

Forum:[Hack Forums](#)

Thread: [Lazy Eval \(Parallel Processing\) in Python](#)

simmer - March 26, 2022, 2:41 am

I have this code that gets me a stack overflow:

```
Code Q = set()
```

```
N = []
```

```
while True:
```

```
    Q = set(Q)
```

```
    N.append(Q)
```

```
v = i = k = n = any(N)
```

```
N_inf = N[-1]
```

```
v.append(i)
```

```
k.append(n)Thanks, I know right, code that not only doesn't work but wastes your time while it figures out it doesn't work
```

I need to focus on N all at once, no trying to count to it, no trying to focus on every individual part of it.

I think that this is called Parallel Programming, and as for the v, i, k, n I need Lazy Evaluation. k, v.append are for context related issues but that is out of the scope of this thread.

Perlo - March 27, 2022, 5:33 am

Your code doesn't make any sense. You have an empty set of Q and you run a while True loop, meaning that whatever code is below, won't ever run. Please be more specific about what your program has to do...

simmer - March 28, 2022, 1:45 pm

I guess what I'm asking is how to reference variables from inside a generator

I need all of N, N[-1], and any(N)

```
Code def W (N):
```

```
    Q = set()
```

```
    N = []
```

```
    while True:
```

```
        N.append(Q)
```

```
        Q = set(Q)
```

```
    yield N
```

```
N = N
```

```
v = i = k = n = any(N)
```

```
N_inf = N[-1]
```

Generators are lazy evaluation methods that do iteration all at once, so it is a good tool to use with infinite values.

The problem is usually when you search for generator values outside of the generator you get individual iterations (for loop, next(), ect)

I think it's appropriate to have a way to retrieve the parameters directly and in full... I will adjust this

thread accordingly.

simmer - April 9, 2022, 2:09 pm

This topic should be redirected to using an "if statement"

Disease - April 9, 2022, 2:19 pm

(Mar 26, 2022 02:41 AM)simmer Wrote: [\[CITE\]](#) I have this code that gets me a stack overflow:

```
Code Q = set()
```

```
N = []
```

```
while True:
```

```
    Q = set(Q)
```

```
    N.append(Q)
```

```
v = i = k = n = any(N)
```

```
N_inf = N[-1]
```

```
v.append(i)
```

```
k.append(n)Thanks, I know right, code that not only doesn't work but wastes your time while it figures out it doesn't work
```

I need to focus on N all at once, no trying to count to it, no trying to focus on every individual part of it.

I think that this is called Parallel Programming, and as for the v, i, k, n I need Lazy Evaluation. k, v.append are for context related issues but that is out of the scope of this thread.

What breaks that while loop? You're creating an infinite loop that will forever append empty sets to the N array. The rest of your code will never execute.

simmer - April 9, 2022, 2:34 pm

lol that was the problem but using an if statement requires no break