

# PERL -X FUNCTION

[http://www.tutorialspoint.com/perl/perl\\_-X.htm](http://www.tutorialspoint.com/perl/perl_-X.htm)

Copyright © tutorialspoint.com

## Syntax

```
-X FILEHANDLE
```

```
-X
```

## Definition and Usage

A file test, where X is one of the letters listed below. This unary operator takes one argument, either a filename or a filehandle, and tests the associated file to see if something is true about it.

If the argument is omitted, tests \$\_

## Return Value

- 1 if condition is true
- 0 if condition is false

```
-r File is readable by effective uid/gid.
-w File is writable by effective uid/gid.
-x File is executable by effective uid/gid.
-o File is owned by effective uid.

-R File is readable by real uid/gid.
-W File is writable by real uid/gid.
-X File is executable by real uid/gid.
-O File is owned by real uid.

-e File exists.
-z File has zero size (is empty).
-s File has nonzero size (returns size in bytes).

-f File is a plain file.
-d File is a directory.
-l File is a symbolic link.
-p File is a named pipe (FIFO), or Filehandle is a pipe.
-S File is a socket.
-b File is a block special file.
-c File is a character special file.
-t Filehandle is opened to a tty.

-u File has setuid bit set.
-g File has setgid bit set.
-k File has sticky bit set.

-T File is an ASCII text file (heuristic guess).
-B File is a "binary" file (opposite of -T).

-M Script start time minus file modification time, in days.
-A Same for access time.
-C Same for inode change time
```

## Example

Try out following example with some file.

```
stat($filename);
print "Readable\n" if -r _;
print "Writable\n" if -w _;
print "Executable\n" if -x _;
```

```
print "Setuid\n" if -u _;  
print "Setgid\n" if -g _;  
print "Sticky\n" if -k _;  
print "Text\n" if -T _;  
print "Binary\n" if -B _;  
  
# Another way of testing  
if( -e $filename ){  
print " File $filename exists\n";  
}
```