PYTHON
SECTION NINE

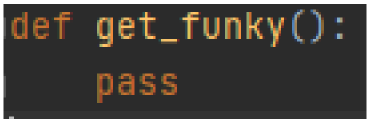
Introduction to Python| Functions | Pragmatic Agility

FUNCTIONS

- Code organizational technique
- DRY concept. Don't Repeat Yourself
- Reusable code blocks
- Define with 0 or more parameters
- Return 0 or more results

DEFINE FUNCTION

Define a function with 'def'

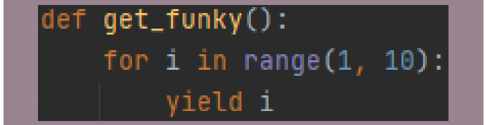


```
def get_funky():
    pass
```

The pass keyword is used to simply leave the object. It indicates to Python that nothing should be done here.

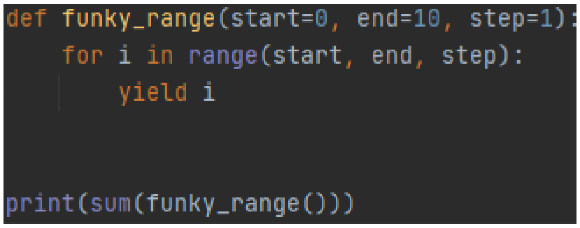
GENERATORS

- Special function that uses yield instead of return
- Used to more efficiently iterate large sequence of objects



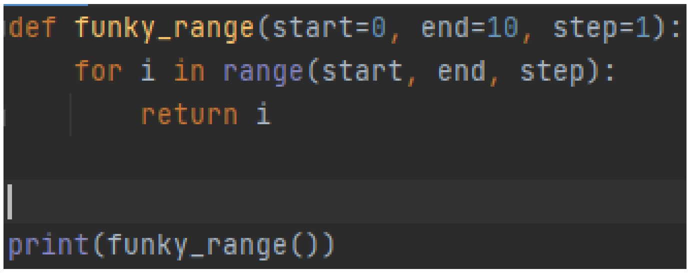
```
def get_funky():
    for i in range(1, 10):
        yield i
```

QUICK EXAMPLES



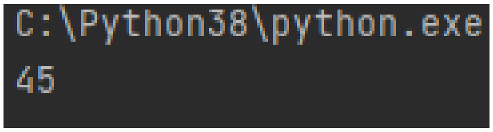
```
def funky_range(start=0, end=10, step=1):
    for i in range(start, end, step):
        yield i


print(sum(funky_range()))
```

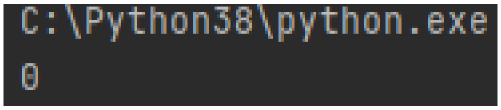


```
def funky_range(start=0, end=10, step=1):
    for i in range(start, end, step):
        return i


print(funky_range())
```



```
C:\Python38\python.exe
45
```



```
C:\Python38\python.exe
0
```

Yield is used to return sequences. If the return command were used, the first value of i would come back alone.

One integer is not iterable. You can not call the sum function on a non-iterable object.