WHAT THE TECH?

Lizzie Ellis
@lizmeister321
May 8th, 2018
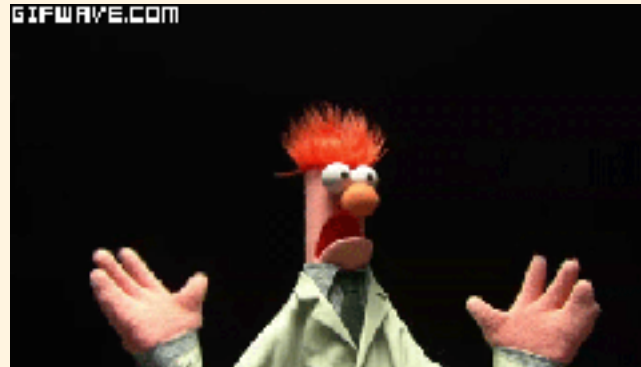
There's so much jargon and totally new concepts in programming that it's easy to freak out when people start talking about it. Our goal to day is to give a little more info behind these ideas so that you feel comfortable learning more about these topics!

WHY ME?

Exhibit A: A Tech Professional

@lizmeister321

Quick bio:

-Background in poli sci, MPP

-took stats in grad school, wanted to learn more, picked up SQL at internship

-Worked for in politics for a while now, including stints at DNC and HFA 2016

- now working at AFSCME using a variety of tools — AWS/cloud, python, databases, data viz

- Also co-lead of Code for DC, a volunteer civic tech group

-No formal "tech training" — no CS degree, no academic work, foundations all came from informal education (friends or on the job)

PART 1:
YOU, THE PROGRAMMER

The true basics. Need to start at the beginning to make sure our foundation is really solid to move forward

COMPUTERS
ARE DUMB

Really!

@lizmeister321

Emphasize that computers are just big calculators that can do math calculations very quickly, and can do very lots of large complex math much faster than humans can.

SO WHY DO THEY SEEM SO SMART?

lots and <u>lots</u> of programmers

over

lots and <u>lots</u> of time

So let's think this through logically then—what's a programmer?

A PERSON WHO WRITES INSTRUCTIONS FOR THE COMPUTER TO FOLLOW

LET'S MAKE A PROGRAM!

@lizmeister321

Walk through pb&j exercise with the group — take your time

At the end, wrap with: Computers don't understand English like you and I do, so we need to find another way to tell the computer exactly what we want it to do. That's where programming languages come in

HTML & CSS
PYTHON
JAVASCRIPT
SQL
RUBY
JAVA
PHP

Some popular programming languages

@lizmeister321

There are a lot of different programming languages, these are just a few that are very popular right now. Have you heard of any of these? All of them have different syntax, and can do slightly different things, but all of them fall under the umbrella of "programming languages"

HTML & CSS are used to control what a web page looks like — color, fonts, shapes, layout, etc
     Python and Ruby are way of having your computer pull information from a data store and present it on your website

     IF PEOPLE SEEM TO BE GETTING IT — FEEL

PAGE O' JARGON

API — *A tool for two programs or computers to easily share data*

THE CLOUD — *Data storage and/or server that can be accessed remotely*

GUI — *A visual way to interact with your computer (like your Desktop)*

HACKATHON — *An event where the goal is to write code (usually with a theme)*

REFACTOR(ING) — *Editing code to run faster or be more human-friendly*

OPEN SOURCE — *Software that anyone can use or modify for free*

@lizmeister321

API: application program interface. You will literally never need to know this again

The Cloud: A computer you access through the internet. Define AWS and why it's so revolutionary here.

GUI: graphic user interface. Pronounced "gooey".

PART 2:
BUILDING YOUR TOOL KIT

Now that we're all programmers with some preliminary knowledge of the tech world, lets get everything together in one place so we can easily use it in the future (just like batgirl has her snazzy tool belt here)

SO I NEED TO KNOW…ALL OF IT?

NO!

What you need to know depends on *what you want to do*

SOOOO…
HOW DO I PICK A LANGUAGE?

Well that didn't answer my question, so how DO you pick?

Remember this slide from a few minutes ago?
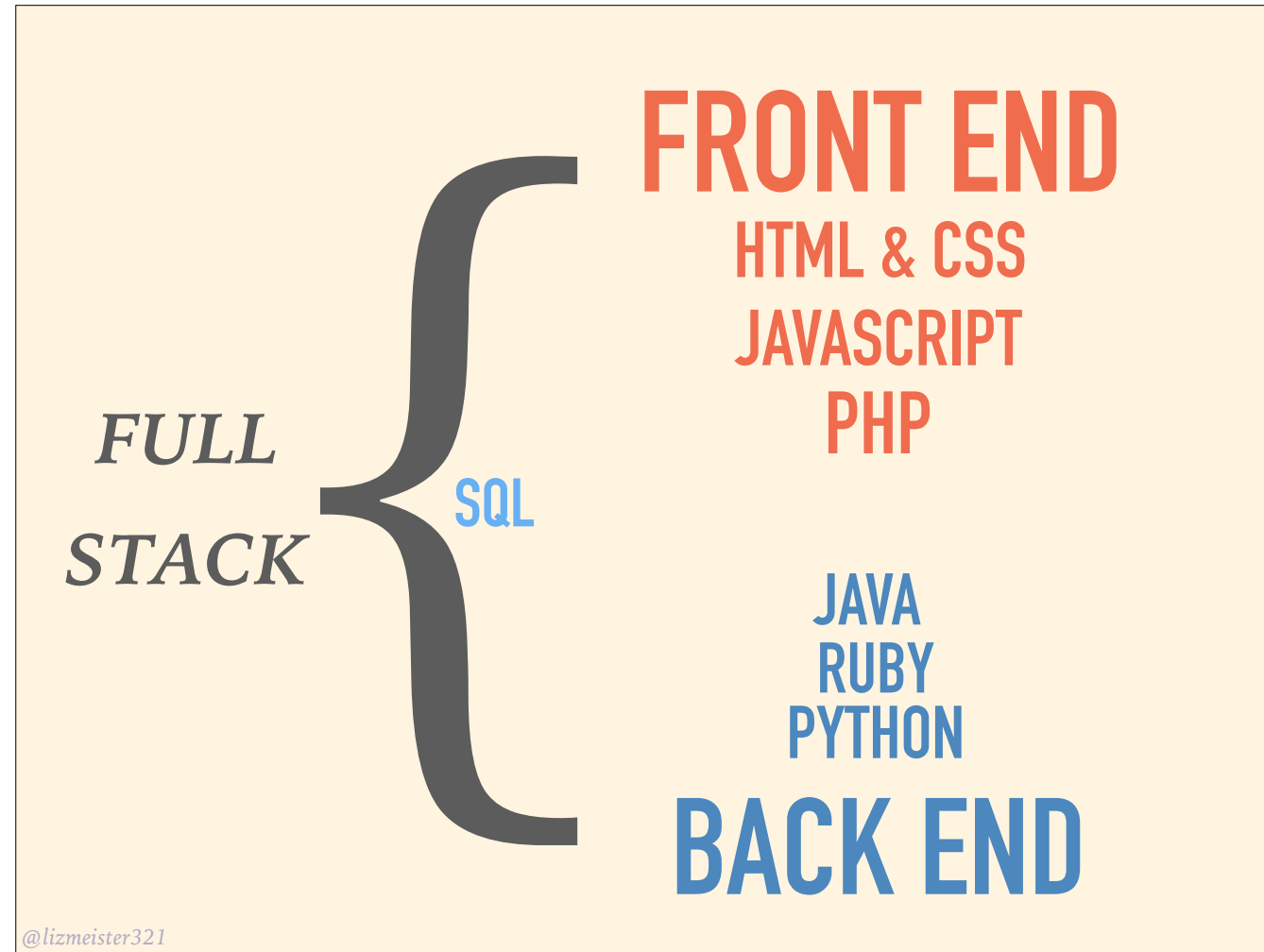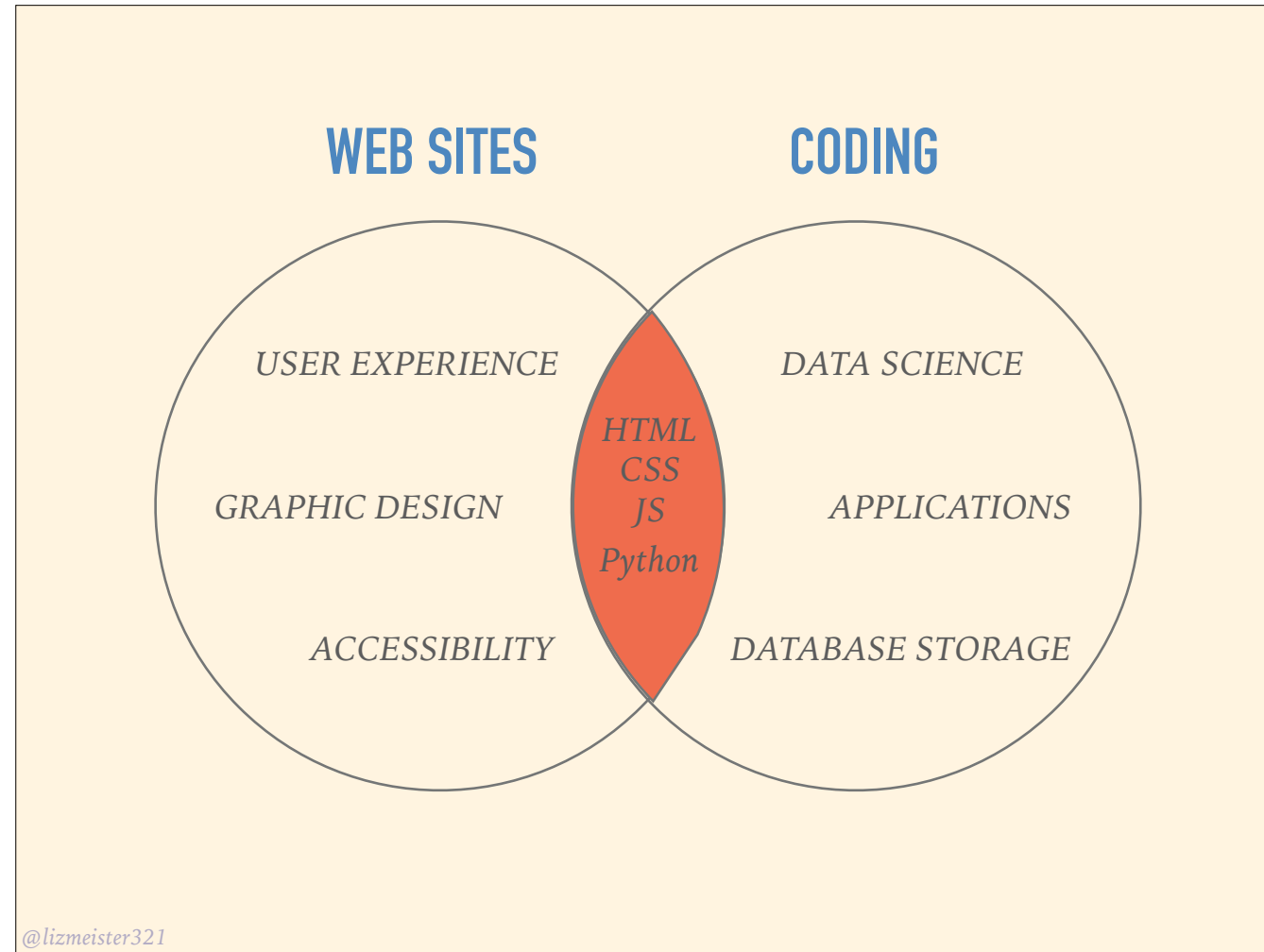
This slide is a bit misleading but let's work through why:

Talk through front vs back end. Full stack means it covers all elements
Many of these languages can be used for full-stack development — whole series of JS variants, Python (Django), Ruby on rails
SQL down in the middle is a bit weird, we'll talk about that in a second

Important to note that programming isn't JUST websites.

Topics to cover here:
—UX/UI
—machine learning/algorithms — R, Python most popular for data science pieces
—Databases (excel comparison) — remember sql from the previous slide? This is one way of working with databases (a back end tool, but not a "back end" language per se). Also other ways to access databases (NoSQL)

GIT · GITHUB · Software that allows for VERSION CONTROL · A website that uses **git** to make software OPEN SOURCE · @lizmeister321

LOT OF CONCEPTS HERE LEAVE TIME FOR THIS,.
Another tool, related to all these other languages, is git and github.
Git is software accessed through the command line (SHOW COMMAND LINE), which is a text-based way of interacting with your computer. It allows for version control, which is basically tracking changes and saving different snapshots of a piece of code or series of files. Version control helps us prevent those "memo1_final_final_Done_final2.pdf" situations

GITHUB is a website that uses this git software and allows people to share their software (or presentations or whatever) with the entire internet. This is a main hub of OPEN SOURCE code, which is basically just code that anyone can view, edit, and submit for changes.

DEMO GITHUB ON YOUR PAGE (octocat is linked)

—key phrases: version control, open source, git vs GitHub

"SPEAK PROGRAMMER"

➤ *Translate jargon*

➤ *Draw diagrams*

➤ *Be specific*

➤ *Google is your best friend*

➤ *Think systems, not specifics*

There's not really a secret "programmer" way of speaking, it's just a stereotype. It may feel overwhelming or confusing to try and manage a group of folks like this but there's a few easy tools to handle it:

—Jargon is everywhere in the industry, and it's designed to keep people out. Always ask for repeat-backs, restate terms in non-jargon language, and avoid using jargon (whether or not you actually know it) in your communications.

—Engineers love whiteboards in part because they allow you to visualize complex systems. Don't understand? Draw it while you're talking so everyone in the meeting can be on the same page for how a thing works. Then, take a picture and share it around!

—Repeat-backs can save you hours. Your definition of "stand up this tool" may be very different from the team's. Clarify and repeat until you're positive!

—GOOGLE IT. Everyone does. Stack Overflow (what is it, why would you use it, why it might be bad but how to deal with it)

—It doesn't actually matter what language the code is written in, as long as the project does what it needs. Be clear on how the different pieces work together, not what the code does or doesn't say. Get clear on the steps each piece of software does.

PART 3:
WHERE DO I SIGN UP?
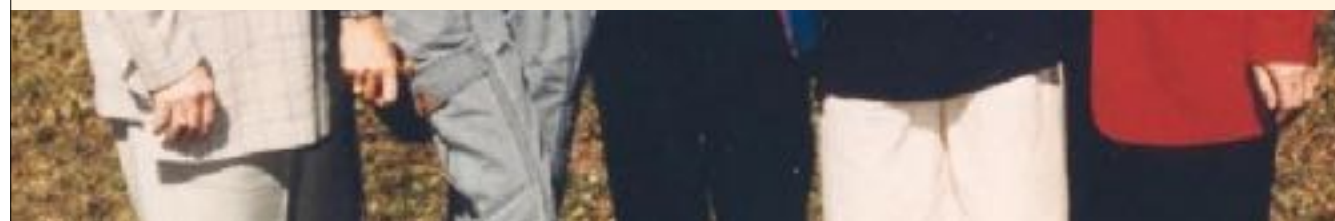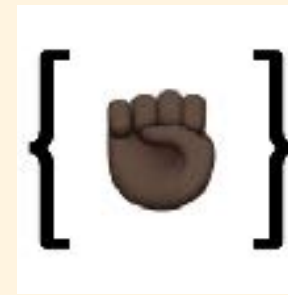
FREE. LOCAL. FUN!

@lizmeister321

GREAT NEWS: so many excellent resources right here in DC!
Other resources: codecademy.com, upskill.com, general assembly, DCFemTech, DC Web Women.

Start with the local/free/flexible if you're really just getting into it. If you've tried a few things and want to learn more, a 1-day class might be worth your money.

Women who code: meet ups at least weekly
DCPL: access to lynda.com, arlington pl has too
DCFemTech Hack for Good: July 2018
TECH LADY HACKATHON HERE (techladyhackathon.org)

A WORD ABOUT BOOTCAMPS

(really, just an emoji)

Bootcamps are a worthy investment if you know you're interested in pursuing this full time. nOT a good first stop. Very expensive, very time consuming if you're just starting to dig in. Many are geared towards career-changers and not all fully set you up for a new job (despite what they'll tell you).

LENNY

The key question any company wants answered is: How can you help me get this job done well? You have to take a step back and say, "What is it that I'm bringing to the table?" Now, you don't want to be in print-media sales, but every single organization has to sell. Selling is an art and a science — it's not easy. You could go in and say, "Hey, I want to get into this new industry. Let me bring my sales understanding to the table to help your business thrive." You could start on a project basis and make a little bit of money, show your value, see if you like them.

It's so important to tell your story across your network. I did this myself. I was 30 when I made the transition from banker into a work-life-flexibility consultant. I looked at my background and I tried to see how I could add value to the two organizations in the whole country that were doing this work at that time. I was able to show one of these entities that I could help them with a client they'd just taken on, which was a bank. So they took a chance on me; they let me have an internship. If you're willing to say, "Look, I'll start from the beginning. I will take a pay cut," that gives them wiggle room to say, "All right, we'll take a chance on you."

@lizmeister321

Taken from Lenny Letter in October — writer switching careers after 19 years

2 important take aways: 1. Be willing to be flexible and open — you're going to be new in this and may need to be willing to do some part-time work to start. Side projects and a portfolio are a great way to prep for this in your free time

2. WORK THE NETWORK. Especially useful in tech where women tend to try and recommend each other. Talk to lots of others, get their stories. This is not news in DC but it's especially helpful in DC where there is such a strong tech lady network.