


TRƯỜNG: ĐH Công thương TPHCM KHOA: CÔNG NGHỆ THÔNG TIN BỘ MÔN: KHDL&TTNT MH: LẬP TRÌNH PYTHON	BUỔI 1. GIỚI THIỆU CĂN BẢN VỀ PYTHON	
--	---	---

A. MỤC TIÊU

- Tạo một chương trình viết bằng ngôn ngữ Python.
- Sử dụng được các hàm xuất, nhập chuẩn; các kiểu dữ liệu; khai báo biến; các phép toán số học, logic.
- Áp dụng được các cấu trúc rẽ nhánh, vòng lặp vào các bài toán cụ thể.

B. DỤNG CỤ - THIẾT BỊ THÍ NGHIỆM CHO MỘT SV

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

I. Tóm tắt lý thuyết

1. Giới thiệu

a. Sử dụng Google Colab để viết chương trình:

B1. Truy cập địa chỉ: <https://colab.research.google.com/>

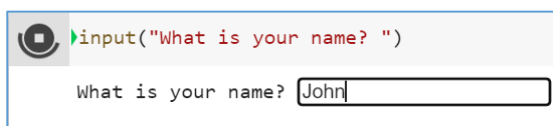
B2. Vào file → new notebook



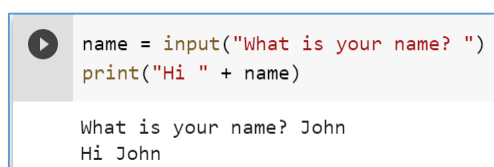
B3. Kết nối máy chủ bằng cách click chuột vào Connect như trên hình.

b. Nhập, xuất trong Python:

Nhập trong Python: **Cú pháp: input("chuỗi ký tự: ")**. Ví dụ:



Xuất trong Python: **Cú pháp: print(...)**. Ví dụ:



c. Biến:

Khái niệm: Biến là vùng chứa để lưu trữ các giá trị dữ liệu.

Tạo biến:

- Python không có lệnh để khai báo một biến.
- Một biến được tạo ngay khi bạn gán giá trị cho nó lần đầu tiên.
- Các biến không cần phải được khai báo với bất kỳ kiểu cụ thể nào và thậm chí có thể thay đổi kiểu sau khi chúng đã được thiết lập.
- Một biến có thể có tên ngắn (như x và y) hoặc tên mô tả hơn (age, carname, total, ...). Quy tắc cho các biến Python:
 - + Tên biến phải bắt đầu bằng một chữ cái hoặc ký tự gạch dưới
 - + Tên biến không thể bắt đầu bằng số
 - + Tên biến chỉ có thể chứa các ký tự chữ-số và dấu gạch dưới (A-z, 0-9 và _)
 - + Tên biến có phân biệt chữ hoa chữ thường (tuổi, Tuổi và TUỔI là ba biến khác nhau)

Ví dụ:

Khai báo biến	Kiểu dữ liệu	Ghi chú
<code>soluong = 10</code>	<code>type(soluong) → int</code>	#khai báo biến số lượng và gán trị là 10.
<code>dongia = 15.50</code>	<code>type(dongia) → float</code>	#khai báo biến đơn giá và gán trị là 15.50
<code>tensanpham = "lemon" hoặc tensanpham = 'lemon'</code>	<code>Type(tensanpham) → str</code>	#khai báo biến tên sản phẩm và gán trị là một chuỗi ký tự 'lemon'
<code>online = True</code>	<code>Type(bool) → bool</code>	#khai báo biến online và gán trị là True

Nhiều giá trị đến nhiều biến:

Python cho phép bạn gán giá trị cho nhiều biến trong một dòng:

```
x, y, z = "Orange", "Banana", "Cherry"
print(x)
print(y)
print(z)
```

Hoặc bạn cũng có thể gán một giá trị cho nhiều biến :

```
x = y = z = "Orange"
print(x)
print(y)
print(z)
```

Nếu ta có một tập hợp giá trị trong một danh sách hoặc một tuple, ...Python cho phép trích xuất mỗi giá trị thành một biến.

```
fruits = ["apple", "banana", "cherry"]
x, y, z = fruits
print(x)
print(y)
print(z)
```

Biến toàn cục:

Các biến được tạo bên ngoài một hàm (như trong tất cả các ví dụ ở trên) được gọi là các biến toàn cục.

Các biến toàn cục có thể được sử dụng bởi tất cả mọi người, cả bên trong hàm và bên ngoài hàm. Ví dụ :

```
x = "awesome"

def myfunc():
    print("Python is " + x)

myfunc()
```

Python is awesome

– Nếu bạn tạo một biến có cùng tên bên trong một hàm, thì biến này sẽ là cục bộ và chỉ có thể được sử dụng bên trong hàm. Biến toàn cục có cùng tên sẽ vẫn như cũ, toàn cục và với giá trị ban đầu. Ví dụ :

```
x = "awesome"

def myfunc():
    x = "fantastic"
    print("Python is " + x)

myfunc()

print("Python is " + x)
```

Python is fantastic
Python is awesome

– Thông thường, khi bạn tạo một biến bên trong một hàm, biến đó là cục bộ và chỉ có thể được sử dụng bên trong hàm đó.

– Để tạo một biến toàn cục bên trong một hàm, bạn có thể sử dụng từ khóa toàn cục.

Ví dụ :

```
def myfunc():
    global x
    x = "fantastic"

myfunc()

print("Python is " + x)
```

Các phép toán:

Toán tử số học: Toán tử số học trong python được thể hiện dưới 7 dạng cơ bản sau: (trong các ví dụ dưới đây thì ta coi a có giá trị là 5 và b có giá trị là 7).

Toán tử	Mô Tả	Ví Dụ
+	Toán tử cộng các giá trị lại với nhau	$a + b = 12$
-	Toán tử trừ các giá trị lại với nhau	$a - b = -2$
*	Toán tử nhân các giá trị lại với nhau	$a * b = 42$

/	Toán tử chia các giá trị cho nhau	$a / b = 0.7142857142857143$
%	Toán tử chia lấy phần dư	$a \% b = 5$
**	Toán tử mũ. $a ** b = ab$	$a ** b = 78125$
//	Toán tử chia làm tròn xuống. VD: $0,57 \Rightarrow 0$ $0.9 \Rightarrow 0$ $-07 \Rightarrow -1$ $-0.1 \Rightarrow -1$	$a // b = 0$

Toán tử quan hệ:

<, <=, >, >=, ==, !=	Các phép toán so sánh
----------------------	-----------------------

Toán tử gán:

Toán Tử	Ví Dụ
=	$c = a$
+=	$c += a \rightarrow c = c + a$
-=	$c -= a \rightarrow c = c - a$
*=	$c *= a \rightarrow c = c * a$
/=	$c /= a \rightarrow c = c / a$
%	$c \% a \rightarrow c = c \% a$
**=	$c ** a \rightarrow c = c ** a$
//=	$c //= a \rightarrow c = c // a$

Toán tử logic: Toán tử logic trong Python hoàn toàn giống như các ngôn ngữ khác. Nó gồm có 3 kiểu cơ bản như sau: and, or, not

2. Kiểu dữ liệu:

Trong lập trình, kiểu dữ liệu là một khái niệm quan trọng. Các biến có thể lưu trữ dữ liệu thuộc nhiều loại khác nhau và các loại khác nhau có thể làm những việc khác nhau. Python có các kiểu dữ liệu sau được tích hợp sẵn theo mặc định, trong các danh mục sau:

Text Type:	str
Numeric Types:	int, float, complex
Sequence Types:	list, tuple, range
Mapping Type:	dict
Set Types:	set, frozenset
Boolean Type:	bool
Binary Types:	bytes, bytearray, memoryview
None Type:	NoneType

Trong Python, kiểu dữ liệu được đặt khi bạn gán giá trị cho một biến:

Example	Data type
x = "Hello World"	str
x = 20	int
x = 20.5	float
x = 1j	complex
x = ["apple", "banana", "cherry"]	list
x = ("apple", "banana", "cherry")	tuple
x = range(6)	range
x = {'name': "John", 'age': 36}	dict
x = {"apple", "banana", "cherry"}	set
x = frozenset(("apple", "banana", "cherry"))	frozenset
x = True	bool

Xét ví dụ nhập vào năm sinh của bạn, tính tuổi và xuất ra màn hình.

```
birth_year = input("Enter your birth year: ")
age = 2022 - birth_year
print(age)
```

Enter your birth year: 1980

TypeError Traceback (most recent call last)

<ipython-input-15-585fa75571e4> in <module>

1 birth_year = input("Enter your birth year: ")

----> 2 age = 2022 - birth_year

3 print(age)

TypeError: unsupported operand type(s) for -: 'int' and 'str'

Hàm input luôn nhận giá trị chuỗi do đó để tính được tuổi ta phải chuyển giá trị của biến *birth_year* sang kiểu cùng kiểu *int* với năm 2022. Khi đó: **age = 2022 – int(birth_year)**

Cú pháp: <kiểu cần chuyển> <tên biến hoặc hằng>

Câu lệnh	Ý nghĩa
int(value)	Chuyển value sang số nguyên.
float(value)	Chuyển value sang số thực.
bool(value)	Chuyển value sang boolean .
str(value)	Chuyển value sang chuỗi.

3. Các câu lệnh điều kiện

a. Lệnh if đơn không có else :

```
x = int(input('Enter a value x = '))
if x < 0:
    print("It's negative")
```

Enter a value x = -10
It's negative

b. Lệnh if đơn có else :

```
x = int(input('Enter a value x = '))
if x < 0:
    print("It's negative")
else:
    print("It's positive")
```

Enter a value x = 10
It's positive

c. Lệnh if lồng nhau:

```
x = int(input('Enter a value x = '))
if x < 0:
    print("It's negative")
elif x == 0:
    print("Equal to zero")
elif x <= 5:
    print("Positive but smaller than or equal to 5")
else:
    print("Positive and larger than 5")
```

Enter a value x = 10
Positive and larger than 5

4. Vòng lặp

a. For: Cấu trúc chuẩn của vòng lặp for:

```
for value in collection:
    # do something with value
```

Ví dụ:

```
for i in range(5):
    print(i)
```

0
1
2
3
4

Tính tổng các giá trị khác None trong list: <pre>sequence = [1, 2, None, 4, None, 5] total = 0 for value in sequence: if value is None: continue total += value</pre>	Tính tổng các giá trị trong list, nếu giá trị là 5 thì thoát khỏi vòng lặp: <pre>sequence = [1, 2, 0, 4, 6, 5, 2, 1] total_until_5 = 0 for value in sequence: if value == 5: break total_until_5 += value</pre>
--	--

Viết chương trình in ra các cặp số $(0,0)$, $(1,0)$, $(1,1)$, $(2,0)$, $(2,1)$, $(2,2)$, $(3,0)$, $(3,1)$, $(3,2)$, $(3,3)$:

```
for i in range(4):
    for j in range(4):
        if j > i:
            break
        print((i, j))
```

b. Cấu trúc chuẩn của vòng lặp while:

```
while điều_kiện:
    Khối lệnh của while
```

Ví dụ:

In lần lượt các số nhỏ hơn 8 :

```
#In và đếm các số từ 0 tới 8:

count = 1
n = 0
while (n < 8):
    print ('Số thứ', count, ' là:', n)
    n = n + 1
    count = count + 1
```

Tính tổng các số :

```
n = int(input("Nhập n: ")) #Nhập số n tùy ý
tong = 0 #khai báo và gán giá trị cho tong
i = 1 #khai báo và gán giá trị cho biến đếm i

while i <= n:
    tong = tong + i
    i = i+1 # cập nhật biến đếm

print("Tổng là", tong)
```

Kết quả : Nhập n: 11 → Tổng là 66

Lệnh break trong while: Với câu lệnh break, chúng ta có thể dừng vòng lặp ngay cả khi điều kiện của while là True:

Ví dụ: Thoát vòng lặp khi i bằng 3:

```
i = 1
while i < 6:
    print(i)
    if i == 3: #kiểm tra điều kiện xem i bằng 3 hay chưa
        break
    i += 1 #cập nhật biến đếm
```

Lệnh continue trong while: Câu lệnh continue trong while sẽ khiến cho vòng lặp bỏ qua lần lặp hiện tại và tiếp tục chạy ở lần lặp tiếp theo.

Ví dụ: In các số từ 1 đến 6 ngoại trừ số 3

```
i = 0
while i < 6:
    i += 1
    if i == 3:
        continue
    print(i)
```

Kết hợp while với else: Giống như vòng lặp for, bạn cũng có thể kết hợp else với while. Trong trường hợp này, khối lệnh của else sẽ được thực hiện khi điều kiện của while là False.

Ví dụ: Minh họa việc sử dụng while kết hợp với else

```
dem = 0
while dem < 3:
    print("Đang ở trong vòng lặp while")
    dem = dem + 1
else:
    print("Đang ở trong else")
```

Đếm và in các số nhỏ hơn 2

```
n = 0
while n < 2:
    print(n, "nhỏ hơn 2")
    n = n + 1
else:
    print(n, "không nhỏ hơn 2")
```

5. Range: hàm range() trả về một tiến trình lặp nhằm tạo ra một chuỗi các số nguyên cách đều nhau. Ví dụ:

```
In [122]: range(10)
Out[122]: range(0, 10)

In [123]: list(range(10))
Out[123]: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9]

In [124]: list(range(0, 20, 2))
Out[124]: [0, 2, 4, 6, 8, 10, 12, 14, 16, 18]

In [125]: list(range(5, 0, -1))
Out[125]: [5, 4, 3, 2, 1]
```

Viết chương trình tính tổng các số từ 1 đến 100.

```
sum = 0
for i in range(101):
    sum += i
print(sum)
```

5050

6. List:

<tên biến> = <[M1, M2, ..., Mn]> , Mi có thể cùng kiểu dữ liệu hoặc khác kiểu.

Ví dụ :

```
list1 = ["apple", "banana", "cherry"]  
list2 = [1, 5, 7, 9, 3]  
list3 = [True, False, False]
```

```
list1 = ["abc", 34, True, 40, "male"]
```

- Chiều dài list được xác định qua hàm len()

```
thislist = ["apple", "banana", "cherry"]  
print(len(thislist))
```

- Chúng ta cũng có thể sử dụng cấu trúc list() để tạo ra một list mới.

```
thislist = list(("apple", "banana", "cherry")) # note the double round-brackets  
print(thislist)
```

- Truy xuất đến các item của list

```
thislist = list(("banana", "apple", 'mango'))  
print(thislist[1])  
print(thislist[-1])  
print(thislist[:2])  
print(thislist[-3:-1])
```

```
apple  
mango  
['banana', 'apple']  
['banana', 'apple']
```

- Lặp thông qua list

```
thislist = ["apple", "banana", "cherry"]  
for x in thislist:  
    print(x)
```

- Lặp thông qua các số chỉ mục

```
thislist = ["apple", "banana", "cherry"]  
for i in range(len(thislist)):  
    print(thislist[i])
```

Các phương thức của list:

Method	Description
<code>append()</code>	Adds an element at the end of the list
<code>clear()</code>	Removes all the elements from the list
<code>copy()</code>	Returns a copy of the list
<code>count()</code>	Returns the number of elements with the specified value
<code>extend()</code>	Add the elements of a list (or any iterable), to the end of the current list
<code>index()</code>	Returns the index of the first element with the specified value
<code>insert()</code>	Adds an element at the specified position
<code>pop()</code>	Removes the element at the specified position
<code>remove()</code>	Removes the item with the specified value
<code>reverse()</code>	Reverses the order of the list
<code>sort()</code>	Sorts the list

II. Bài tập hướng dẫn mẫu

1. Viết chương trình nhập và xuất ra tên, tuổi của một khách hàng.

```
name = input("Your name is: ")
age = input("Age : ")
print(f"\nMy name is {name}")
print("\nAge = " + age)
```

```
Your name is: Lan
Age : 42
```

```
My name is Lan
```

```
Age = 42
```

2. Viết chương trình tính tiền thuê Taxi. Biết rằng, mỗi khách hàng tính 2£ và 1.5£ cho 1km.

```
def taxi():
    passengers = (int)(input("\nNhập vào số người: "))
    distance = (float)(input("\nhập vào số km: "))
    total = 2*passengers + 1.5* distance
    print("\nTotal: ", total)
```

```
taxi()
```

```
Nhập vào số người: 4
```

```
hập vào số km: 3
```

```
Total: 12.5
```

3. Viết chương trình nhập vào một chuỗi ký tự. Xuất tất cả các ký tự ở vị trí chẵn.

```
string = input("\nstring enter: ")
print(string[::2])
```

```
string enter: abcdefg
aceg
```

4. Viết chương trình tính tổng các số từ 0 đến n mà là bội của 3 hoặc 5.

```

sum = 0
n = 11
for i in range(n):
    if i%3 == 0 or i%5 == 0:
        sum += i
print(f"\nTong cac so tu 0 den {n} ma la boi 3 hoac 5 la: {sum}")

```

Tong cac so tu 0 den 11 ma la boi 3 hoac 5 la: 33

III. Bài tập ở lớp

1. Viết chương trình nhập vào 2 giá trị a, b và tính tổng, hiệu, tích thương của a và b.
2. Lấy các ký tự từ chỉ số 2 đến chỉ số 4 (llo) của chuỗi "Hello World".
3. Xóa khoảng trắng thừa ở đầu và cuối chuỗi " Hello World ". Chuyển chuỗi sang dạng hoa và thường. Thay ký tự H thành ký tự J.
4. Viết chương trình giải phương trình bậc nhất $aX+b=0$
5. Viết chương trình giải phương trình bậc 2: $aX^2+bX+c=0$
6. Viết chương trình tính tổng các số từ 0 đến 999 mà là bội của 3 hoặc 5.
7. Tính tổng các số chẵn của n ($s = 2+4+6+\dots+n$)
8. Yêu cầu người dùng nhập nhiệt độ bằng độ C. Chương trình sẽ in một tin nhắn dựa trên về nhiệt độ:
 - Nếu nhiệt độ nhỏ hơn -273,15, in ra rằng nhiệt độ không hợp lệ vì nó dưới độ không tuyệt đối.
 - Nếu chính xác là -273,15, in ra nhiệt độ tuyệt đối bằng 0.
 - Nếu nhiệt độ nằm trong khoảng từ -273,15 đến 0, in ra nhiệt độ dưới điểm đóng băng.
 - Nếu bằng 0, in ra nhiệt độ đang ở điểm đóng băng.
 - Nếu nằm trong khoảng từ 0 đến 100, in ra nhiệt độ ở mức bình thường.
 - Nếu là 100, in ra nhiệt độ đang ở điểm sôi.
 - Nếu trên 100, in ra nhiệt độ cao hơn điểm sôi.
9. Trong học chế tín chỉ, thang điểm đánh giá được chia thành điểm không đạt và điểm đạt.
 - **Điểm không đạt:** 0 - 3,9 thang điểm 10 tương đương điểm F= 0 thang điểm 4.
 - **Điểm đạt:**
 - + 4,0 -5,4 thang điểm 10 tương đương điểm D =1 thang điểm 4.
 - + 5,5 -6,9 thang điểm 10 tương đương điểm C= 2 thang điểm 4.
 - + 7,0 -8,4 thang điểm 10 tương đương điểm B =3 thang điểm 4.
 - + 8,5 - 10 thang điểm 10 tương đương điểm A= 4 thang điểm 4.

Viết chương trình nhập vào điểm của sinh viên sau đó hãy cho biết sinh viên đó có đạt hay không đạt và điểm theo thang điểm 4 thuộc điểm nào?

Bài tập về nhà

1. Tính tổng các số lẻ ($s = 1+3+5+\dots+n$)
2. Tính tích các số chẵn (tích = $2 \times 4 \times 6 \times \dots \times n$)
3. Tính tích các số lẻ (tích = $1 \times 3 \times 5 \times \dots \times n$)
4. Tính $n!$ (n giai thừa)
5. Nhập số nguyên n . Viết chương trình để tạo ra một dãy số chứa ($i : i*i$) như là số nguyên từ 1 đến n (bao gồm cả 1 và n) sau đó in ra dãy số này. Ví dụ: số n là 8 thì đầu ra sẽ là: 1: 1 2: 4 3: 9 4: 16 5: 25 6: 36 7: 49 8: 64
6. Viết chương trình tìm tất cả các số chia hết cho 7 nhưng không phải bội số của 5, nằm trong đoạn 2000 và 3200 (tính cả 2000 và 3200). Các số thu được sẽ được in thành chuỗi trên một dòng, cách nhau bằng dấu phẩy.
7. Một cửa hàng tính phí 12 USD cho mỗi mặt hàng nếu bạn mua ít hơn 10 mặt hàng. Nếu bạn mua từ 10 đến 99 các mặt hàng, chi phí là \$10 cho mỗi mặt hàng. Nếu bạn mua 100 món trở lên, giá là 7\$ một món. Viết một chương trình hỏi người dùng họ đang mua bao nhiêu mặt hàng và in tổng chi phí.
8. Những năm nào chia hết cho 4 được và không chia hết cho 100 được coi là năm nhuận (ví dụ năm 2100 không phải là năm nhuận, 2104 là năm nhuận). Viết chương trình hỏi người dùng một năm và in ra xem đó có phải là năm nhuận hay không.
9. Hãy dùng vòng lặp in ra các dấu * theo mẫu:

Nhập vào chiều rộng và chiều cao. In ra hình chữ nhật theo mẫu:

```
*****
*****
*****
*****
```

Nhập vào chiều rộng và chiều cao. In ra hình chữ nhật theo mẫu:

```
*****
*                                     *
*                                     *
*****
```

In một hình tam giác như hình bên. Cho phép người dùng chỉ định mức độ cao hình tam giác:

```
*
**
***
****
```

In hình thoi như hình bên, cho phép người dùng chỉ định mức độ cao kim cương:

```

*
**
***
****
*****
****
***
**
*
```