LESSON 12 FUNCTIONAL STUFF!!!

but before we get into functional programming...

SPLATS!!!

↳ what if I want to dynamically pass an array (list, sequence) ← iterable
of positional arguments to a function

```
foods = ['spam', 'eggs']

'I ate {0} and {1} for dinner'.format(*foods)
```

↳ Now repeat this with combinations of positional arguments, wrong number in iterable, etc...

Double Splat! unpack a mapping (dict) into keyword parameters

```
'I ate {main} and {side} for dinner'.format(**foods)

foods {'main': 'spam', 'side': 'eggs'
```

How do I make a function, like 'format' which receives any number of positional or keyword arguments?

```
def foo(*args, **kwargs):
    print('I got {} and {}'.format(args, kwargs))
```

↳ simplest example.

Also do examples mixing ~~or~~ specific arguments and splats.

KEYWORD ONLY ARGUMENTS: (Python 3)

```
def foo(x, *, y):
    return x+y
```

← Try calling this, also do it without the *

Functional Programming → Motivate

- Functions are first class ~~functions~~ !
  ↳ can pass around functions like any other object

Example:

```
def make_hello(target):
    def hello():
        print('Hello {}'.format(target))
    return hello
```

↳ make a list of hello functions, iterate over it and call them

RECURSION !

- Generally should be avoided in python if more than a few levels deep
- Default stack is 1000 levels
- Stack trace is important!
- Iteration is the way

```
def fib(n):
    if n == 0:
        return 0
    if n == 1:
        return 1
    return fib(n-1) + fib(n-2)
```

} the poor performing Example

```
def fac(n):
    if n == 1:
        return 1
    return n * fac(n-1)
```

↳ How the stack!

↳ Higher order functions → lets memoize both of these!

```
def memo(fun):
    results = {}
    def memoized_fun(var):
        if var in results:
            return results[var]
        output = fun(var)
        results[var] = output
        return output
    return memoized_fun
```

do decorator "by hand" then use the pie operator

do examples, then discuss docstring issue, use functools wraps

---

Lambdas: Useful Example sorting a grocery list by price

```
l = [('apples', 2, 1.5), ('bananas', 30, 0.5), ('spam', 1, 2.5)]
```

want to sort by total price (quantity * unit price)

sort method can sort by a key extraction function

```
def keyfunc(elem):
    return elem[1] * elem[2]

l.sort(key = keyfunc)
```

do as lambda ⟶ Explain lambda restrictions