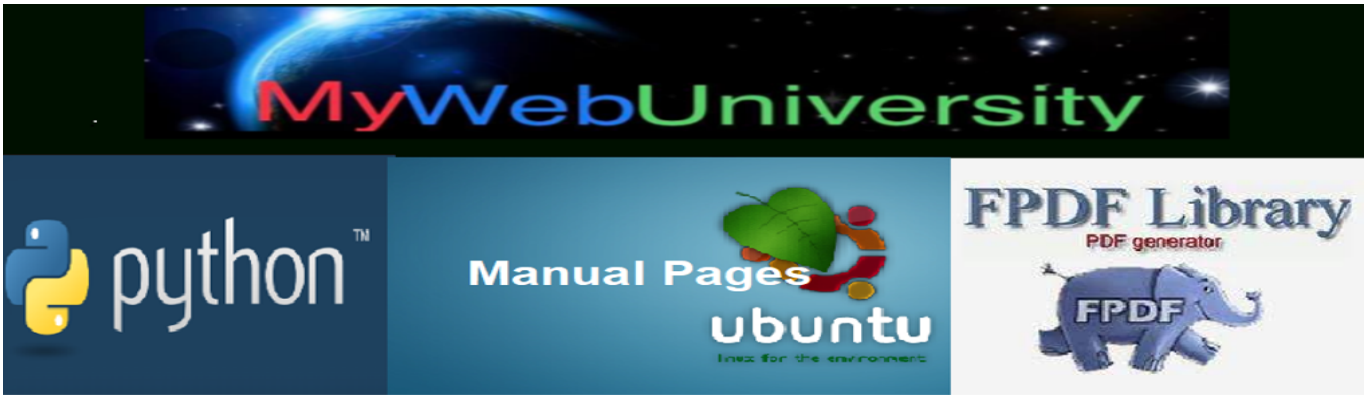

Rocky Enterprise Linux 9.2 Manual Pages on command 'Crypt::Digest::BLAKE2s_256.3pm'

$ man Crypt::Digest::BLAKE2s_256.3pm

Crypt::Digest::BLAKE2s_256(3pm)User Contributed Perl DocumentationCrypt::Digest::BLAKE2s_256(3pm)

NAME

Crypt::Digest::BLAKE2s_256 - Hash function BLAKE2s [size: 256 bits]

SYNOPSIS

```
### Functional interface:
use Crypt::Digest::BLAKE2s_256 qw( blake2s_256 blake2s_256_hex blake2s_256_b64 blake2s_256_b64u
                                   blake2s_256_file blake2s_256_file_hex blake2s_256_file_b64 blake2s_256_file_b64u );

# calculate digest from string/buffer
$blake2s_256_raw  = blake2s_256('data string');
$blake2s_256_hex  = blake2s_256_hex('data string');
$blake2s_256_b64  = blake2s_256_b64('data string');
$blake2s_256_b64u = blake2s_256_b64u('data string');
# calculate digest from file
$blake2s_256_raw  = blake2s_256_file('filename.dat');
$blake2s_256_hex  = blake2s_256_file_hex('filename.dat');
$blake2s_256_b64  = blake2s_256_file_b64('filename.dat');
$blake2s_256_b64u = blake2s_256_file_b64u('filename.dat');
# calculate digest from filehandle
$blake2s_256_raw  = blake2s_256_file(*FILEHANDLE);
$blake2s_256_hex  = blake2s_256_file_hex(*FILEHANDLE);
```

```
$blake2s_256_b64  = blake2s_256_file_b64(*FILEHANDLE);
$blake2s_256_b64u = blake2s_256_file_b64u(*FILEHANDLE);

### OO interface:
use Crypt::Digest::BLAKE2s_256;

$d = Crypt::Digest::BLAKE2s_256->new;
$d->add('any data');
$d->addfile('filename.dat');
$d->addfile(*FILEHANDLE);
$result_raw  = $d->digest;     # raw bytes
$result_hex  = $d->hexdigest;  # hexadecimal form
$result_b64  = $d->b64digest;  # Base64 form
$result_b64u = $d->b64udigest; # Base64 URL Safe form
```

DESCRIPTION

Provides an interface to the BLAKE2s_256 digest algorithm.

EXPORT

Nothing is exported by default.

You can export selected functions:

```
use Crypt::Digest::BLAKE2s_256 qw(blake2s_256 blake2s_256_hex blake2s_256_b64 blake2s_256_b64u
                                  blake2s_256_file blake2s_256_file_hex blake2s_256_file_b64 blake2s_256_file_b64u);
```

Or all of them at once:

```
use Crypt::Digest::BLAKE2s_256 ':all';
```

FUNCTIONS

blake2s_256

Logically joins all arguments into a single string, and returns its BLAKE2s_256 digest

encoded as a binary string.

```
$blake2s_256_raw = blake2s_256('data string');
#or
$blake2s_256_raw = blake2s_256('any data', 'more data', 'even more data');
```

blake2s_256_hex

Logically joins all arguments into a single string, and returns its BLAKE2s_256 digest encoded as a hexadecimal string.

```
$blake2s_256_hex = blake2s_256_hex('data string');
#or
$blake2s_256_hex = blake2s_256_hex('any data', 'more data', 'even more data');
```

blake2s_256_b64

Logically joins all arguments into a single string, and returns its BLAKE2s_256 digest encoded as a Base64 string, with trailing '=' padding.

```
$blake2s_256_b64 = blake2s_256_b64('data string');
#or
$blake2s_256_b64 = blake2s_256_b64('any data', 'more data', 'even more data');
```

blake2s_256_b64u

Logically joins all arguments into a single string, and returns its BLAKE2s_256 digest encoded as a Base64 URL Safe string (see RFC 4648 section 5).

```
$blake2s_256_b64url = blake2s_256_b64u('data string');
#or
$blake2s_256_b64url = blake2s_256_b64u('any data', 'more data', 'even more data');
```

blake2s_256_file

Reads file (defined by filename or filehandle) content, and returns its BLAKE2s_256 digest encoded as a binary string.

```
$blake2s_256_raw = blake2s_256_file('filename.dat');
#or
$blake2s_256_raw = blake2s_256_file(*FILEHANDLE);
```

blake2s_256_file_hex

Reads file (defined by filename or filehandle) content, and returns its BLAKE2s_256 digest encoded as a hexadecimal string.

```
$blake2s_256_hex = blake2s_256_file_hex('filename.dat');
#or
$blake2s_256_hex = blake2s_256_file_hex(*FILEHANDLE);
```

BEWARE: You have to make sure that the filehandle is in binary mode before you pass it as argument to the addfile() method.

blake2s_256_file_b64

Reads file (defined by filename or filehandle) content, and returns its BLAKE2s_256 digest encoded as a Base64 string, with trailing '=' padding.

```
$blake2s_256_b64 = blake2s_256_file_b64('filename.dat');
#or
$blake2s_256_b64 = blake2s_256_file_b64(*FILEHANDLE);
```

blake2s_256_file_b64u

Reads file (defined by filename or filehandle) content, and returns its BLAKE2s_256 digest encoded as a Base64 URL Safe string (see RFC 4648 section 5).

```
$blake2s_256_b64url = blake2s_256_file_b64u('filename.dat');
#or
$blake2s_256_b64url = blake2s_256_file_b64u(*FILEHANDLE);
```

The OO interface provides the same set of functions as Crypt::Digest.

new

```
$d = Crypt::Digest::BLAKE2s_256->new();
```

clone

```
$d->clone();
```

reset

```
$d->reset();
```

add

```
$d->add('any data');
#or
$d->add('any data', 'more data', 'even more data');
```

addfile

```
$d->addfile('filename.dat');
#or
$d->addfile(*FILEHANDLE);
```

add_bits

```
$d->add_bits($bit_string);   # e.g. $d->add_bits("111100001010");
#or
$d->add_bits($data, $nbits); # e.g. $d->add_bits("\xF0\xA0", 16);
```

hashsize

```
$d->hashsize;
#or
Crypt::Digest::BLAKE2s_256->hashsize();
#or
Crypt::Digest::BLAKE2s_256::hashsize();
```

digest

```
$result_raw = $d->digest();
```

hexdigest

```
$result_hex = $d->hexdigest();
```

b64digest

```
$result_b64 = $d->b64digest();
```

b64udigest

```
$result_b64url = $d->b64udigest();
```

SEE ALSO

- CryptX, Crypt::Digest
- <https://blake2.net/>
- <https://tools.ietf.org/html/rfc7693>