

# List Comprehensions For NOObS and N3rds

Chris Calloway



Triangle Zope and Python  
Users Group: TriZPUG

# What is a List Comprehension?

**A special list expression.**

**Remember, Python lists are:**

- a) Mutable,**
- b) Ordered,**
- c) Sequences of arbitrary objects**



**Triangle Zope and Python  
Users Group: TriZPUG**

# Are There Other List Expressions?

**Yes.**

***Some others are:***

- a) slices,**
- b) ranges,**
- c) filters,**
- d) maps,**
- e) zips**



# What's Special About a List Comprehension?

**It's an idiom for a FOR loop.**

```
result = []  
for target in iterable:  
    if condition:  
        result.append(expression)
```



**Triangle Zope and Python  
Users Group: TriZPUG**

# What's Special About a List Comprehension?

```
result = []  
for target in iterable:  
    if condition:  
        result.append(expression)
```

Same as:

```
[expression for target in iterable if condition]
```



# Do List Comprehensions Violate the Zen of Python?

**Maybe.**

- **Explicit is better than implicit.**
- **Sparse is better than dense.**
- **Special cases aren't special enough to break the rules.**
- **There should be one-- and preferably only one --obvious way to do it.**
- **If the implementation is hard to explain, it's a bad idea.**



**Triangle Zope and Python  
Users Group: TriZPUG**

# Do List Comprehensions Violate the Zen of Python?

**Maybe.**

- **Beautiful is better than ugly.**
- **Complex is better than complicated.**
- **Flat is better than nested.**
- **Readability counts.**
- **Practicality beats purity.**



**Triangle Zope and Python  
Users Group: TriZPUG**

# Do List Comprehensions Violate the Zen of Python?

**Probably.**

**In the face of ambiguity, refuse the temptation to guess.**

```
>>> import this
```



**Triangle Zope and Python  
Users Group: TriZPUG**



# OK Silly, Be Serious...

[expression for target in iterable lc-clauses]

**Where lc-clauses are:**

for target in iterable

**Or**

if condition



**Triangle Zope and Python  
Users Group: TriZPUG**

# OK Serious, Simplify It...

```
[x for x in 'abcd' if x < 'd']
```

Same as:

```
result = []  
for x in 'abcd':  
    if x < 'd':  
        result.append(x)
```

Both result in:

```
['a', 'b', 'c']
```



Triangle Zope and Python  
Users Group: TriZPUG

# OK Simple, Get Real...

Expression can mutate target:

```
[x**2 for x in xrange(5)]
```

Results in:

```
[0, 1, 4, 9, 16]
```



Triangle Zope and Python  
Users Group: TriZPUG

# OK Real, Show Me Complex...

Any number of lc-clauses:

```
[x**y for x in xrange(5) for y in xrange(3)]
```

Results in:

```
[0, 0, 0, 0, 1, 2, 0, 2, 4, 0, 3, 6, 0, 4, 8]
```



Triangle Zope and Python  
Users Group: TriZPUG

# OK Complex, What's Complicated?...

ANY number of lc-clauses:

```
[x**y for x in xrange(5)
    if not x % 2
    for y in xrange(3)
    if y % 2]
```

Results in:

```
[0, 2, 4]
```



Triangle Zoep and Python  
Users Group: TriZPUG

## In General...

```
result = []  
for target in iterable:  
    lc-clause-1:  
        ...  
        lc-clause-n:  
            result.append(expression)
```



Triangle Zope and Python  
Users Group: TriZPUG

# What use is a List Comprehension?

- a) return values
- b) parameters in function calls
- c) lambda implementations
- d) function implementations



Triangle Zope and Python  
Users Group: TriZPUG

# List Comprehension as Return Value

```
def series(xlist, ylist):  
    return [x**y for x in xlist for y in ylist]
```



Triangle Zope and Python  
Users Group: TriZPUG



# List Comprehension as Parameter

```
import sys
```

```
myfunc([x.lower() for x in sys.argv[:-1] if x[0] <> '-'])
```



Triangle Zope and Python  
Users Group: TriZPUG

# List Comprehension as Lambda Implementation

```
lambda xlist, ylist: [x**y for x in xlist for y in ylist]
```



**Triangle Zope and Python  
Users Group: TriZPUG**

# List Comprehension as Function Implementation

```
def filter(func, seq):  
    return [item for item in seq if func(item)]
```

```
def map(func, seq):  
    return [func(item) for item in seq]
```



Triangle Zope and Python  
Users Group: TriZPUG

# Are List Comprehensions Optimized?

No.



Triangle Zope and Python  
Users Group: TriZPUG