Execute the following command in the Terminal to download and install these components:

```
xcode-select --install
```

You'll be prompted to start the installation, and then prompted again to accept a software license. Then the tools will download and install automatically.
You can now install Homebrew.

Step 3 — Installing and Setting Up Homebrew

To install Homebrew, you'll download an installation script and then execute the script.

First, download the script to your local machine by typing the following command in your Terminal window:

```
curl -fsSL -o install.sh
https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh
```

The command uses `curl` to download the Homebrew installation script from Homebrew's Git repository on GitHub.

Let's walk through the flags that are associated with the `curl` command:

- The `-f` or `--fail` flag tells the Terminal window to give no HTML document output on server errors.
- The `-s` or `--silent` flag mutes `curl` so that it does not show the progress meter, and combined with the `-S` or `--show-error` flag it will ensure that `curl` shows an error message if it fails.
- The `-L` or `--location` flag will tell `curl` to handle redirects. If the server reports that the requested page has moved to a different location, it'll automatically execute the request again using the new location.
- The `-o` switch specifies a local filename for the file. Rather than displaying the contents to the screen, the `-o` switch saves the contents into the file you specify.

Before running a script you've download from the Internet, you should review its contents so you know what the script will do. Use the `less` command to review the installation script so you understand what it will do"

```
less install.sh
```

Once you're comfortable with the contents of the script, execute the script with the `bash` command:

```
/bin/bash install.sh
```

The installation script will explain what it will do and will prompt you to confirm that you want to do it. This lets you know exactly what Homebrew is going to do to your system before you let it proceed. It also ensures you have the prerequisites in place before it continues.

You'll be prompted to enter your password during the process. However, when you type your password, your keystrokes will not display in the Terminal window. This is a security measure and is something you'll see often when prompted for passwords on the command line. Even though you don't see them, your keystrokes are being recorded by the system, so press the `RETURN` key once you've entered your password.

Press the letter `y` for "yes" whenever you are prompted to confirm the installation.

Once the installation process is complete, you will want to put the directory Homebrew uses to store its executables at the front of the `PATH` environment variable. This ensures that Homebrew installations will be called over the tools that macOS includes.

The file you'll modify depends on which shell you're using. ZSH is the default shell on macOS Mojave and higher. The Bash shell is a popular shell that older versions of macOS used as the default, and if you've upgraded your OS, you may still be using Bash.

Execute the following command to determine your shell:

```
echo $0
```

You'll see either `bash` or `zsh`.

If you're using ZSH, you'll open the file `~/.zshrc` in your editor:

```
nano ~/.zshrc
```

If you're using the Bash shell, you'll use the file `~/.bash_profile`:

```
nano ~/.bash_profile
```

Once the file opens up in the Terminal window, add the following lines to the end of the file:

~/.zshrc

```
# Add Homebrew's executable directory to the front of the PATH
export PATH=/usr/local/bin:$PATH
```

The first line is a comment that will help you remember what this does if you open this file in the future.

To save your changes, hold down the `CTRL` key and the letter `O`, and when prompted, press the `RETURN` key. Then exit the editor by holding the `CTRL` key and pressing `X`. This will return you to your Terminal prompt.

To activate these changes, close and reopen your Terminal app. Alternatively, use the `source` command to load the file you modified.

If you modified `.zshrc`, execute this command:

```
source ~/.zshrc
```

If you modified `.bash_profile`, execute this command:

```
source ~/.bash_profile
```

Once you have done this, the changes you have made to the `PATH` environment variable will take effect. They'll be set correctly when you log in again in the future, as the configuration file for your shell is executed automatically when you open the Terminal app.

Now let's verify that Homebrew is set up correctly. Execute this command:

```
brew doctor
```

If no updates are required at this time, you'll see this in your Terminal:

```
Output
Your system is ready to brew.
```

Otherwise, you may get a warning to run another command such as `brew update` to ensure that your installation of Homebrew is up to date. Follow any on-screen instructions to fix your environment before moving on.

Step 4 — Installing, Upgrading, and Removing Packages

Now that Homebrew is installed, use it to download a package. The `tree` command lets you see a graphical directory tree and is available via Homebrew.

Install `tree` with the `brew install` command:

```
brew install tree
```

Homebrew will update its list of packages and then download and install the `tree` command:

```
Output

Updating Homebrew...

==> Downloading
https://homebrew.bintray.com/bottles/tree-1.8.0.catalina.bottle.tar.gz
######################################################################## 100.0%
==> Pouring tree-1.8.0.catalina.bottle.tar.gz
🍺  /usr/local/Cellar/tree/1.8.0: 8 files, 117.2KB
```

Homebrew installs files to `/usr/local` by default, so they won't interfere with future macOS updates. Verify that `tree` is installed by displaying the command's location with the `which` command:

```
which tree
```

The output shows that `tree` is located in `/usr/local/bin`:

```
Output
```

```
/usr/local/bin/tree
```

Run the `tree` command to see the version:

```
tree --version
```

The version prints to the screen, indicating it's installed:

```
Output
```

```
tree v1.8.0 (c) 1996 - 2018 by Steve Baker, Thomas Moore, Francesc Rocher, Florian Sesser, Kyosuke Tokoro
```

Occasionally, you'll want to upgrade an existing package. Use the `brew upgrade` command, followed by the package name:

```
brew upgrade tree
```

You can run `brew upgrade` with no additional arguments to upgrade all programs and packages Homebrew manages.

When you install a new version, Homebrew keeps the older version around. After a while, you might want to reclaim disk space by removing these older copies. Run `brew cleanup` to remove all old versions of your Homebrew-managed software.

To remove a package you're no longer using, use `brew uninstall`. To uninstall the `tree` command, execute this command:

```
brew uninstall tree
```

The output shows that the package was removed:

```
Output
Uninstalling /usr/local/Cellar/tree/1.8.0... (8 files, 117.2KB)
```

You can use Homebrew to install desktop applications too.

Step 5 — Installing Desktop Applications

You're not restricted to using Homebrew for command-line tools. Homebrew Cask lets you install desktop applications. This feature is included with Homebrew, so there's nothing additional to install.

Test it out by using Homebrew to install Visual Studio Code. Execute the following command in your terminal:

```
brew install visual-studio-code
```

The application will install:

```
Output
==> Downloading https://update.code.visualstudio.com/1.58.2/darwin/stable
==> Downloading from
https://az764295.vo.msecnd.net/stable/c3f126316369cd610563c75b1b1725e0679a
dfb3/VSCode-darwin.zip
######################################################################## 
100.0%
==> Installing Cask visual-studio-code
==> Moving App 'Visual Studio Code.app' to '/Applications/Visual Studio
Code.app'
==> Linking Binary 'code' to '/usr/local/bin/code'
🍺  visual-studio-code was successfully installed!
```

You'll find the application in your `Applications` folder, just as if you'd installed it manually.

To remove it, use `brew uninstall`:

```
brew uninstall visual-studio-code
```

Homebrew will remove the installed software:

```
Output
```

```
==> Uninstalling Cask visual-studio-code
==> Backing App 'Visual Studio Code.app' up to
'/usr/local/Caskroom/visual-studio-code/1.58.2/Visual Studio Code.app'
==> Removing App '/Applications/Visual Studio Code.app'
==> Unlinking Binary '/usr/local/bin/code'
==> Purging files for version 1.58.2 of Cask visual-studio-code
```

It performs a backup first in case the removal fails, but once the program is fully uninstalled, the backup is removed as well.

Step 6 — Uninstalling Homebrew

If you no longer need Homebrew, you can use its uninstall script.

Download the uninstall script with `curl`:

```
curl -fsSL -o uninstall.sh
https://raw.githubusercontent.com/Homebrew/install/master/uninstall.sh
```

As always, review the contents of the script with the `less` command to verify the script's contents:

```
less uninstall.sh
```

Once you've verified the script, execute the script with the `--help` flag to see the various options you can use:

```
bash uninstall.sh --help
```

-

The options display on the screen:

```
Output
```

```
Homebrew Uninstaller
Usage: uninstall.sh [options]
    -p, --path=PATH  Sets Homebrew prefix. Defaults to /usr/local.
        --skip-cache-and-logs
                     Skips removal of HOMEBREW_CACHE and HOMEBREW_LOGS.
    -f, --force      Uninstall without prompting.
    -q, --quiet      Suppress all output.
    -d, --dry-run    Simulate uninstall but don't remove anything.
    -h, --help       Display this message.
```

Use the `-d` flag to see what the script will do:

```
bash uninstall.sh -d
```

-

The script will list everything it will delete:

```
Output
```

```
Warning: This script would remove:
/Users/brianhogan/Library/Caches/Homebrew/
/Users/brianhogan/Library/Logs/Homebrew/
/usr/local/Caskroom/
/usr/local/Cellar/
/usr/local/bin/brew -> /usr/local/bin/brew
==> Removing Homebrew installation...
Would delete:
....
```

When you’re ready to remove everything, execute the script without any flags:

```
bash uninstall.sh
```

-

This removes Homebrew and any programs you’ve installed with it.

Conclusion

In this tutorial you installed and used Homebrew on your Mac. You can now use Homebrew to install command line tools, programming languages, and other utilities you'll need for software development.

Homebrew has many packages you can install. Visit the official list to search for your favorite programs.