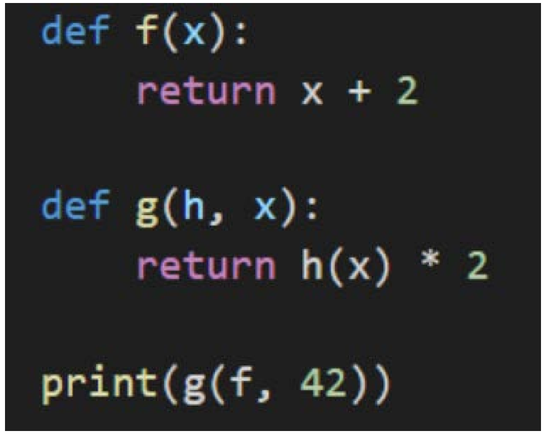
Summing Up

Gerald Britton

IT SOLUTIONS DESIGNER

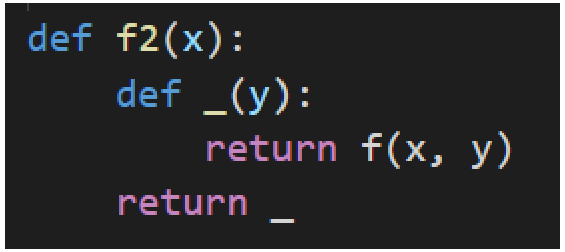
@GeraldBritton www.linkedin.com/in/geraldbritton

Higher-order functions



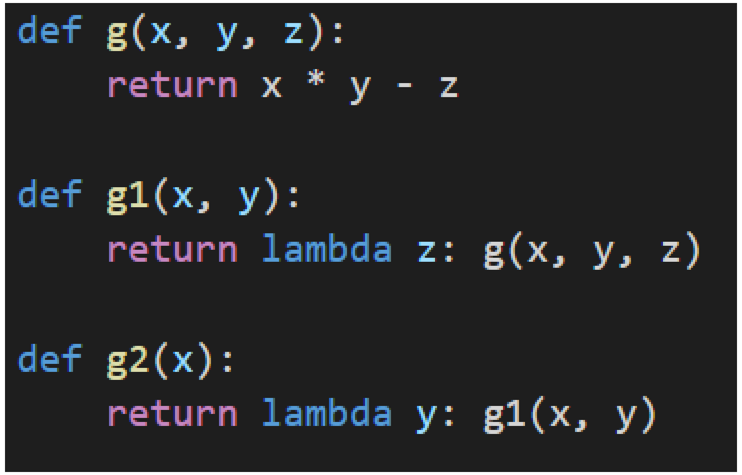
```
def f(x):
    return x + 2

def g(h, x):
    return h(x) * 2

print(g(f, 42))
```

Composition



```
def f2(x):
    def _(y):
        return f(x, y)
    return _
```

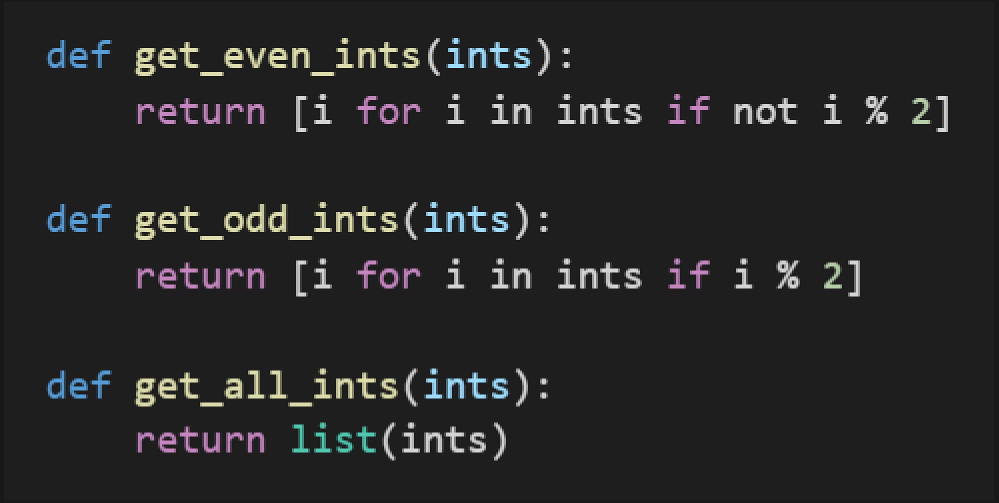
Closure



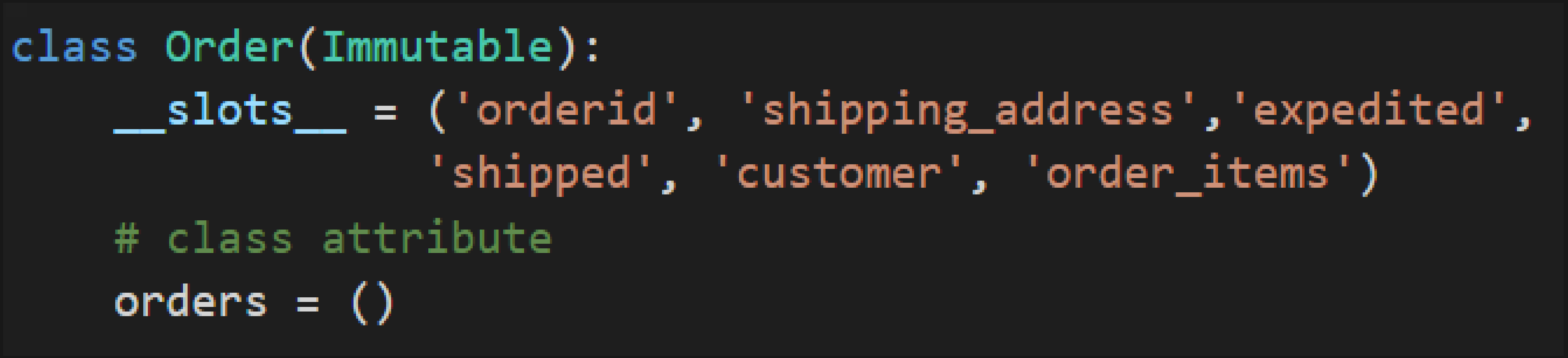
```
def g(x, y, z):
    return x * y - z

def g1(x, y):
    return lambda z: g(x, y, z)

def g2(x):
    return lambda y: g1(x, y)
```

Currying

Pure Functions



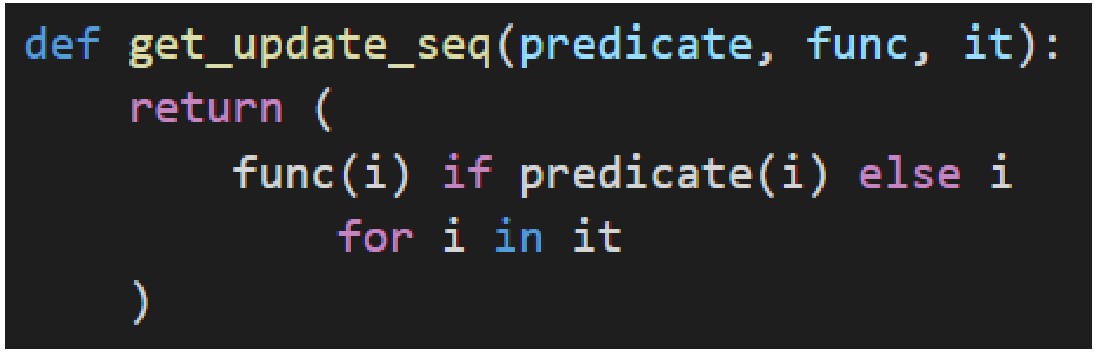
```python
def get_even_ints(ints):
    return [i for i in ints if not i % 2]

def get_odd_ints(ints):
    return [i for i in ints if i % 2]

def get_all_ints(ints):
    return list(ints)
```



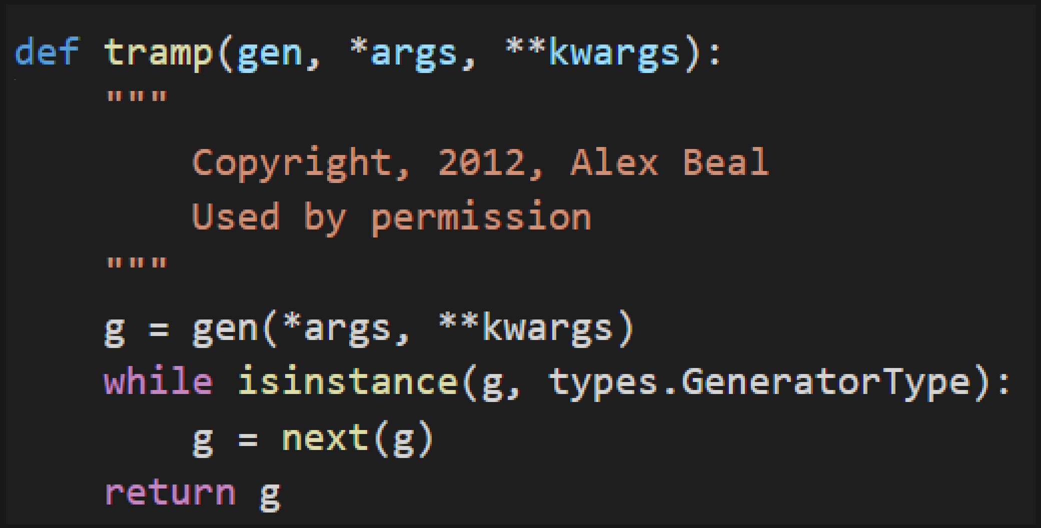
```
class Order(Immutable):
    __slots__ = ('orderid', 'shipping_address','expedited',
                 'shipped', 'customer', 'order_items')
    # class attribute
    orders = ()
```

Immutability

Lazy Evaluation

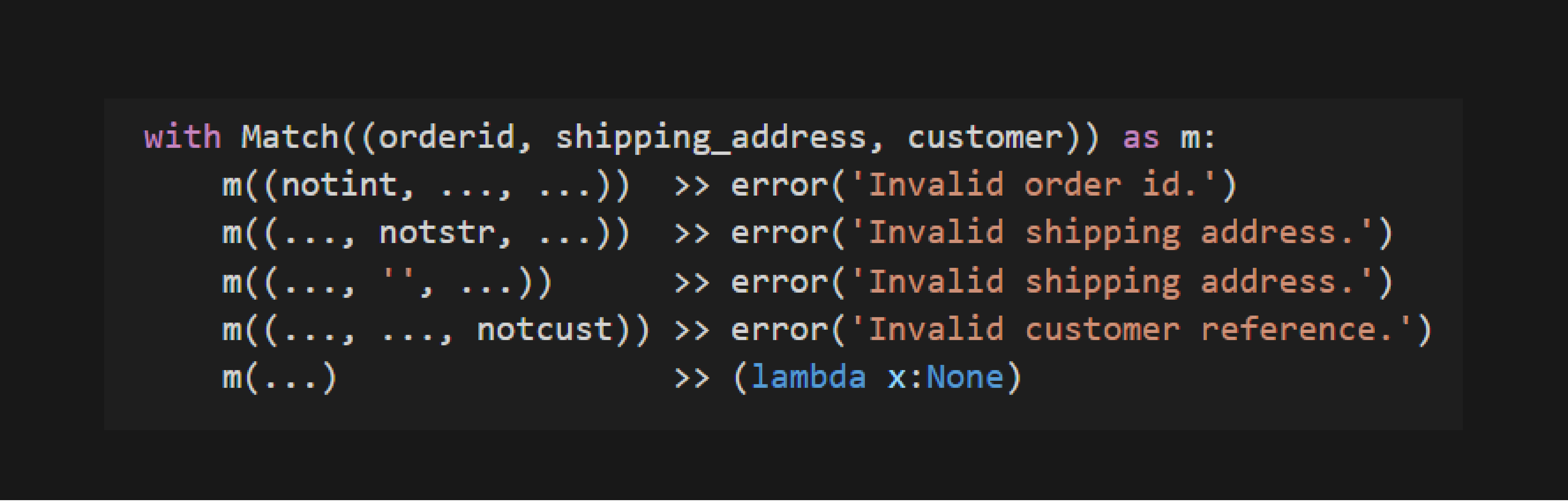


```python
def get_update_seq(predicate, func, it):
    return (
        func(i) if predicate(i) else i
            for i in it
    )
```



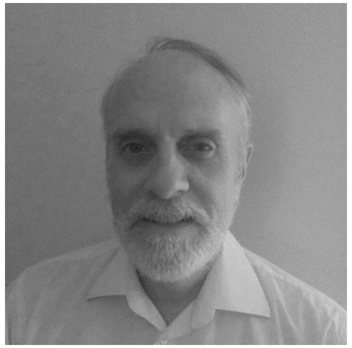
```python
def tramp(gen, *args, **kwargs):
    """
        Copyright, 2012, Alex Beal
        Used by permission
    """
    g = gen(*args, **kwargs)
    while isinstance(g, types.GeneratorType):
        g = next(g)
    return g
```

Recursion



```
with Match((orderid, shipping_address, customer)) as m:
    m((notint, ..., ...))  >> error('Invalid order id.')
    m((..., notstr, ...))  >> error('Invalid shipping address.')
    m((..., '', ...))      >> error('Invalid shipping address.')
    m((..., ..., notcust)) >> error('Invalid customer reference.')
    m(...)                 >> (lambda x:None)
```

Matching



- Functional Programming with Python
- Design Patterns with Python
- Building More Python Design Patterns

Gerald Britton
IT Solutions designer
@GeraldBritton
www.linkedin.com/in/geraldbritton