Data types useful for Internet programming.

Modules

[logging](#) [signal](#) [struct](#) [sys](#)

Classes

[DomainName](#)
[InetAddr](#)

class **DomainName**

Representation of a DNS domain name.

Methods defined here:

__c

__co
Return a copy of thi _____ ct that loses any
pointer information   this object's rep.

__hash__(self)
Returns a hash of this object based on its string rep.

__init__(self, s='')
Initialize from a user-supplied string (optionally empty)
.

__len__(self)
Returns the length of the [RFC 1035](#)-compliant binary
representation, taking into account pointers.

__repr__(self)

__str__(self)
Returns a '.'-terminated string representation of this
object.

pack(self)
Packs this object into [RFC 1035](#)-compliant data and return
s the
resulting binary-packed string.

parent(self)
Return a [DomainName](#) object that represents the parent dom

```
    ain.
```

Static methods defined here:

fromData(data, offset=0)
```
    Return a DomainName object from user-supplied packed bina
    ry
    data, with an optional offset modifier.
```

class **InetAddr**

```
    Internet address type.
```

Methods defined here:

__cmp__(self, other)

__hash__(self)
```
    Returns a hash of this object based on its string rep.
```

__init__(self, s)
```
    Initialize from a user-supplied dotted quad string.
```

__re...

__str__(self)

toNetwork(self)
```
    Convert to network d                    packed binary bytes)
    .
```

Static methods defined here:

fromNetwork(n)

inet_aton(...)
```
    inet_aton(string) -> packed 32-bit IP representation

    Convert an IP address in string format (123.45.67.89) to the
    32-bit packed
    binary format used in low-level network functions.
```

inet_ntoa(...)
```
    inet_ntoa(packed_ip) -> ip_address_string

    Convert an IP address from 32-bit packed binary format to st
    ring format
```

Data

DEBUG1 = 9
DEBUG2 = 8
FILTER = '.............................. !"#$%&\'()*+,-./..'
ch = <logging.StreamHandler instance>
chformatter = <logging.Formatter instance>
fh = <logging.handlers.RotatingFileHandler instance>
fhformatter = <logging.Formatter instance>
logfile = './ncsdns.log'
logger = <logging.RootLogger instance>
x = 255

Assignment Project Exam Help

https://eduassistpro.github.io/

Add WeChat edu_assist_pro