Author: Jay Jitesh Chavan
LinkedIn

Phone Number Analysis Using Python

```
import phonenumbers
from phonenumbers import (
    timezone,
    geocoder,
    carrier,
    NumberParseException,
    PhoneNumberType,
    is_valid_number,
    is_possible_number,
    number_type
)

# Function to parse and analyze phone number
def analyze_phone_number(number):
    try:
        # Parse the phone number using phonenumbers library
        phone = phonenumbers.parse(number)

        # Check if the number is valid and callable
        # is_possible_number checks if the number can exist based on
the numbering plan
        if not is_possible_number(phone):
            print("This number is not possible based on the numbering
plan.")
            return
        # is_valid_number checks if the number is actually valid for
the country
        if not is_valid_number(phone):
            print("This number is not valid.")
            return

        # Mapping of phone number types to human-readable
descriptions
        number_type_desc = {
            PhoneNumberType.MOBILE: "Mobile",
            PhoneNumberType.FIXED_LINE: "Landline",
            PhoneNumberType.FIXED_LINE_OR_MOBILE: "Fixed Line or
Mobile",
            PhoneNumberType.TOLL_FREE: "Toll Free",
            PhoneNumberType.VOIP: "VoIP",
            PhoneNumberType.UNKNOWN: "Unknown"
        }

        # Determine the type of the phone number
        num_type = number_type(phone)
        num_type_str = number_type_desc.get(num_type, "Unknown")
        # Extract timezone, carrier, and geographical region
        time_zones = timezone.time_zones_for_number(phone)
        carrier_name = carrier.name_for_number(phone, "en")  # 
Carrier name in English
        region = geocoder.description_for_number(phone, "en")  # 
Region description in English

        # Format the phone number in international and national 
formats
        intl_format = phonenumbers.format_number(phone, 
phonenumbers.PhoneNumberFormat.INTERNATIONAL)
        national_format = phonenumbers.format_number(phone, 
phonenumbers.PhoneNumberFormat.NATIONAL)

        # Print the analysis results
        print("\n=== Phone Number Analysis ===")
        print(f"Phone number: {intl_format}")  # Display 
international format
        print(f"National format: {national_format}")  # Display 
national format
        print(f"Valid number: Yes")  # Confirm validity
        print(f"Number type: {num_type_str}")  # Show the type of the 
number
        print(f"Time zones: {', '.join(time_zones)}")  # Display 
associated time zones
        print(f"Carrier: {carrier_name if carrier_name else 
'Unknown'}")  # Display carrier name
        print(f"Region: {region if region else 'Unknown'}")  # 
Display region

        # Log results to a text file for record-keeping
        with open("phone_number_log.txt", "a") as log_file:
            log_file.write(f"Phone: {intl_format}, National: 
{national_format}, Type: {num_type_str}, Timezones: {', 
'.join(time_zones)}, Carrier: {carrier_name}, Region: {region}\n")

    except NumberParseException as e:
        # Handle exceptions specifically for number parsing
        print(f"Error: {str(e)}")
    except Exception as ex:
        # Handle any other exceptions that may occur
        print(f"An unexpected error occurred: {str(ex)}")

# Take input from the user
number_input = input("Enter your phone number with country code 
(e.g., +11234567890): ")
analyze_phone_number(number_input)
```