops get executed is the multiplication of number of times each loop runs.

```
for i in range(3):  
    for j in range(2):  
        print("six")
```

```
six  
six  
six  
six  
six  
six  
six
```

Nested while loop

- Similar to for loop, while loop can be nested

```
while [boolean expression]:  
    [some statements]  
    while [another boolean expression]:  
        [some statements]
```

- Be careful about the indentation

Mixing for and while

```
for [loop variable] in [sequence]:  
    [some statements]  
    while [boolean expression]:  
        [some statements]
```

```
while [boolean expression]:  
    [some statements]  
    for [loop variable] in [sequence]:  
        [some statements]
```

Common nested loop bug

- Accidentally type wrong looping variable

```
for i in range(10):  
    for i in range(5):  
        [some statments.]
```



Finding primes

- Prime number is a number that has exactly two divisors: 1 and itself
- Finding prime is not an easy task
 - Similarly checking if a number is prime is also difficult (takes a lot of time)
- We will study an algorithm for finding prime called Sieve of Eratosthenes

Sieve of Erastosthenes

- Ancient algorithm for finding prime numbers from 2 to upto any given limit
- Main idea
 - Create a list of number from 2 to n
 - Iteratively mark the multiple of unmarked number starting from 2 and so on
 - Stop the marking when the square of current number is greater than n
 - The remaining unmarked numbers are primes

Finding primes from 2 to 49

Step 1: Numbers from 2 ... 49

	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

Step 2: Eliminated multiples of 2

	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

Primes:
2

Finding primes from 2 to 49

[2]

Step 3: Eliminated multiples of 3

	2	3	4	5	6	7	Primes:
8	9	10	11	12	13	14	2, 3
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	31	32	33	34	35	
36	37	38	39	40	41	42	
43	44	45	46	47	48	49	

Step 4: Eliminated multiples of 5

	2	3	4	5	6	7	Primes:
8	9	10	11	12	13	14	2, 3, 5
15	16	17	18	19	20	21	
22	23	24	25	26	27	28	
29	30	31	32	33	34	35	
36	37	38	39	40	41	42	
43	44	45	46	47	48	49	

Finding primes from 2 to 49

[3]

Step 5: Eliminated multiples of 7

	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

Primes:
2, 3, 5, 7

Step 6: Remaining are prime.

	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31	32	33	34	35
36	37	38	39	40	41	42
43	44	45	46	47	48	49

Primes:
2, 3, 5, 7,
11, 13, 17,
19, 23, 29,
31, 37, 41,
43, 47

Pseudocode

Input: an integer $n > 1$.

Let A be an array of **Boolean** values, indexed by integers 2 to n , initially all set to **true**.

```
for  $i = 2, 3, 4, \dots$ , not exceeding  $\sqrt{n}$ :  
  if  $A[i]$  is true:  
    for  $j = i^2, i^2+i, i^2+2i, i^2+3i, \dots$ , not exceeding  $n$ :  
       $A[j] := \text{false}$ .
```

Output: all i such that $A[i]$ is true.

Exercise

- How many times does the following loop run ?

```
for i in range(4):  
    for j in range(i):  
        print("hi")
```

Exercise

- Create a nested loop that produces the following pattern

```
1
22
333
4444
55555
```