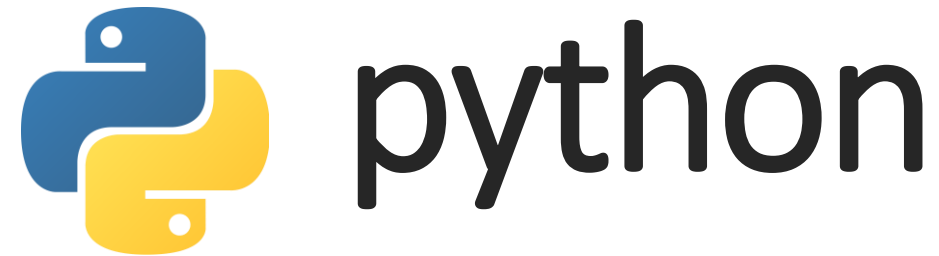




Dipartimento di  
Ingegneria Elettrica Elettronica  
e Informatica

# Advanced Programming Languages



---

**MARCO GRASSIA**

MARCO.GRASSIA@UNICT.IT

# Advanced Programming Languages



# Iterators

---

[HTTPS://SCIPY-LECTURES.ORG/ADVANCED/ADVANCED\\_PYTHON/INDEX.HTML](https://scipy-lectures.org/advanced/advanced_python/index.html)  
[HTTPS://STACKABUSE.COM/INTRODUCTION-TO-PYTHON-ITERATORS/](https://stackabuse.com/introduction-to-python-iterators/)  
[HTTPS://REALPYTHON.COM/INTRODUCTION-TO-PYTHON-GENERATORS/](https://realpython.com/introduction-to-python-generators/)

# Iterators

---

**Objects** that implement the `__next__` method. They

- Hold the state (position) of the iteration
- Allow looping just once and must be reinitialized to loop again

The ***next* method** returns the next item in the sequence

- If there is nothing to return, raises the ***StopIteration*** exception

An **iterable** is an object capable of returning an iterator

- Must implement the `__iter__` method, callable using the ***iter*** function



# Quick example

# Generators

---

**Functions** containing the keyword *yield*

*yield*:

- returns an object when called, just like *return* ...
- **BUT** the state of the function is saved, so that execution resumes where it was left off when *next()* is called again

Do not return values when initialized

Example #1

# Generators

---

They are **lazy iterators**:

- Dynamic values generation
- No `__len__` method (i.e., no `len()`)
- Help with Out of Memory issues
- Bidirectional communication

Concurrent and recursive invocations are allowed

Not thread safe (per-se)

# Generators

---

## Bidirectional communication:

- Allows to send values **to** the generator
- Three generator methods:
  - `.send(...)`: sends the value to the generator **and** returns the next value (like `next()`)
  - `.throw(...)`: throws the passed exception **after** resuming the generator, that will handle it
  - `.close()`: stops the generator. Equivalent to catching a `throw(GeneratorExit())`
- ***yield*** can be used in expressions to assign values to generator's variables
- Values will be assigned when the generator resumes from *yield*



# Example #2