

PYTHON INHERITANCE



Introduction

Inheritance allows us to define a class that inherits all the methods and properties from another class.

Parent class is the class being inherited from, also called base class.

Child class is the class that inherits from another class, also called derived class.

CREATING A PARENT OR CHILD CLASS

PARENT

```
class Person:  
    def __init__(self, fname, lname):  
        self.firstname = fname  
        self.lastname = lname  
  
    def printname(self):  
        print(self.firstname, self.lastname)  
  
#Use the Person class to create an object, and then execute  
#the printname method:  
  
x = Person("John", "Doe")  
x.printname()
```

CHILD

Create a class named Student, which will inherit the properties and methods from the Person class:

```
class Student(Person):
```

pass

Use the pass keyword when you do not want to add any other properties or methods to the class.

Now the Student class has the same properties and methods as the Person class.

Use the student class to create an object, and then execute the printname method:

```
x = Student("Mike", "Olsen")  
x.printname()
```



Add the `__init__()` Function

So far we have created a child class that inherits the properties and methods from its parent.

We want to add the `__init__()` function to the child class (instead of the `pass` keyword).

Note: The `__init__()` function is called automatically every time the class is being used to create a new object.

Example

Add the `__init__()` function to the `Student` class:

```
class Student(Person):
    def __init__(self, fname, lname):
        #add properties etc.
```

When you add the `__init__()` function, the child class will no longer inherit the parent's `__init__()` function.

Note: The child's `__init__()` function overrides the inheritance of the parent's `__init__()` function.

To keep the inheritance of the parent's `__init__()` function, add a call to the parent's `__init__()` function:

Example

```
class Student(Person):
    def __init__(self, fname, lname):
        Person.__init__(self, fname, lname)
```



Add Properties

Add a property called graduationyear to the Student class:

```
class Student(Person):
    def __init__(self, fname, lname):
        super().__init__(fname, lname)
        self.graduationyear = 2019
```



Add METHODS

Add a method called welcome to the Student class:

```
class Student(Person):
    def __init__(self, fname, lname, year):
        super().__init__(fname, lname)
        self.graduationyear = year
```

```
    def welcome(self):
        print("Welcome", self.firstname, self.lastname, "to the class of",
              self.graduationyear)
```

If you add a method in the child class with the same name as a function in the parent class, the inheritance of the parent method will be overridden.



Thanks!