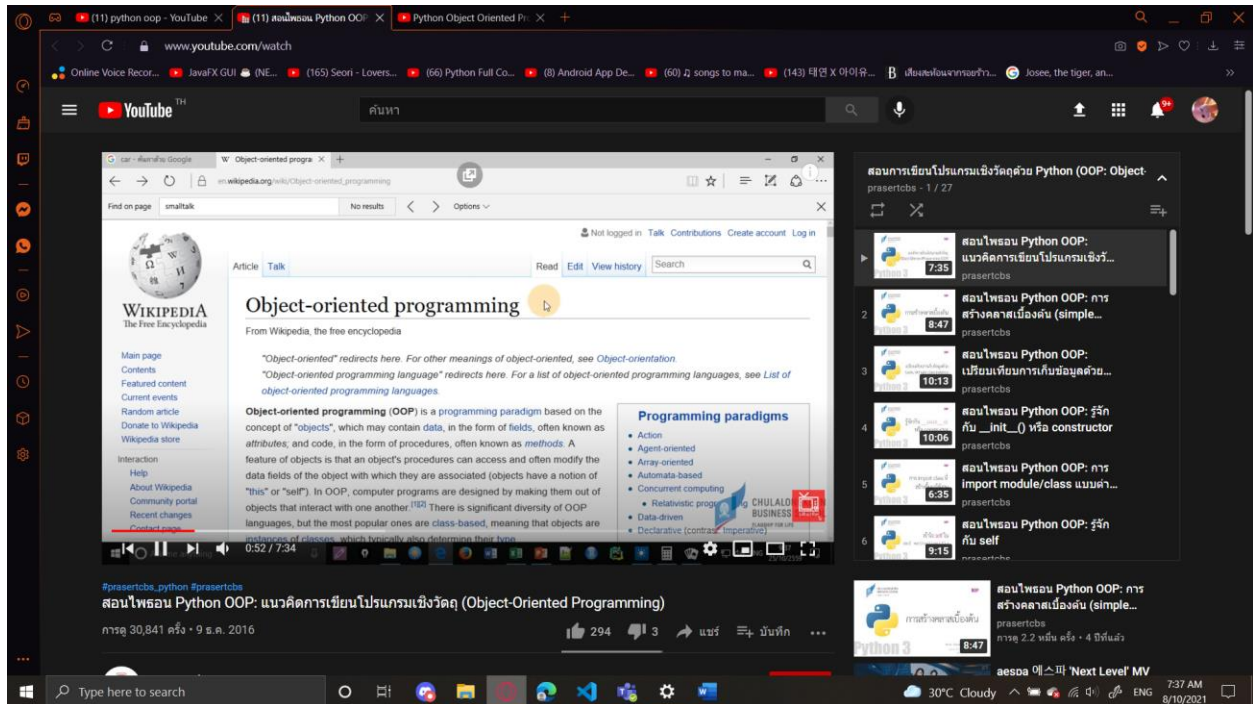
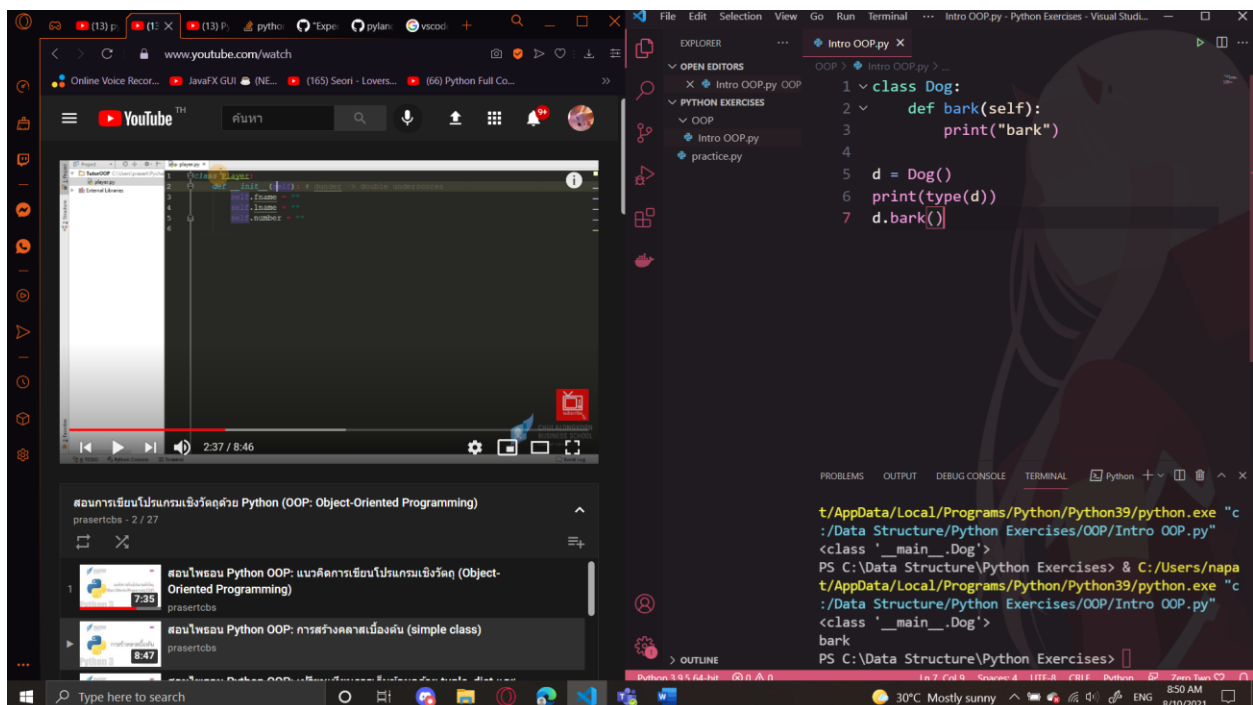


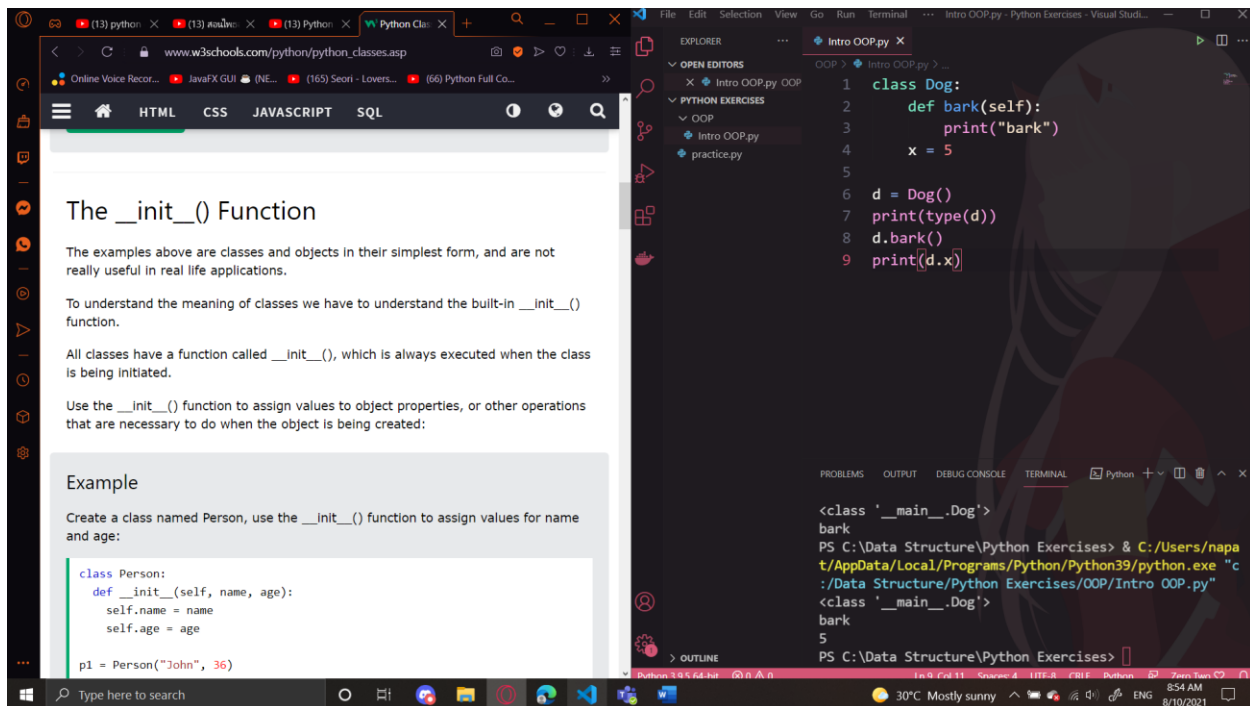
1) ศึกษาแนวคิดเกี่ยวกับ Python : OOP



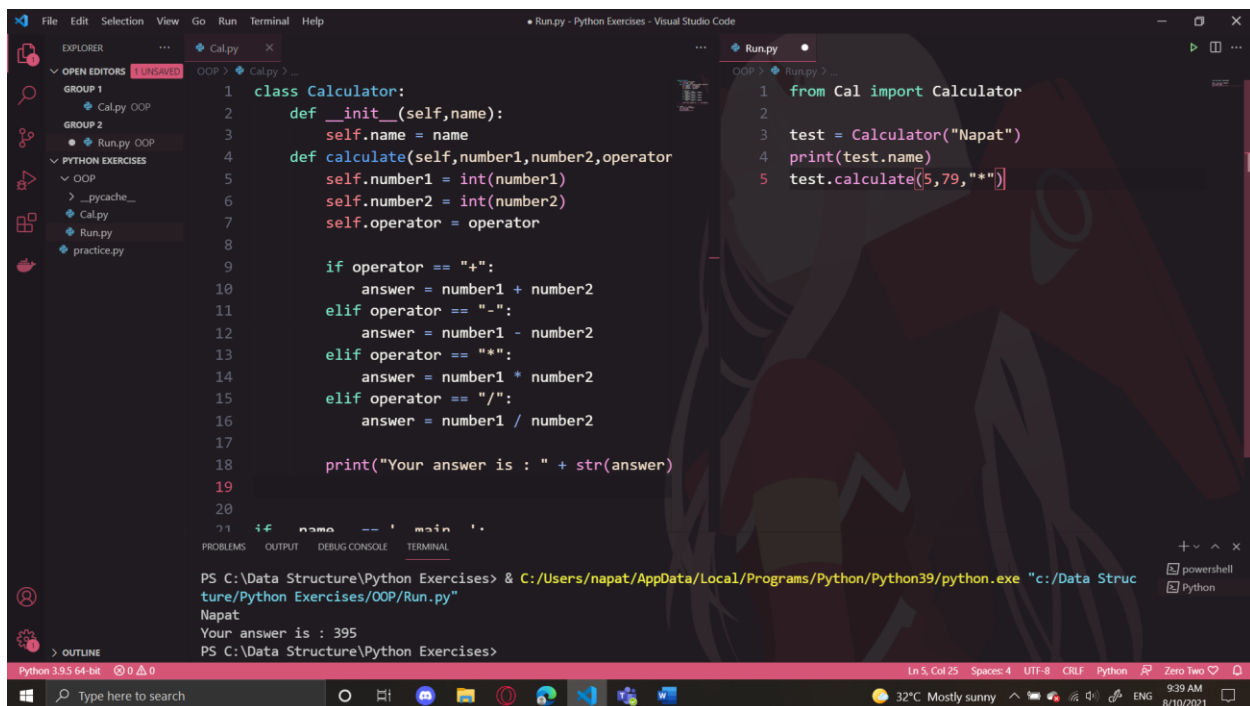
2) ลองสร้าง Class ใน Python ครั้งแรก



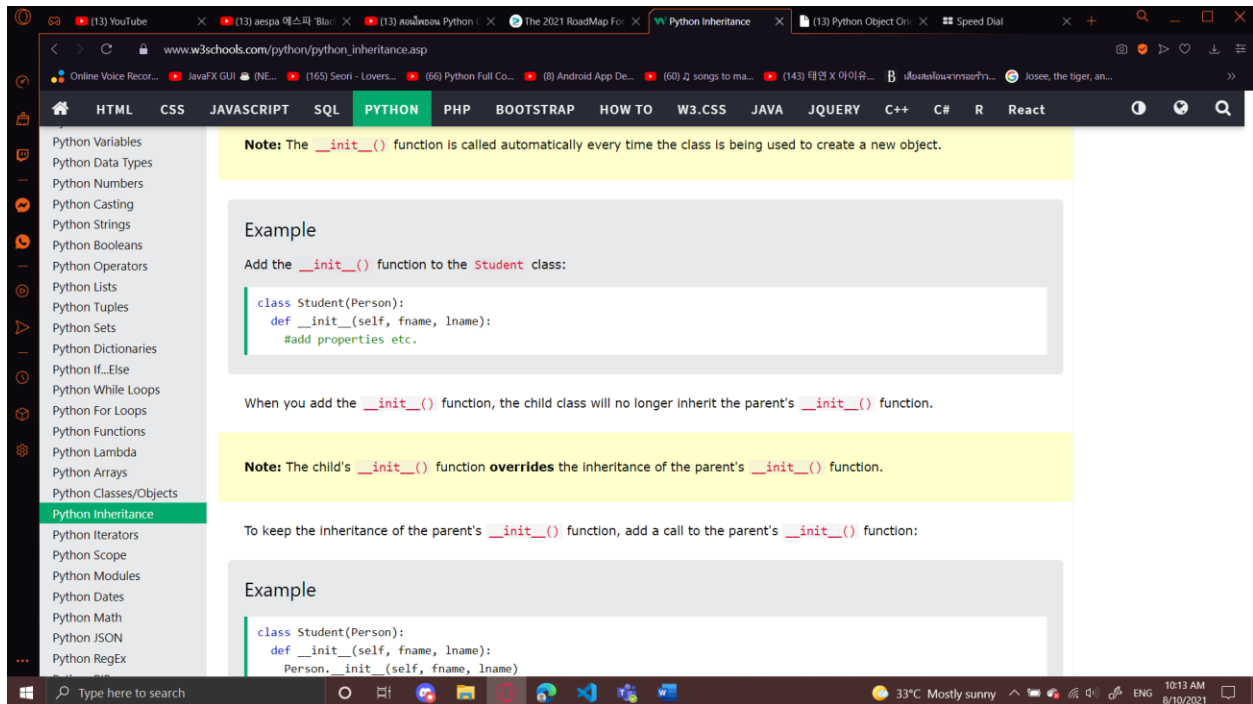
3) ศึกษาเพิ่มเติมเรื่องฟังก์ชัน `__init__` function ใน w3schools



4) ลองทำโปรแกรมเครื่องคิดเลขแบบง่ายด้วย Python : OOP



5) ศึกษาเพิ่มเติมเรื่องของ Inheritance



The screenshot shows a web browser window with the URL www.w3schools.com/python/python_inheritance.asp. The page is titled "Python Inheritance" and contains the following content:

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.

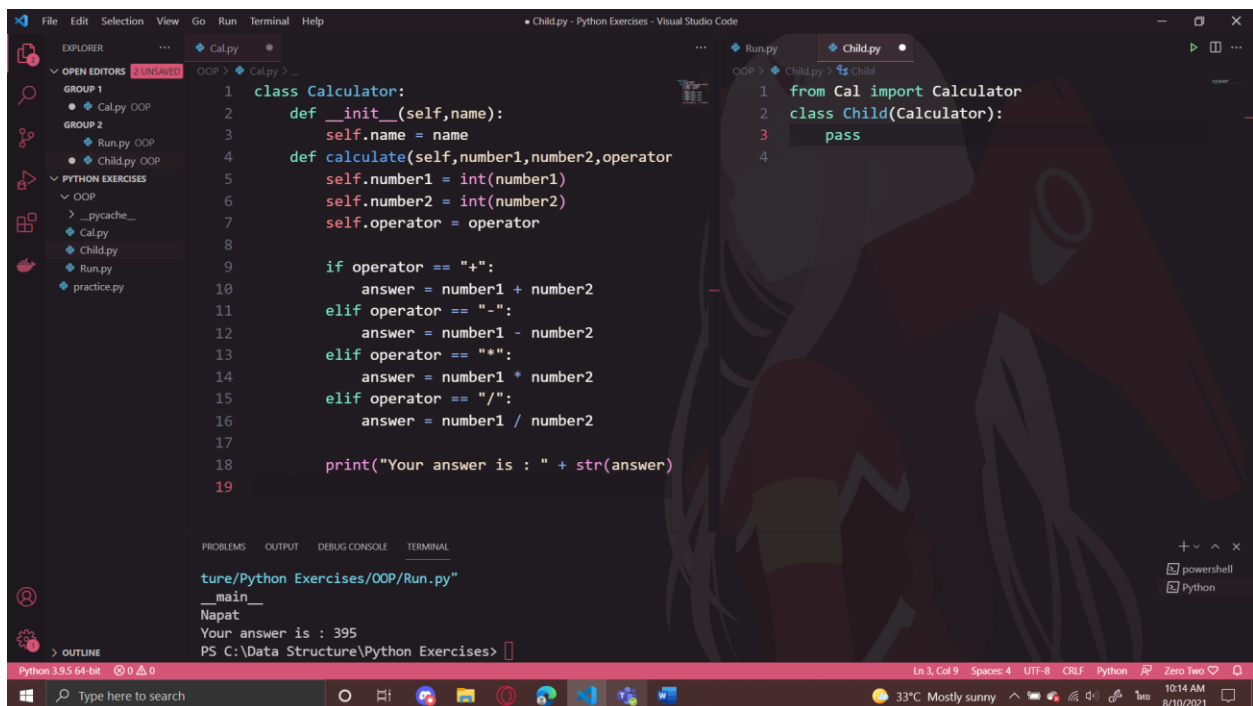
Note: The child's `__init__()` function **overrides** the inheritance of the parent's `__init__()` function.

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)
```

6) ลองทำการ Inherit จาก Class แม่



The screenshot shows a Visual Studio Code editor window with the file `Child.py` open. The code defines a `Calculator` class and a `Child` class that inherits from it.

```
class Calculator:
    def __init__(self, name):
        self.name = name
    def calculate(self, number1, number2, operator):
        self.number1 = int(number1)
        self.number2 = int(number2)
        self.operator = operator
        if operator == "+":
            answer = number1 + number2
        elif operator == "-":
            answer = number1 - number2
        elif operator == "*":
            answer = number1 * number2
        elif operator == "/":
            answer = number1 / number2
        print("Your answer is : " + str(answer))

class Child(Calculator):
    pass
```

The terminal output shows the execution of the code:

```
Python 3.9.5 64-bit
PS C:\Data Structure\Python Exercises>
Your answer is : 395
```

```
File Edit Selection View Go Run Terminal Help
Child.py - Python Exercises - Visual Studio Code

EXPLORER
  OPEN EDITORS
    GROUP 1
      Cal.py
    GROUP 2
      Run.py
  PYTHON EXERCISES
    OOP
      Child.py
      Run.py
      practice.py

Cal.py
1 class Calculator:
2     def __init__(self, name):
3         self.name = name
4     def calculate(self, number1, number2, operator):
5         self.number1 = int(number1)
6         self.number2 = int(number2)
7         self.operator = operator
8
9         if operator == "+":
10            answer = number1 + number2
11        elif operator == "-":
12            answer = number1 - number2
13        elif operator == "*":
14            answer = number1 * number2
15        elif operator == "/":
16            answer = number1 / number2
17
18        print("Your answer is : " + str(answer))
19

Terminal
PS C:\Data Structure\Python Exercises> C:\Users\napat\AppData\Local\Programs\Python\Python39\python.exe "c:/Data Structure/Python Exercises/OOP/Child.py"
Moo
Your answer is : 4.0
PS C:\Data Structure\Python Exercises>
```

```
File Edit Selection View Go Run Terminal Help
Child.py - Python Exercises - Visual Studio Code

EXPLORER
  OPEN EDITORS
    GROUP 1
      Cal.py
    GROUP 2
      Run.py
  PYTHON EXERCISES
    OOP
      Child.py
      Run.py
      practice.py

Child.py
1 from typing import Mapping
2 from Cal import Calculator
3 class Child(Calculator):
4     def __init__(self, fname, lname):
5         self.fname = fname
6         self.lname = lname
7         print("Hi " + fname + " " + lname)
8     if __name__ == "__main__":
9         test = Child("Napat", "Jirarattanakunchai")
10        test.calculate(52, 13, "/")
11
12

Terminal
PS C:\Data Structure\Python Exercises> C:\Users\napat\AppData\Local\Programs\Python\Python39\python.exe "c:/Data Structure/Python Exercises/OOP/Child.py"
Hi Napat Jirarattanakunchai
Your answer is : 4.0
PS C:\Data Structure\Python Exercises>
```

Use the `super()` function

Python also has a `super()` function that will make the child class inherit all the methods and properties from its parent:

Example

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

[Try it Yourself »](#)

By using the `super()` function, you do not have to use the name of the parent element, it will automatically inherit the methods and properties from its parent.

Add Properties

Add a property called `graduationyear` to the `Student` class:

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

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)
```

[Try it Yourself »](#)

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.

Test Yourself With Exercises