PERL PIPE FUNCTION

Description

This function opens a pair of connected communications pipes: READHANDLE for reading and WRITEHANDLE for writing. YOu may need to set $| to flush your WRITEHANDLE after each command.

Syntax

Following is the simple syntax for this function −

```
pipe READHANDLE, WRITEHANDLE
```

Return Value

This function returns 0 on failure and 1 on success.

Example

Following is the example code showing its basic usage −

```perl
#!/usr/bin/perl -w

use IO::Handle;

pipe(PARENTREAD, PARENTWRITE);
pipe(CHILDREAD, CHILDWRITE);

PARENTWRITE->autoflush(1);
CHILDWRITE->autoflush(1);

if ($child = fork) # Parent code
{
   close CHILDREAD; # We don't need these in the parent
   close PARENTWRITE;
   print CHILDWRITE "34+56;\n";
   chomp($result = <PARENTREAD>);
   print "Got a value of $result from child\n";
   close PARENTREAD;
   close CHILDWRITE;
   waitpid($child,0);
}else{
   close PARENTREAD; # We don't need these in the child
   close CHILDWRITE;
   chomp($calculation = <CHILDREAD>);
   print "Got $calculation\n";
   $result = eval "$calculation";
   print PARENTWRITE "$result\n";
   close CHILDREAD;
   close PARENTWRITE;
   exit;
}
```

It will produce following results: You can see that the calculation is sent to CHILDWRITE, which is then read by the child from CHILDREAD. The result is then calculated and sent back to the parent via PARENTWRITE, where the parent reads the result from PARENTREAD.

```
Got 34+56;
Got a value of 90 from child
```