

Wrap text with fmt

```
$ openssl ciphers
ECDHE-RSA-AES256-GCM-SHA384: ECDHE-ECDSA-AES256-GCM-SHA384: ECDHE-RSA-AES256-SHA384: ECDHE-ECDSA-AES256-SHA384: ECDHE-RSA-AES256-SHA: ECDHE-ECDSA-AES256-SHA: SRP-DSS-AES-256-CBC-SHA: SRP-RSA-AES-256-CBC-SHA: SRP-AES-256-CBC-SHA: DH-DSS-AES256-GCM-SHA384: DHE-DSS-AES256-GCM-SHA384: DH-RSA-AES256-GCM-SHA384: DHE-RSA-AES256-GCM-SHA384: DHE-RSA-AES256-SHA256: DHE-DSS-AES256-SHA256: DH-RSA-AES256-SHA256: DH-DSS-AES256-SHA256: DHE-RSA-AES256-SHA: DHE-DSS-AES256-SHA: DH-RSA-AES256-SHA: DH-DSS-AES256-SHA: DHE-RSA-CAMELLIA256-SHA: DHE-DSS-CAMELLIA256-SHA: DH-RSA-CAMELLIA256-SHA: DH-DSS-CAMELLIA256-SHA: ECDH-RSA-AES256-GCM-SHA384: ECDH-ECDSA-AES256-GCM-SHA384: ECDH-RSA-AES256-SHA384: ECDH-ECDSA-AES256-SHA384: ECDH-RSA-AES256-SHA: ECDH-ECDSA-AES256-SHA: AES256-GCM-SHA384: AES256-SHA256: AES256-SHA: CAMELLIA256-SHA: PSK-AES256-CBC-SHA: ECDHE-RSA-AES128-GCM-SHA256: ECDHE-ECDSA-AES128-GCM-SHA256: ECDHE-RSA-AES128-SHA256: ECDHE-ECDSA-AES128-SHA256: ECDHE-RSA-AES128-SHA: ECDHE-ECDSA-AES128-SHA: SRP-DSS-AES-128-CBC-SHA: SRP-RSA-AES-128-CBC-SHA: SRP-AES-128-CBC-SHA: DH-DSS-AES128-GCM-SHA256: DHE-DSS-AES128-GCM-SHA256: DH-RSA-AES128-GCM-SHA256: DHE-RSA-AES128-GCM-SHA256: DHE-RSA-AES128-SHA256: DHE-DSS-AES128-SHA256: DH-RSA-AES128-SHA256: DH-DSS-AES128-SHA256: DHE-RSA-AES128-SHA: DHE-DSS-AES128-SHA: DH-RSA-AES128-SHA: DH-DSS-AES128-SHA: DHE-RSA-SEED-SHA: DHE-DSS-SEED-SHA: DH-RSA-SEED-SHA: DH-DSS-SEED-SHA: DHE-RSA-CAMELLIA128-SHA: DHE-DSS-CAMELLIA128-SHA: DH-RSA-CAMELLIA128-SHA: DH-DSS-CAMELLIA128-SHA: ECDH-RSA-AES128-GCM-SHA256: ECDH-ECDSA-AES128-GCM-SHA256: ECDH-RSA-AES128-SHA256: ECDH-ECDSA-AES128-SHA256: ECDH-RSA-AES128-SHA: ECDH-ECDSA-AES128-SHA: AES128-GCM-SHA256: AES128-SHA256: AES128-SHA: SEED-SHA: CAMELLIA128-SHA: PSK-AES128-CBC-SHA: ECDHE-RSA-RC4-SHA: ECDHE-ECDSA-RC4-SHA: ECDH-RSA-RC4-SHA: ECDH-ECDSA-RC4-SHA: RC4-SHA: RC4-MD5: PSK-RC4-SHA: ECDHE-RSA-DES-CBC3-SHA: ECDHE-ECDSA-DES-CBC3-SHA: SRP-DSS-3DES-EDE-CBC-SHA: SRP-RSA-3DES-EDE-CBC-SHA: SRP-3DES-EDE-CBC-SHA: EDH-RSA-DES-CBC3-SHA: EDH-DSS-DES-CBC3-SHA: DH-RSA-DES-CBC3-SHA: DH-DSS-DES-CBC3-SHA: ECDH-RSA-DES-CBC3-SHA: ECDH-ECDSA-DES-CBC3-SHA: DES-CBC3-SHA: PSK-3DES-EDE-CBC-SHA: EDH-RSA-DES-CBC-SHA: EDH-DSS-DES-CBC-SHA: DH-RSA-DES-CBC-SHA: DH-DSS-DES-CBC-SHA: DES-CBC-SHA
$ openssl ciphers |
> sed 's:/:/g' | # Separate words with space
> fmt | # Format words in lines
> head
ECDHE-RSA-AES256-GCM-SHA384 ECDHE-ECDSA-AES256-GCM-SHA384
ECDHE-RSA-AES256-SHA384 ECDHE-ECDSA-AES256-SHA384 ECDHE-RSA-AES256-SHA
ECDHE-ECDSA-AES256-SHA SRP-DSS-AES-256-CBC-SHA SRP-RSA-AES-256-CBC-SHA
SRP-AES-256-CBC-SHA DH-DSS-AES256-GCM-SHA384 DHE-DSS-AES256-GCM-SHA384
DH-RSA-AES256-GCM-SHA384 DHE-RSA-AES256-GCM-SHA384
DHE-RSA-AES256-SHA256 DHE-DSS-AES256-SHA256 DH-RSA-AES256-SHA256
DH-DSS-AES256-SHA256 DHE-RSA-AES256-SHA DHE-DSS-AES256-SHA
DH-RSA-AES256-SHA DH-DSS-AES256-SHA DHE-RSA-CAMELLIA256-SHA
DHE-DSS-CAMELLIA256-SHA DH-RSA-CAMELLIA256-SHA DH-DSS-CAMELLIA256-SHA
ECDH-RSA-AES256-GCM-SHA384 ECDH-ECDSA-AES256-GCM-SHA384
$
```

Send plain text email

```
$ sendmail john.smith@example.com <<\EOF
> From: Alice Jones <alice.jones@example.com>
> To: John Smith <john.smith@example.com>
> Subject: Hi there
>
> I'm learning how to send email from the command line.
> EOF
$
```

Tailor email

```
$ cat send-connections.sh
#!/bin/sh
sendmail john.smith@example.com <<EOF
```

From: Diomidis Spinellis <dds@aueb.gr>
To: John Smith <john.smith@example.com>
Date: \$(date -R)
Subject: Current network connections

These are the currently active network connections.

```
$(netstat)
```

```
EOF
```

```
$ sh send-connections.sh
```

```
$
```

Convert attachments into text

```
$ dd if=/dev/random of=data count=32 bs=1
```

```
32+0 records in
```

```
32+0 records out
```

```
32 bytes (32 B) copied, 0.0688745 s, 0.5 kB/s
```

```
$ more data
```

```
^Z<FB><AF>^_w[<FB><<99>f<92><A2><9A>.<82>b<AB><87>^P>^G<D7><E4>N6<8A>^_<F4>t<A7><95>^K
```

```
$ base64 data >data.base64
```

```
$ cat data.base64
```

```
GvuvH3db+zyZZpKimi6CYquHED4H1+RONoof9HSnlQs=
```

```
$ base64 -d <data.base64 >data.decoded
```

```
$ cmp data data.decoded && echo Files are the same
```

```
Files are the same
```

```
$
```

Send attachments

```
$ cat send-fixed-email.sh
```

```
#!/bin/sh
```

```
sendmail john.smith@example.com <<\EOF
```

```
To: John Smith <john.smith@example.com>
```

```
From: Diomidis Spinellis <dds@example.com>
```

```
Subject: Here are your Hadoop results
```

```
Date: Sun, 3 Apr 2016 09:58:48 +0300
```

```
MIME-Version: 1.0
```

```
Content-Type: multipart/mixed;
```

```
boundary="-----030303090406090809090501"
```

This is a multi-part message in MIME format.

```
-----030303090406090809090501
```

```
Content-Type: text/plain; charset=utf-8; format=flowed
```

```
Content-Transfer-Encoding: 7bit
```

The attachment contains your Hadoop results.

```
-----030303090406090809090501
```

```
Content-Type: application/octet-stream;
```

```
name="data"
```

```
Content-Transfer-Encoding: base64
```

```
Content-Disposition: attachment;
```

```
filename="data"
```

```
HviDNR105+2Tr0+0fsx30yzNueQqPuAXL9IUraf0i7Y=
```

```
-----030303090406090809090501--
```

```
EOF
```

```
$ cat send-variable-email.sh
```

```
#!/bin/sh
```

```
sendmail john.smith@example.com <<EOF
```

```
To: John Smith <john.smith@example.com>
```

From: Diomidis Spinellis <dds@example.com>
Subject: Here are your Hadoop results
Message-ID: <5700BF28.2070500@example.com>
Date: \$(date -R)
MIME-Version: 1.0
Content-Type: multipart/mixed;
boundary="-----030303090406090809090501"

This is a multi-part message in MIME format.
-----030303090406090809090501
Content-Type: text/plain; charset=utf-8; format=flowed
Content-Transfer-Encoding: 7bit

The attachment contains your Hadoop results.

-----030303090406090809090501
Content-Type: application/octet-stream;
name="data"
Content-Transfer-Encoding: base64
Content-Disposition: attachment;
filename="data"

\$(base64 binary-data-file.dat)
-----030303090406090809090501--
EOF
\$