def

for 1n

vield i

3 # Doesn't start the function!

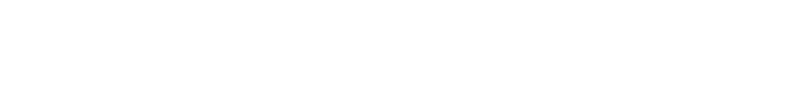
=> <class 'generator'>



=> 1

=> 2

raises StopIteration



Simple Generator

The yield keyword tells Python to

```
treat the function as a generator
    for i in range(n):
        yield i
q = qenerate ints(3) # Doesn't start the function!
type(g) # => <class 'generator'>
next(g) # => 0
next(g) # => 1
next(g) # => 2
next(g) # raises StopIteration
```

def generate_ints(n):

Another Generator

```
def generate_fibs():
    a, b = 0, 1
    while True:
        a, b = b, a + b
    yield a
```

Infinite data stream of Fibonacci numbers