# Environment Variables

You have already been introduced to environment variables and have put them to good use. To recap, an environment variable is a storage location that has a name and a value. They often effect the way programs behave. For example, you learned how to inform various programs about your preferred editor by defining the $EDITOR environment variable.

#### Common Environment Variables

| Variable | Description |
|---|---|
| EDITOR | The program to run to perform edits. |
| HOME | The Home directory of the user. |
| LOGNAME | The login name of the user. |
| MAIL | The location of the user's local inbox. |
| OLDPWD | The previous working directory. |
| PATH | A colon separated list of directories to search for commands. |
| PAGER | This program may be called to view a file. |
| PS1 | The primary prompt string. |
| PWD | The present working directory. |
| USER | The username of the user. |

#### Viewing Environment Variables

If you know the name of the environment variable that you want to examine, you can run `echo $VARIABLE_NAME` or `printenv VARIABLE_NAME`. If you want to examine all the environment variables that are set, use the `env` or `printenv` commands.

`printenv` - Print all or part of environment.

```
$ printenv HOME
/home/bob
$ echo $HOME
/home/bob
$ printenv
TERM=xterm-256color
SHELL=/bin/bash
USER=bob
PATH=/usr/local/bin:/usr/bin:/bin
MAIL=/var/mail/bob
PWD=/home/bob
LANG=en_US.UTF-8
HOME=/home/bob
LOGNAME=bob
$ env
TERM=xterm-256color
SHELL=/bin/bash
USER=bob
PATH=/usr/local/bin:/usr/bin:/bin
MAIL=/var/mail/bob
PWD=/home/bob
LANG=en_US.UTF-8
HOME=/home/bob
LOGNAME=bob
$
```

#### Exporting Environment Variables

When a process is started it inherits the exported environment variables of the process that spawned it. A variable that is set or changed only effects the current running process unless it is exported. The variables that are not exported are called local variables. The `export` command allows variables to be used by subsequently executed commands. Here is an example.

```
$ echo $PAGER

$ PAGER=less
$ echo $PAGER
less
$ bash
$ echo $PAGER

$ exit
exit
$ export PAGER=less
$ bash
$ echo $PAGER
less
$ exit
exit
$
```

In the above example `PAGER` was defined in the current environment. When you start a child process it inherits all the environment variables that were exported in your current environment. Since PAGER was not exported it was not set in the spawned bash shell. When you exported PAGER you saw that it was indeed available in the child process.

## Removing Variables from the Environment

You can use `unset` to remove or delete an environment variable.

```
$ echo $PAGER
less
$ unset PAGER
$ echo $PAGER

$
```