

## สรุปการเข้าร่วมอบรมหลักสูตร Python Programming

โครงการพัฒนาระบบคลาวด์กลางภาครัฐ (Government Data Center and Cloud Service: GDCC)

จัดโดย สำนักงานคณะกรรมการดิจิทัลเพื่อเศรษฐกิจและสังคมแห่งชาติ (สดช.)

วันที่ ๑๔ - ๑๖ พฤษภาคม ๒๕๖๗ เวลา ๐๘.๓๐ - ๑๖.๓๐ น.

ณ ห้องแชฟไฟร์ ๑๑๒ อาคารอิมแพ็คฟอรัม เมืองทองธานี จังหวัดนนทบุรี

**วิทยากร** นายกิตติพัฒน์ เจียมรัตน์ ประทีป

**ผู้เข้าร่วมอบรม** นางสาวพัชรวรรณ วงษา และนายปฏิพล เขียววิสัย เจ้าหน้าที่ระบบงานคอมพิวเตอร์  
กลุ่มพัฒนานวัตกรรมดิจิทัล ศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร

### วัตถุประสงค์ของการจัดอบรม

- เพื่อสร้างความเข้าใจเกี่ยวกับพื้นฐานของภาษา Python Programming ที่จำเป็นต่อการจัดการข้อมูล และวิเคราะห์ข้อมูล
- เพื่อเรียนรู้เกี่ยวกับ Library ต่างๆ ของ Python ที่ช่วยต่อการนำไปใช้ในเชิงคณิตศาสตร์และสถิติ ให้สามารถจัดการข้อมูลและทำความสะอาดข้อมูลสำหรับการวิเคราะห์ได้อย่างถูกต้อง
- สามารถนำข้อมูลไป Plot เป็น Visualization ในรูปแบบต่างๆ ได้

### ความรู้พื้นฐานด้านการเขียนโปรแกรมด้วยภาษา Python

#### ความสำคัญของ Python Programming

ภาษา Python เป็นภาษาที่ใช้ในการเขียนโปรแกรมซึ่งพัฒนามาจากภาษาซี มีลักษณะเป็นภาษา สคริปต์ สนับสนุนแนวคิด OOP(Object Oriented Programming) มีความเป็น Open Source และไม่ยึดติดกับแพลตฟอร์ม ทำให้นำไปรันได้บนหลายระบบปฏิบัติการ นอกจากนี้ ยังง่ายต่อการเรียนรู้และใช้งาน รวมทั้งมี Library ให้ใช้งานอย่างหลากหลาย

#### ชนิดข้อมูลและหลักการการเขียนเบื้องต้น

ชนิดข้อมูลมาตรฐานของ Python มีอยู่ ๕ ชนิด ได้แก่ Numbers, String, List, Tuple และ Dictionary ข้อมูลจะเป็นชนิด Dynamic จึงสามารถเปลี่ยนชนิดข้อมูลได้ง่ายและสะดวก และไม่ต้องประกาศตัวแปร ก่อนกำหนดค่า แต่มีความเป็น Case Sensitive จึงควรระวังตัวอักษรพิมพ์ใหญ่ ตัวพิมพ์เล็ก และอักขระพิเศษ รวมไปถึง Reserved Words ส่วนรูปแบบการเขียนโปรแกรมภาษา Python เบื้องต้นนั้น จะแบ่งบล็อกของโค้ดด้วยย่อหน้า ซึ่งต่างจากในหลายๆ ภาษาที่แบ่งด้วยเครื่องหมายวงเล็บปีกกา และใช้เครื่องหมาย # หรือ Triple Quotes (```) สำหรับ Comment คำสั่งใน Python

#### ตัวอย่างคำสั่งและฟังก์ชันพื้นฐาน

input() สำหรับนำเข้าข้อมูล	print() สำหรับแสดงผลออกทางหน้าจอ
open() สำหรับเปิดไฟล์	int() สำหรับแปลงค่าเป็นเลขจำนวนเต็ม
float() สำหรับแปลงค่าเป็นเลขทศนิยม	upper() สำหรับแปลงเป็นตัวอักษรพิมพ์ใหญ่
count() สำหรับนับจำนวน	isdigit() สำหรับตรวจสอบว่าเป็นตัวเลขหรือไม่
len() สำหรับหาความยาวตัวอักษร	isalpha() สำหรับตรวจสอบว่าเป็นตัวอักษรหรือไม่
insert() สำหรับแทรกข้อมูล	split() สำหรับการตัดคำหรือแยกข้อความออกเป็นส่วนๆ

## ตัวอย่างแบบฝึกหัดในการฝึกอบรม

การคำนวณแปลงอุณหภูมิจากองศาเซลเซียสเป็นฟาเรนไฮต์

โค้ด

```
celsius = float(input("Please enter Celsius "))
fahrenheit = (celsius * 9)/5+32
print("%.2f Celsius is equal to %.2f Farenheit" % (celsius,fahrenheit))
```

ผลลัพธ์

```
Please enter Celsius 40
40.00 Celsius is equal to 104.00 Farenheit
```

การหาจำนวนตัวอักษรและตัวเลขในประโยค

โค้ด

```
1  inputString = "10th Anniversary"
2  #print(inputString[5])
3  sumd = 0
4  sumw = 0
5  # digit = alpha = 0
6  for s in inputString:
7      if s.isdigit() == True:
8          sumd = sumd + 1
9      elif s.isalpha() == True:
10         sumw = sumw + 1
11  print("The number of letters are {}".format(sumw))
12  print("The number of digits are {}".format(sumd))
13
```

ผลลัพธ์

```
The number of letters are 13
The number of digits are 2
```

การสร้างฟังก์ชันคำนวณหาจำนวนวัน ชั่วโมง นาที และวินาที จากข้อมูลนำเข้าวินาที

โค้ด

```
1  def ConversionSeconds2Days(seconds):
2      Days = seconds // (24 * 60 * 60)
3      #print(Days)
4      seconds = seconds % (24 * 60 * 60)
5      #print(seconds)
6      Hours = seconds // (60 * 60)
7      #print(Hours)
8      seconds = seconds % (60 * 60)
9      #print(seconds)
10     Minutes = seconds // 60
11     #print(Minutes)
12     seconds = seconds % 60
13     #print(seconds)
14     result = str(Days) + ":" + str(Hours) + \
15         ":" + str(Minutes) + ":" + str(seconds)
16     return result
17     seconds = int(input("Fill out seconds : "))
18     result = ConversionSeconds2Days(seconds)
19     print("Days:Hours:Minutes:Seconds = ",result)
```

ผลลัพธ์

```
Fill out seconds : 123456
Days:Hours:Minutes:Seconds = 1:10:17:36
```

## การสร้างคลาสกระเป๋าเงินและฟังก์ชันฝากถอน

โค้ด

```
1 import datetime
2 class Wallet:
3     __amount = 0
4     def __init__(self,owner):
5         self.owner = owner
6     def checkBalance(self):
7         return self.__amount
8     def deposit(self,currency,depamount):
9         if currency == "TH":
10            self.__amount = self.__amount + depamount
11        elif currency == "USD":
12            self.__amount = self.__amount + (depamount * 35)
13        else:
14            print("Invalid Currency")
15            self.recordTransaction("ERR1",depamount)
16            self.recordTransaction("DEP",depamount)
17            return self.checkBalance()
18
19        def withdraw(self,currency,wamount):
20            w_amountTH = 0
21            if currency == "TH":
22                w_amountTH = wamount
23            elif currency == "USD":
24                w_amountTH = (wamount * 35)
25            else:
26                print("Invalid Currency")
27                self.recordTransaction("ERR1",0)
28            return
29            if self.__amount >= w_amountTH:
```

## การคำนวณหาวันที่ถัดไปจากวันที่กำหนด

โค้ด

```
year = int(input("Input a year: "))
leap_year = False
if year % 400 == 0:
    leap_year = True
elif year % 100 == 0:
    leap_year = False
elif year % 4 == 0:
    leap_year = True
#print(leap_year)
month = int(input("Input a month : "))
max_day = 30
if month in (1,3,5,7,8,10,12):
    max_day = 31
elif month == 2:
    if leap_year:
        max_day = 29
    else:
        max_day = 28
```

```
#print(max_day)
day = int(input("Input a day : "))
if day < max_day:
    day = day + 1
elif day == max_day:
    day = 1
    if month == 12:
        month = 1
        year = year + 1
    else:
        month = month + 1
else:
    print("Date Invalid")
print("The next date is {}-{}-{}".format(day,month,year))
```

ผลลัพธ์

```
Input a year: 2000
Input a month : 3
Input a day : 31
The next date is 1-4-2000
```

ภาพการดำเนินงานอบรม



ภาพที่ ๑ งานอบรม Python Programming  
ที่มา: <https://gocc.gdcc.onde.go.th/news/48>