Installing Git

```
$ echo "Data Sciences Institute"
```

Installing Git

Typically, Git is already installed on our system but we can check for that using the `git` command:

```
$ git --version
```

Does anyone not see a version?

Installing on Linux

If you're on Ubantu:

```
$ sudo apt install git
```

If you're on Fedora, RHEL or CentOS:

```
$ sudo dnf install git
```

```
$ sudo yum install git
```

Installing on Mac

You can install Git via Homebrew, if you have Homebrew installed (https://brew.sh/).

```
$ brew install git
```

Finally, you can install Git from source at this link:

https://sourceforge.net/projects/git-osx-installer/

Installing on Windows

The download will start automatically through this link: https://git-scm.com/download/win

Questions?

`Git Setup`

The first thing to do now that we have Git installed on our system is to customize it. These changes will remain despite any upgrades to Git that we install.

Using the command `git config`, we can set configuration variables that control all aspects of how Git looks and operates.

Checking Configurations

Before we change any of our global configurations, we can check what they are:

```
$ git config --list
```

If we haven't configured Git, we can do that now!

Identity

First, we'll set our username and email address. Git uses this information every time we commit.

```
$ git config --global user.name "Jane Doe"
$ git config --global user.email "jane.doe@gmail.com"
```

The option `--global` means that we only have to pass this through once.

11

Editor

Next, we'll configure our the default text editor for when Git needs to type in a message. Git uses our system's default editor (usually Vi or Vim) but we can change it if we prefer. If we want to change the editor to emacs, we would do so below:

```
$ git config --global core.editor emacs
```

Diff Tool

We can also set the default diff tool which is used to resolve merge conflicts:

```
$ git config --global merge.tool vimdiff
```

Checking the Setting

We can use the `git config --list` command to see all Git settings. See the values of a specific specific setting:

```
$ git config user.name
```

Help

If we ever need help, even offline, we can access the manual page three ways:

1. `$ git help <verb>`

2. `$ git <verb> --help`

3. `$ man git-<verb>`

For example, we can get help for the `config` command:

```
$ git help config
```

Questions?