

# How to get a Google Internship

Evan H. Carmi

Learn what it takes to get a software engineering internship with companies such as Google, The New York Times, and startups.

## Chapter 5

# Interviewing

So you've heard back from some companies and they want to speak with you. Wonderful! This is your chance to show them how great you are. In addition to a company learning more about you, this is a time for you to ask questions and decide whether this is really a place where you'd want to work. Internships often lead to full time jobs, so this can be a decision that will start your career. Some questions are good to ask at the beginning of the process; others you may want to hold off on until after you have a job offer in writing. For example, wait until after you have an offer if you are wondering whether you can go on a weeklong road trip with some friends midway through your internship.

Once you begin speaking with different companies, there are a few things that can happen next. You may be asked to answer some questions by email or do a small programming project. When applying to [Atlassian](#) I was asked the following questions at the very early stages of the process:

Hi Evan,

Thank you for taking the time to apply for an Intern role at Atlassian. We've reviewed your application and are excited to invite you to move ahead to the next stage of the process. Before I dig into the details, congratulations on making it to the next stage!

The next stage involves responding to questions. These questions will give us some more insight into your suitability for the position.

Please return your completed responses to me at your earliest convenience.

- What languages do you have experience with? (at university or otherwise?)
- What experience, if any, do you have with web technologies (js, css, html, etc)?
- Do you code in your spare time? What coding projects have you been involved in?
- What's your favorite part of coding?
- What's your least favourite part of coding? (if you don't say bug fixing I'll be surprised! ;)

- What would you like to learn more about?
- What are you interested in?
- What part of coding do you feel you're strongest at? What makes you strong in this area?

All the best!

These questions provide an easy way to explain your experience and see if this is a good position. Feel free to ask any questions you have at this point in time.

You may be asked to write some code. Since you can write it offline, these problems are less stressful than a coding interview. Take your time and be sure to use best practices. Test the code with different versions of software and make sure there are no external dependencies that will cause problems.

When I was applying for an internship position at The New York Times – which I later decided to accept – I was asked the following:

Evan,

Brian passed along your name as an intern candidate for this summer. By now he should have been in touch to schedule a phone interview; ahead of that conversation, we'd like to get your answers to a few quick points:

1. 100 words or less: What's the best online news project (could be a site, an interactive, an app, anything really) you've seen recently? What made it great?
2. In another 100 words, what's most in need of improvement on NYTimes.com?
3. Short scripting exercise:

Using the language/technology of your choice (Ruby strongly preferred, but anything runnable on OS X okay), write a short script that given the URL to an article on NYTimes.com prints the author's byline to standard output.

For example, I should be able to run:

```
ruby byline.rb "http://www.nytimes.com/2011/03/02/world/middleeast/02yemen.html?hp"
and get back
```

```
"LAURA KASINOF"
```

If you run into questions on edge cases just make a best guess and note the limitations in the comments at the top of your script. This should be quick; no need for a 45-file inheritance chain.

4. What's the best example of your work? If available on the internet, would be best to pass a link but also okay to attach. What did you like most about it?

Look forward to speaking soon!

Ben

I decided to write my scripting response in Python since I was more comfortable with it than Ruby at the time. I thought it was better to write something cleanly in idiomatic Python than a mediocre solution in Ruby. Use your judgement to show off your best coding skills. Here's the script I sent back to The New York Times:

```

"""
New York Times Pop Quiz
Invocation syntax:
    python byline.py http://www.nytimes.com/2011/03/02/world/middleeast/02yemen.html?hp

For articles with two authors, I simply print both authors names.

Evan Carmi
"""

import urllib, sys
from sgmlib import SGMLParser

def main(argv):
    """
    Parse invocation parameters and call byline finding function.
    """
    argv_length = len(argv)
    if argv_length < 2:
        print __doc__
    elif (argv_length == 3 and argv[2] == '--regex'):
        byline_parse(argv[2], regex=True)
    else:
        byline_parse(argv[1])

def byline_parse(url, regex=False):
    """
    Get and parse URL.
    """
    f = urllib.urlopen(url)
    s = f.read()
    f.close()

    parser = BylineParser()
    parser.feed(s)
    parser.close()
    print ''' + parser.get_author() + '''

    sys.exit()

class BylineParser(SGMLParser):
    """

```

```

Define specific functions for a subclass of SGMLParser.
"""
def __init__(self, verbose=0):
    SGMLParser.__init__(self, verbose)
    self.inside_byline = 0
    self.data = ''

# Called to process beginning of h6 tag.
def start_h6(self, attrs):
    for name, value in attrs:
        if name == "class" and value == "byline":
            self.inside_byline = True

# Called to process end of h6 tag.
def end_h6(self):
    self.inside_byline = False

# Called to handle data inside tags.
def handle_data(self, data):
    if self.inside_byline:
        self.data += data

# Get author of read data
def get_author(self):
    # Remove beginning 'By '; remove outside whitespace.
    return str.strip(self.data.replace('By ', ''))

if __name__ == '__main__':
    main(sys.argv)

```

I was sure to test my code on multiple computers and make sure it didn't have any requirements for a specific version of Python or any installed packages. I also was clear about how to invoke the script.

If all goes well, the next step is an interview. The interview is not something to be feared. Getting good at interviews simply takes practice. It can be good to apply to some companies that are not your top choices at the beginning of your search. This will give you the chance to practice in interviews that you are less invested in. By the time you get an interview with your dream job, you will have had experience interviewing and be more calm and prepared. You do not want your interview with Google to be the first interview of your life.

## Before the Interview

So you have an interview scheduled. Congratulations! This is a great step. If you get an offer, that's awesome. If this one doesn't work out, you'll have gained some valuable experience that will help you with the next company. Before the time of your interview, make sure you are prepared.

## **Find a distraction-free place**

If your interview is not in person, you'll want a good place for your part of the process. You need to find a distraction-free space with a good phone and/or internet connection. If you are at a university, there are often rooms you can reserve for an interview. Ask the Career Resource Center or similar organization what is available. If you are just doing it at your home, be sure to let roommates/family/neighbors know so they do not make noise or interrupt you during the interview.

## **Schedule time before and after**

Make sure you are free before and after your interview in case the interview goes a bit longer than planned.

## **Ensure you have proper equipment**

If you are going to have a phone conversation, make sure that you have a phone with good reception. If you have access to a landline phone, even better. Landlines often have much better sound quality than a cell phone. If you will be using your computer and internet, make sure you have a good connection. It may be worthwhile to find an ethernet cable and plug in directly if you sometimes have WiFi connectivity issues. You want to make sure there will not be any distractions or technical problems during the interview.

## **Research the company**

Before your interview, research the company. Know what the company does. Ask questions that are not trivially answered on their homepage. If the company has a product, be sure to try it out. Be familiar with what they are doing. The people interviewing you have spent time on this problem or product, so you should be somewhat familiar with it before the interview. If the company is large and has many products, ask about the job and title of your interviewer beforehand.

## **The Interview**

The time has arrived. You are set up in a good location without any distractions. You've researched the company and are familiar with them. Now you are ready to actually have the interview. Be online at least a few minutes before the interview starts. Often the interviewer will need to email you to check about your phone number or some other info. If the interview is on Skype, make sure you are online and have accepted their contact request.

Take a deep breath and have a conversation. As you go through the problems, try to think out loud. Be sure to explain the way you are approaching a problem. You don't need to answer the question correctly; you need to explain the way you think about a problem and how you solve it.

Sometimes it can be valuable to take notes during and after the interview. Write down what you were asked, especially the questions you didn't solve. These may be good problems to practice with later. Write down any questions they answered for you. These notes will come in handy later. When comparing offers you don't want to mix up organizations.

## Coding Interviews

You will often be asked to write code in an interview. There are lots of different tools that allow you and the interviewer to modify a document online. Sometimes people just use a Google Document. Regardless, writing code while someone is watching can be stressful. The best way to get more comfortable doing this is practice. Ask a friend to do a mock programming interview where they give a problem and you have to code up a solution to it on the spot as they watch and ask questions. If you are doing an in-person interview you may be asked to write on a whiteboard instead of a shared online document, but the same techniques apply.

## Freezing up in an interview

Sometimes you get asked a question that you don't know. Interviewers are often just trying to get a sense of you. It's not always important whether correctly you answer the exact question they ask. Rather, they want to see how you think. So if you are unsure of a question, explain the parts that you do understand. Try to talk out loud at the beginning of the question. If you are still stuck, ask for a hint. Explain how you would go about solving this question if you had to work on it outside of an interview. Who would you ask? What would you search for? How would you solve a problem that you don't know the answer to? If it's something you could find the answer to online, explain that process. Googling is a skill.

At the end of the interview you can ask the interviewer questions. Try to think about some questions beforehand. Maybe you want to know what programming languages and technical stack the company uses. Maybe you are wondering how the company is structured and run and what kinds of projects interns work on. Asking some personal questions can also be very informative. For example, you can ask how long your interviewer has been with the company, where they worked previously, and how the jobs compare. You might ask if they like their current job, and what is their most and least favorite part of it. Finally, be sure to ask when you should hear back about the next steps in the process.

## After the Interview

After the interview you can send a follow up email thanking them for interviewing you. Then you wait.

When I applied to the New York Times I asked when I should hear back from the team with a decision. They said they would decide by the following Monday. However, that day passed without any word. So I sent them this email as a gentle nudge:

Hi Brian,

It was great chatting with you guys last Thursday. I was told you should have made decisions by yesterday (Monday) regarding internships, so I just wanted check in.

Thanks, Evan

You don't want to be annoying, but a brief reminder can be helpful.

Companies are always busy and very often will miss deadlines getting back to you. Be sure to remind them about any deadlines. It's a good way for them to also remember your name.

## Offered a position

If the interview goes well, hopefully you will be offered a position. Sometimes a company will call to offer you an internship or job over the phone rather than sending an email. When I was offered a job at The New York Times, I missed the phone call. Then I got an email like this:

Hey Evan,

Tried to give you a ring, but couldn't get through. Love to talk to you about coming to work with us this summer. Give me a ring when you have a minute at the number below.

It wasn't quite an offer, but it sounded good. It's rare that a company will call you to tell you you've been rejected. Phone calls are generally good signs.





## Chapter 8

# Conclusion

Now it is your turn. You have to stop reading and start doing. Start applying, use the skills discussed in this book. Build a website. Build some projects. Get the internship you want!

If you've enjoyed this book write about it, tweet about it, recommend it to others, and let me know.

I can be reached by email at [evan@ecarmi.org](mailto:evan@ecarmi.org), on twitter [@ehc](https://twitter.com/ehc) and via my website [ecarmi.org](http://ecarmi.org).

## About the Author

Evan Carmi is an experienced software engineer and designer, currently working as [Brewster](#), a startup in New York City. He has developed applications in Django, Ruby on Rails, and Flask across numerous platforms including Amazon Web Services and Rackspace. Before joining Brewster he spent two summers at The New York Times building their 2012 London Olympics site, created software for an award-winning professor researching Nonviolent Civil Resistance, and managed a team of programmers developing software in multimedia classrooms.

Evan graduated from Wesleyan University in 2013. At Wesleyan, Evan majored in Computer Science, studied three foreign languages, worked on a Vandercook letterpress, made a book, and took a course titled Acro-Yoga in the Age of Mechanicity.

He was awarded the Michael Rice Prize for Excellence in Computer Science.