

IE221 - BÀI THỰC HÀNH 2

Họ tên: Nguyễn Trần Bảo Anh

MSSV: 22520066

Lớp: IE221.Q11.1

Bài tập 1.



```
1 # Bài tập 1
2 so_nguyen = 42
3 print(f"Giá trị: {so_nguyen}, Kiểu: {type(so_nguyen)}")
4
5 so_thuc = 3.14159
6 print(f"Giá trị: {so_thuc}, Kiểu: {type(so_thuc)}")
7
8 chuoi = "Kỹ thuật lập trình Python"
9 print(f"Giá trị: {chuoi}, Kiểu: {type(chuoi)}")
10
11 danh_sach = [1, 2.5, "ba", True]
12 print(f"Giá trị: {danh_sach}, Kiểu: {type(danh_sach)}")
13
14 tu_dien = {'mon_hoc': 'IE221', 'nam': 2023}
15 print(f"Giá trị: {tu_dien}, Kiểu: {type(tu_dien)}")
16
17 tup = (1, 2, 3)
18 print(f"Giá trị: {tup}, Kiểu: {type(tup)}")
```



```
Giá trị: 42, Kiểu: <class 'int'>
Giá trị: 3.14159, Kiểu: <class 'float'>
Giá trị: Kỹ thuật lập trình Python, Kiểu: <class 'str'>
Giá trị: [1, 2.5, 'ba', True], Kiểu: <class 'list'>
Giá trị: {'mon_hoc': 'IE221', 'nam': 2023}, Kiểu: <class 'dict'>
Giá trị: (1, 2, 3), Kiểu: <class 'tuple'>
```

Bài tập 2.

```
1 # Bài tập 2
2 chieu_dai = float(input("Nhập chiều dài: "))
3 chieu_rong = float(input("Nhập chiều rộng: "))
4 dien_tich = chieu_dai * chieu_rong
5 print(f"Diện tích hình chữ nhật là: {dien_tich} cm2")
```

```
➡ Nhập chiều dài: 5
   Nhập chiều rộng: 3.2
   Diện tích hình chữ nhật là: 16.0 cm2
```

Bài tập 3.

```
1 # Bài tập 3
2 # List
3 print("List:")
4 sinh_vien = [22520066, "Nguyen Tran Bao Anh", "IE221.Q11.1"]
5 sinh_vien.append(2025)
6 print("List đầy đủ:", sinh_vien)
7 print("Slicing 2 phần tử đầu:", sinh_vien[0:2])
8
9 # Dict
10 print("\nDict:")
11 diem_so = {'Toan': 9, 'Ly': 8}
12 diem_so['Hoa'] = 9
13 print("Dict đầy đủ:", diem_so)
14 print("Giá trị Toan:", diem_so.get('Toan'))
15
16 # Tuple
17 print("\nTuple:")
18 lop, nam = 'IE221', 2023
19 lop, nam = 'Lớp', 'Năm'
20 print("Lớp:", lop, "Năm:", nam)
```

```
➡ List:
   List đầy đủ: [22520066, 'Nguyen Tran Bao Anh', 'IE221.Q11.1', 2025]
   Slicing 2 phần tử đầu: [22520066, 'Nguyen Tran Bao Anh']

   Dict:
   Dict đầy đủ: {'Toan': 9, 'Ly': 8, 'Hoa': 9}
   Giá trị Toan: 9

   Tuple:
   Lớp: IE221 Năm: 2023
```

Bài tập 4.

1. Phân tích từng case sau, nếu sai thì nêu lý do và sửa (chạy code để kiểm tra):

- Case 1: age = 20; print(age).
- Case 2: 1age = 20
- Case 3: if = "condition"
- Case 4: x = 5; x = "five"
- Case 5: tup = (1,2,3); tup[0] = 4

Case 1: Đúng.

```
1 age = 20
2 print(age)
```

⇒ 20

Case 2: Sai.

Lý do: Lỗi cú pháp. Tên biến không được bắt đầu bằng chữ số.

```
1 1age = 20
```

⇒ File "/tmp/ipython-input-2455107249.py", line 1
1age = 20
^
SyntaxError: invalid decimal literal

Sửa lỗi

```
1 _age = 20
2 print(_age)
```

⇒ 20

Case 3: Sai.

Lý do: Lỗi cú pháp. Tên biến không được trùng với keyword Python (keyword **if**).

```
1 if = "condition"
```

⇒ File "/tmp/ipython-input-908500339.py", line 1
if = "condition"
^
SyntaxError: invalid syntax

Sửa lỗi

```
1 iff = "condition"
2 print(iff)
```

condition

Case 4: Đúng.

```
1 x = 5
2 x = "five"
3 print(x)
```

five

Case 5: Sai.

Lý do: Lỗi kiểu dữ liệu. Tuple trong Python là immutable (không thay đổi được).

```
1 tup = (1,2,3)
2 tup[0] = 4
```

TypeError Traceback (most recent call last)
/tmp/ipython-input-54431359.py in <cell line: 0>()
1 tup = (1,2,3)
----> 2 tup[0] = 4

TypeError: 'tuple' object does not support item assignment

Sửa lỗi

Nếu muốn thay đổi phần tử, cần dùng List.

```
1 lst = [1,2,3]
2 lst[0] = 4
3 print(lst)
```

[4, 2, 3]

2. Phân biệt list và tuple (Nêu giống và khác). Cho ví dụ minh họa so sánh bằng code.

Giống nhau:

- Đều là cấu trúc dữ liệu có thể chứa nhiều phần tử.
- Các phần tử có thể khác kiểu dữ liệu (int, float, string, object...).
- Đều có thể truy cập phần tử bằng chỉ số (index).

- Có thể chứa phần tử trùng lặp.
- Hỗ trợ duyệt bằng vòng lặp và các toán tử như in, not in.

Khác nhau:

Tiêu chí	List	Tuple
Khai báo	Dùng []	Dùng ()
Tính thay đổi	Mutable – có thể thêm, xóa, sửa phần tử	Immutable – không thể thay đổi sau khi tạo
Ứng dụng	Dùng khi dữ liệu cần thay đổi	Dùng khi dữ liệu cố định, không đổi

Ví dụ:



```

1 # List
2 my_list = [1, 2, 3]
3 print("List ban đầu:", my_list)
4
5 my_list[0] = 10      # sửa phần tử
6 my_list.append(4)    # thêm phần tử
7 print("List sau khi sửa:", my_list)
8
9 # Tuple
10 my_tuple = (1, 2, 3)
11 print("\nTuple ban đầu:", my_tuple)
12
13 # Thử sửa phần tử
14 try:
15     my_tuple[0] = 10
16 except TypeError as e:
17     print("Lỗi khi sửa tuple:", e)
18
19 # Truy cập giống list
20 print("Phần tử thứ 2 của tuple:", my_tuple[1])

```



List ban đầu: [1, 2, 3]
List sau khi sửa: [10, 2, 3, 4]

Tuple ban đầu: (1, 2, 3)
Lỗi khi sửa tuple: 'tuple' object does not support item assignment
Phần tử thứ 2 của tuple: 2

3. Lập bảng so sánh các kiểu dữ liệu trong Python (Kiểu dữ liệu, Có thứ tự (Ordered), Cho phép trùng lặp, Mutable (có thể thay đổi), Cú pháp tạo, Truy cập phần tử, Ứng dụng chính, Ví dụ).

Tiêu chí / Kiểu dữ liệu	Number	String	List
Có thứ tự (Ordered)	Không	Có	Có
Cho phép trùng lặp	Không (chỉ 1 giá trị số, không có khái niệm lặp lại)	Có	Có
Mutable (Có thể thay đổi)	Không	Không	Có
Cú pháp tạo	a = 10	"abc"	[1,2,3]
Truy cập phần tử	Không	s[0]	lst[0]
Ứng dụng chính	Tính toán	Xử lý văn bản	Lưu trữ, thay đổi linh hoạt

Tiêu chí / Kiểu dữ liệu	Tuple	Dictionary	Set
Có thứ tự (Ordered)	Có	Có	Không
Cho phép trùng lặp	Có	Keys: Không Values: Có	Không
Mutable (Có thể thay đổi)	Không	Có	Có
Cú pháp tạo	(1,2,3)	{"key": "value"}	{1,2,3}
Truy cập phần tử	tup[0]	dict["key"]	Không qua index (chỉ duyệt)
Ứng dụng chính	Dữ liệu cố định	Key-value, tra cứu nhanh	Tập hợp duy nhất, toán tử tập hợp

Ví dụ:



```
1 # Numbers
2 a = 10; b = 3.14
3 print("Numbers:", a, b)
4
5 # String
6 s = "Python"
7 print("String:", s[0])    # P
8
9 # List
10 lst = [1, 2, 3]
11 lst.append(4)
12 print("List:", lst)
13
14 # Tuple
15 tup = (1, 2, 3)
16 print("Tuple:", tup[1])  # 2
17
18 # Dictionary
19 d = {"name": "Anh", "class": "IE221"}
20 print("Dictionary:", d["name"])
21
22 # Set
23 s = {1, 2, 2, 3}
24 print("Set:", s)        # {1, 2, 3}
```



```
Numbers: 10 3.14
String: P
List: [1, 2, 3, 4]
Tuple: 2
Dictionary: Anh
Set: {1, 2, 3}
```