The world's fastest framework for building websites

(Or at least that's what they say)

GSPS 8/11/2018 - Dylan Cromer

ALL OF MY RESOURCES IN ONE PLACE

▸ Dylan's GitHub repo for this talk

▸ Github pages docs

▸ Hugo tutorial

▸ Hugo tutorial for Github pages integration

▸ Github pages domain name info

WHAT YOU WILL GET OUT OF THIS

▸ A working webpage, like jakevdp.github.io

▸ If you want, one with a custom domain,
  like dylancromer.com

▸ Your website will be integrated into a Git repository

  ▸ This means it is version controlled (which is good!)

▸ In general you will come closer to Nirvana

MAYBE YOUR FIRST DECISION

▸ Let's look at all the themes:

   ▸ Go to themes.gohugo.io with me

DEFINITELY THE FIRST DECISION

▸ Do you want

    ▸ A GitHub site that looks like yourbeautifulname.github.io

    ▸ or one that looks like yourbeautifulproject.github.io

    ▸ You can have only one of the first, but many like the second

▸ If you're using a custom domain it really won't matter

FIRST STEP

▸ Whichever choice, make a GitHub repository for your site. **Make sure you create a readme for this first one**

    ▸ yourprojectname, or

    ▸ yourusername.github.io

▸ Now make another, either

    ▸ yourprojectname-dev, or

    ▸ yourusername.github.io-dev

OK, LET'S GO TO MY GITHUB REPOSITORY NOW

github.com/dylancromer/hugolaunch

NOTES ON MY SHELL SCRIPT

▸ They aren't amazingly good (although one is Hugo's, not mine…)

▸ They will get the job done

▸ You should probably learn what the commands mean if you don't already…
                    … Eventually

OK WE RAN THE SCRIPT, WHAT NOW

▸ What we have now is

   ▸ A GitHub repository to serve the site

   ▸ A repository to manage your edits before deploying them

   ▸ (Maybe) a theme

▸ Now, can edit some files and then run deploy script

▸ If you installed a theme try copying the sample site files into the main directory

HOW DOES HUGO WORK?

▸ Basically you edit markdown files (very basic syntax, much easier than say LaTeX)

▸ Can create new pages, add files, etc. with hugo command

▸ Hugo generates a site folder which GitHub then serves to the public

▸ The deploy.sh script automates transferring your changes onto your real site

FINAL STEPS TO GO ONLINE

▸ Go to the settings for your *public* Github site repository (not the -dev one)

▸ Enable GitHub Pages setting

▸ Try going to the url! e.g. yourname.github.io

WANT A CUSTOM DOMAIN?

▸ Buy a domain name, like ChristopherRooneyIsDaBest.com (or go to the registrar for the one you already have)

▸ If it's NameCheap.com, they have a nice tutorial

▸ Another for Google Domains (skip to step 6)

▸ Basically the same process everywhere

   ▸ Make A records pointing to IPs

   ▸ Make CNAME record pointing to yoursite.github.io

LAST STEP FOR CUSTOM DOMAIN

▸ Make a CNAME file in your public repository's static/ folder

▸ it needs to contain yourdomain.yourTLD and ***nothing else***

▸ Hard learned lesson:

  ▸ Make sure you go into your -dev repo's `config.toml` file

  ▸ change `baseurl` to be "`//yourname.yourTLD/`"

  ▸ the two slashes are ***VERY*** important

  ▸ **if you don't do this you will *not* be a happy camper**