Python Interview Questions

What is the difference between a list and a tuple in Python? **2**

How do you handle exceptions in Python? **2**

How do you open and read a file in Python? **3**

How do you create a dictionary in Python? **3**

How do you create a class in Python? **4**

How do you iterate over a list in Python? **5**

How do you check if a key exists in a dictionary in Python? **5**

What is a generator in Python? **6**

How do you remove duplicates from a list in Python? **7**

How do you sort a list in Python? **7**

What is the difference between a list and a tuple in Python?

Answer: A list is mutable while a tuple is immutable. In other words, a list can be modified after it is created, while a tuple cannot be modified. Here is an example that demonstrates the difference:

```
my_list = [1, 2, 3]
my_tuple = (1, 2, 3)

my_list[0] = 4   # this is allowed
my_tuple[0] = 4  # this will result in a TypeError
```

How do you handle exceptions in Python?

Answer: Exceptions can be handled in Python using the try-except block. Here is an example:

```
try:
    x = 1 / 0
except ZeroDivisionError:
    print("Cannot divide by zero")
```

This code tries to divide 1 by 0, which will result in a ZeroDivisionError exception. The except block handles this exception and prints a message.

How do you open and read a file in Python?

Answer: You can use the open() function to open a file, and the read() method to read its contents. Here is an example:

```
with open("example.txt", "r") as f:
    contents = f.read()

print(contents)
```

This code opens the file example.txt in read mode, and reads its contents into the contents variable. The with statement ensures that the file is properly closed when we are done reading from it.

How do you create a dictionary in Python?

Answer: A dictionary is created using curly braces {} and key-value pairs separated by colons. Here is an example:

```
my_dict = {"apple": 1, "banana": 2, "orange": 3}
```

This creates a dictionary with keys "apple", "banana", and "orange", and values 1, 2, and 3 respectively.

How do you create a class in Python?

Answer: A class is created using the class keyword, followed by the name of the class and a colon. Here is an example:

```
class Person:
    def __init__(self, name, age):
        self.name = name
        self.age = age

    def say_hello(self):
        print(f"Hello, my name is {self.name} and I am
{self.age} years old")

person = Person("John", 25)
person.say_hello()
```

This code creates a class called Person with a constructor that takes a name and an age, and a method called say_hello that

prints a message. It then creates an instance of the Person class with name "John" and age 25, and calls the say_hello method.

How do you iterate over a list in Python?

Answer: You can use a for loop to iterate over a list. Here is an example:

```
my_list = [1, 2, 3]

for item in my_list:
    print(item)
```

This code iterates over the list my_list and prints each item.

How do you check if a key exists in a dictionary in Python?

Answer: You can use the in keyword to check if a key exists in a dictionary. Here is an example:

```
my_dict = {"apple": 1, "banana": 2, "orange": 3}

if "banana" in my_dict:
```

```
    print("The key 'banana' exists in the dictionary")
```

This code checks if the key "banana" exists in the dictionary my_dict, and prints a message if it does.

What is a generator in Python?

Answer: A generator is a special type of function that allows you to iterate over a sequence of values. Instead of returning a value and exiting like a normal function, a generator can yield a value and then pause its execution until the next value is requested. Here is an example:

```
def my_generator():
    yield 1
    yield 2
    yield 3

for value in my_generator():
    print(value)
```

This code defines a generator called my_generator that yields the values 1, 2, and 3. It then iterates over the generator using a for loop and prints each value.

How do you remove duplicates from a list in Python?

Answer: You can remove duplicates from a list using the set() function. Here is an example:

```
my_list = [1, 2, 3, 2, 1]

new_list = list(set(my_list))

print(new_list)
```

This code creates a list with duplicates, and then creates a new list with the duplicates removed using the set() function. The list() function is used to convert the set back to a list.

How do you sort a list in Python?

Answer: You can sort a list in Python using the sort() method or the sorted() function. Here is an example:

```
my_list = [3, 1, 2]
```

```
my_list.sort()

print(my_list)

new_list = sorted(my_list)

print(new_list)
```

This code sorts a list using both the sort() method and the sorted() function, and prints the sorted lists. The sort() method modifies the original list, while the sorted() function returns a new sorted list.