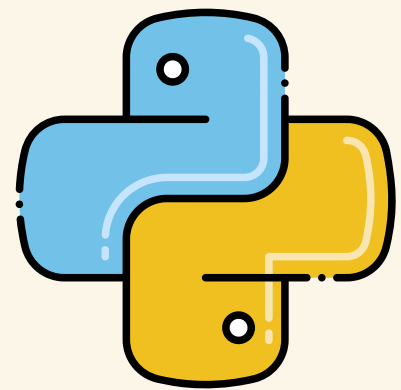




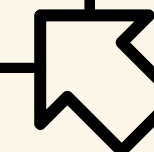
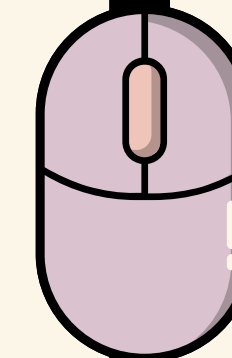
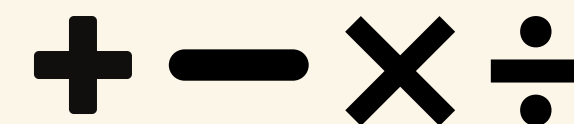
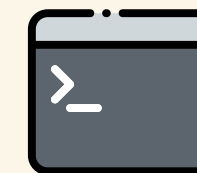
GIÁO TRÌNH LẬP TRÌNH PYTHON

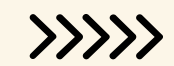
Phần Nâng Cao
(trẻ em)

Khám phá thế giới lập trình với ngôn ngữ Python



Nguyễn Duy Thiên





MỤC LỤC



01 CÁC THAO TÁC VỚI KIỂU DỮ LIỆU
STRING

02 KIỂU DỮ LIỆU LIST, DICTIONARY

03 ĐỊNH NGHĨA VÀ CÁC THAO TÁC
VỚI HÀM

04 SỬ DỤNG CÁC THƯ VIỆN CƠ BẢN
CÓ SẴN: RANDOM, MATH,
DATETIME

05 THƯ VIỆN TURTLE

CHƯƠNG 1: CÁC THAO TÁC VỚI KIỂU DỮ LIỆU STRING

A. Mục đích nghiên cứu

- Tạo, truy cập và thao tác chuỗi cơ bản.
- Sử dụng các phương thức xử lý chuỗi để làm sạch, định dạng, tìm kiếm và thay thế văn bản.
- Định dạng chuỗi một cách chuyên nghiệp bằng f-string và .format().

B. Nội dung nghiên cứu

Bài 1. Cơ bản về chuỗi (String)

Bài 1. Các phương thức xử lý chuỗi thường dùng

Bài 1. Định dạng chuỗi (String Formatting)

C. Bài tập tổng hợp

Bài 1. Cơ bản về chuỗi (String)

a. Chuỗi là gì?

- Chuỗi là dãy các ký tự (chữ, số, dấu câu, khoảng trắng...) được đặt trong dấu nháy đơn ('...') hoặc nháy kép ("...").
- Chuỗi là immutable: không thể thay đổi ký tự trực tiếp.

b. Tạo chuỗi

```
Python NC > chương1 > bai1.py > multi_line
1  name = "Lan" # Dùng cặp nháy kép để tạo chuỗi
2
3  message = 'Chào bạn!' # Dùng cặp nháy đơn để tạo chuỗi
4
5  # Dùng 3 nháy kép """ để tạo chuỗi xuống dòng
6  multi_line = """Dòng 1
7  Dòng 2"""
```

c. Ký tự đặc biệt

- `\n` → xuống dòng
- `\t` → tab
- `\"` hoặc `\'` → chèn dấu nháy trong chuỗi
- `\\` → in dấu gạch chéo trong chuỗi

d. Truy cập qua vị trí

```
Python NC > chương1 > bai1.py > ...
1  s = "Python"
2  # Danh sách kí tự trong chuỗi bắt đầu từ vị trí 0
3  print(s[0])      # lấy kí tự đầu tiên qua vị trí 0 là 'P'
4  print(s[-1])     # lấy kí tự cuối qua vị trí -1 là 'n'
```

Bài 2. Các phương thức xử lý chuỗi thường dùng

a. Chuyển đổi chữ hoa/thường:

- `s.upper()` → in hoa tất cả
- `s.lower()` → in thường tất cả
- `s.capitalize()` → viết hoa chữ cái đầu
- `s.title()` → viết hoa đầu mỗi từ

```
bai2.py x
Python NC > chương1 > bai2.py > ...
1 hello = 'hello'
2 print(hello.upper()) # in ra HELLO
3
4 python = 'PYTHON'
5 print(python.lower()) # in ra python
6
7 dochoi = "xe ô tô đồ chơi"
8 print(dochoi.capitalize()) # in ra Xe ô tô đồ chơi (chữ X được viết hoa)
9
10 ten = 'nguyen van an'
11 print(ten.title()) # in ra Nguyen Van An
```

b. Loại bỏ khoảng trắng và kiểm tra độ dài:

- `s.strip()` → bỏ khoảng trắng đầu/cuối
- `len(s)` → trả về độ dài của chuỗi

```
bai2.py x
Python NC > chương1 > bai2.py > ...
1 ten = "    Nguyen Van A    "
2 ten_da_xoa_khoang_trang = ten.strip()
3 print(ten_da_xoa_khoang_trang) # In ra Nguyen Van A đã được xóa khoảng trắng ở đầu và cuối
4 print(len(ten_da_xoa_khoang_trang)) # In ra 12
```

c. Tìm kiếm và thay thế:

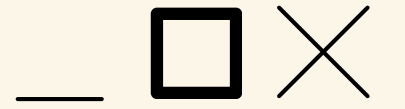
- `s.find(sub)` → trả về vị trí đầu tiên của sub, hoặc -1 nếu không tìm thấy
- `s.replace(old, new)` → thay thế tất cả old bằng new

```
Python NC > chương2 > bai2.py > ...
1  hoc_python = "Học mã python"
2
3  # tìm vị trí của kí tự t
4  vi_tri_t = hoc_python.find("t")
5  print(vi_tri_t) # in ra 9
6
7  # thay thế mã thành code
8  hoc_python_moi = hoc_python.replace("mã", "code")
9  print(hoc_python_moi) # in ra Học code python
```

d. Kiểm tra nội dung:

- `s.isalpha()` → kiểm tra toàn bộ chuỗi có là chữ cái hay không
- `s.isdigit()` → kiểm tra toàn bộ chuỗi có là số hay không
- `s.isalnum()` → kiểm tra toàn bộ chuỗi có vừa số và chữ hay không

```
Python NC > chương1 > bai2.py > ...
1  saoviet = "saoviet" # chỉ chứa chữ
2  print(saoviet.isalpha()) # in ra True
3
4  hotline = "0812114345" # chỉ chứa số
5  print(hotline.isdigit()) # in ra True
6
7  saoviet_hotline = "abc0812114345" # vừa chứa số vừa chứa chữ
8  print(saoviet_hotline.isalnum()) # in ra True
```



Bài 3. Định dạng chuỗi (String Formatting)

a. Phương thức `.format()`

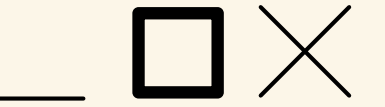
```
Python NC > bai3.py
1 print("Xin chào, {}! Bạn {} tuổi.".format("Lan",14))
2 # format truyền giá trị qua dấu {} (phải theo đúng thứ tự)
3 print("Tên: {name}, Tuổi: {age}".format(age=13, name="Nam"))
4 # có thể dùng biến để truyền giá trị (không cần đúng thứ tự)
```

b. f-string

```
Python NC > bai3.py
1 name = "Lan"
2 age = 14
3 print(f"Xin chào, {name}! Bạn {age} tuổi.")
4 # Truyền biến trực tiếp vào chuỗi string (thêm chữ f vào trước chuỗi)
```

c. Định dạng số trong f-string:

```
Python NC > bai3.py
1 pi = 3.14159
2 print(f"Pi ≈ {pi:.2f}") # 3.14 (:.2f là 2 số thập phân)
3 price = 123456
4 print(f"Giá: {price:,} VNĐ") # "Giá: 123,456 VNĐ" (thêm dấu phân cách phần nghìn)
```



Bài tập tổng hợp

Bài tập 1: Kiểm tra tên

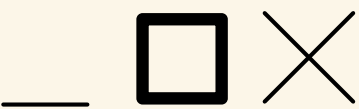
- Viết chương trình yêu cầu người dùng nhập tên
- Kiểm tra tên có chứa số và khoảng trống hay không

Bài tập 2: Chuẩn hóa họ tên

- Em hãy viết chương trình:
- Cho biến `ten_nguoi_dung` = " tên của em "
- Xóa các dấu cách thừa ở đầu và cuối chuỗi.
- Viết hoa chữ cái đầu của mỗi từ trong tên
- In kết quả ra màn hình.

Bài tập 3: Kiểm tra mật khẩu

- Viết chương trình yêu cầu người dùng nhập mật khẩu, sau đó kiểm tra:
- Nếu mật khẩu dài dưới 8 ký tự → in ra "Mật khẩu quá ngắn!"
- Nếu mật khẩu dài từ 8 ký tự trở lên → in ra "Mật khẩu hợp lệ!"



Kết quả ví dụ

Bài tập 1

```
Nhập họ tên: NguyenVanAn1
Tên người dùng có số và không có khoảng trắng
Nhập họ tên: Nguyen Van An
Tên người dùng không có số và có khoảng trắng
```

Bài tập 3

```
Nhập mật khẩu: Abc1234
MẬT KHẨU QUÁ YẾU
Nhập mật khẩu: Abc12345678
MẬT KHẨU MẠNH
```

Bài tập 2

```
Tên người dùng:      nguyên văn an
Tên người dùng sau khi xóa dấu cách thừa: nguyên văn an
Tên người dùng sau khi viết hoa chữ cái đầu: Nguyễn Văn An
```

CHƯƠNG 2: KIỂU DỮ LIỆU LIST, DICTIONARY

A. Mục đích nghiên cứu

- Tạo, truy cập, thêm/sửa/xóa dữ liệu trong list và dictionary.
- Áp dụng các phương thức và kỹ thuật xử lý nâng cao.
- Lựa chọn đúng cấu trúc dữ liệu phù hợp với từng bài toán thực tế (quản lý danh sách, lưu thông tin có nhãn, phân tích dữ liệu đơn giản...).

B. Nội dung nghiên cứu

Bài 4. Kiểu dữ liệu list

Bài 5. Kiểu dữ liệu dictionary

C. Bài tập tổng hợp

Bài 4. Kiểu dữ liệu list

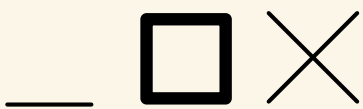
a. Khái niệm

list là một danh sách có thứ tự, có thể thay đổi giá trị, có thể lưu trữ các giá trị có cùng hoặc khác kiểu dữ liệu

```
a = [2, 5, 10, 1, 3]
```

vị trí	0	1	2	3	4
danh sách a	2	5	10	1	3





b. Tạo và truy cập

Tạo danh sách bằng cách gán dữ liệu và in giá trị thông qua vị trí

```
Python NC > chương2 > bai4.py > ...
1  fruits = ["táo", "chuối", "cam"]
2  print(fruits[0])      # "táo"
3  print(fruits[-1])     # "cam" (truy cập giá trị cuối)
```

Tạo danh sách với vòng lặp

```
Python NC > chương2 > bai4.py > ...
1  ds_a = [] # Khởi tạo danh sách trống
2  n = int(input("Nhập độ dài danh sách: "))
3  for i in range(n): # Tạo vòng lặp từ 0 đến n - 1
4      gia_tri = int(input("Nhập giá trị: ")) # Nhập giá trị từ bàn phím
5      ds_a.append(gia_tri) # Thêm giá trị vừa nhập vào danh sách
6  print(ds_a) # in ra danh sách sau khi nhập các giá trị
```

Truy cập từng phần tử trong danh sách với vòng lặp

1. Truy cập qua từng giá trị trong danh sách

```
Python NC > chương2 > bai4.py > ...
1  ds_a = [10,1,95,2,3] # Khởi tạo danh sách
2  for gia_tri in ds_a: # Lặp qua từng phần tử trong danh sách
3      print(gia_tri) # In phần tử ra màn hình
```

2. Truy cập các giá trị qua vị trí

```
Python NC > chương2 > bai4.py > ...
1  ds_a = [10,1,95,2,3] # Khởi tạo danh sách
2  for i in range(0,len(ds_a)): # Lặp qua từng vị trí trong danh sách
3      print(ds_a[i]) # In phần tử ở từng vị trí ra màn hình
```



Bài 5. Kiểu dữ liệu dictionary

a. Khái niệm

dict (dictionary) là cấu trúc lưu dữ liệu dưới dạng cặp khóa–giá trị (**key: value**).

→ Khóa (key) phải là bất biến không thể thay đổi, giá trị (value) có thể là bất kỳ kiểu nào

Cú pháp

```
dict = {  
    key1: value1,  
    key2: value2,  
    ...  
}
```

b. Tạo và truy cập giá trị thông qua key

```
bai5.py x  
Python NC > chuong2 > bai5.py > ...  
1 student = {"name": "Lan", "age": 14, "grade": 8}  
2 print(student["name"]) # In ra giá trị "Lan" thông qua key "name"
```

c. Các phương thức

```
bai5.py x  
Python NC > chuong2 > bai5.py > ...  
1 student = {"name": "Lan", "age": 14, "grade": 8}  
2  
3 student["name"] = "Thông" # sửa giá trị của key "name" thành Thông  
4  
5 del student["grade"] # xóa key "grade" và giá trị
```

Duyệt qua các cặp key-value

1. Truy cập qua từng key

```
Python NC > chương2 > bai5.py > ...  
1  student = {"name": "Lan", "age": 14, "grade": 8}  
2  
3  for key in student: # duyệt qua từng key trong student  
4      print(student[key]) # in từng giá trị thông qua key
```

2. Truy cập qua từng key và value

```
Python NC > chương2 > bai5.py > ...  
1  student = {"name": "Lan", "age": 14, "grade": 8}  
2  
3  for key,value in student.items(): # duyệt qua từng key và value trong student  
4      print(key) # in key  
5      print(value) # in value
```

Bài tập tổng hợp

Bài tập 1: Quản lý thông tin học sinh

- Viết chương trình đơn giản lưu thông tin: ten, tuoi, lop.
- In ra các thông tin.

Bài tập 2: Tìm và sắp xếp số

- Viết chương trình xử lý một danh sách số nguyên:
 1. Nhập các phần tử vào danh sách
 2. Tìm và in ra màn hình số nhỏ nhất và lớn nhất trong danh sách
 3. Sắp xếp danh sách theo thứ tự tăng dần và in ra màn hình danh sách sau khi được sắp xếp

Bài tập 3: Danh sách mua đồ chơi

- Viết chương trình tạo danh sách chứa các đồ chơi, mỗi mặt hàng có ten và so_luong (kết hợp list và dictionary)
- Chương trình xử lý:
 1. Thêm mặt hàng (nhập tên và số lượng sau đó lưu vào dictionary)
 2. Xem danh sách
 3. Thoát

Kết quả ví dụ

Bài tập 1

```
Nhap ten: Thien
Nhap tuoi: 25
Nhap lop: Python
=== THÔNG TIN HỌC SINH ===
Ten: Thien
Tuoi: 25
Lop: Python
```

Bài tập 2

```
Nhap so thu 1: 5
Nhap so thu 2: 1
Nhap so thu 3: 6
Nhap so thu 4: 7
Nhap so thu 5: 10
So nho nhat: 1
So lon nhat: 10
Danh sach sap xep tang dan: [1, 5, 6, 7, 10]
```

Bài tập 3

```
=== MENU ===
1. Them do choi
2. Xem danh sach
3. Thoat
Chon chuc nang (1-3): 1
Ten mat hang: robot
So luong: 2
Da them robot vao danh sach.
```

CHƯƠNG 3: ĐỊNH NGHĨA VÀ CÁC THAO TÁC VỚI HÀM

A. Mục đích nghiên cứu

- Chương này giúp học viên nắm vững khái niệm hàm trong lập trình, bao gồm cách định nghĩa, gọi hàm, truyền tham số, trả về giá trị.
- Hiểu và áp dụng các quy tắc khi sử dụng hàm.
- Học viên có thể viết mã rõ ràng, dễ đọc, dễ bảo trì và có khả năng tái sử dụng cao – nền tảng thiết yếu cho mọi dự án lập trình.

B. Nội dung nghiên cứu

Bài 6. Khái niệm và vai trò của hàm

Bài 7. Định nghĩa, giá trị trả về và gọi hàm

C. Bài tập tổng hợp

Bài 6. Khái niệm và vai trò của hàm

a. Hàm là gì?

- Hàm là một khối lệnh có tên, thực hiện một nhiệm vụ cụ thể và có thể được gọi (sử dụng) nhiều lần trong chương trình.

b. Tại sao cần dùng hàm?

- Tránh lặp lại mã.
- Chia nhỏ chương trình thành các phần logic, dễ quản lý.
- Dễ kiểm thử, gỡ lỗi và mở rộng.

c. Cấu trúc cơ bản của hàm trong Python

```
def tên_hàm(tham_số):  
    # khối lệnh  
    return giá_trị # (tùy chọn)
```

Bài 7. Định nghĩa, giá trị trả về và gọi hàm

a. Định nghĩa

Cú pháp def – định nghĩa hàm

Từ khóa def dùng để khai báo hàm, theo sau là tên hàm và danh sách tham số trong dấu ngoặc. Bên dưới là khối lệnh xử lý

```
Python NC > chuong3 > bai7.py > greet
1  def greet(name):
2      print('xin chào', name)
```

Trong đó:

def: từ khóa khai báo hàm

greet: tên hàm (nên đặt tên mô tả chức năng của hàm)

name: tham số đầu vào (tùy chọn)

b. Giá trị trả về và gọi hàm

Sử dụng return để trả về kết quả:

Lệnh return kết thúc hàm và gửi giá trị được chỉ định ra ngoài nơi gọi hàm.

```
Python NC > chương3 > bai7.py > multiply
1 def multiply(a, b):
2     return a * b
3 result = multiply(3, 4) # result = 12
```

Cách gọi hàm và truyền tham số:

```
Python NC > chương3 > bai7.py > ...
1 def greet(name):
2     print("Xin chào!", name)
3
4 greet("Minh") # Truyền vào tham số name giá trị "Minh"
5 # In ra "Xin chào!, Minh"
```

Hàm không có return → trả về None:

Trong Python, mọi hàm đều trả về một giá trị. Nếu không có lệnh return, Python tự động trả về đối tượng đặc biệt None.

```
Python NC > chương3 > bai7.py > ...
1 def say_hello():
2     print("Xin chào!")
3
4 x = say_hello() # In ra "Xin chào!", nhưng x = None
```

Bài tập tổng hợp

Bài tập 1: Kiểm tra số chẵn/lẻ

Viết hàm `is_even_number(n)`:

- Nhận vào một số nguyên `n`
- Trả về `True` nếu `n` chẵn, `False` nếu lẻ

Sau đó nhập `n` từ bàn phím và gọi hàm kiểm tra `n` là số chẵn hay số lẻ.

Bài tập 2: Tính tổng từ 1 đến `n`

Viết hàm `sum_from_1_to_n(n)`:

- Trả về tổng: $1 + 2 + 3 + \dots + n$
- Nếu $n \leq 0$, trả về 0

Nhập `n` từ bàn phím và sử dụng hàm để tính tổng từ 1 đến `n`

Bài tập 3: In bảng cửu chương của `n`

Viết hàm `print_multiplication_table(n)`:

- in ra bảng cửu chương của `n` (từ 1 đến 10), mỗi phép tính trên một dòng.

Bài tập 4: Máy tính cơ bản

Viết 4 hàm tính toán cơ bản:

- `sum(a, b)`
- `subtract(a, b)`
- `multiply(a, b)`
- `divide(a, b)`

Sau đó nhập `n` từ bàn phím (1 - 4) để chọn phép tính theo thứ tự và in ra kết quả. Nếu $n < 1$ hoặc $n > 4$ thì in ra “không tồn tại phép tính” và yêu cầu nhập lại.

Kết quả ví dụ

Bài tập 1

```
Nhap so nguyen duong n: 4
4 la so chan
```

Bài tập 2

```
Nhap so nguyen duong n: 10
Tong tu 1 den 10 la: 55
```

Bài tập 3

```
Nhap so nguyen duong n: 5
Bang cuu chuong cua 5 la:
5 x 1 = 5
5 x 2 = 10
...
5 x 9 = 45
5 x 10 = 50
```

Bài tập 4

```
1 - Cong
2 - Tru
3 - Nhan
4 - Chia
Hay chon phep tinh (1-4): 2
Nhap so a: 10
Nhap so b: 15
Ket qua: -5
```

CHƯƠNG 4: SỬ DỤNG CÁC THƯ VIỆN CƠ BẢN CÓ SẴN: RANDOM, MATH, DATETIME

A. Mục đích nghiên cứu

- Làm quen và sử dụng hiệu quả các thư viện chuẩn (built-in modules) trong Python như random, math, datetime.
- Giải quyết các bài toán thực tế liên quan đến toán học, số ngẫu nhiên, xử lý thời gian.
- Hiểu cách import và sử dụng hàm/thuộc tính từ thư viện một cách an toàn và rõ ràng.
- Xây dựng chương trình có tính ứng dụng cao, gần gũi với đời sống (ví dụ: trò chơi, lịch, quản lý thời gian,...).

B. Nội dung nghiên cứu

Bài 8. Giới thiệu thư viện và cách import

Bài 9. Thư viện random – làm việc với số ngẫu nhiên

Bài 10. Thư viện math – các phép toán nâng cao

Bài 11. Thư viện datetime – làm việc với ngày và giờ

C. Bài tập tổng hợp

Bài 8. Giới thiệu thư viện và cách import

a. Thư viện (module) trong Python là gì?

Là tập hợp các hàm, hằng số, lớp được viết sẵn, giúp thực hiện các tác vụ phổ biến mà không cần viết lại từ đầu.

b. Cú pháp import cơ bản:

- `import module_name` → truy cập qua `module_name.function()`

```
Python NC > chuong4 > bai8.py
1 import math
2 print(math.sqrt(16))
```

- Đặt tên rút gọn (alias):

```
Python NC > chuong4 > bai8.py > now
1 import datetime as dt
2 now = dt.datetime.now()
```

- `from module_name import function_name` → gọi trực tiếp `function_name()`

```
Python NC > chuong4 > bai8.py
1 from math import sqrt
2 print(sqrt(16))
```

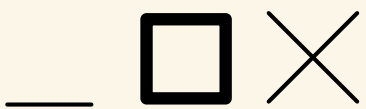
Bài 9. Thư viện random – làm việc với số ngẫu nhiên

Dùng để sinh số/giá trị ngẫu nhiên

Hàm / Hằng số	Mô tả
random.randint(a, b)	Trả về số nguyên ngẫu nhiên trong đoạn [a, b] (kể cả a và b)
random.choice(sequence)	Chọn ngẫu nhiên một phần tử từ chuỗi, list,...

Ví dụ cách sử dụng hàm:

```
bai9.py x
Python NC > chuong4 > bai9.py > ...
1  import random
2  so_ngau_nhien = random.randint(1,10)
3  print(so_ngau_nhien) # in ra 1 số ngẫu nhiên trong khoảng 1 đến 10
```



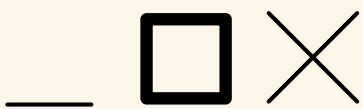
Bài 10. Thư viện math – các phép toán nâng cao

Các hàm và hằng số toán học

Hàm / Hằng số	Mô tả
math.sqrt(x)	Căn bậc hai của x
math.pi	Hằng số $\pi \approx 3.14159...$
math.factorial(n)	Giai thừa của số nguyên không âm n
math.pow(x, y)	Trả về x mũ y (kết quả là số thực)

Ví dụ cách sử dụng hàm:

```
Python NC > chương4 > bai10.py > ...
1  import math
2  sopi = math.pi
3  print(sopi) # in ra 1 số 3.14159...
```



Bài 11. Thư viện datetime – làm việc với ngày và giờ

Làm việc với ngày, giờ, thời gian → Thường dùng lớp datetime từ module datetime

Hàm / Thuộc tính	Mô tả
datetime.datetime.now()	Trả về ngày và giờ hiện tại (có cả giây và microgiây)
datetime.date.today()	Chỉ trả về ngày hiện tại (không có giờ)

Ví dụ cách sử dụng hàm:

```
Python NC > chương4 > bai11.py > ...
1 import datetime
2 now = datetime.datetime.now()
3 print(now) # in ra ngày hiện tại
```

```
Python NC > chương4 > bai11.py > ...
1 import datetime
2 now = datetime.datetime.now()
3 hour = now.hour
4 print(hour) # in ra giờ hiện tại
```

Bài tập tổng hợp

Bài tập 1: Mô phỏng xúc xắc

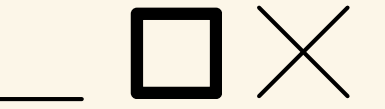
Viết chương trình:

- Mô phỏng việc tung một con xúc xắc n lần
- In ra tổng số điểm sau n lần tung
- Không trả về giá trị, chỉ in ra

Bài tập 2: Tính diện tích và chu vi các hình

Viết ba hàm riêng biệt cho:

1. `hinh_tron(r)` → in ra "Hình tròn bán kính $r=...$, chu vi= $...$, diện tích= $...$ "
2. `hinh_cau(r)` → in ra "Hình cầu bán kính $r=...$, thể tích= $...$ " (công thức: $\frac{4}{3}\pi r^3$)
3. `hinh_tru(r, h)` → in ra "Hình trụ bán kính $r=...$, chiều cao $h=...$, thể tích= $...$ " (công thức: $\pi r^2 h$)



Bài tập 3: Hiển thị thời gian hiện tại theo định dạng dễ đọc

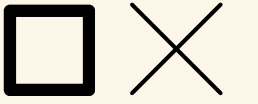
Viết chương trình:

- Lấy thời gian hiện tại bằng `datetime.datetime.now()`
- In ra theo định dạng:
- "Bây giờ là [giờ]:[phút] ngày [ngày]/[tháng]/[năm]"
- Ví dụ: "Bây giờ là 14:30 ngày 20/12/2025"

Bài tập 4: Trò chơi đoán số ngẫu nhiên

Viết chương trình:

- Dùng `random` để tạo số bí mật:
 - Viết một hàm nhận số và kiểm tra số người chơi "đoán"
 - So sánh với số bí mật
 - In "Đúng rồi!", "Quá cao!", hoặc "Quá thấp!"



Kết quả ví dụ

Bài tập 1

```
Lan tung thu 1 duoc diem: 5
Lan tung thu 2 duoc diem: 6
Lan tung thu 3 duoc diem: 4
Tong diem sau 3 lan tung: 15
```

Bài tập 2

```
Hinh tron ban kinh r = 4 , chu vi = 25.13 , dien tich = 50.27
Hinh cau ban kinh r = 4 , the tich = 268.08
Hinh tru ban kinh r = 4 , chieu cao h = 10 , the tich = 502.65
```

Bài tập 3

```
Bay gio la 9 : 36 ngay 22 / 12 / 2025
```

Bài tập 4

```
=====
Toi da nghi ra mot so tu 1 den 100.
Ban hay doan xem do la so gi!
=====

Nhap so cua ban: 50
So cua ban thap hon so bi mat. Hay thu so cao hon!
Nhap so cua ban: 75
So cua ban cao hon so bi mat. Hay thu so thap hon!
Nhap so cua ban: 51
Chuc mung! Ban da doan dung.
```

CHƯƠNG 5: THƯ VIỆN TURTLE

A. Mục đích nghiên cứu

- Biến code khô khan thành hình ảnh sinh động, tạo hứng thú học tập cho trẻ.
- Làm quen với trình tự lệnh, tọa độ, vòng lặp và hàm qua thao tác điều khiển rùa.
- củng cố kiến thức hình học và phát triển tư duy thẩm mỹ, sáng tạo.

B. Nội dung nghiên cứu

Bài 12. Thư viện turtle, khởi động thư viện

Bài 13. Các câu lệnh turtle cơ bản

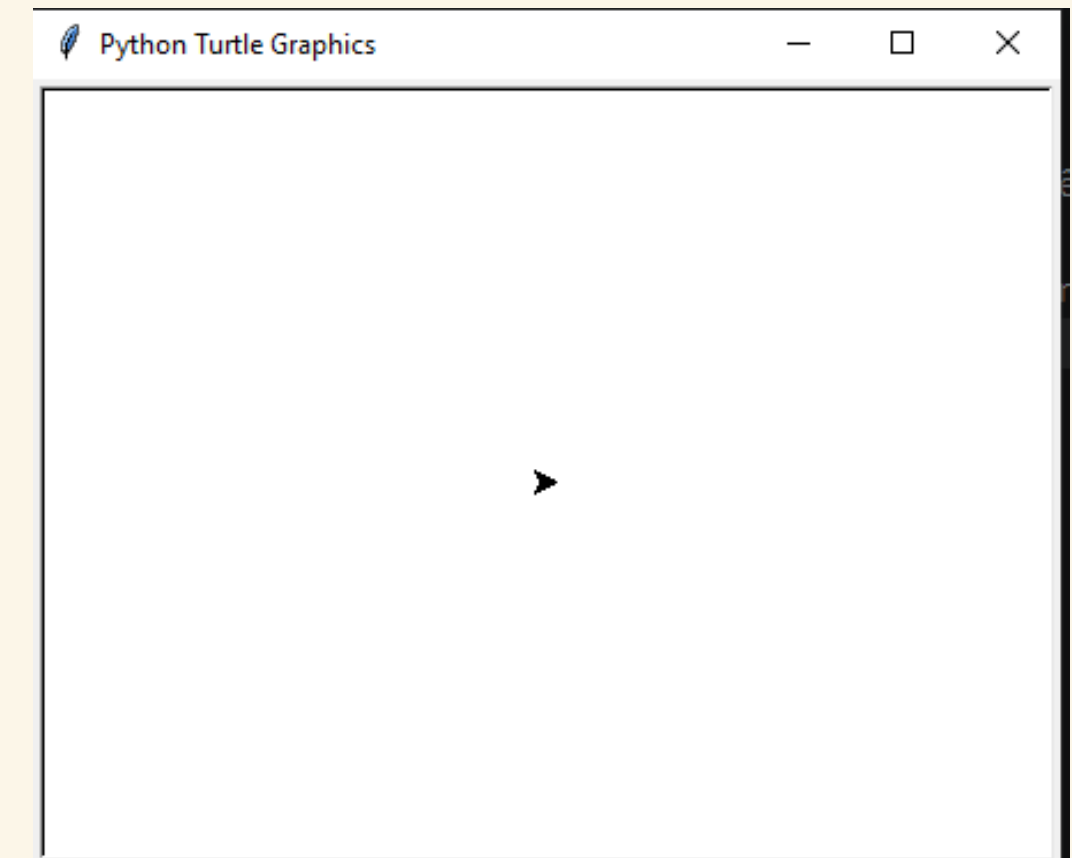
C. Bài tập tổng hợp

Bài 12. Thư viện turtle, khởi động thư viện

Là một thư viện tạo một chú robot hình con rùa cầm một chiếc bút mực. Em ra lệnh cho nó đi tới, đi lùi, rẽ trái, rẽ phải... và trên đường đi, nó sẽ để lại những nét vẽ tuyệt đẹp trên màn hình!

Khởi động thư viện turtle

```
bai12.py x
Python NC > chuong5 > bai12.py > ...
1  import turtle # 1. Gọi thư viện turtle
2
3  # 2. Tạo chú rùa
4  but_ve = turtle.Turtle() # Tạo chú rùa (đặt tên biến là but_ve)
5
6  # (Đừng quên dòng này ở cuối chương trình để màn hình không tự tắt nhé!)
7  turtle.done()
```



Bài 13. Các câu lệnh turtle cơ bản

a. Nhóm lệnh di chuyển:

- `but_ve.forward(100)` → đi **thẳng** 100 bước
- `but_ve.backward(50)` → đi **lùi** 50 bước
- `but_ve.right(90)` → xoay **phải** 90 độ
- `but_ve.left(90)` → xoay **trái** 90 độ
- `but_ve.circle(90)` → vẽ **hình tròn** bán kính 90

b. Nhóm lệnh điều khiển bút:

- `but_ve.penup()` → **Nhấc** bút lên
- `but_ve.pendown()` → **Hạ** bút xuống

c. Nhóm lệnh cho màn hình:

- `turtle.bgcolor("lightblue")` → Đổi màu nền của cửa sổ (có thể dùng: "blue", "green", "yellow", "pink", "orange"...).

d. Nhóm lệnh làm đẹp:

- `but_ve.color("red")` → Đổi màu đường nét
- `but_ve.begin_fill()` → bắt đầu theo dõi vùng tô màu
- `but_ve.end_fill()` → kết thúc và bắt đầu tô màu vùng vừa vẽ
- `but_ve.pensize(5)` → Đổi độ dày của nét vẽ (số càng to nét càng đậm).
- `but_ve.speed(5)` → Chỉnh tốc độ vẽ (từ 1 đến 10. Số 1 là chậm, số 10 là nhanh nhất. Số 0 là nhanh như chớp).
- `but_ve.shape("turtle")` → Biến chú rùa thành hình con rùa thật

Bài tập tổng hợp

Bài tập 1: Vẽ hình vuông nhiều màu

- Vẽ một hình vuông có cạnh 100 bước, mỗi cạnh một màu khác nhau (đỏ, xanh lá, xanh dương, vàng).

Bài tập 2: Vẽ hình chữ nhật nét đậm

- Vẽ một hình chữ nhật có chiều dài 150 bước, chiều rộng 80 bước, nét vẽ dày 5, màu tím.

Bài tập 3: Vẽ hai hình vuông xa nhau

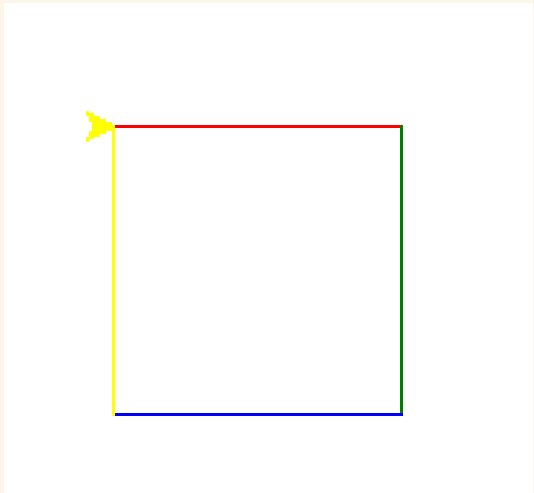
- Vẽ một hình vuông (cạnh 80) ở vị trí ban đầu.
- Sau đó di chuyển rùa sang phải 200 bước (không vẽ)
- Vẽ thêm một hình vuông nữa (cạnh 80, màu khác).

Bài tập 4: Vẽ hình tam giác và đổi màu nền

- Đổi màu nền màn hình thành màu xanh nhạt ("lightblue").
- Sau đó vẽ một hình tam giác đều có cạnh 120 bước, nét vẽ dày 3, nền và viền màu đỏ.

Kết quả ví dụ

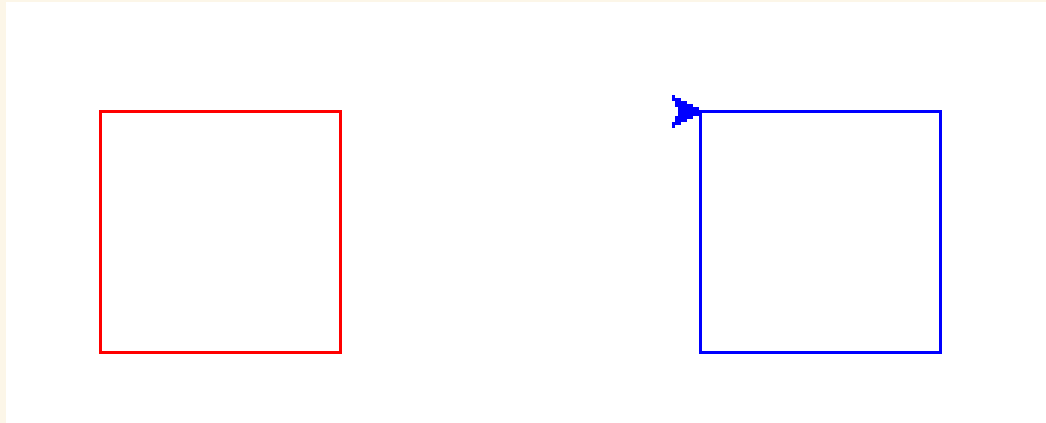
Bài tập 1



Bài tập 2



Bài tập 3



Bài tập 4

