Install Hugo in Windows System

Assumptions

- You will use `C:\Hugo\Sites` as the starting point for your new project.
- You will use `C:\Hugo\bin` to store executable files.

Set up Your Directories

You'll need a place to store the **Hugo** executable, your content, and the generated Hugo website:

- Open Windows Explorer.
- Create a new folder: `C:\Hugo`, assuming you want Hugo on your *C* drive, although this can go anywhere
- Create a subfolder in the Hugo folder: `C:\Hugo\bin`
- Create another subfolder in Hugo: `C:\Hugo\Sites`

Technical Users

- Download the latest zipped **Hugo** executable from Hugo Releases.
- Extract all contents to your `..\Hugo\bin` folder.
- The hugo executable will be named as *hugo_hugo-version_platform_arch.exe*. Rename the executable to *hugo.exe* for ease of use.
- In PowerShell or your preferred CLI, add the *hugo.exe* executable to your *PATH* by navigating to `C:\Hugo\bin` (or the location of your *hugo.exe* file) and use the command `set PATH=%PATH%;C:\Hugo\bin`. If the hugo command does not work after a reboot, you may have to run the command prompt as administrator.

Less-technical Users

1. Go to the **Hugo** Releases page.
2. The latest release is announced on top. Scroll to the bottom of the release announcement to see the downloads. They're all ZIP files.
3. Find the Windows files near the bottom (they're in alphabetical order, so Windows is last) – download either the 32-bit or 64-bit file depending on whether you have 32-bit or 64-bit Windows. (If you don't know, see here.)
4. Move the ZIP file into your `C:\Hugo\bin` folder.
5. Double-click on the ZIP file and extract its contents. Be sure to extract the contents into the same `C:\Hugo\bin` folder – Windows will do this by default unless you tell it to extract somewhere else.
6. You should now have three new files: The hugo executable (*hugo.exe*), LICENSE, and README.md. Now you need to add **Hugo** to your Windows *PATH* settings:

For Windows 10 Users:

- Right click on the Start button.
- Click on System.
- Click on Advanced System Settings on the left.
- Click on the Environment Variables... button on the bottom.
- In the User variables section, find the row that starts with *PATH* (PATH will be all caps).
- Double-click on *PATH*.
- Click the New... button.
- Type in the folder where hugo.exe was extracted, which is `C:\Hugo\bin` if you went by the instructions above. The *PATH* entry should be the folder where **Hugo** lives and not the binary. Press Enter when you're done typing.
- Click OK at every window to exit.

For Windows 7 and 8.x users:

Windows 7 and 8.1 do not include the easy path editor included in Windows 10, so non-technical users on those platforms are advised to install a free third-party path editor like Windows Environment Variables Editor or Path Editor.

Verify the Executable

Run a few commands to verify that the executable is ready to run, and then build a sample site to get started.

1. Open a Command Prompt

At the prompt, type hugo help and press the *Enter* key. You should see output that starts with:

```
hugo is the main command, used to build your Hugo site.

Hugo is a Fast and Flexible Static Site Generator
built with love by spf13 and friends in Go.

Complete documentation is available at https://gohugo.io/.
```

If you do, then the installation is complete. If you don't, double-check the path that you placed the hugo.exe file in and that you typed that path correctly when you added it to your *PATH* variable. If you're still not getting the output, search the Hugo discussion forum to see if others have already figured out our problem. If not, add a note—in the "Support" category—and be sure to include your command and the output.

At the prompt, change your directory to the `Sites` directory.

```
C:\Program Files> cd C:\Hugo\Sites
C:\Hugo\Sites>
```

2. Run the Command

Run the command to generate a new site. I'm using example.com as the name of the site.

```
C:\Hugo\Sites> hugo new site example.com
```

You should now have a directory at `C:\Hugo\Sites\example.com`. Change into that directory and list the contents. You should get output similar to the following:

```
C:\Hugo\Sites> cd example.com
C:\Hugo\Sites\example.com> dir
Directory of C:\hugo\sites\example.com

04/13/2015  10:44 PM    <DIR>          .
04/13/2015  10:44 PM    <DIR>          ..
04/13/2015  10:44 PM    <DIR>          archetypes
04/13/2015  10:44 PM                83 config.toml
04/13/2015  10:44 PM    <DIR>          content
04/13/2015  10:44 PM    <DIR>          data
04/13/2015  10:44 PM    <DIR>          layouts
04/13/2015  10:44 PM    <DIR>          static
               1 File(s)             83 bytes
               7 Dir(s)   6,273,331,200 bytes free
```