Illustrate the difference between tuple and list with examples python?

Mutability:

Lists are mutable, meaning their contents can be changed after creation.

Tuples are immutable, meaning their contents cannot be changed after creation.

Syntax:

Lists are defined using square brackets [].

Tuples are defined using parentheses ().

Use Cases:

Lists are used when you need a collection of items that can change.

Tuples are used when you need a collection of items that should not change.

Performance:

Tuples can be slightly more memory efficient than lists because they are immutable.

Here's an example to illustrate these differences:

Lists

```
# Creating a list
my_list = [1, 2, 3, 4, 5]

# Accessing elements
print(my_list[0])  # Output: 1

# Modifying elements
my_list[0] = 10
print(my_list)  # Output: [10, 2, 3, 4, 5]

# Adding elements
my_list.append(6)
```

```
print(my_list)  # Output: [10, 2, 3, 4, 5, 6]

# Removing elements
my_list.remove(3)
print(my_list)  # Output: [10, 2, 4, 5, 6]
```

Tuple

```
# Creating a tuple
my_tuple = (1, 2, 3, 4, 5)

# Accessing elements
print(my_tuple[0])  # Output: 1

# Tuples are immutable, so the following operations would raise an error:
my_tuple[0] = 10  # TypeError: 'tuple' object does not support item assignment
my_tuple.append(6)  # AttributeError: 'tuple' object has no attribute 'append'

# However, you can create a new tuple if needed
new_tuple = my_tuple + (6,)
print(new_tuple)  # Output: (1, 2, 3, 4, 5, 6)
```

Summary

Lists are used when you need a mutable sequence of items.

Tuples are used when you need an immutable sequence of items.

2. How do you know the type of variable?

```
# Define some variables
a = 10
b = 3.14
c = "Hello, World!"
```

```
d = [1, 2, 3]
e = (4, 5, 6)
f = {'name': 'Alice', 'age': 30}
g = {7, 8, 9}
h = True

# Determine and print the type of each variable
print(type(a))  # Output: <class 'int'>
print(type(b))  # Output: <class 'float'>
print(type(c))  # Output: <class 'str'>
print(type(d))  # Output: <class 'list'>
print(type(e))  # Output: <class 'tuple'>
print(type(f))  # Output: <class 'dict'>
print(type(g))  # Output: <class 'set'>
print(type(h))  # Output: <class 'bool'>
```