dunder

give me a starter tutorial on python dunder methods. what are the most common and useful ones? Explain with examples

Dunder methods, short for "double underscore" methods, are special methods in Python that begin and end with double underscores. They are also known as magic methods and are meant to be invoked indirectly. Most of these methods provide a way to define the behavior of objects with respect to the built-in Python language operations and functions. Here's a brief tutorial covering some of the most common and useful dunder methods:

1. `__init__(self, ...)`
    - Initializes a new instance of a class.
    - Example: Creating a new object with specific initial values.

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y
p = Point(1, 2)
```

2. `__str__(self)`
    - Returns a human-readable string representation of an object.
    - Example: Printing an object in a user-friendly format.

```
class Point:
    def __str__(self):
        return f"Point({self.x}, {self.y})"
```

3. `__repr__(self)`
    - Returns an unambiguous string representation of an object, often used for debugging.
    - Example: Official string representation that can be used to recreate the object.

```
class Point:
    def __repr__(self):
        return f"Point({self.x}, {self.y})"
```

4. `__eq__(self, other)`
    - Defines the behavior of the equality operator `==`.
    - Example: Comparing two objects for equality based on their content.

```
class Point:
    def __eq__(self, other):
        return self.x == other.x and self.y == other.y
```

5. `__add__(self, other)`

- Defines the behavior of the addition operator `+`.
- Example: Adding two objects together.

```
class Point:
    def __add__(self, other):
        return Point(self.x + other.x, self.y + other.y)
```

6. `__len__(self)`
   - Returns the length of the container. Called by the `len()` function.
   - Example: Getting the number of elements in a container.

```
class MyCollection:
    def __len__(self):
        return len(self.my_container)
```

7. `__getitem__(self, key)`
   - Allows access to elements using square brackets.
   - Example: Accessing an element at a particular position in a container.

```
class MyCollection:
    def __getitem__(self, key):
        return self.my_container[key]
```

8. `__iter__(self)`
   - Returns an iterator for the container.
   - Example: Allowing iteration over the container.

```
class MyCollection:
    def __iter__(self):
        return iter(self.my_container)
```

9. `__call__(self, ...)`
   - Allows the instance of a class to be called as a function.
   - Example: Making an object callable.

```
class Adder:
    def __call__(self, x, y):
        return x + y
adder = Adder()
result = adder(2, 3)  # result is 5
```

These are some of the most common dunder methods you'll encounter and use in Python. They allow for more Pythonic code and enable objects to integrate seamlessly with the language features.