Intersecting Sequences

def (arg1, arg2,... argN): ... return

```
def times(x, y): # Create and assign function
     return x * y # Body executed when called
```

```
# Calling a function
times(2, 4) # Arguments in parentheses
```

```
8
```

```
x = times(3.14, 4) # Save the result object
x
```

```
12.56
```

```
times('Ni', 4) # Functions are "typeless"
```

```
'NiNiNiNi'
```

```
def intersect(seq1, seq2):
    res = [] # Start empty
    for x in seq1: # Scan seq1
        if x in seq2: # Common item?
           res.append(x) # Add to end
    return res
```

```
times('Ni', 4) # Functions are "typeless"
```

```
s1 = "SPAM"
s2 = "SCAM"
intersect(s1, s2) # Strings
```

```
['S', 'A', 'M']
```

```
#  the function could be replaced with a single list comprehension expression, as it exhibits the classic loop collector code pattern:
```

```
[x for x in s1 if x in s2]
```

```
['S', 'A', 'M']
```

Start coding or generate with AI.