

Python Syntactic Sugar — Practice

CSCI 40

Fall 2026

Total Score: _____/4

LLM evaluation

2026-08-26

Re-evaluation pending after quiz-bank corrections.

Python Syntactic Sugar

Note. Syntactic sugar is a way for programming languages to make certain common operations more convenient. Python is famous for having a lot of syntactic sugar that can making learning the language hard. All of the “idioms” you will find in this packet do not allow you to write “more powerful” programs, but they will allow you to write programs that have fewer lines of code to accomplish the same task.

You will never be required in this class to use any particular syntactic sugar to accomplish a task. But these idioms are very widely used in python, and so you need to understand them in order to understand the code that other people (or AIs) write.

1 Sequence Unpacking

Note. We can place a sequence of assignment targets on the left-hand side of an equals sign as long as:

1. every item on the left is an assignable target (`SyntaxError` if violated),
2. the value on the right is iterable (`TypeError` if violated), and
3. it supplies exactly as many values as there are targets (`ValueError` if violated).

Problem 1. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 [a, b] = [1, 2]
2 print('a=', a)
```

Problem 2. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 [a, b, c] = [1, 2]
2 print('a=', a)
```

Problem 3. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 [a, b] = 12
2 print('a=', a)
```

Problem 4. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 [a, b] = '12'
2 print('a=', a)
```

Note. Recall that in python, if you have commas separating values without square brackets `[]` or curly braces `{}`, then this creates a *tuple*. (Tuples behave like immutable lists.) A common use case for tuples is sequence unpacking.

Problem 5. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 a, b = [1, 2]
2 print('a=', a)
```

Problem 6. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 a = 10
2 b = 5
3 a, b = b, a
4 print('a=', a)
```

Problem 7. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 a = 5
2 b = 10
3 for i in range(2, 5):
4     a, b = b, a
5 print('a=', a)
```

Note. It is common to combine sequence unpacking with for loops.

Problem 8. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 total = 0
2 for a, b in [(1, 2), (3, 4)]:
3     total += a
4 print('total=', total)
```

Problem 9. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 total = 0
2 for a, b in [(1, 2), (3, 4), 5]:
3     total += a
4 print('total=', total)
```

Problem 10. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 total = 0
2 for a, b in [(1, 2), (3, 4, 5)]:
3     total += a
4 print('total=', total)
```

2 Special Functions

Note. There are a number of special built-in functions for making iteration more convenient.

1. The `zip` function returns a lazy iterator of tuples drawn from two or more iterables. It stops when the shortest input is exhausted.

```
1 >>> list(zip('hello', 'world'))
2 [('h', 'w'), ('e', 'o'), ('l', 'r'), ('l', 'l'), ('o', 'd')]
3 >>> list(zip('hello', 'world', 'hola', 'mundo'))
4 [('h', 'w', 'h', 'm'), ('e', 'o', 'o', 'u'), ('l', 'r', 'l', 'n'), ('l', 'l', 'a', 'd')]
5 >>> list(zip(range(1000), 'world'))
6 [(0, 'w'), (1, 'o'), (2, 'r'), (3, 'l'), (4, 'd')]
```

2. The `enumerate` function returns a lazy iterator of tuples whose first item is an index and whose second item is a value from the original iterable. It is similar to combining `zip` and `range`.

```
1 >>> list(enumerate('hello'))
2 [(0, 'h'), (1, 'e'), (2, 'l'), (3, 'l'), (4, 'o')]
3 >>> enumerate('hello')
4 <enumerate object at 0x7f4521f6c220>
```

(Notice that I use the `list` function above only to consume the lazy iterator and display all of its values.)

Problem 11. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 result = ''
2 for a, b in zip('hello', 'world'):
3     result += a
4     result += b
5 print('result=', result)
```

Problem 12. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 names = ['alice', 'bob', 'charlie', 'dave', 'eve']
2 accumulator = []
3 for i, name in enumerate(names):
4     text = name + ' is number ' + str(i)
5     accumulator.append(text)
6 print('accumulator[3]=', accumulator[3])
```

Problem 13. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 names = ['alice', 'bob', 'charlie', 'dave', 'eve']
2 accumulator = []
3 for i,name in enumerate(names):
4     text = 'number ' + str(i) + ' is ' + name
5     accumulator.append(text)
6 print('accumulator[-1]=', accumulator[-1])
```

Problem 14. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 sentence = 'python is *weird*'
2 accumulator = ''
3 for i,char in enumerate(sentence):
4     if char == '*' and sentence[i-1] == ' ':
5         accumulator += '_'
6     elif char == '*':
7         accumulator += '!'
8     else:
9         accumulator += char
10 print('accumulator=', accumulator)
```

3 Comprehensions

Note. Any list or dictionary that contains the keyword `for` inside of it is a *comprehension*. It is very easy to get `IndexErrors` or `TypeError`s inside a list comprehension, and the error messages can be hard to debug. (Expect me to try to trick you on the quiz this way...)

Problem 15. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 names = ['alice', 'bob', 'charlie', 'dave', 'eve']
2 greetings = ['hello ' + name for name in names]
3 greeting = greetings[2]
4 print('greeting=', greeting)
```

Problem 16. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 xs = [ x*x for x in range(10) ]
2 num = xs[5]
3 print('num=', num)
```

Problem 17. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 xs = [ x*x for x in range(10) if x%2 ]
2 num = xs[3]
3 print('num=', num)
```

Problem 18. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 xs = [ x**x for x in range(-3,3) ]
2 num = xs[5]
3 print('num=', num)
```

Problem 19. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 xs = [ x for x in range(-3,3) if x ]
2 num = xs[3]
3 print('num=', num)
```

Problem 20. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 sentence = "This is an example sentence with a few words in it."
2 small_words = [ word.lower() for word in sentence.split() if len(word) <= 2]
3 print('len(small_words)=', len(small_words))
```

Problem 21. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 sentence = "This is an example sentence with a few words in it."
2 small_words = [ word.lower() for word in sentence.split() if word <= 2]
3 print('len(small_words)=', len(small_words))
```

Problem 22. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 sentence = "This is an example sentence with a few words in it."
2 small_words = [ word.lower() for word in sentence if len(word) <= 2]
3 print('len(small_words)=', len(small_words))
```

Problem 23. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 names = ['alice', 'bob', 'charlie', 'dave', 'eve']
2 greetings = [ name[0].upper() + name[1:].lower() for name in names ]
3 greeting = greetings[1]
4 print('greeting=', greeting)
```

Problem 24. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 names = ['alice', 'bob', 'charlie', 'dave', 'eve']
2 greetings = [ names[0].upper() + names[1:].lower() for name in names ]
3 greeting = greetings[1]
4 print('greeting=', greeting)
```

Problem 25. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 sentence = "This is an example sentence with a few words in it."
2 words = [ word.lower() for word in sentence.split() if 't' in word ]
3 print('len(words)=', len(words))
```

Problem 26. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 sentence = "This is an example sentence with a few words in it."
2 xs = [ x.upper() for x in sentence if 't' in x ]
3 print('xs[1]=', xs[1])
```

Problem 27. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 sentence = "This is an example sentence with a few words in it."
2 small_words = [ word.lower() for word in sentence.split() if len(word) <= 2]
3 print('len(small_words)=', len(small_words))
```

Problem 28. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 xss = [ [ i for i in range(x) ] for x in [2, 3, 4] ]
2 x = xss[-1][-2]
3 print('x=', x)
```

Problem 29. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 xss = [ [ i for i in range(x) ] for x in [2, 3, 4] if x%2 == 0 ]
2 x = xss[-1][-2]
3 print('x=', x)
```

Problem 30. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 xss = [ [ i for i in range(x) ] for x in [2, 3, 4] if x%2 == 0 ]
2 x = xss[-1:][-2][1]
3 print('x=', x)
```

Problem 31. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 xss = [ [ i for i in range(x) ] for x in [2, 3, 4] if x%2 == 0 ]
2 x = xss[-1][-2:][1]
3 print('x=', x)
```

Problem 32. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 xss = [ [ i for i in range(x) if i%3== 1 ] for x in [4, 5, 6] if x%2 == 1 ]
2 x = xss[-1][-1]
3 print('x=', x)
```

Problem 33. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 xss = [ [ i for i in range(x) if x%3== 1 ] for x in [4, 5, 6] if x%2 == 1 ]
2 print('xss=', xss)
```

Problem 34. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 xss = [ i for x in [2, 3, 4] if x%2 == 0 for i in range(x) ]
2 x = xss[-1][-2]
3 print('x=', x)
```

Problem 35. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 xss = [ i for x in [4, 5, 6] if x%2 == 1 for i in range(x) if i%3== 1 ]
2 x = xss[-1]
3 print('x=', x)
```

Problem 36. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 xss = [ i for x in [4, 5, 6] if x%2 == 1 for i in range(x) if x%3== 1 ]
2 print('xss=', xss)
```

4 Comprehensions on Twitter Data

Note. Comprehensions are commonly used for processing datasets. The examples below provide additional practice filtering a list of dictionaries.

Problem 37. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 tweets = [  
2     { 'source': 'Twitter Web Client'  
3       , 'text': ('From Donald Trump: Wishing everyone a wonderful holiday & '  
4                 'a happy, healthy, prosperous New Year. Let\u2019s think like '  
5                 'champions in 2010!')  
6       , 'retweet_count': 28  
7     },  
8     { 'source': 'Twitter Web Client'  
9       , 'text': ('Trump International Tower in Chicago ranked 6th tallest '  
10                'building in world by Council on Tall Buildings & Urban Habitat '  
11                'http://bit.ly/sqvQq')  
12       , 'retweet_count': 33  
13     },  
14     { 'source': 'Twitter Web Client'  
15       , 'text': 'Wishing you and yours a very Happy and Bountiful Thanksgiving!'  
16       , 'retweet_count': 13  
17     },  
18     { 'source': 'Twitter for iPhone'  
19       , 'text': ('RT @realDonaldTrump: Happy Birthday @DonaldJTrumpJr!\n'  
20                 'https://t.co/uRxyCD3hBz')  
21       , 'retweet_count': 9529  
22     },  
23     { 'source': 'Twitter for iPhone'  
24       , 'text': ('Happy Birthday @DonaldJTrumpJr!\n'  
25                 'https://t.co/uRxyCD3hBz')  
26       , 'retweet_count': 9529  
27     },  
28     { 'source': 'Twitter for Android'  
29       , 'text': ('Happy New Year to all, including to my many enemies and '  
30                 'those who have fought me and lost so badly they just don\'t '  
31                 'know what to do. Love!')  
32       , 'retweet_count': 141853  
33     },  
34     { 'source': 'Twitter for Android'  
35       , 'text': ('Russians are playing @CNN and @NBCNews for such fools - '  
36                 'funny to watch, they don\'t have a clue! @FoxNews totally '  
37                 'gets it!')  
38       , 'retweet_count': 23213  
39     },  
40     { 'source': 'Twitter for iPhone'  
41       , 'text': ('Join @American32, founded by Hall of Fame legend '  
42                 '@JimBrownNFL32 on 1/19/2017 in Washington, D.C.\u2026 '  
43                 'https://t.co/9WJZ8iTCQV')  
44       , 'retweet_count': 7366  
45     }]  
46  
47 trump_tweets = [tweet for tweet in tweets if 'trump' in tweet['text'].lower()]  
48 print('len(trump_tweets)=', len(trump_tweets))
```

Problem 38. The following code (circle one)

terminates without error

throws an exception

runs forever

If the code terminates without error, write the output. If the code throws an exception, state the exception.

```
1 tweets = [  
2     { 'source': 'Twitter Web Client'  
3       , 'text': ('From Donald Trump: Wishing everyone a wonderful holiday & '  
4                 'a happy, healthy, prosperous New Year. Let\u2019s think like '  
5                 'champions in 2010!')  
6       , 'retweet_count': 28  
7     },  
8     { 'source': 'Twitter Web Client'  
9       , 'text': ('Trump International Tower in Chicago ranked 6th tallest '  
10                'building in world by Council on Tall Buildings & Urban Habitat '  
11                'http://bit.ly/sqvQq')  
12       , 'retweet_count': 33  
13     },  
14     { 'source': 'Twitter Web Client'  
15       , 'text': 'Wishing you and yours a very Happy and Bountiful Thanksgiving!'  
16       , 'retweet_count': 13  
17     },  
18     { 'source': 'Twitter for iPhone'  
19       , 'text': ('RT @realDonaldTrump: Happy Birthday @DonaldJTrumpJr!\n'  
20                 'https://t.co/uRxyCD3hBz')  
21       , 'retweet_count': 9529  
22     },  
23     { 'source': 'Twitter for iPhone'  
24       , 'text': ('Happy Birthday @DonaldJTrumpJr!\n'  
25                 'https://t.co/uRxyCD3hBz')  
26       , 'retweet_count': 9529  
27     },  
28     { 'source': 'Twitter for Android'  
29       , 'text': ('Happy New Year to all, including to my many enemies and '  
30                 'those who have fought me and lost so badly they just don\'t '  
31                 'know what to do. Love!')  
32       , 'retweet_count': 141853  
33     },  
34     { 'source': 'Twitter for Android'  
35       , 'text': ('Russians are playing @CNN and @NBCNews for such fools - '  
36                 'funny to watch, they don\'t have a clue! @FoxNews totally '  
37                 'gets it!')  
38       , 'retweet_count': 23213  
39     },  
40     { 'source': 'Twitter for iPhone'  
41       , 'text': ('Join @AmerIcan32, founded by Hall of Fame legend '  
42                 '@JimBrownNFL32 on 1/19/2017 in Washington, D.C.\u2026 '  
43                 'https://t.co/9WJZ8iTCQV')  
44       , 'retweet_count': 7366  
45     }]  
46  
47 popular_tweets = [tweet for tweet in tweets if tweet['retweet_count'] > 100]  
48 print('len(popular_tweets)=', len(popular_tweets))
```