파이썬 문법 학습

https://realpython.com/python-dicts/

- 검색 realpython dict

딕셔너리 학습

Dictionaries in Python – Real Py

realpython.com/python-dicts/

Real Python | Start Here | Learn Python | Store | More | Join

Stuck at home? Enjoy free courses, on us →

- If the key is already present in `d`, the corresponding value in `d` for that key is updated to the value from `<obj>`.

Here is an example showing two dictionaries merged together:

Python >>>

```
>>> d1 = {'a': 10, 'b': 20, 'c': 30}
>>> d2 = {'b': 200, 'd': 400}

>>> d1.update(d2)
>>> d1
{'a': 10, 'b': 200, 'c': 30, 'd': 400}
```

In this example, key `'b'` already exists in `d1`, so its value is updated to `200`, the value for that key from `d2`. However, there is no key `'d'` in `d1`, so that key-value pair is added from `d2`.

`<obj>` may also be a sequence of key-value pairs, similar to when the `dict()` function is used to define a dictionary. For example, `<obj>` can be specified as a list of tuples:

Python >>>

```
>>> d1 = {'a': 10, 'b': 20, 'c': 30}
>>> d1.update([('b', 200), ('d', 400)])
>>> d1
{'a': 10, 'b': 200, 'c': 30, 'd': 400}
```

Or the values to merge can be specified as a list of keyword arguments:

Python >>>

Table of Contents

- Defining a Dictionary
- Accessing Dictionary Values
- Dictionary Keys vs. List Indices
- Building a Dictionary Incrementally
- Restrictions on Dictionary Keys
- Restrictions on Dictionary Values
- Operators and Built-in Functions
- → Built-in Dictionary Methods
- Conclusion

Tweet | Share | Email

Recommended Video Course

Dictionaries in Python

High Quality Python Video Courses

Watch Now »

Help

https://srv.realpython.net/click/64754707389/?c=31941813441&p=58946116052&r=81003

딕셔너리 종료

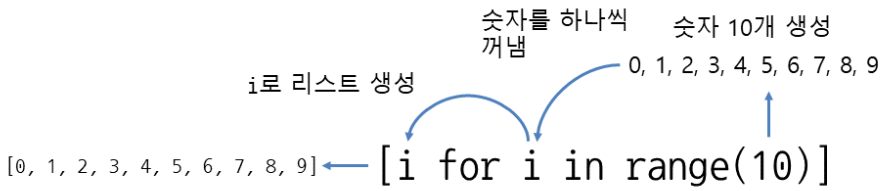
- 학습 후 퀴즈

컴프리헨션:
내장, 내포, 축약, 해석

리스트, 셋, 사전

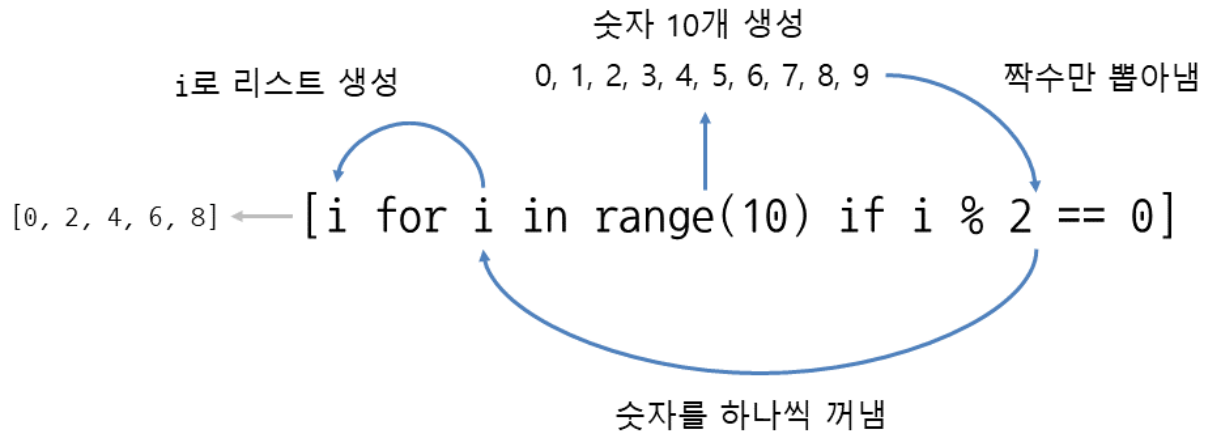
컴프리헨션(1)

- **국내 사이트**
  - 컴프리헨션 으로 검색
  - https://wikidocs.net/22805
  - https://doorbw.tistory.com/174
  - https://ddanggle.gitbooks.io/interpy-kr/ch15-comprehension.html
  - https://dojang.io/mod/page/view.php?id=2285



컴프리헨션(2)

- **조건식이 있는 내포**



컴프리헨션(3)

- https://realpython.com/list-comprehension-python/

When to Use a List Comprehension in Python

by James Timmins Nov 06, 2019 16 Comments basics python

Tweet Share Email

Table of Contents

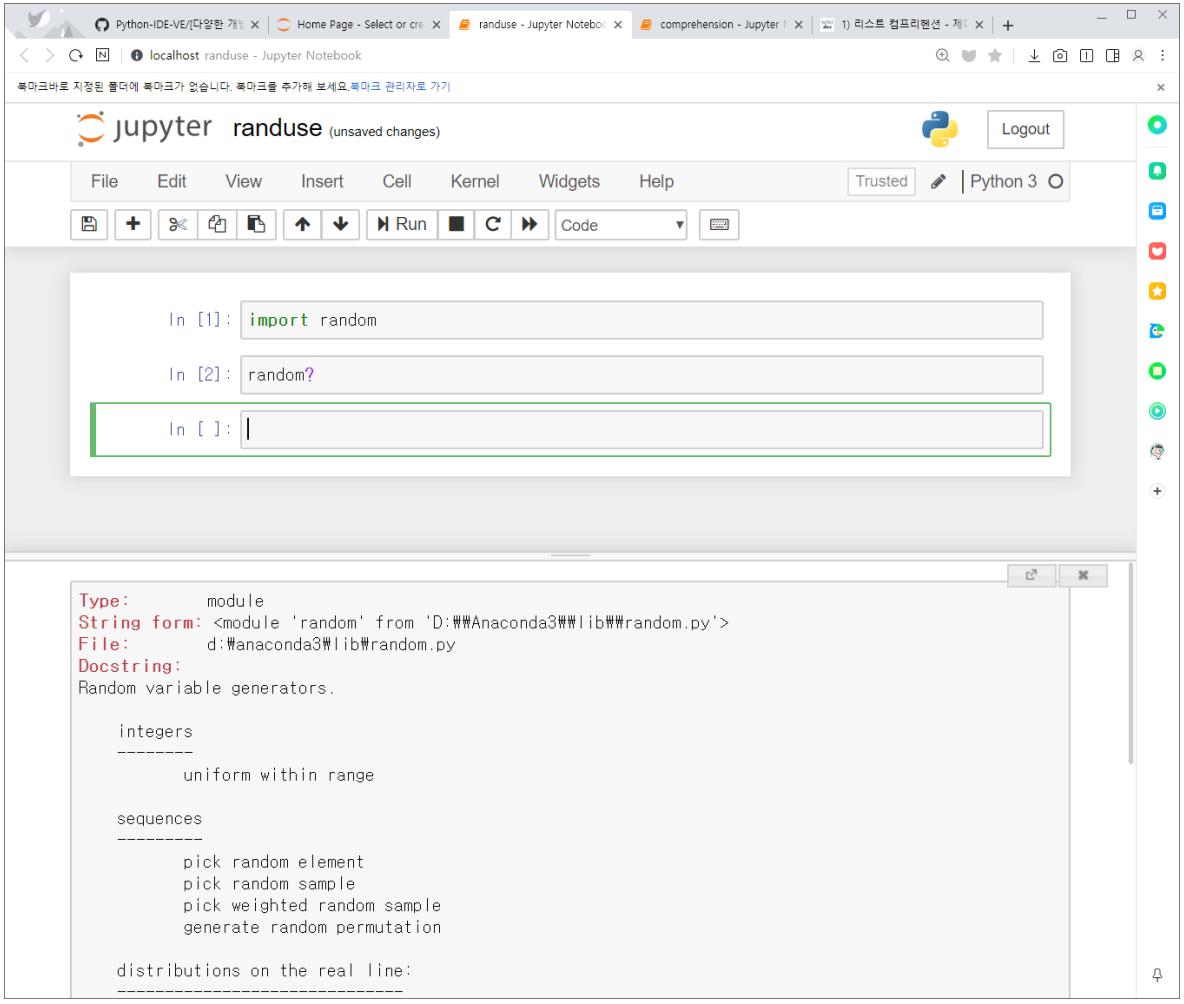
내포의 특징

- **파이썬스러운(pythonic) 코딩 방식**
- **효용성**
  - 한 번 알아두면 쉽게 코딩
  - 속도는 반복보다 빠름
    - **내장 함수 map() 보다는 느림**

모듈 random

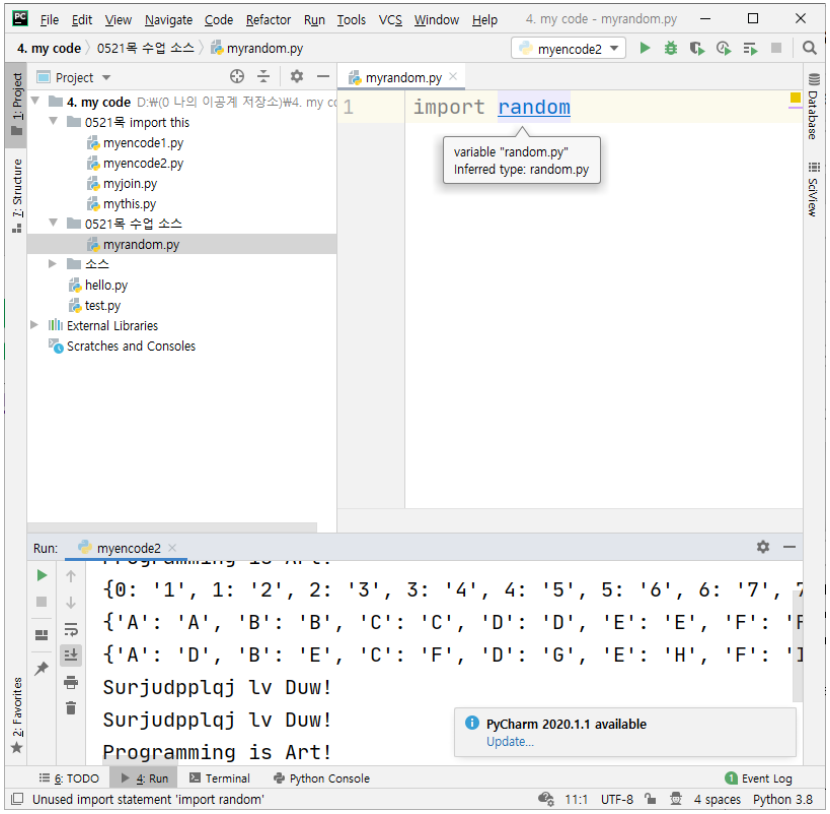
모듈 random

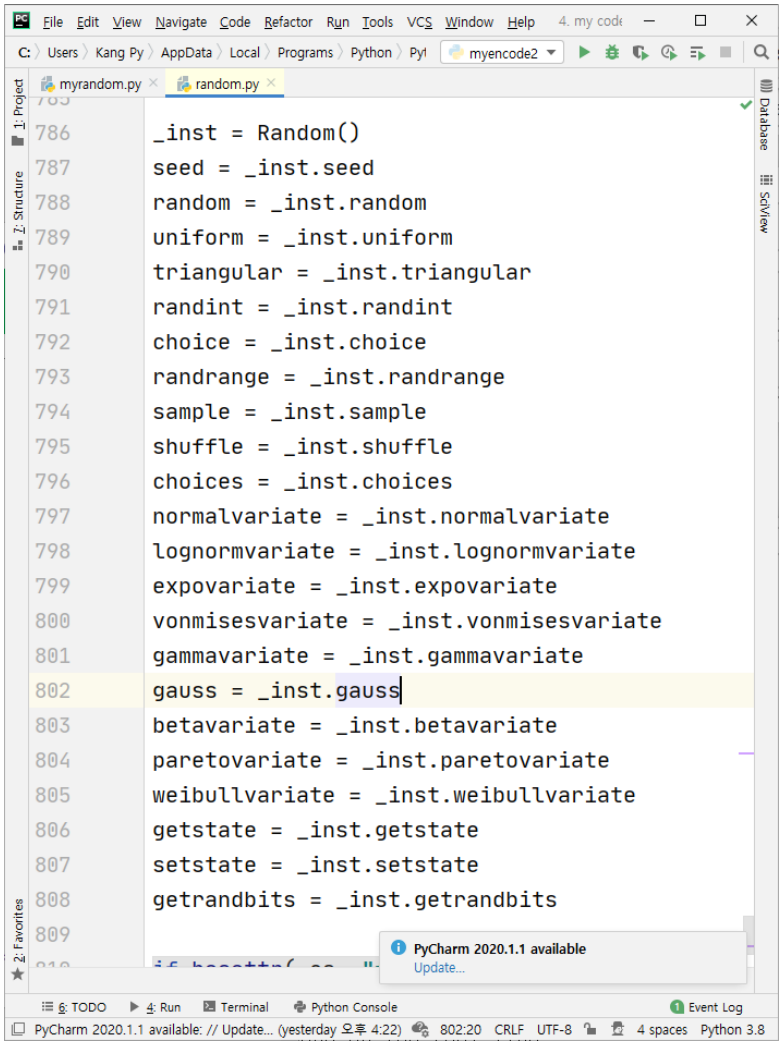
- **lib₩random.py**



모듈 random.py 소스

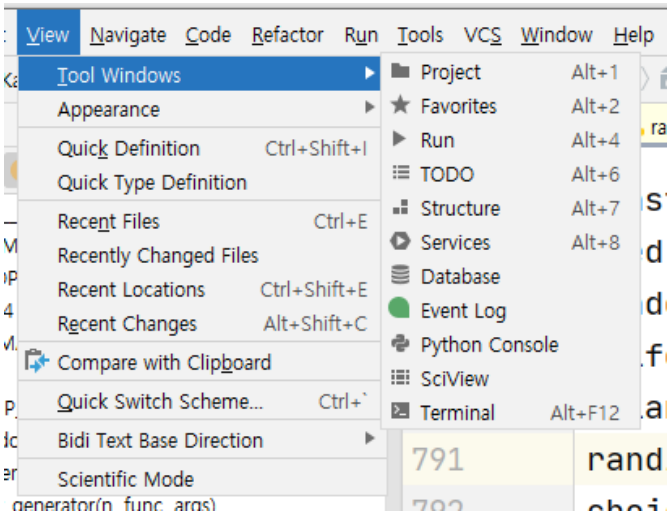
- **소스 바로 가기**
  - Ctrl + 마우스 클릭

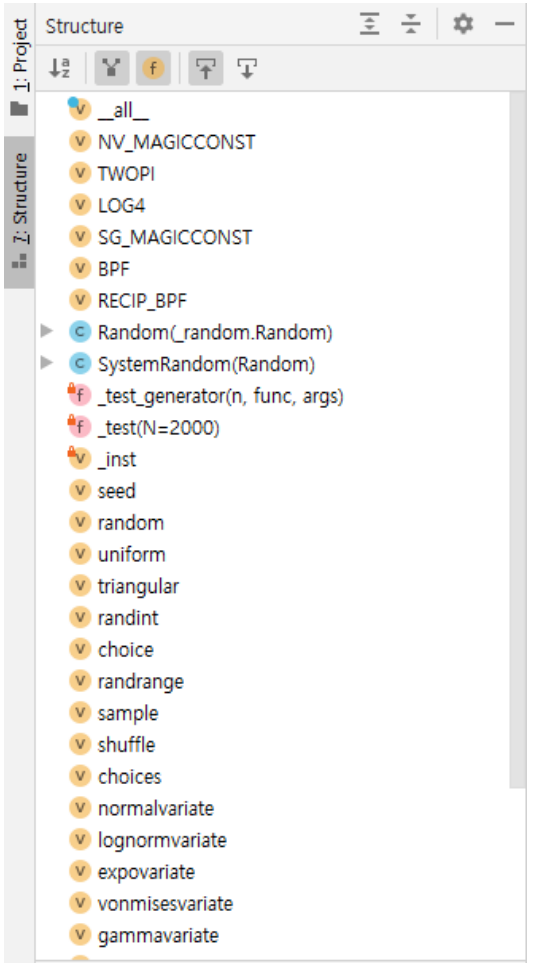




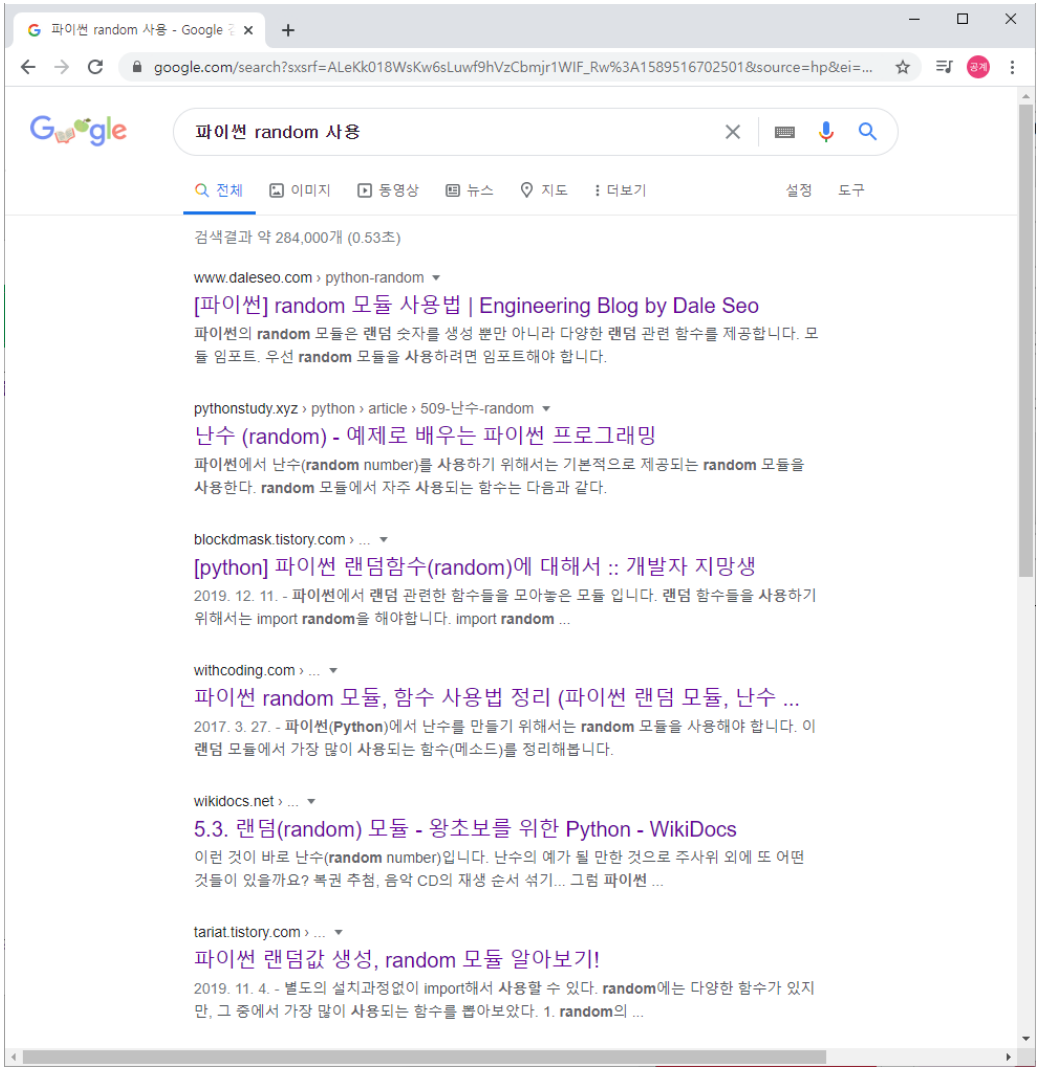
소스의 구조 보기

- **Alt + 7**





검색


파이썬 random 사용 - Google
google.com/search?sxsrf=ALeKk018WsKw6sLuwf9hVzCbmjr1WIF_Rw%3A1589516702501&source=hp&ei=...
Google
파이썬 random 사용
전체
이미지
동영상
뉴스
지도
더보기
설정
도구
검색결과 약 284,000개 (0.53초)
www.daleseo.com › python-random
[파이썬] random 모듈 사용법 | Engineering Blog by Dale Seo
파이썬의 random 모듈은 랜덤 숫자를 생성 뿐만 아니라 다양한 랜덤 관련 함수를 제공합니다. 모듈 임포트. 우선 random 모듈을 사용하려면 임포트해야 합니다.
pythonstudy.xyz › python › article › 509-난수-random
난수 (random) - 예제로 배우는 파이썬 프로그래밍
파이썬에서 난수(random number)를 사용하기 위해서는 기본적으로 제공되는 random 모듈을 사용한다. random 모듈에서 자주 사용되는 함수는 다음과 같다.
blockdmask.tistory.com › ...
[python] 파이썬 랜덤함수(random)에 대해서 :: 개발자 지망생
2019. 12. 11. - 파이썬에서 랜덤 관련한 함수들을 모아놓은 모듈 입니다. 랜덤 함수들을 사용하기 위해서는 import random을 해야합니다. import random ...
withcoding.com › ...
파이썬 random 모듈, 함수 사용법 정리 (파이썬 랜덤 모듈, 난수 ...
2017. 3. 27. - 파이썬(Python)에서 난수를 만들기 위해서는 random 모듈을 사용해야 합니다. 이 랜덤 모듈에서 가장 많이 사용되는 함수(메소드)를 정리해봅니다.
wikidocs.net › ...
5.3. 랜덤(random) 모듈 - 왕초보를 위한 Python - WikiDocs
이런 것이 바로 난수(random number)입니다. 난수의 예가 될 만한 것으로 주사위 외에 또 어떤 것들이 있을까요? 복권 추첨, 음악 CD의 재생 순서 섞기... 그럼 파이썬 ...
tariat.tistory.com › ...
파이썬 랜덤값 생성, random 모듈 알아보기!
2019. 11. 4. - 별도의 설치과정없이 import해서 사용할 수 있다. random에는 다양한 함수가 있지만, 그 중에서 가장 많이 사용되는 함수를 뽑아보았다. 1. random의 ...

다양한 함수

- **random() 함수**
  - 0부터 1사이의 랜덤 실수를 리턴
- **uniform() 함수**
  - 2개의 숫자 사이의 랜덤 실수를 리턴
- **randint(a, b) 함수**
  - 2개의 숫자 사이의 랜덤 정수를 리턴
- **randrange(a, b) 함수**
  - range(start, stop, step) 함수로 만들어지는 정수 중에 하나를 랜덤하게 리턴
- **choice(리스트) 함수**
  - 랜덤하게 하나의 원소를 선택
- **sample(리스트, 갯수) 함수**
  - 랜덤하게 갯수 원소를 유니크하게 선택
- **shuffle(리스트) 함수**
  - 원소의 순서를 랜덤하게 바꿈
- **seed(정수) 함수**
  - 시드 값 지정, 난수가 일정하게 생성

shuffle과 컴프리헨션의 조합

```
In [28]: items = [1, 2, 3, 4, 5, 6, 7]
         random.shuffle(items)
         items

Out[28]: [5, 4, 6, 3, 1, 7, 2]
```

```
In [30]: items = [1, 2, 3, 4, 5, 6, 7]
         [random.shuffle(items) for _ in range(5)]

Out[30]: [None, None, None, None, None]
```

```
In [106]: items = [1, 2, 3, 4, 5, 6, 7]

          def myshuffle(lst):
              random.shuffle(lst)
              return lst
```

```
In [120]: myshuffle(items)
          items

Out[120]: [7, 6, 3, 1, 2, 5, 4]
```

```
In [125]: [myshuffle(items) for _ in range(5)]

Out[125]: [[2, 4, 5, 6, 1, 3, 7],
           [2, 4, 5, 6, 1, 3, 7],
           [2, 4, 5, 6, 1, 3, 7],
           [2, 4, 5, 6, 1, 3, 7],
           [2, 4, 5, 6, 1, 3, 7]]
```

다음 코드의 결과 예상은?

- **목적**
  - 1에서 7까지의 정수를 shuffle를 5번하여 매번 결과를 리스트로 생성

Write code in Python 3.6

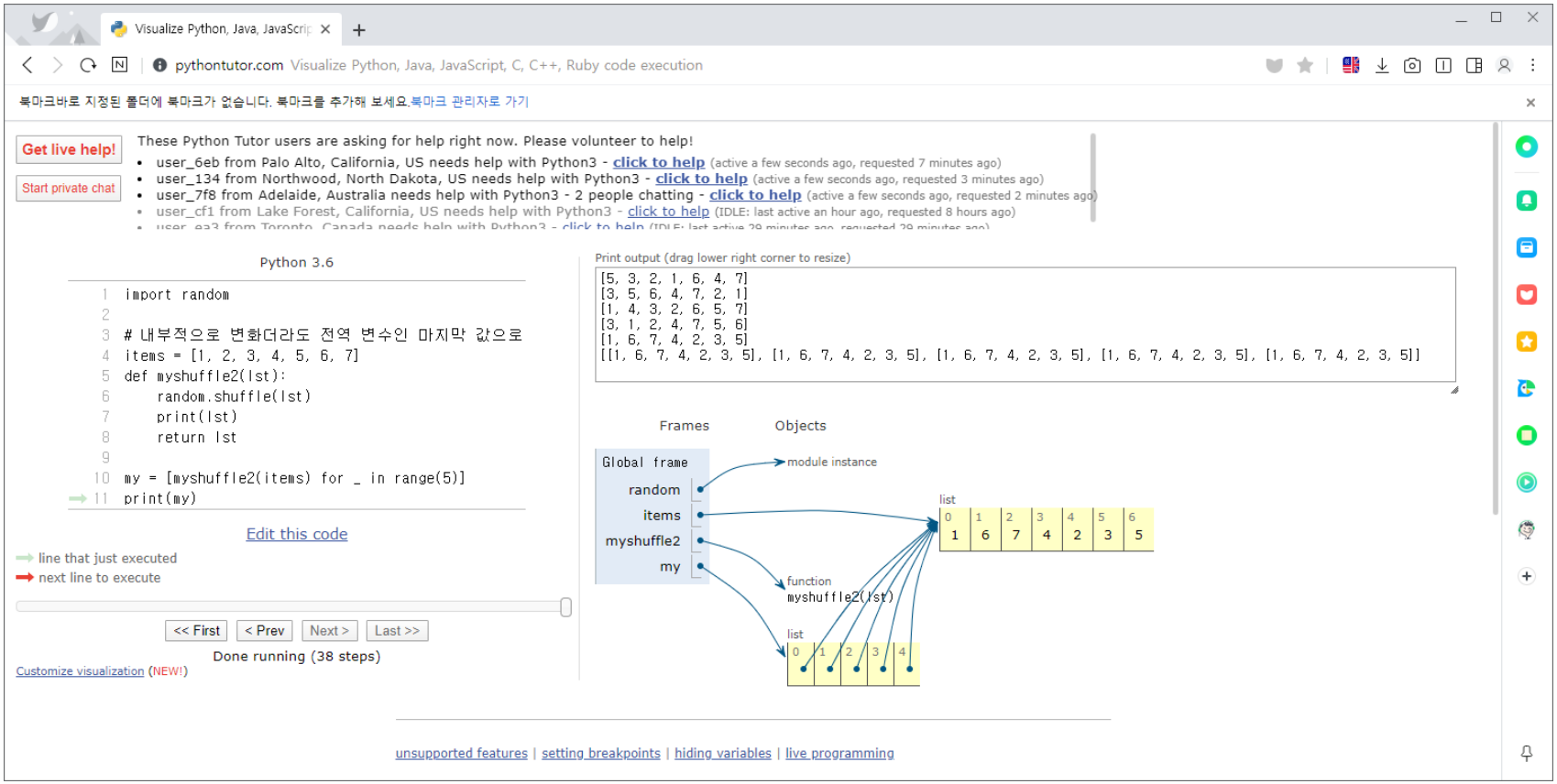
```
1  import random
2
3  # 내부적으로 변화더라도 전역 변수인 마지막 값으로
4  items = [1, 2, 3, 4, 5, 6, 7]
5  def myshuffle2(lst):
6      random.shuffle(lst)
7      print(lst)
8      return lst
9
10 my = [myshuffle2(items) for _ in range(5)]
11 print(my)
```

- **예상과는 다른 결과**

[[2, 4, 5, 6, 1, 3, 7], [2, 4, 5, 6, 1, 3, 7], [2, 4, 5, 6, 1, 3, 7], [2, 4, 5, 6, 1, 3, 7], [2, 4, 5, 6, 1, 3, 7]]

메모리 내부 모습 참조

- **Pythontutor.com 에서 보기**



다음으로 수정해 출력

- **리스트를 깊은 복사로 새로운 리스트 반환**

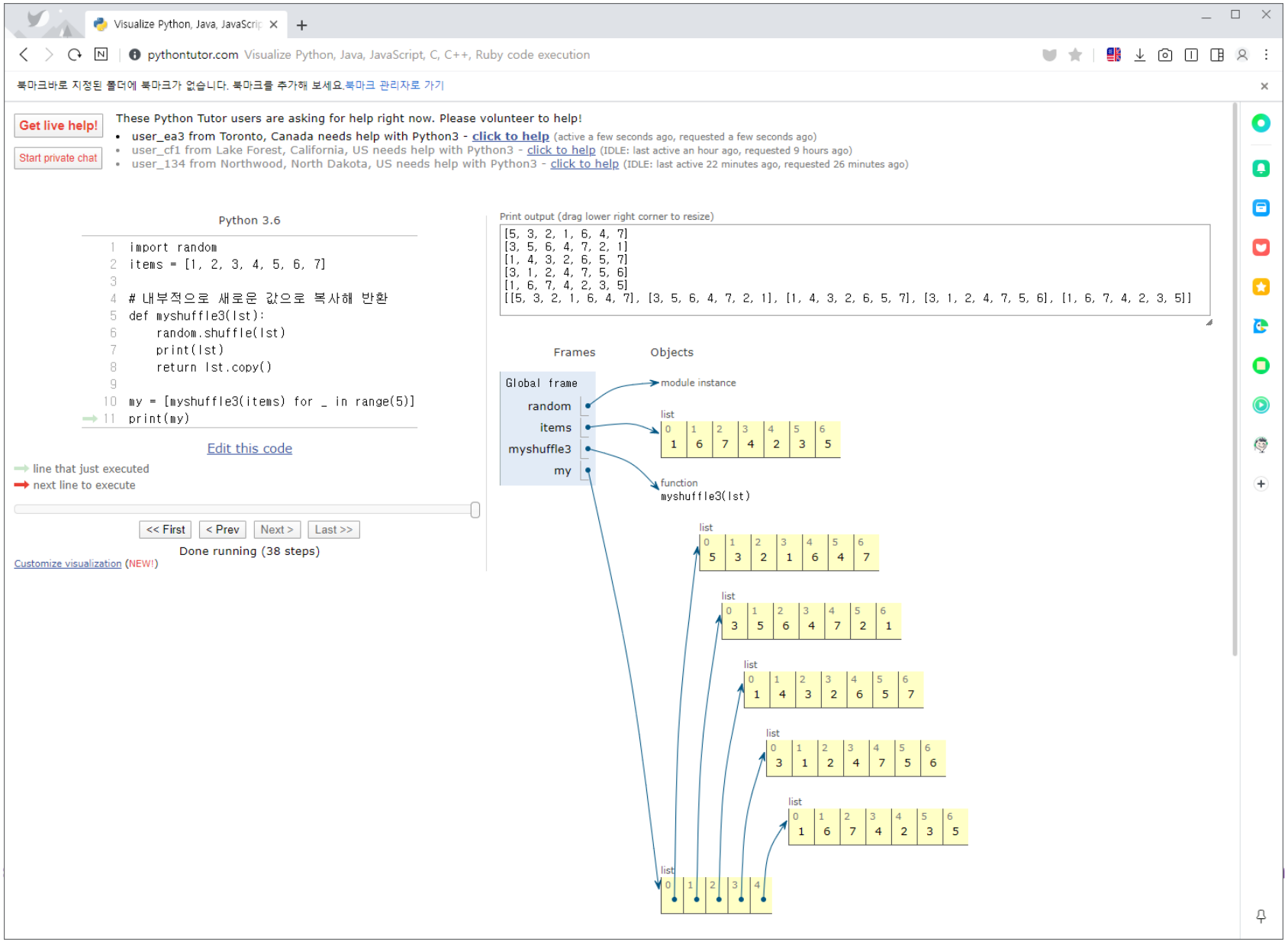
```
import random
items = [1, 2, 3, 4, 5, 6, 7]

# 내부적으로 새로운 값으로 복사해 반환
def myshuffle3(lst):
    random.shuffle(lst)
    print(lst)
    return lst.copy()

my = [myshuffle3(items) for _ in range(5)]
print(my)
```

```
[1, 4, 6, 7, 3, 2, 5]
[7, 4, 2, 5, 6, 3, 1]
[7, 5, 1, 3, 2, 4, 6]
[3, 1, 4, 5, 6, 2, 7]
[1, 7, 6, 2, 5, 3, 4]
[[1, 4, 6, 7, 3, 2, 5], [7, 4, 2, 5, 6, 3, 1], [7, 5, 1, 3, 2, 4, 6], [3, 1, 4, 5, 6, 2, 7], [1, 7, 6, 2, 5, 3, 4]]
```

메모리 내부 모습 확인


Visualize Python, Java, JavaScrip
pythontutor.com Visualize Python, Java, JavaScript, C, C++, Ruby code execution
북마크바로 지정된 폴더에 북마크가 없습니다. 북마크를 추가해 보세요.북마크 관리자로 가기
Get live help!
Start private chat
These Python Tutor users are asking for help right now. Please volunteer to help!
user_ea3 from Toronto, Canada needs help with Python3 - click to help (active a few seconds ago, requested a few seconds ago)
user_cf1 from Lake Forest, California, US needs help with Python3 - click to help (IDLE: last active an hour ago, requested 9 hours ago)
user_134 from Northwood, North Dakota, US needs help with Python3 - click to help (IDLE: last active 22 minutes ago, requested 26 minutes ago)
Python 3.6
1 import random
2 items = [1, 2, 3, 4, 5, 6, 7]
3
4 # 내부적으로 새로운 값으로 복사해 반환
5 def myshuffle3(lst):
6 random.shuffle(lst)
7 print(lst)
8 return lst.copy()
9
10 my = [myshuffle3(items) for _ in range(5)]
11 print(my)
Edit this code
line that just executed
next line to execute
<< First
< Prev
Next >
Last >>
Done running (38 steps)
Customize visualization (NEW!)
Print output (drag lower right corner to resize)
[5, 3, 2, 1, 6, 4, 7]
[3, 5, 6, 4, 7, 2, 1]
[1, 4, 3, 2, 6, 5, 7]
[3, 1, 2, 4, 7, 5, 6]
[1, 6, 7, 4, 2, 3, 5]
[[5, 3, 2, 1, 6, 4, 7], [3, 5, 6, 4, 7, 2, 1], [1, 4, 3, 2, 6, 5, 7], [3, 1, 2, 4, 7, 5, 6], [1, 6, 7, 4, 2, 3, 5]]
Frames
Objects
Global frame
random
items
myshuffle3
my
module instance
list
function
myshuffle3(lst)