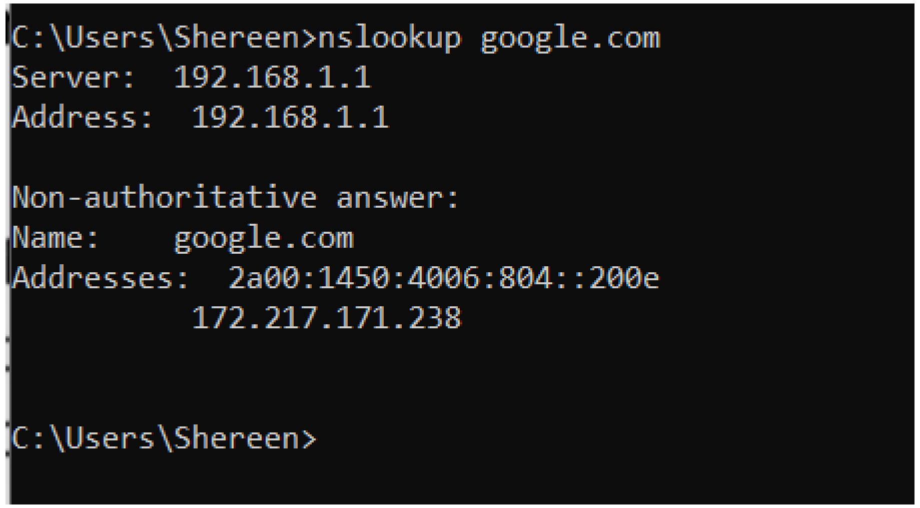
The DNS Protocol

The DNS Protocol

- ***The** Domain Name System (DNS)* is the scheme by which millions of Internet hosts cooperate to answer the question of what hostnames resolve to which IP addresses.
-



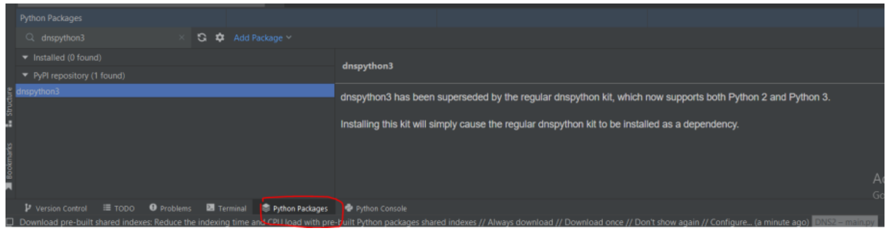
```
C:\Users\Shereen>nslookup google.com
Server:  192.168.1.1
Address:  192.168.1.1

Non-authoritative answer:
Name:    google.com
Addresses:  2a00:1450:4006:804::200e
          172.217.171.238


C:\Users\Shereen>
```

Making a DNS Query from Python

- **Installing dnspython3**
  - **From CMD use:** pip install dnspython3
  - From Pycharm: search for dnspyhtons in python package then click install



dnspython3

dnspython3 has been superseded by the regular dnspython kit, which now supports both Python 2 and Python 3.

Installing this kit will simply cause the regular dnspython kit to be installed as a dependency.

Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always download // Download once // Don't show again // Configure... (a minute ago)

Making a DNS Query from Python

```
import dns.resolver

def lookup(name):
    for qtype in 'A', 'AAAA', 'CNAME', 'MX', 'NS':
        answer = dns.resolver.query(name, qtype, raise_on_no_answer=False)
        #for val in answer:
         #   print(val)
        if answer.rrset is not None:
            print(answer.rrset)
if __name__ == '__main__':
    lookup('google.com')
```

DNS Record types

- **Commonly used record types**
  - **A** (Host address)
  - **AAAA** (IPv6 host address)
  - **CNAME** (Canonical name for an alias) is a type of record in the Domain Name System (DNS) used to map a domain name as an alias for another domain. CNAME records always point to another domain name and never directly to an IP address.

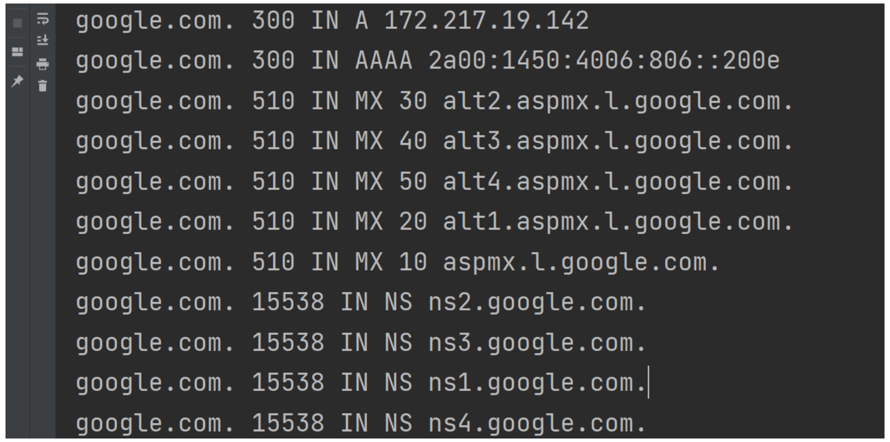
DNS Record types

- **MX Record**
  - A MX record also called mail exchanger record is a resource record in the Domain Name System that specifies a mail server responsible for accepting email messages on behalf of a recipient's domain. It also sets the preference value used to prioritizing mail delivery if multiple mail servers are available.
- **NS** (Name Server):Directs to name servers, where to ask if you want know about a subdomain
- **TXT** (Descriptive text):used to put miscellaneous info, used to verify ownership of domain

- raise_on_no_answer: when no answer not return error

The Answer of DNS Query

- **In order, the keys that get printed on each line are as follows:**
  - The name looked up.
  - The time in seconds that you are allowed to cache the name before it expires.
  - The “class” like IN, which indicates that you are being returned Internet address responses.
  - The “type” of record. Some common ones are A for an IPv4 address, AAAA for an IPv6 address, NS for a record that lists a name server, and MX for a reply giving the mail server that should be used for a domain.
  - Finally, the “data” provides the information you need to connect to or contact a service



```
google.com. 300 IN A 172.217.19.142
google.com. 300 IN AAAA 2a00:1450:4006:806::200e
google.com. 510 IN MX 30 alt2.aspmx.l.google.com.
google.com. 510 IN MX 40 alt3.aspmx.l.google.com.
google.com. 510 IN MX 50 alt4.aspmx.l.google.com.
google.com. 510 IN MX 20 alt1.aspmx.l.google.com.
google.com. 510 IN MX 10 aspmx.l.google.com.
google.com. 15538 IN NS ns2.google.com.
google.com. 15538 IN NS ns3.google.com.
google.com. 15538 IN NS ns1.google.com.
google.com. 15538 IN NS ns4.google.com.
```

Resolving Mail Domains

- **resolve_email_domain(domain) Method**

"For an email address `name@domain` find its mail server IP addresses."

- Make query for MX record type
  - If exits :
    1. Sort records based on priority from lowest to highest
    2. For each record
       - Get record server name: **name= record.exchange.to_text(omit_final_dot=True)**
       - call **resolve_hostname** (name)
  - else call **resolve_hostname** (domain name)

resolve_hostname method

- **def resolve_hostname(hostname):**
- "Print an A or AAAA record for `hostname`; follow CNAMEs if necessary."

1. Make query using dns.resolver.query with hostname and rType=A
2. If exits :
   - For each record print address
   - return
3. Else : Make query using dns.resolver.query with hostname and rType= AAAA
4. If exits :
   - For each record print address
   - return

resolve_hostname method con’t

5. Else: Make query using dns.resolver.query with hostname and rType= CNAME
6. If exits :
    - For each record call resolve_hostname(cname)
    - Return
7. Else print(indent, 'ERROR: no A, AAAA, or CNAME records for', hostname)

Python sorted() Function

- The **sorted**() function returns a sorted list of the specified iterable object.
- **Syntax**
  - sorted(*iterable*, key=*key*, reverse=*reverse*)
- **Parameter Values**

Parameter	Description
iterable	Required. The sequence to sort, list, dictionary, tuple etc.
Key	Optional. A Function to execute to decide the order. Default is None
reverse	Optional. A Boolean. False will sort ascending, True will sort descending. Default is False

resolve_mail method

```
def resolve_mail(domain):
    answer=dns.resolver.query(domain,'MX')
    if answer.rrset is not None:
        records = sorted(answer)
        for record in records:
            print(record.preference) # print priority
            print(record.exchange)  # print name
            name = record.exchange.to_text(omit_final_dot=True)
            resolve_host(name)
    else:
        print('No exchange name')
        resolve_host(domain)
        return
resolve_mail('google.com')
```

resolve_host Method

```
import dns.resolver

def resolve_host(hostname):
    answer=dns.resolver.query(hostname,'A')
    if answer.rrset is not None:
        for record in answer:
            print('hosname ',hostname,'has address',record.address)
        return
```

resolve_host Method con’t

```
else:
    answer=dns.resolver.query(hostname,'AAAA')
    if answer.rrset is not None:
        for record in answer:
            print(record.address)
        return
    else:
```

resolve_host Method con’t

```
else:
    answer = dns.resolver.query(hostname, 'CNAME')
    if answer.rrset is not None:
        record = answer[0]
        cname = record.address
        resolve_host(record.address)
    else:
        print('error no a,AAAA or CNAME for host name ')
```

Thank You