



Python OOP: Inheritance (Attributes)



Inheritance (Attributes)



Key Takeaways

- Inheritance (Attributes)

- Take advantage of natural hierarchies between objects and concepts by creating classes that “inherit” attributes and behaviors from other classes.
 - ✓ For example: a SavingsAccount is a type of account, so it could inherit the attributes and functionality of the Account class.
- Advantages:
 - ✓ Reuse existing code.
 - ✓ Write more maintainable and scalable code.
 - ✓ Improve project structure and design.
- Key Concepts:
 - Parent class (superclass): the class from which another class inherits attributes and behaviors.
 - Child class (subclass): the class that inherits attributes and behaviors from another class.
- You can create multilevel hierarchies. Classes can inherit from classes that inherit from classes that inherit from classes and so on...



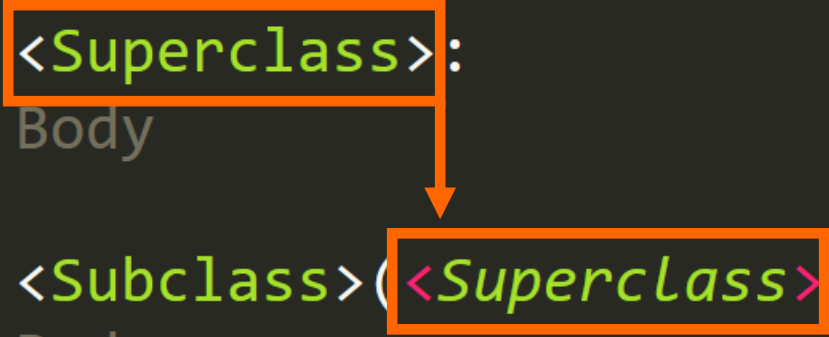
Inheritance (Attributes)



Key Takeaways

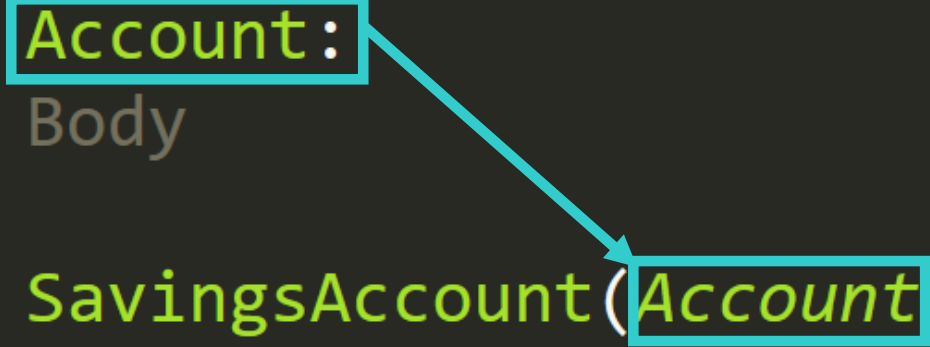
- General Syntax (First Step – Parent & Child)

```
class <Superclass>:  
    # Body  
  
class <Subclass>(<Superclass>):  
    # Body
```



- Example

```
class Account:  
    # Body  
  
class SavingsAccount(Account):  
    # Body
```





Inheritance (Attributes)



Key Takeaways

- General Syntax (Second Step - Attributes)

```
class <Superclass>:  
    # Body
```

```
class <Subclass>(<Superclass>):
```

Attributes are automatically inherited if you don't define `__init__()` in the subclass

```
def __init__(self, <parameters>):  
    <Superclass>.__init__(self, <arguments_for_parent_class>)  
    # Other instance attributes
```

```
# Methods
```

- Example

```
class Account:
```

```
    accounts_created = 0
```

```
    def __init__(self, number, client, balance):  
        self.number = number  
        self.client = client  
        self.balance = balance  
        accounts_created += 1
```

```
    def display_balance(self):  
        print(self.balance)
```

```
class SavingsAccount(Account):
```

```
    def __init__(self, number, client, balance, interest_rate):  
        Account.__init__(self, number, client, balance)  
        self.interest_rate = interest_rate
```

```
    def display_interest_rate(self):  
        print(self.interest_rate)
```



Inheritance (Attributes)



Key Takeaways

- Example in More Detail

```
class Account:

    accounts_created = 0


    def __init__(self, number, client, balance):
        self.number = number
        self.client = client
        self.balance = balance
        accounts_created += 1

    def display_balance(self):
        print(self.balance)

class SavingsAccount(Account):

    def __init__(self, number, client, balance, interest_rate):
        Account.__init__(self, number, client, balance)
        self.interest_rate = interest_rate

    def display_interest_rate(self):
        print(self.interest_rate)
```



SavingsAccount
inherits from
Account