# Getting Started: Using Git and GitHub for contributing to the W3C Note SVG 2.0 Accessibility (using Git and GitHub)

## Intro

Doug Schepers (chair) came up with the idea for creating the W3C Accessible SVG Community Group for improving of SVG and updating the W3C Note that Charles McCathie Nevile and Marja-Riitta Koivunen authored in 2000. Charles set up our GitHub presence (SVG-access-W3CG) including the repository, and eventually will have the support files to get us started.

Doug: SVG, as a text format for graphics, already has a lot of potential for accessibility. It’s not just that SVG can be made accessible, it’s that SVG can be much more accessible than other graphics formats, with very little extra effort on the part of developers and designers.

[Chaals McCathie-Nevile](https://twitter.com/Chaals) (Yandex), co-chair for W3C Accessible SVG Community Group, wrote the “[Accessibility Features of SVG](http://www.w3.org/TR/SVG-access/)” specification many years ago with some great tips for basic accessibility and some mapping between SVG 1.0 and WCAG 1.0. That document defines authoring and content requirements, rather than requirements on *user agents* (browsers and screen readers).

To specify requirements on user agents, we will look to SVG2 and related modules. [Rich Schwerdtfeger](http://www-03.ibm.com/able/news/rich_schwerdtfeger.html) (IBM) has been working hard in the SVG Working Group on porting existing ARIA attributes and some HTML5 accessibility features into SVG 2.  We also have plans for adding even more accessibility requirements directly into SVG.

The idea is (for each dev participant) to create a test file that will test a \*single\* concept (say, the aria-hidden attribute). Some will create test files, while others will complete a form and run the test files in the configurations (browser/AT/OS) they have on their devices (laptop, mobile), and record the results. To help in this effort you can be a developer, a tester/user, or both.

W3C Accessible SVG Community Group: <http://www.w3.org/community/svga11y/>

Mailing List: [public-svga11y@w3.org](mailto:public-svga11y@w3.org)



## Git and GitHub (for Dummies)

### Developers

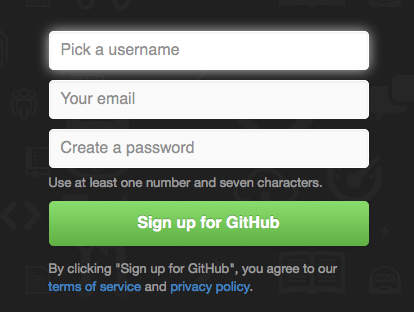
To create files and add them to the repository, you need two things:

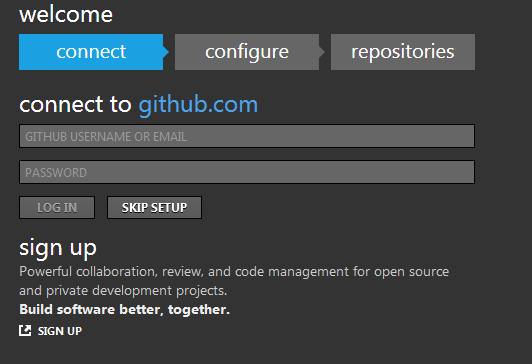
1. **Git** (local: versioning application you download to your computer)
2. **GitHub** (remote: is an on-line workspace that works with Git)
   1. You can store and access your work online, on GitHub, but only if you EXPLICITLY SEND IT THERE - via a Git ***push***
   2. What you do locally on your computer (in Git) doesn’t appear remotely online (in GitHub); until you create the pathways and make the explicit commands

### Steps for those without a GitHub Account

Please follow these steps in this order:

1. Quickly read about these terms (Blame, Branch, Clone, Commit, Fork, Merge, Pull Request, Push, Repository) in the [GitHub Glossary](https://help.github.com/articles/github-glossary)
2. GitHub: Go to <https://github.com/> and sign-up for a new GitHub account (It is Free)



1. Git: Download the app for [Windows](https://windows.github.com/), or for [Mac](https://mac.github.com/), or [Linux and/or Unix](http://git-scm.com/download/linux) (each Free)
2. After install, open the downloaded app
3. Log-in to the local app, the login should look something like this:  
   
4. Enter your GitHub login details (you created in Step #2) using the **email address** and password.  
   (If you use your username instead of the email address, things could get screwy)
5. Once you are back at the main screen you should see the repositories (Repos) that are available for you to Clone. On the left, you'll see your GitHub account, as well as any organizations you're a part of. Clicking on a name will show you which repositories are available.
6. Click the **Clone** button. (Clicking on **Clone** brings the repository to your computer.)
7. Now open up your Documents folder and you should see a new folder called GitHub, in there will be your cloned Repo. You now have a local copy of your GitHub repository.
8. Read our [Contributing](https://github.com/Open-A11y-Testing/Test-Triage/blob/master/contributing.md) document (to learn how to create and structure the test files)
9. Fork the Repo ([Open-A11y-Testing /Test-Triage](https://github.com/Open-A11y-Testing/Test-Triage/blob/master/contributing.md)), by clicking the "Fork" button in the GitHub.com repository.  
   Represents the Pull Request, Watch, Star and Fork buttons
10. Get the [Git Cheat Sheet](https://github.com/github/training-materials/blob/master/downloads/github-git-cheat-sheet.pdf?raw=true) (if using command line)
11. Validate your HTML, CSS and JavaScript

??

### Developer Steps for those with a GitHub Account

Please follow these steps in this order:

1. ??
2. ??
3. ??

## SVG-access-W3CG URLs

Contributing Doc:

Repo (Repository): <https://github.com/SVG-access-W3CG/new-note-draft>

## Resources

YouTube Video: GitHub for Beginners: <https://www.youtube.com/watch?v=LXoWxrTdXkM>  
GitHub Glossary: <https://help.github.com/articles/github-glossary>