

CALLABLES

What are callables?

any object that can be called using the `()` operator

callables **always** return a value

like **functions** and **methods** but it goes beyond just those two...

many other objects in Python are also callable

To see if an object is callable, we can use the built-in function: `callable`

```
callable(print) → True
```

```
callable('abc'.upper) → True
```

```
callable(str.upper) → True
```

```
callable(callable) → True
```

```
callable(10) → False
```


Different Types of Callables

built-in functions

`print` `len` `callable`

built-in methods

`a_str.upper` `a_list.append`

user-defined functions

created using `def` or `lambda` expressions

methods

functions `bound` to an object

classes

`MyClass(x, y, z)`

→ `__new__(x, y, z)` → creates the new object

→ `__init__(self, x, y, z)`

→ returns the object (reference)

class instances

if the class implements `__call__` method

generators, coroutines, asynchronous generators

Code