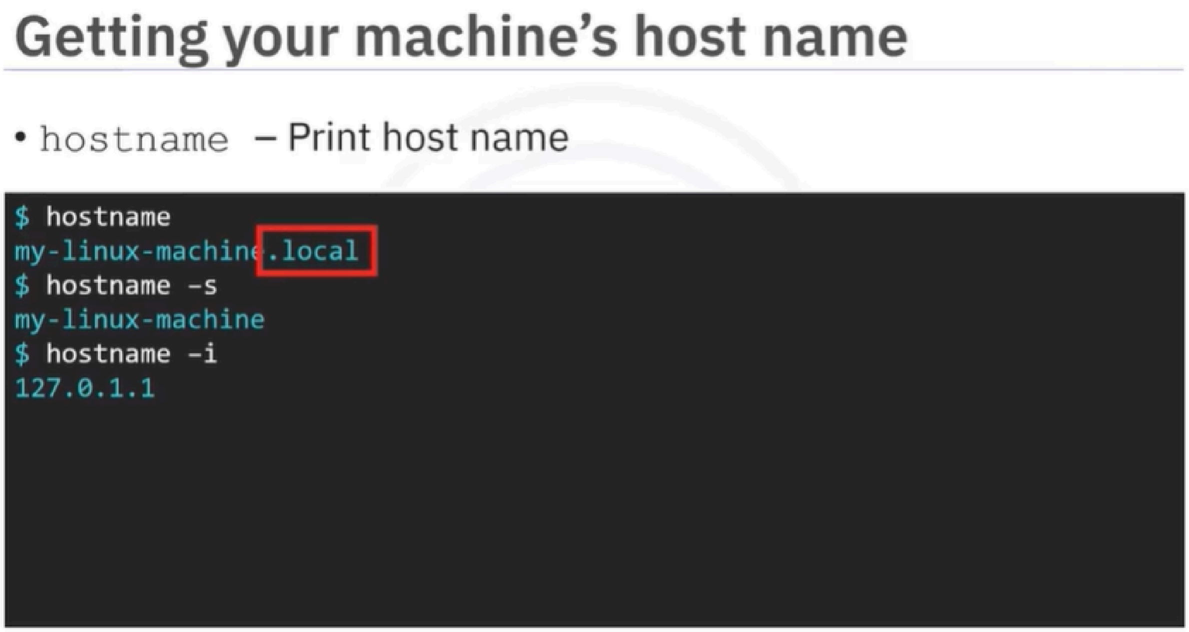
Networking Commands

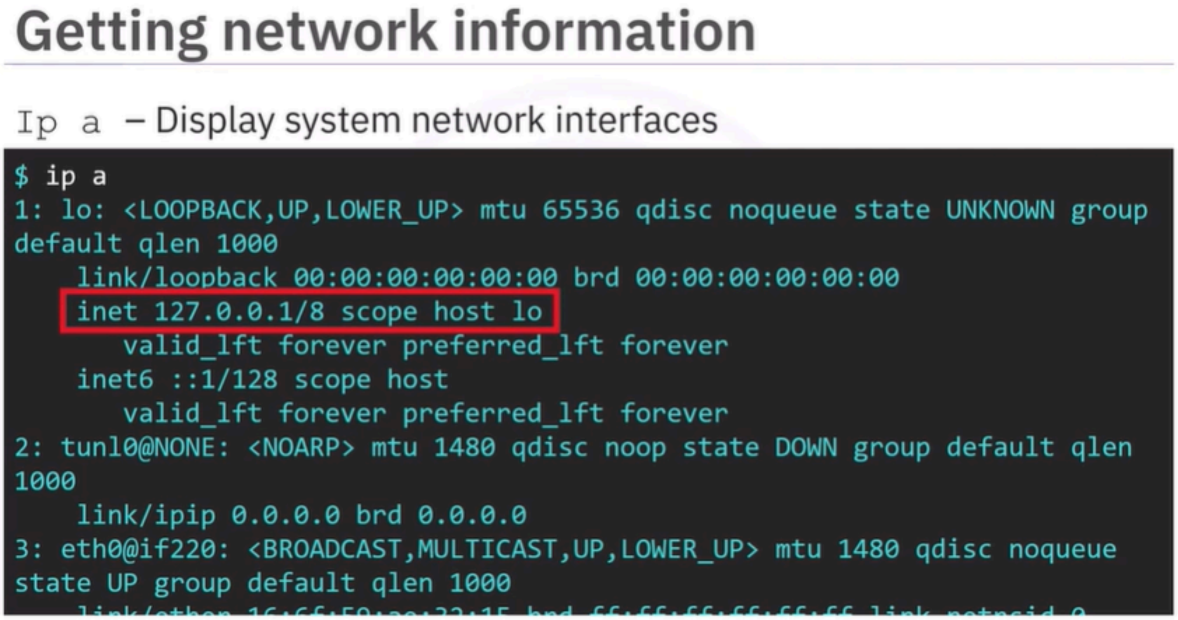


Getting your machine's host name

- `hostname` – Print host name

```
$ hostname
my-linux-machine.local
$ hostname -s
my-linux-machine
$ hostname -i
127.0.1.1
```

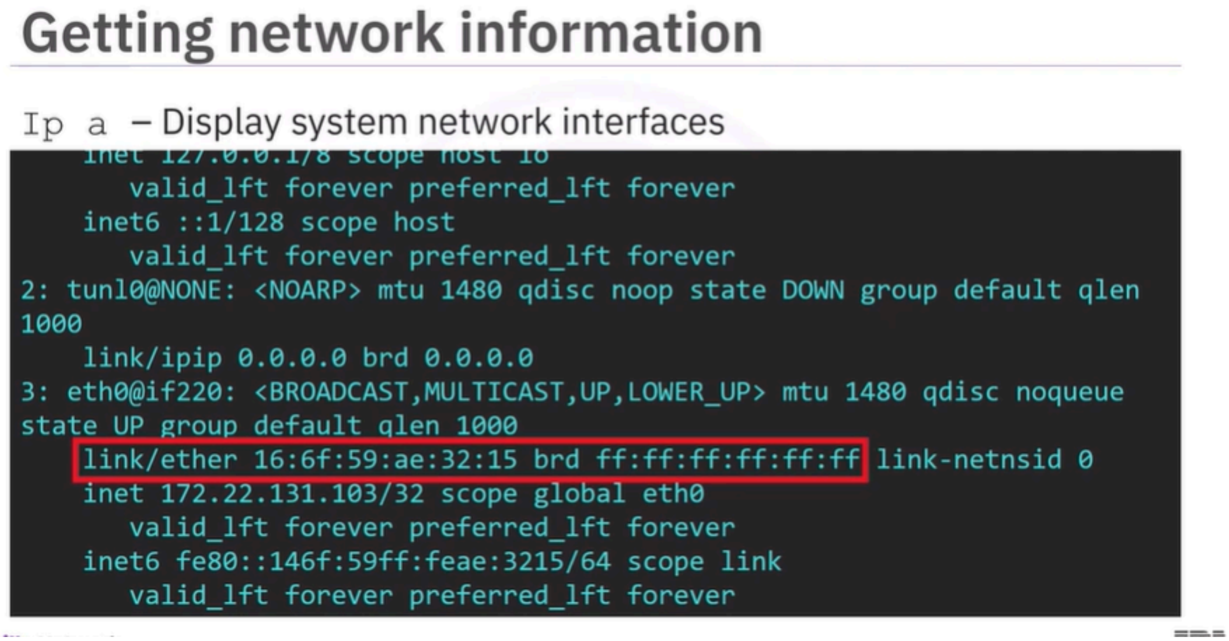
- The hostname command is used to get or set The hostname and other information which uniquely identifies your computer. Entering the hostname command without any options Returns the hostname of your machine, Which in this example is mylinuxmachine.local. The .local suffix appears in the hostname If your machine has local domain set. To drop the domain suffix, you can include the "-s option. If you use the "-i option, Hostname will provide the IP address of the hostname.



Getting network information

`Ip a` – Display system network interfaces

```
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen
1000
    link/ipip 0.0.0.0 brd 0.0.0.0
3: eth0@if220: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1480 qdisc noqueue
state UP group default qlen 1000
    link/ether 16:6f:59:ae:32:15 brd ff:ff:ff:ff:ff:ff link-netnsid 0
```

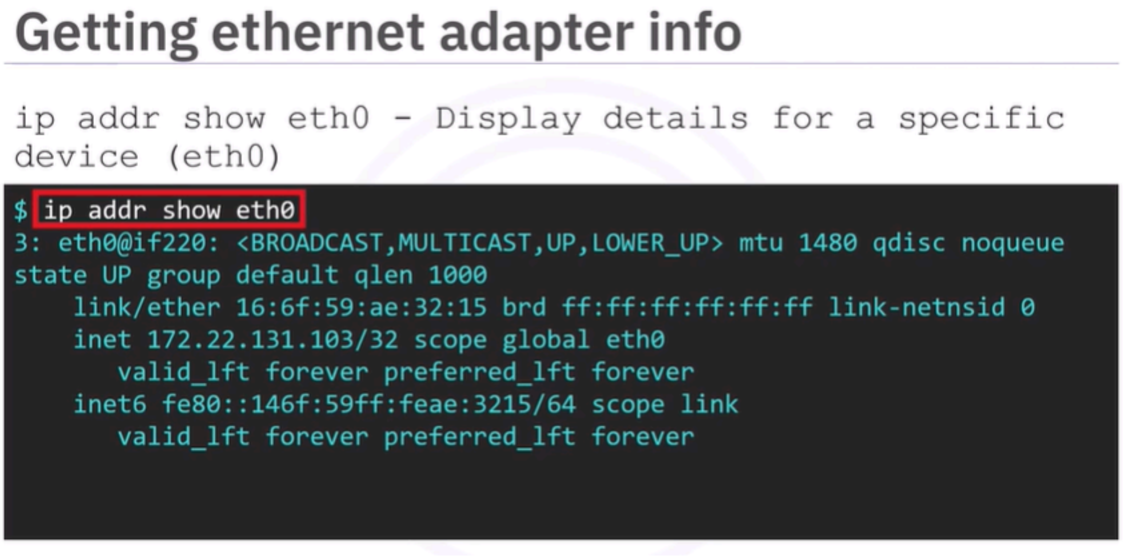


Getting network information

`Ip a` – Display system network interfaces

```
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: tunl0@NONE: <NOARP> mtu 1480 qdisc noop state DOWN group default qlen
1000
    link/ipip 0.0.0.0 brd 0.0.0.0
3: eth0@if220: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1480 qdisc noqueue
state UP group default qlen 1000
    link/ether 16:6f:59:ae:32:15 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.22.131.103/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::146f:59ff:feae:3215/64 scope link
       valid_lft forever preferred_lft forever
```

- The ip command is a powerful utility For configuring and displaying network interface information. To view all details of your device's communication interfaces, Use the command 'ip a', Which provides comprehensive information Including IP addresses, MAC addresses, And other interface-specific details. This command is particularly useful For system administrators managing network configurations.

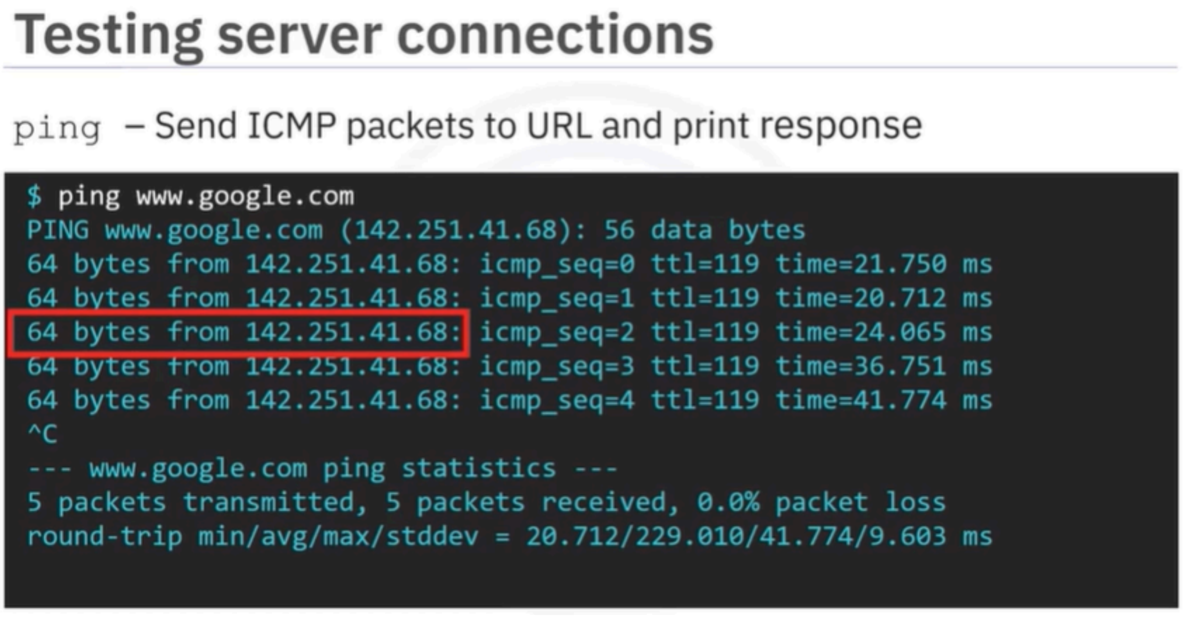


Getting ethernet adapter info

`ip addr show eth0` - Display details for a specific device (eth0)

```
$ ip addr show eth0
3: eth0@if220: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1480 qdisc noqueue
state UP group default qlen 1000
    link/ether 16:6f:59:ae:32:15 brd ff:ff:ff:ff:ff:ff link-netnsid 0
    inet 172.22.131.103/32 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::146f:59ff:feae:3215/64 scope link
       valid_lft forever preferred_lft forever
```

- To display details for a specific device, Such as an Ethernet adapter named ETH0, Use the command IP address show eth0. This will show information about the adapter, Including its IP address, The number of packets received and transmitted, And key metrics such as errors, dropped packets, And total data sent and received.

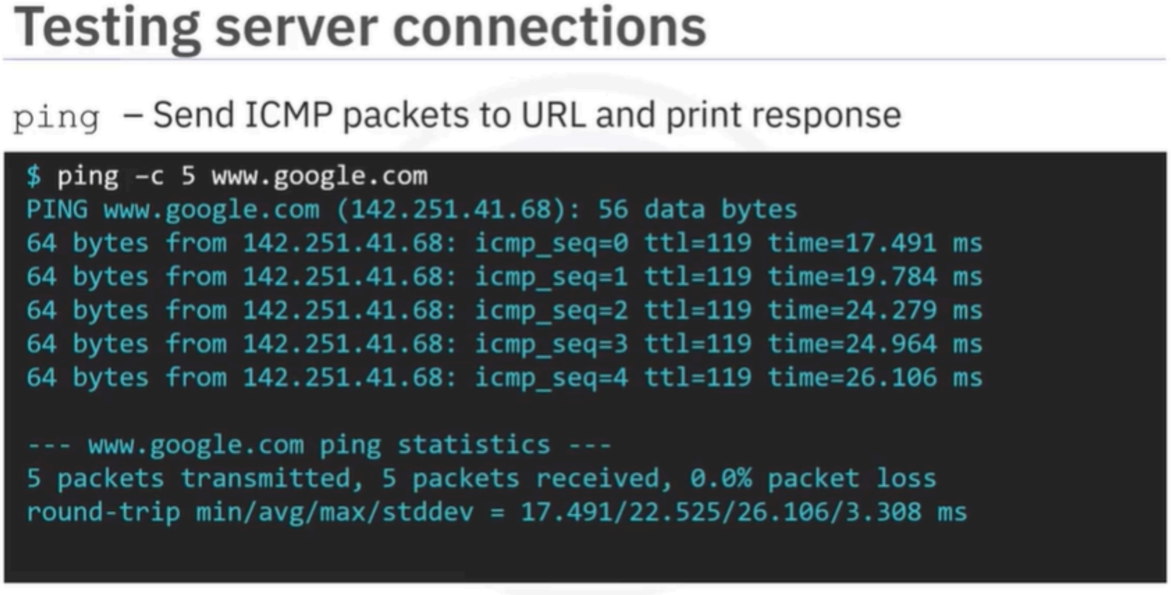


Testing server connections

`ping` – Send ICMP packets to URL and print response

```
$ ping www.google.com
PING www.google.com (142.251.41.68): 56 data bytes
64 bytes from 142.251.41.68: icmp_seq=0 ttl=119 time=21.750 ms
64 bytes from 142.251.41.68: icmp_seq=1 ttl=119 time=20.712 ms
64 bytes from 142.251.41.68: icmp_seq=2 ttl=119 time=24.065 ms
64 bytes from 142.251.41.68: icmp_seq=3 ttl=119 time=36.751 ms
64 bytes from 142.251.41.68: icmp_seq=4 ttl=119 time=41.774 ms
^C
--- www.google.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 20.712/229.010/41.774/9.603 ms
```

- You can use the ping command To test connectivity to a host or IP address. Ping sends packets, known as ICMP, Internet Control Message Protocol requests, To the server, listens for a response, And prints a result. For example, by entering ping google dot com, Ping returns a line of info For each successful response to an echo request, And continues until you abort with control c. And after ping aborts, It prints summary statistics for the ping results. For each echo request, Ping reports useful information, Including the IP address of the given URL, 142.251.41.68, And the total round-trip time in milliseconds. The provided statistics at the end include How many packets were transmitted and received, The percentage of packets dropped, And the minimum, average, maximum, And standard deviation of the round-trip times in milliseconds. If you would like the ping command To return a set number of ping results,



Testing server connections

`ping` – Send ICMP packets to URL and print response

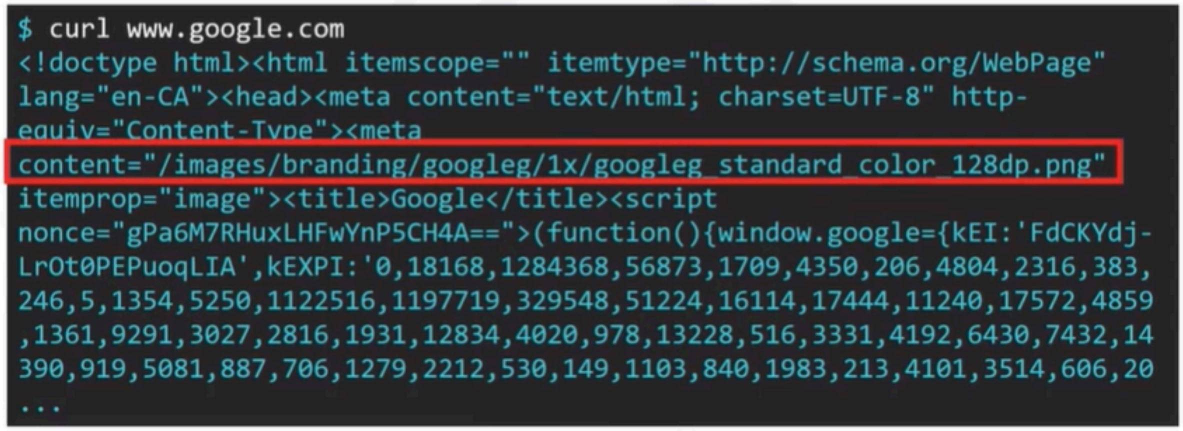
```
$ ping -c 5 www.google.com
PING www.google.com (142.251.41.68): 56 data bytes
64 bytes from 142.251.41.68: icmp_seq=0 ttl=119 time=17.491 ms
64 bytes from 142.251.41.68: icmp_seq=1 ttl=119 time=19.784 ms
64 bytes from 142.251.41.68: icmp_seq=2 ttl=119 time=24.279 ms
64 bytes from 142.251.41.68: icmp_seq=3 ttl=119 time=24.964 ms
64 bytes from 142.251.41.68: icmp_seq=4 ttl=119 time=26.106 ms

--- www.google.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 17.491/22.525/26.106/3.308 ms
```

- You can use the minus c option. Entering ping minus c five google dot com Returns the five ping results, Aborts, and then prints the same statistics It would print without the minus c option.

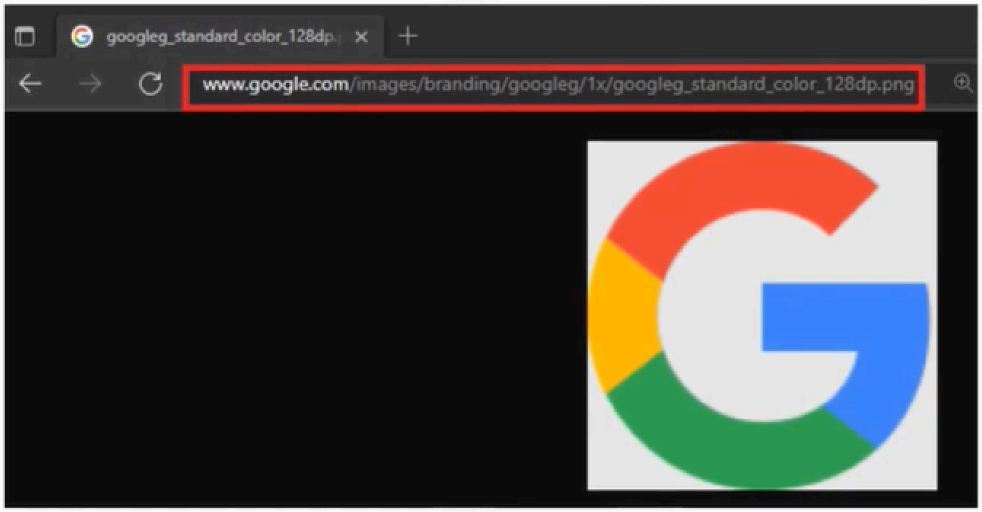
Web scraping with curl

`curl` (Client URL) – Transfer data to and from URL(s)

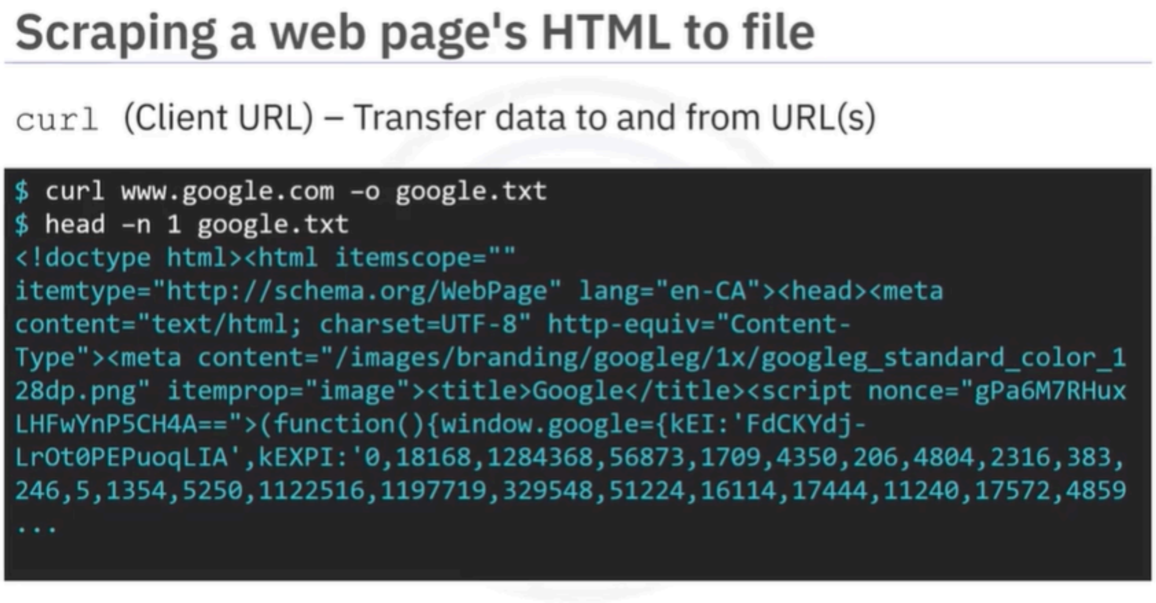


```
$ curl www.google.com
<!doctype html><html itemscope="" itemtype="http://schema.org/WebPage"
lang="en-CA"><head><meta content="text/html; charset=UTF-8" http-
equiv="Content-Type"><meta
content="/images/branding/googleg/1x/googleg_standard_color_128dp.png"
itemprop="image"><title>Google</title><script
nonce="gPa6M7RHuxLHFwYnP5CH4A==">(function(){window.google={kEI:'FdCKYdj-
LrOt0PEPuoqLIA',kEXPI:'0,18168,1284368,56873,1709,4350,206,4804,2316,383,
246,5,1354,5250,1122516,1197719,329548,51224,16114,17444,11240,17572,4859
,1361,9291,3027,2816,1931,12834,4020,978,13228,516,3331,4192,6430,7432,14
390,919,5081,887,706,1279,2212,530,149,1103,840,1983,213,4101,3514,606,20
...
```

Web scraping with curl


googleg_standard_color_128dp.
www.google.com/images/branding/googleg/1x/googleg_standard_color_128dp.png

- The curl command is a powerful tool That enables you to transfer data to and from URLs And supports many different protocols. Entering curl www.google.com Returns the entire HTML content of the landing page At www.google.com Using the default HTTP protocol. You can see, for example, The path to a PNG file for the Google G logo.
- You can render the logo by appending its path To google.com in your browser.

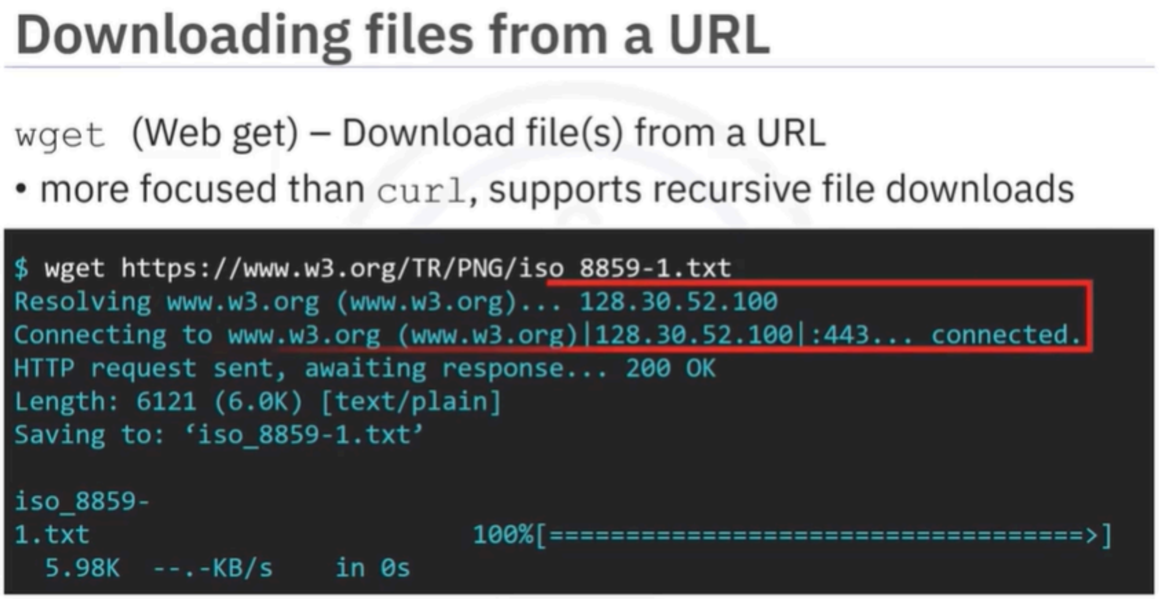


Scraping a web page's HTML to file

`curl` (Client URL) – Transfer data to and from URL(s)

```
$ curl www.google.com -o google.txt
$ head -n 1 google.txt
<!doctype html><html itemscope=""
itemtype="http://schema.org/WebPage" lang="en-CA"><head><meta
content="text/html; charset=UTF-8" http-equiv="Content-
Type"><meta content="/images/branding/googleg/1x/googleg_standard_color_1
28dp.png" itemprop="image"><title>Google</title><script nonce="gPa6M7RHux
LHFwYnP5CH4A==">(function(){window.google={kEI:'FdCKYdj-
LrOt0PEPuoqLIA',kEXPI:'0,18168,1284368,56873,1709,4350,206,4804,2316,383,
246,5,1354,5250,1122516,1197719,329548,51224,16114,17444,11240,17572,4859
...
```

- You can even get curl to write the contents of a URL To a local file. This is done using the minus o option. For example, you can enter curl www.google.com Together with the minus o option And a file name such as google.txt. You can then view the contents of google.txt Using the head command And verify that the file contents indeed Match the previous output.



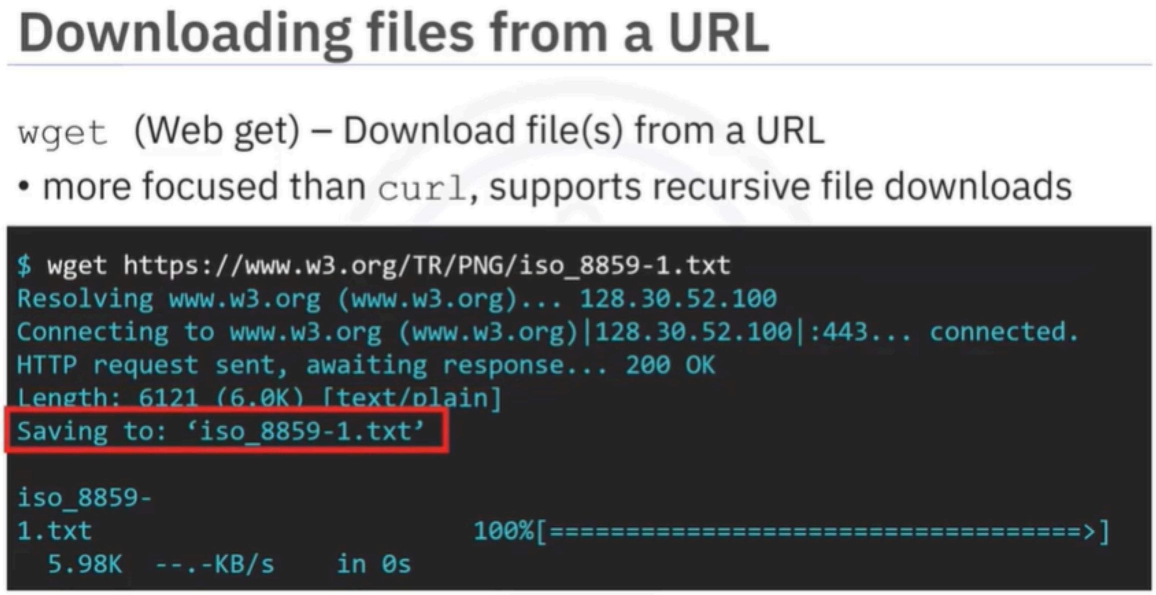
Downloading files from a URL

`wget` (Web get) – Download file(s) from a URL

- more focused than `curl`, supports recursive file downloads

```
$ wget https://www.w3.org/TR/PNG/iso_8859-1.txt
Resolving www.w3.org (www.w3.org)... 128.30.52.100
Connecting to www.w3.org (www.w3.org)|128.30.52.100|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6121 (6.0K) [text/plain]
Saving to: 'iso_8859-1.txt'

iso_8859-
1.txt                          100%[===================================>]
  5.98K  --.-KB/s    in 0s
```



Downloading files from a URL

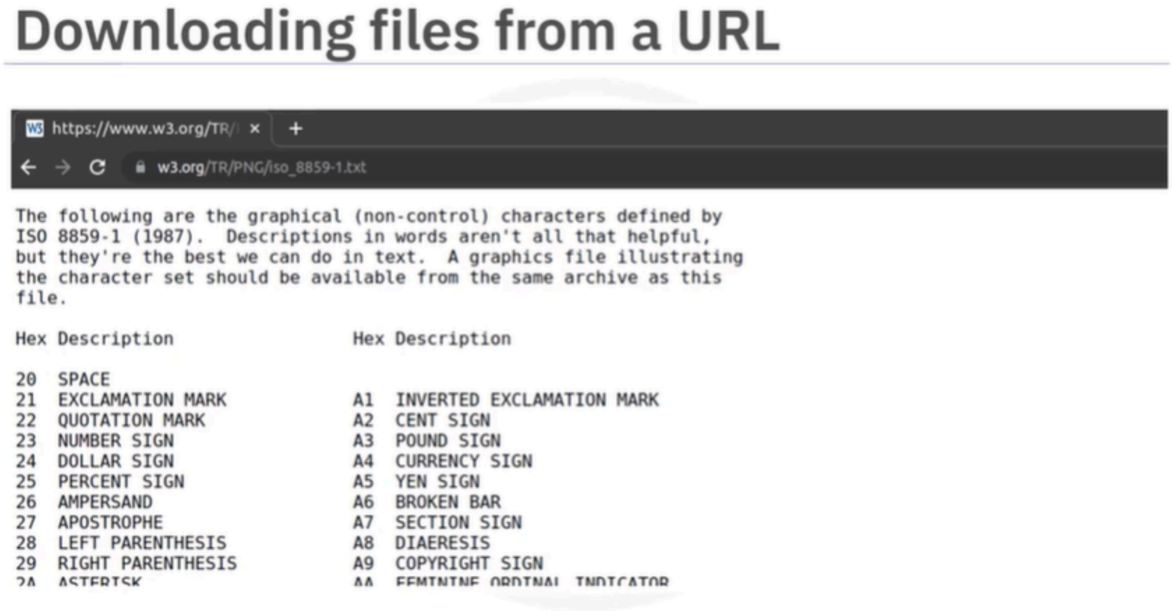
`wget` (Web get) – Download file(s) from a URL

- more focused than `curl`, supports recursive file downloads

```
$ wget https://www.w3.org/TR/PNG/iso_8859-1.txt
Resolving www.w3.org (www.w3.org)... 128.30.52.100
Connecting to www.w3.org (www.w3.org)|128.30.52.100|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 6121 (6.0K) [text/plain]
Saving to: 'iso_8859-1.txt'

iso_8859-
1.txt                          100%[===================================>]
  5.98K  --.-KB/s    in 0s
```

- Similar to curl, the wget command Is used to retrieve files located at a URL. Wget is like curl in that it can retrieve a file Located at a URL or the HTML code for a web page. But it's more specialized in its protocol support And has recursive downloading capabilities. This is useful when a URL might point to a folder That contains several files. Here you use wget to download a single test file Called iso underscore eight eight five nine dash one dot text Which is hosted by w3.org. Wget returns information while it is downloading Such as resolving and connecting to the target server. HTTP requests sent, awaiting a response And saving the file, which it automatically names for you To the current directory.

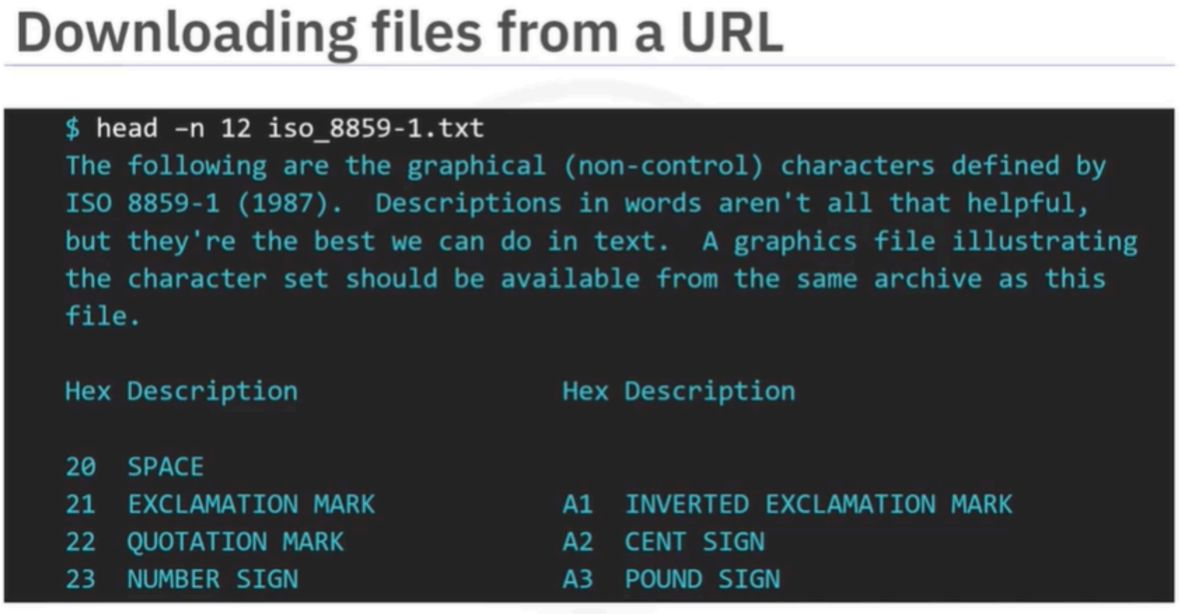


Downloading files from a URL

```
The following are the graphical (non-control) characters defined by
ISO 8859-1 (1987).  Descriptions in words aren't all that helpful,
but they're the best we can do in text.  A graphics file illustrating
the character set should be available from the same archive as this
file.

Hex Description                 Hex Description

20  SPACE
21  EXCLAMATION MARK            A1  INVERTED EXCLAMATION MARK
22  QUOTATION MARK              A2  CENT SIGN
23  NUMBER SIGN                 A3  POUND SIGN
24  DOLLAR SIGN                 A4  CURRENCY SIGN
25  PERCENT SIGN                A5  YEN SIGN
26  AMPERSAND                   A6  BROKEN BAR
27  APOSTROPHE                  A7  SECTION SIGN
28  LEFT PARENTHESIS            A8  DIAERESIS
29  RIGHT PARENTHESIS           A9  COPYRIGHT SIGN
2A  ASTERISK                    AA  FEMININE ORDINAL INDICATOR
```



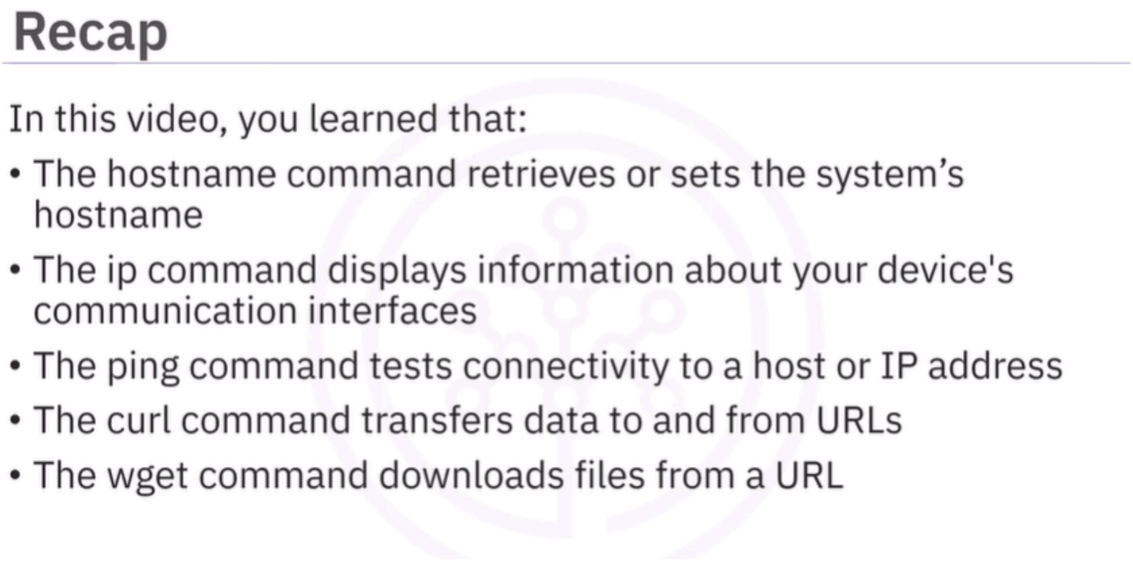
Downloading files from a URL

```
$ head -n 12 iso_8859-1.txt
The following are the graphical (non-control) characters defined by
ISO 8859-1 (1987).  Descriptions in words aren't all that helpful,
but they're the best we can do in text.  A graphics file illustrating
the character set should be available from the same archive as this
file.

Hex Description                 Hex Description

20  SPACE
21  EXCLAMATION MARK            A1  INVERTED EXCLAMATION MARK
22  QUOTATION MARK              A2  CENT SIGN
23  NUMBER SIGN                 A3  POUND SIGN
```

- For reference, here is what your browser shows you When viewing looking at the data located at the URL. It's a simple text file. By entering head minus twelve iso underscore eight eight five nine Dash one dot text You see the first twelve lines of the contents of the downloaded file. As expected, the file contains the exact same data That you just saw on the previous slide.



Recap

In this video, you learned that:

- The hostname command retrieves or sets the system’s hostname
- The ip command displays information about your device's communication interfaces
- The ping command tests connectivity to a host or IP address
- The curl command transfers data to and from URLs
- The wget command downloads files from a URL