

# Unloop Program Overview

**Mission:** Enable people who have been in prison to succeed in careers in tech

What Unloop does:

- Education programs
- Mentorship
- Employer Network

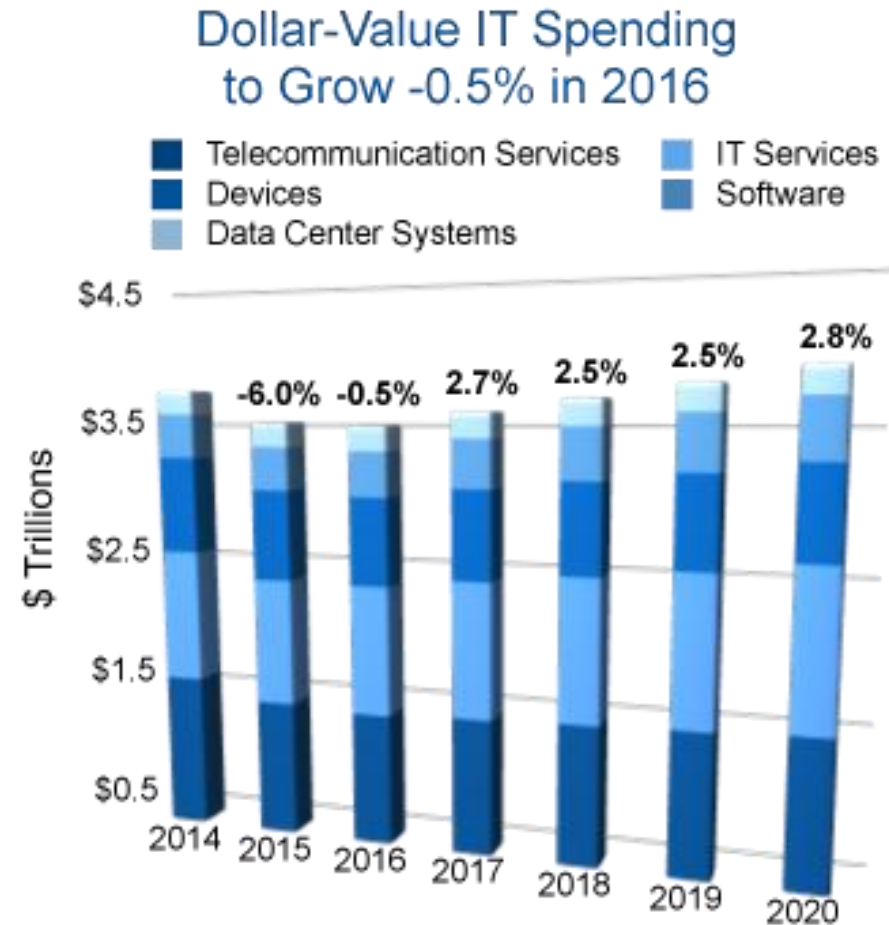
Program at Monroe Correctional Complex

- 1 year, 40 credit certificate program in web development
- Teaches skills you would need to be entry level web developer
- Workshops & events with industry professionals
- Selection criteria = 75% formula & 25% discretionary:
  - Grade in Intro to Tech Careers (50%)
  - Written application (30%)
  - Computer basics & logic assessment (20%)

# What is Technology? What is the “Tech Industry”?

**Technology:** the use of science in industry, engineering, etc., to invent useful things or to solve problems

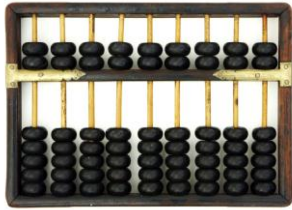
**Tech Industry:** hardware, software, services



Source: Gartner Market Data Book, 1Q16 Update

# What is a computer?

**Computer:** An electronic device that stores and manipulates information – a computer is a counting device!



# Hardware vs. Software

Hardware: the physical parts of a computer system



Software: the set of instructions & stored data that make computers useful



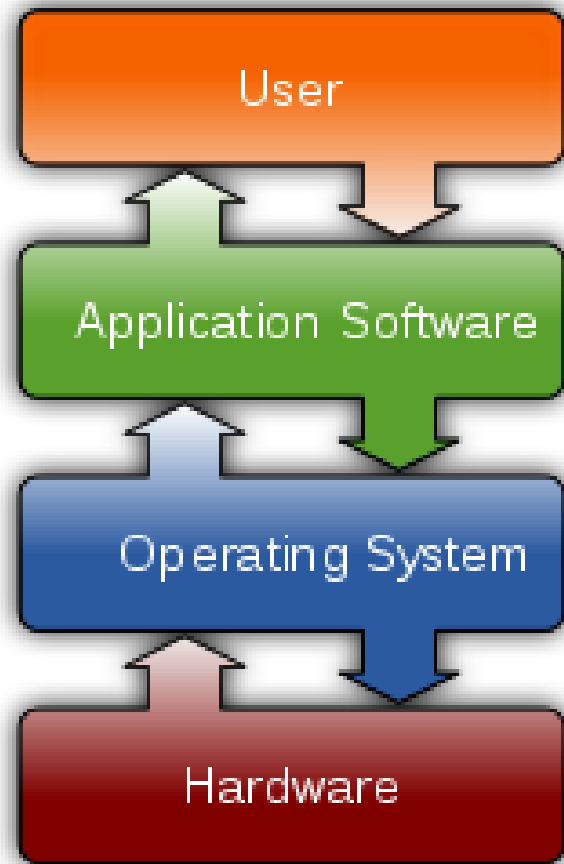
Focus of this class

# How does software work?

When software “runs” on computer hardware, it’s using electric pulses (a lot of them!) to represent information on a computer’s circuits

- **Binary:** Electric pulses (often represented by 1’s and 0’s) that make up the information that is stored & processed by a computer
  - Bits, bytes, kilobytes, megabytes, gigabytes, etc.
  - Quick translation: “b” in English = 01100010 in binary!
  - When someone uses it for programming = “machine code”
- **Programming language:** an almost-English language that helps someone tell a computer what to do
  - Simpler than trying to write in binary (they used to do stuff like that!)
  - Examples: HTML, JavaScript, Python, Ruby, etc.
- **Program:** a set of instructions for a computer, written in a programming language

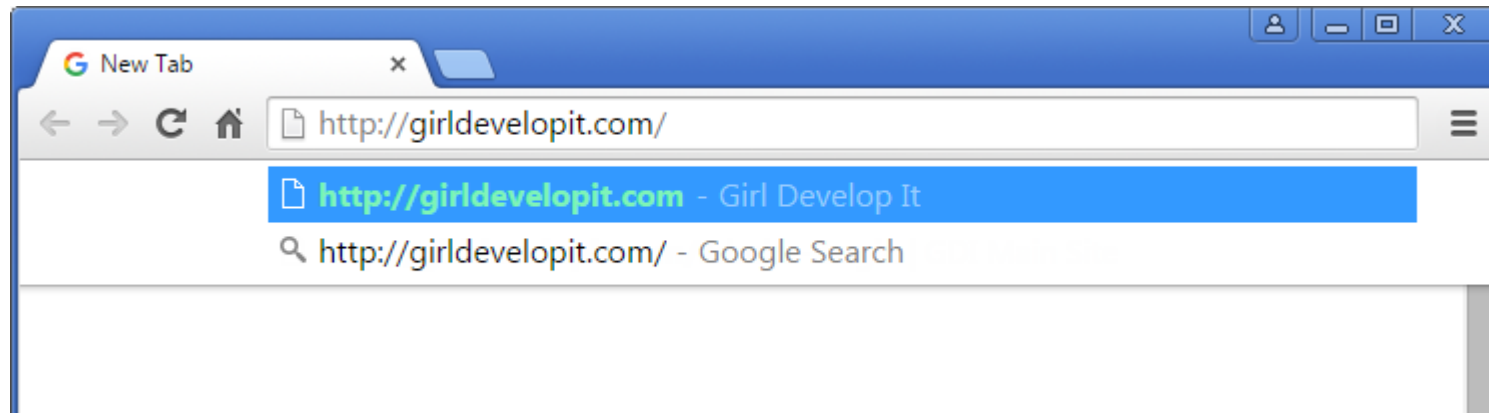
# Important types of software



- **Operating system:** performs basic tasks, such as recognizing input from the keyboard, sending output to the display screen, and launching applications
  - Examples?
- **Application:** a program that runs on an operating system
  - Examples?
- **Browser:** A specific type of application that processes and displays HTML files
  - Examples?

# What happens when I visit a website?

**Step 1:** Type an address into the search bar in your browser



# What happens when I visit a website?

**Step 2:** Your computer performs a DNS lookup

- Each website has a unique code, called an IP address, that is like a phone number
- DNS (or domain name servers), are like phone books. The DNS servers will connect you to the right server  
girldevelopit.com → 205.178.189.129

**Step 3:** Your computer connects to the server

- Your computer uses that IP address to find the correct web server and sends a request for a page



# What happens when I visit a website?

**Step 4:** The server responds to your request

- The server processes your request and finds the right files. Sometimes it has to "compile" a page from code and other sources, like a database.

**Step 5:** The server sends your computer the files

- This is usually an HTML page, with some CSS, JavaScript, or media files.

**Step 6:** Your browser displays the files

- Your browser interprets the code and assembles all the files into a page you can see and use. Each browser does this slightly differently.

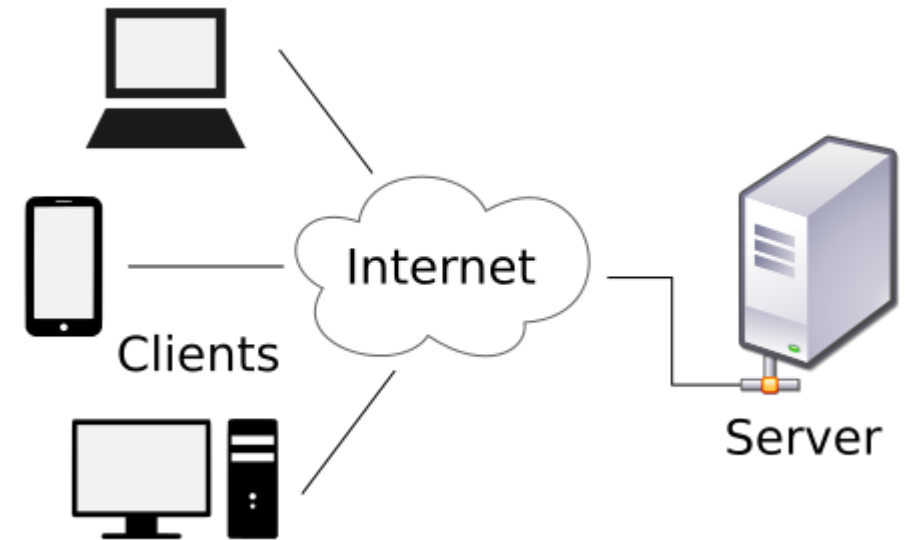
# Important internet concepts

**Client:** Accesses services (e.g. an HTML file) made available by a server

**Server:** provides services for other programs or devices (client)

**HTTP:** hyper-text transfer protocol – the rules for how to send data over the internet

**IP address:** the unique number of a computer connected to the internet



# Basic Programming Concepts

Refresh: What is a computer program?

Blockly programs will teach:

- *If/Else Statements*
- *For Loops*
- *While Loops*
- *Variables (we will cover this next session)*

Why Blockly:

- We don't have to worry about Syntax!
  - Syntax: Spelling and correct formatting of code

# If Statement

Test a Condition, and execute a statement(s) based on that condition.

*If an obstacle is on the right, turn left.*

## **If/Else Statement:**

An If Statement, with a “Else Clause” attached. The “Else Clause” allows the program to take an action when the condition is NOT true

*If an obstacle is on the right, turn left. Otherwise, go forward.*

## **Nested If Statement:**

Checks several conditions, one after another, and executes statement (s) based on those conditions

*If an obstacle is on the right, turn left, else if obstacle is on the left, flash red lights. Otherwise go forward.*

# For Loop

Repeat a section of code a number of times

**Example:**

*Walk forward 5 steps, turn left. Repeat 4 times.*

# While Loop

Similar to a For Loop, a While Loop lets a section of code be repeated.

A While Loop repeats the code, while a condition exists

Example:

*While obstacles are in front of the robot, flash red lights.*

*While location does not equal destination, walk forward.*

Discussion: what would happen if we had a while loop with a condition that was always true?

# Programming is a different type of Edu

- Embrace “failure” – it’s part of the process
- Learn to be at home with not knowing, but stay curious!
- Collaboration is encouraged – don’t reinvent the wheel, work together
- There is rarely one solution to a problem – be creative
- Experiment with different types of problem solving

# TSA (think, search, ask) approach to problem solving

## 1. Think:

- What is the problem I am having?
- What do I know about the problem?
- What do I not know about the problem?
- What could I try on my own to solve the problem?
- If I can't solve it after some effort, how can I phrase it as a question? (be as specific as possible)

## 2. Search

- What resources do I have where I could get information about my question?
- Do any of the resources I can access have information that can answer my question?
- If not, do any resources have information that allow me to make my question more specific?

## 3. Ask

- Do any of the people around me know the answer to my question?
- If not, have any of the people around me run into the same problem?
- Do any of the people around me have information that allow me to make my question more specific?
- Last step: ask a TA or your teacher!

### Why?

1. This is how people will expect you to solve problems in tech
2. Fewer bottlenecks when you are asking for help
3. People will be happier to help you
4. Clearly stating problems is an invaluable skill

### Be mindful of:

1. Time: don't just bang your head against the wall for forever, if you're stuck, it's ok to ask!
2. Collaboration: be eager to help and be helped! If someone helps you along the way, share what you learn



# The Internet of Things (IoT) discussion

What is the internet of things?

Why is the IoT important?

What is an industry that the IoT might impact? How?

Who will profit from the IoT?

What could be some negative impacts or risks of the IoT?

# Recap of Last Week's Topics

What is computer programming?

What is the difference between hardware & software?

What is a programming language?

What happens behind the scenes when you visit a website?

What is an IP address?

What is the difference between client and server?

# What is a software developer?

**Software developer:** Someone that uses one or more programming languages to build software

Also often called a “dev”, programmer or software engineer

Some different types of developers:

- Web developer (front-end, back-end, full stack)
- Applications developer (can include web developers)
- Systems software developer

# What is a designer (in software)?

**Designer:** Someone that envisions and creates the look and feel of software

Design is VERY broad! Includes visual & interactive elements

Some different types of designers:

- User Experience (UX)
- User interface (UI)
- Web designer

# What is a product/program manager (PM)?

**Product manager:** responsible for orchestrating the various activities associated with ensuring that a product is delivered that meets users' needs.

Helps design, development, and marketing work together to deliver a product on time & on spec

Sometimes considered the “CEO” of a product – does whatever needs to be done!

# Example Software Developer JD - Expedia

## **Responsibilities:**

- You will design and implement software applications for our lodging systems and detect/alert anomalies.
- You will collaborate with product managers and other engineering teams to prototype creative solutions quickly and implement the technical vision.
- Engineer innovative solution with modern technologies to improve platform performance
- Drive continuous improvement in our software development process within an agile development team.
- You will write automated unit, system and acceptance tests as to support our continuous integration pipelines.

## **Qualifications:**

- 1+ years overall development experience with demonstrated growth in producing web
- Able to present technical information in a clear and concise manner.
- Strong technical proficiency in programming languages, JavaScript/HTML/CSS
- Clean technical designs are part of your DNA and you have the ability to instill the same drive in others.
- Critical thinking: there are always options; you will be adept at identifying them, evaluating options and concisely communicate them to your peers.
- Experience in agile/Scrum software development practices and a track record of learning from your actions to make positive adjustments.
- BS Degree in Computer Science or equivalent work experience.

# Example Software Developer JD - Nordstrom

## Key Responsibilities:

- Provision, maintain, and iterate on a common data pipeline
- Gather, store, and process raw data at scale
- Collaborate with software developers and data scientists to serve their data monitoring and analytic needs
- Make decisions based on metrics and measurable data
- Adjust positively to quickly changing priorities and shifting goals
- Automate yourself out of a job

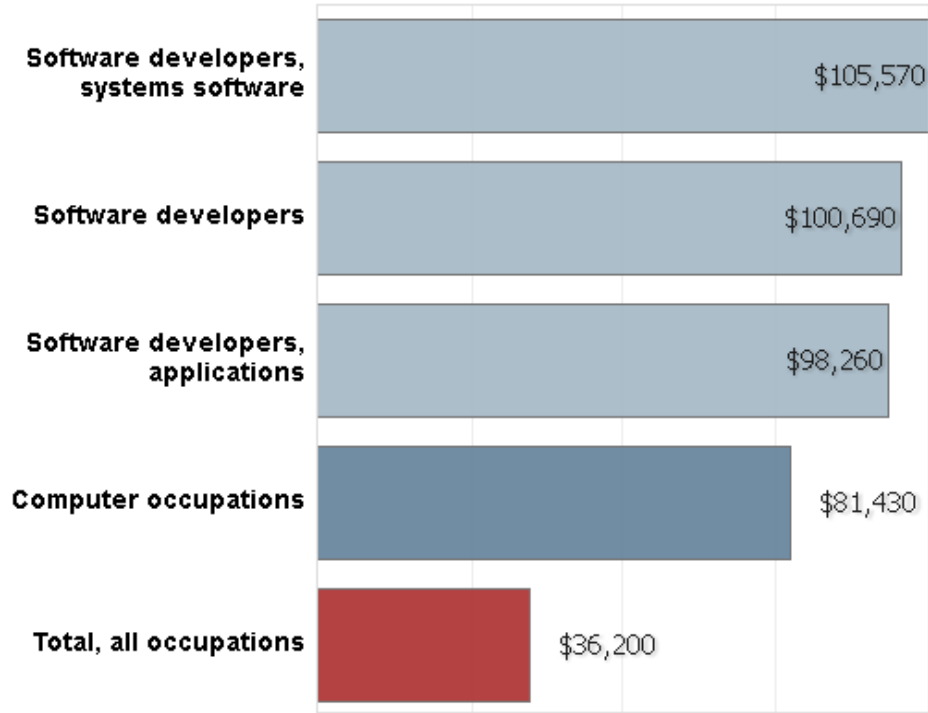
## Qualifications:

- Degree in Computer Science or equivalent experience
- Agile software development experience
- Excellent problem solving, debugging and troubleshooting skills
- Familiarity with Linux, cloud computing (i.e., AWS), and container technology (i.e., Docker, Kubernetes)
- Fluency with at least one of Java, Go, Python, NodeJS, or other modern language stack
- Experience with tools that promote Infrastructure-as-code (i.e., Cloudformation, Terraform, Chef, Ansible)
- Interest in analytics and streaming platforms (i.e., Elasticsearch, Kafka, Spark, Hadoop, etc) is a big plus

# Software developer job growth & pay

## Software Developers

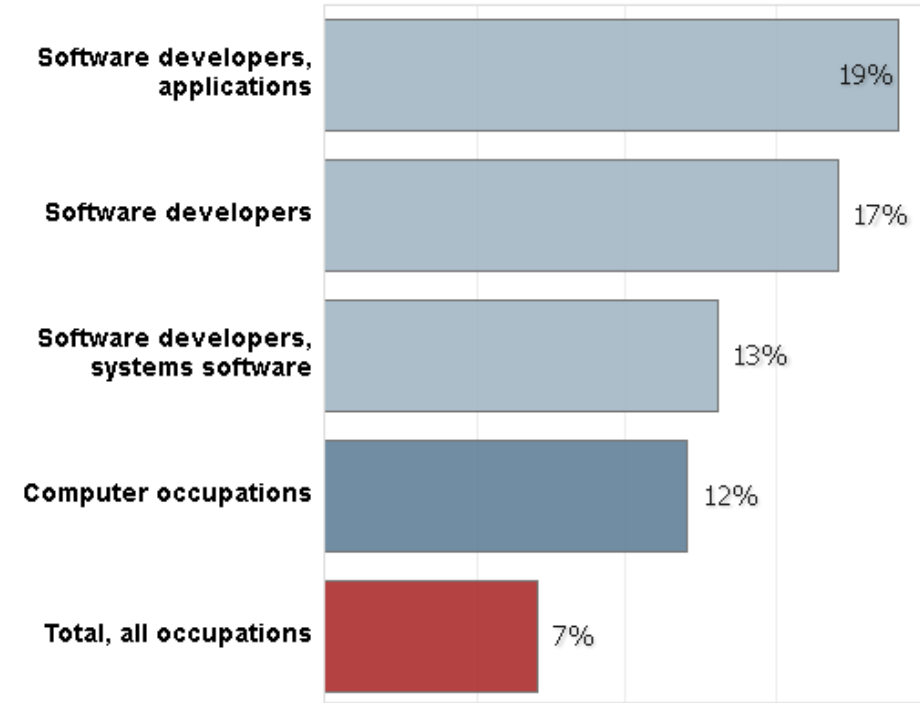
Median annual wages, May 2015



Note: All Occupations includes all occupations in the U.S. Economy.  
Source: U.S. Bureau of Labor Statistics, Occupational Employment Statistics

## Software Developers

Percent change in employment, projected 2014-24



Note: All Occupations includes all occupations in the U.S. Economy.  
Source: U.S. Bureau of Labor Statistics, Employment Projections program

Note: Wage data is MEDIAN – including experienced developers! Expect to make much less as entry level



# Two main working arrangements

## Working for a company

- Pros:
  - Focus on coding, not the other stuff
  - Structure, support, and mentorship are easier to find
  - Stable income: you know how much money you will make, usually great benefits included
  - Built in networking
  - Social proof of your ability
- Cons:
  - Potential for conflict when learning new culture, norms, expectations
  - Commuting/the grind
  - (Potentially) less upside in earning potential
  - More restrictions on people with records

## Working for yourself

- Pros:
  - Can be really rewarding to build your own thing (even if it fails)
  - Potentially higher payoff (professionally and financially)
  - You will learn how to do a lot more than coding
  - Freedom in building your own schedule, network, commute, culture, etc.
- Cons:
  - Income instability
  - Emotionally draining
  - Requires skill in a lot of areas outside of coding
  - Can be isolating
  - For startups – very high likelihood of failure

# I want to work for a company... but which one?

Big Company	<ul style="list-style-type: none"><li>- You know a lot of these:<ul style="list-style-type: none"><li>- Google</li><li>- Facebook</li><li>- Microsoft (mostly)</li></ul></li><li>- Maybe less well known:<ul style="list-style-type: none"><li>- Accenture</li><li>- Wipro</li><li>- Deloitte</li></ul></li></ul>	
Small Company	<ul style="list-style-type: none"><li>- Startups mostly begin here! Some Seattle based ones:<ul style="list-style-type: none"><li>- EnergySavvy</li><li>- Socrata</li><li>- Textio</li></ul></li><li>- Freelancers, agencies, etc. Some are in Unloop's network<ul style="list-style-type: none"><li>- Substantial</li><li>- FuseIQ</li><li>- Loxley Digital</li></ul></li></ul>	
	Product	Service

# I want to work for myself.. but how?

## 1. Freelance

- a) Cross between working for others & working for yourself
- b) Find clients, keep them happy, get paid
- c) Requires a depth and breadth of skill not required in a company
- d) Upshot: can be crafted to suit your lifestyle, solo or through an agency

## 2. Entrepreneurship

- a) The riskiest option – should you quit your day job?
- b) But arguably the biggest upside financially, professionally, and personally
- c) Really, really hard – the average entrepreneur fails 7 times before succeeding
- d) A mythological force in the tech industry – a lot of people build towards this

# Working arrangement recap

What are the pros and cons of working for a big company vs. a startup?

Why do so many people want to do a startup if so many fail?

What option makes the most sense for you?

# What interests, passions, and skills help people succeed in tech/CS?

- Curiosity
- Hard Work
- Teamwork
- Logical Thinking
- Patience
- Empathy

# Can I get a job with a background?

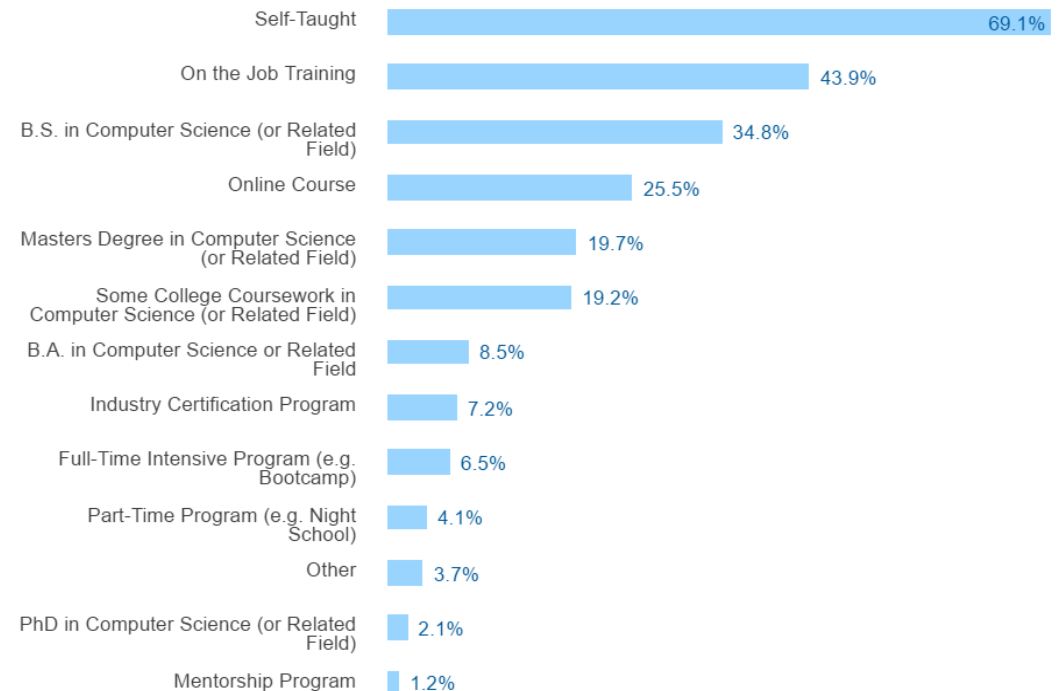
- Yes – people do it, even if it isn't easy
- The stigma you will face is real
- In our conversations with employers, what they expect regarding criminal record:
  - Honesty – be upfront
  - Context/circumstance about the crime
  - Accepting responsibility
  - Evidence of significant progress & a new direction
- Some questions to think about:
  - How can you demonstrate that an employer can trust you?
  - What strengths has your experience in prison given you that would be an asset to an employer?

# What are my education options for CS?

Regardless of path of study, skills and ability to “do the job” trumps educational pathway

- Bootcamp
- Community College
- Four-year degree
- Self-study (Free!)

## Developer Education Today



40,183 responses from non-student developers

# How to get hired?

- Develop clear goals about where you want to go
- Build things, all the time, and show them publicly
- Meet people, build relationships and ask for advice
- Degrees and certifications: they don't hurt
- Though everything, be curious about how things work



# How do I demonstrate my skills?

- Portfolio: what goes in it?
  - Personal projects
  - Open source contributions
- How to make your portfolio public
  - Make a website with your best work
  - Put your code on GitHub
- Technical Interviews



# Build relationships in the Tech Community

- What connections do you need?
  - Employers, Mentors, Community
- How do you get them?
  - Go to meetups and events
  - Research and reach out
  - Open Source contributions



# Variable

So far, we've talked about code based on reacting to "in the moment" conditions. But robots (and programs) can store data, and use that to help them to decide what to do.

A variable is a value that can change, depending on conditions or on information passed to the program. This variable can then be used in programs. Variables store different types of data, below are three common types:

- **String** - A string variable is a string of alphanumeric characters and allowed symbols that are contained within quotation marks. For example, "Hello world, I'm 102 years old today!" is an example of a string. Strings are basically used for storing text.
- **Number** - A number variable couldn't be more straightforward because all number variables store are numbers. You don't store them within quotes like strings, instead, numbers can just be written as they are. If you want to store the number 9 in a variable, you just write 9. Today we will only be using numerical variables.
- **Boolean** - A boolean variable is one of two things: *true* or *false*. This data type is kind of like an on and off switch, so you can ask true or false questions in your code. For example, you might ask "is the video currently playing?" The response you'd get would be a boolean variable. *True* would mean the video is currently playing and *false* would mean it is not.

# Variable

These variables allow us to 'save' the values from our coding, in order to write more complex code. We do this by defining a variable with a specific, case-sensitive name. Once you create (or **declare**) a variable as having a particular name, you can then call up that value by typing the variable name.

**First step, name your variable. Name should be:**

- Meaningful (distancetotheobstacleonleft)
- Short (distleftobstacle)
- Readable (DistLeftObstacle)

**Set a value for your variable.**

*DistLeftObstacle = 100*

**Adjust and use value for your variable**

*DistLeftObstacle = DistLeftObstacle + 100*

*If DistLeftObstacle is less than DistRightObstacle, turn right. Otherwise turn left.*