Course Summary



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

```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)
```

Pure Functions

```python
from dataclasses import dataclass, field
@dataclass(frozen=True)
class Order:
    # class attribute
    orders: tuple = field(init=False)
```

Immutability

Dataclasses
Read-only collections

Lazy Evaluation

```
@staticmethod
def get_order_details(orders):
    d = namedtuple('OrderDetails', 'orderid, cu
    return (d(o.orderid, o.customer.name, o.exp
        for o in orders for i in o.order_items)
```

```
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

```
for i in range(5):
    match i:
        case 1 | 3 | 5 if i < 4:
            print (f'{i} is odd and less than 4')
        case 1 | 3 | 5 if i > 4:
            print (f'{i} is odd and greater than 4')
        case 2 | 4:
            print (f'{i} is even’)
```

Matching

Match statement in Python 3.10



- Functional Programming with Python 3
- Design patterns with Python 3

Gerald Britton

IT Solutions Designer

@GeraldBritton www.linkedin.com/in/geraldbritton