***PYTHON phonenumbers module***

Python is a very powerful language and also very rich in libraries. **phonenumbers** is one of the modules that provides numerous features like providing basic information of a phone number, validation of a phone number etc. Here, we will learn how to use phonenumbers module just by writing simple Python programs. This is a Python port of Google’s libphonenumber library.

***Installation***

Install the phonenumbers module by typing the following command in command prompt.

pip install phonenumbers

**Getting Started**

**1. Convert String to phonenumber format:**To explore the features of phonenumbers module, we need to take the phone number of a user in phonenumber format. Here we will see how to convert the user phone number to phonenumber format. Input must be of string type and country code must be added before phone number.

* Python3

|  |
| --- |
| # Program to convert input to  # phonenumber format    import phonenumbers    # Parsing String to Phone number  # Phone number format: (+Countrycode)xxxxxxxxxx  phoneNumber = phonenumbers.parse("+919876543210")    # This will print the phone number and  # it's basic details.  print(phoneNumber) |

**Output:**

Country Code: 91 National Number: 9876543210

**2. Get Timezone:**Here is the simple Python program to get the timezone of a phone number using phonenumbers module. First, we do parse the string input to phonenumber format, and then we use an inbuilt function to get the timezone of a user. It gives the output for valid numbers only.

* Python3

|  |
| --- |
| # Program to get timezone a phone number    import phonenumbers  from phonenumbers import timezone    # Parsing String to Phone number  phoneNumber = phonenumbers.parse("+919876543210")    # Pass the parsed phone number in below function  timeZone = timezone.time\_zones\_for\_number(phoneNumber)    # It print the timezone of a phonenumber  print(timeZone) |

**Output:**

('Asia/Calcutta',)

**3. Extract phone numbers from text:**We can extract phone numbers that are present in a text/paragraph using this module. You can iterate over it to retrieve a sequence of phone numbers. For this, **PhoneNumberMatcher** object provides the relevant function.

* Python3

|  |
| --- |
| # Program to extract phone numbers from a text  import phonenumbers    # Text Input  text = "Contact us at +919876543210 or +14691234567"    # Pass the text and country code to the below function  numbers = phonenumbers.PhoneNumberMatcher(text, "IN")    # Printing the phone numbers matched with country code  # and also the indexes of the phone numbers in the string input  for number in numbers:      print(number) |

**Output:**

PhoneNumberMatch [14,27) +919876543210

**4. Carrier and Region of a Phone Number:**Here we will learn how to find the carrier and region of a phone number using the geocoderand carrierfunctions of this module.

* Python3

|  |
| --- |
| # Program to find carrier and region  # of a phone number  import phonenumbers  from phonenumbers import geocoder, carrier    # Parsing String to Phone number  phoneNumber = phonenumbers.parse("+919876543210")    # Getting carrier of a phone number  Carrier = carrier.name\_for\_number(phoneNumber, 'en')    # Getting region information  Region = geocoder.description\_for\_number(phoneNumber, 'en')    # Printing the carrier and region of a phone number  print(Carrier)  print(Region) |

**Output:**

Airtel

India

**5. Validation of a phone number:**A simple python program, to check whether a given phone number is valid or not (e.g. it’s in an assigned exchange), and to check whether a given phone number is possible or not (e.g. it has the right number of digits).

* Python3

|  |
| --- |
| # Program to check whether a phone number is  # valid or not  import phonenumbers    # Parsing String to Phone number  phone\_number = phonenumbers.parse("+91987654321")    # Validating a phone number  valid = phonenumbers.is\_valid\_number(phone\_number)    # Checking possibility of a number  possible = phonenumbers.is\_possible\_number(phone\_number)    # Printing the output  print(valid)  print(possible) |

**Output:**

False

True