# Web Development

*by Devmountain Tutorials*

| :≡ | WHAT'S COVERED |
|----|----------------|

This section will explore the basic building blocks of web development by discussing:

1. GOOGLE AND THE HISTORY OF WEB DEVELOPMENT
2. SOFTWARE ENGINEERING
3. WEB DEVELOPMENT
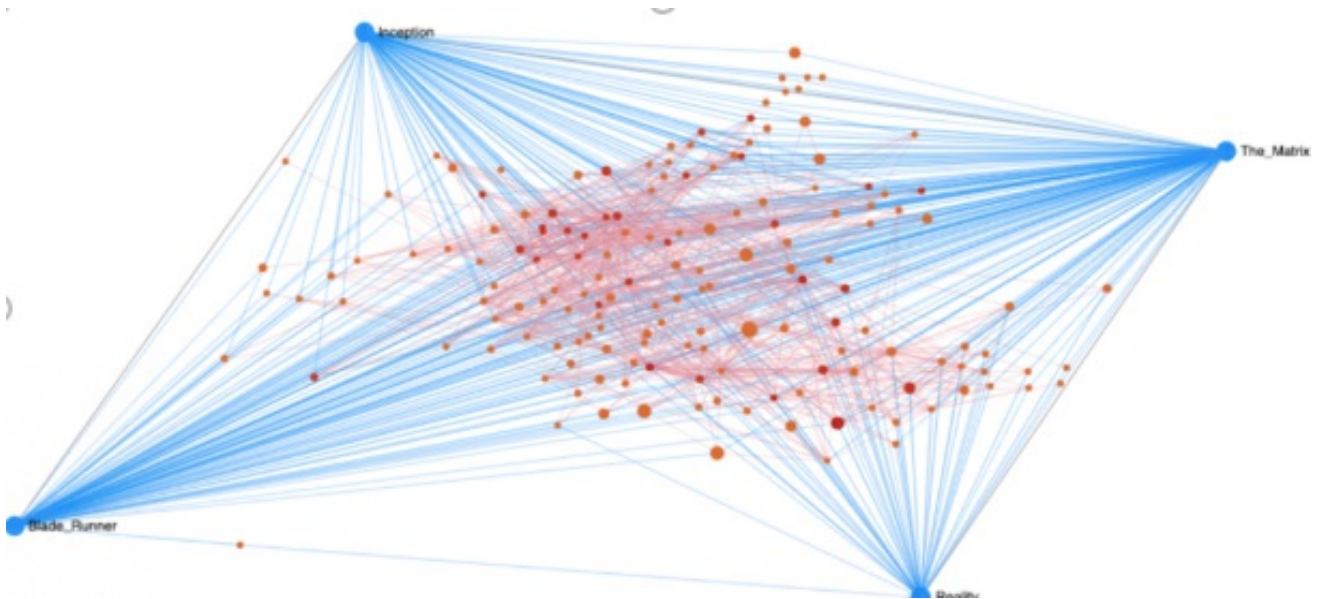4. HOW THE WEB WORKS

# 1. GOOGLE AND THE HISTORY OF WEB DEVELOPMENT



**Image of a graph of the links between four Wikipedia articles: "Inception", "The Matrix", "Blade Runner", and "Reality"**

In 1996, Stanford University PhD students, Larry Page and Sergey Brin, started working on a research project nicknamed "BackRub". Their original motivation for the project was to explore a relatively new technology — **hyperlinks**. They wanted to visualize links between different websites on the Internet as a huge, interconnected graph.

Hyperlinks (or simply, **links**) are those pieces of text on a website that connect you from one page to another. Traditionally, links are displayed in blue, underlined text to help differentiate them from regular text.

Soon, Page and Brin realized that their humble research project could be something more. In fact, they realized that the programs they'd developed for research purposes could become an entirely new product — a search engine for the Internet. They called it Google Search.

In the late 1990s, the Internet was so small that Page and Brin were able to operate Google off a single computer housed in a container they built out of Legos.
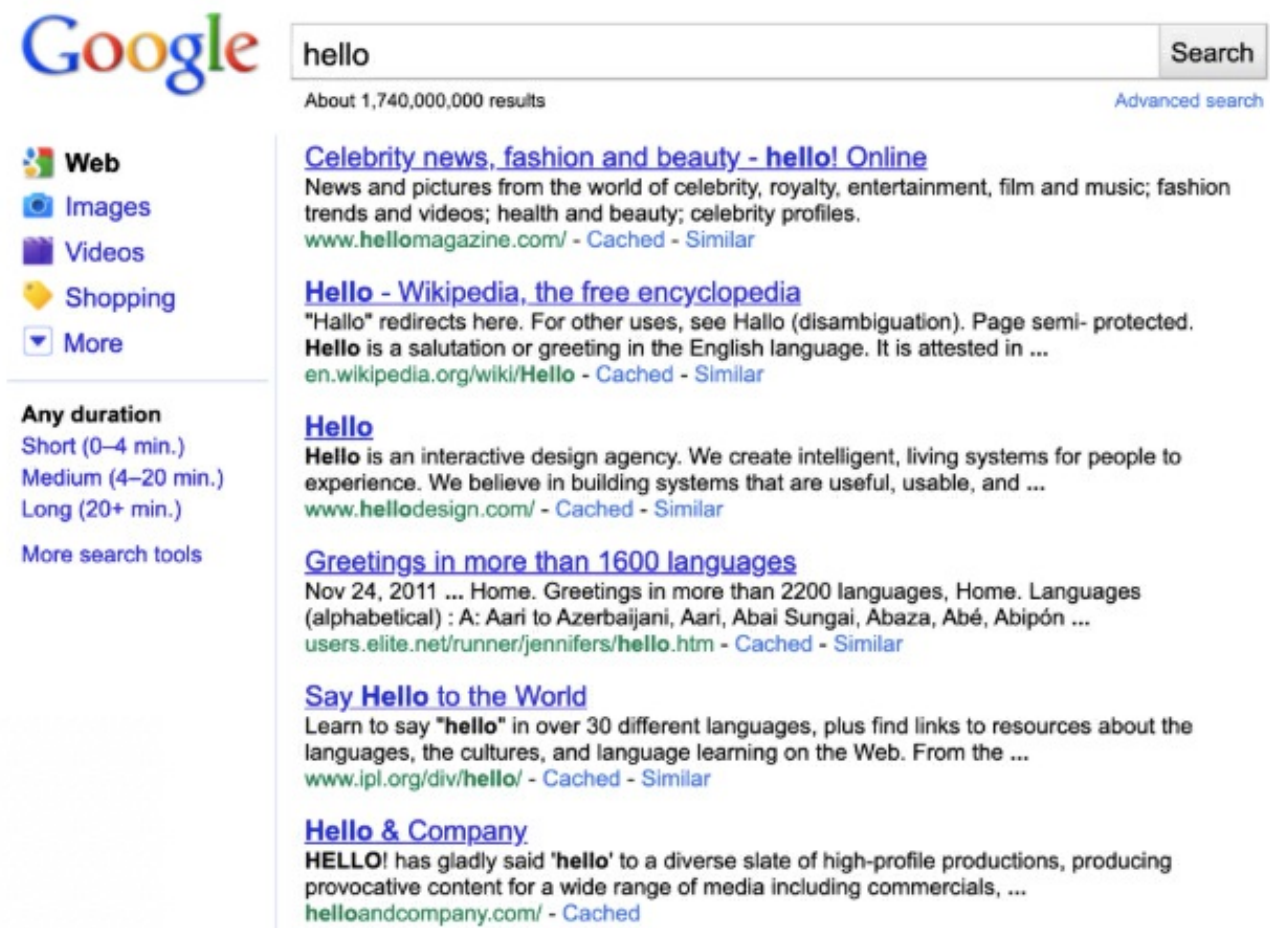
**Computer Housed in Legos**

**A screenshot of what www.google.com looked like in 2000**

Google's homepage had similarly humble beginnings. This is what google.com looked like in 2000:



**A screenshot of Google's search results page from 2000**

Compare early Google to modern Google! A single computer is no longer sufficient for processing the amount of data that Google handles on a daily basis. Nowadays, Google needs entire data centers filled with computers to keep their website running.
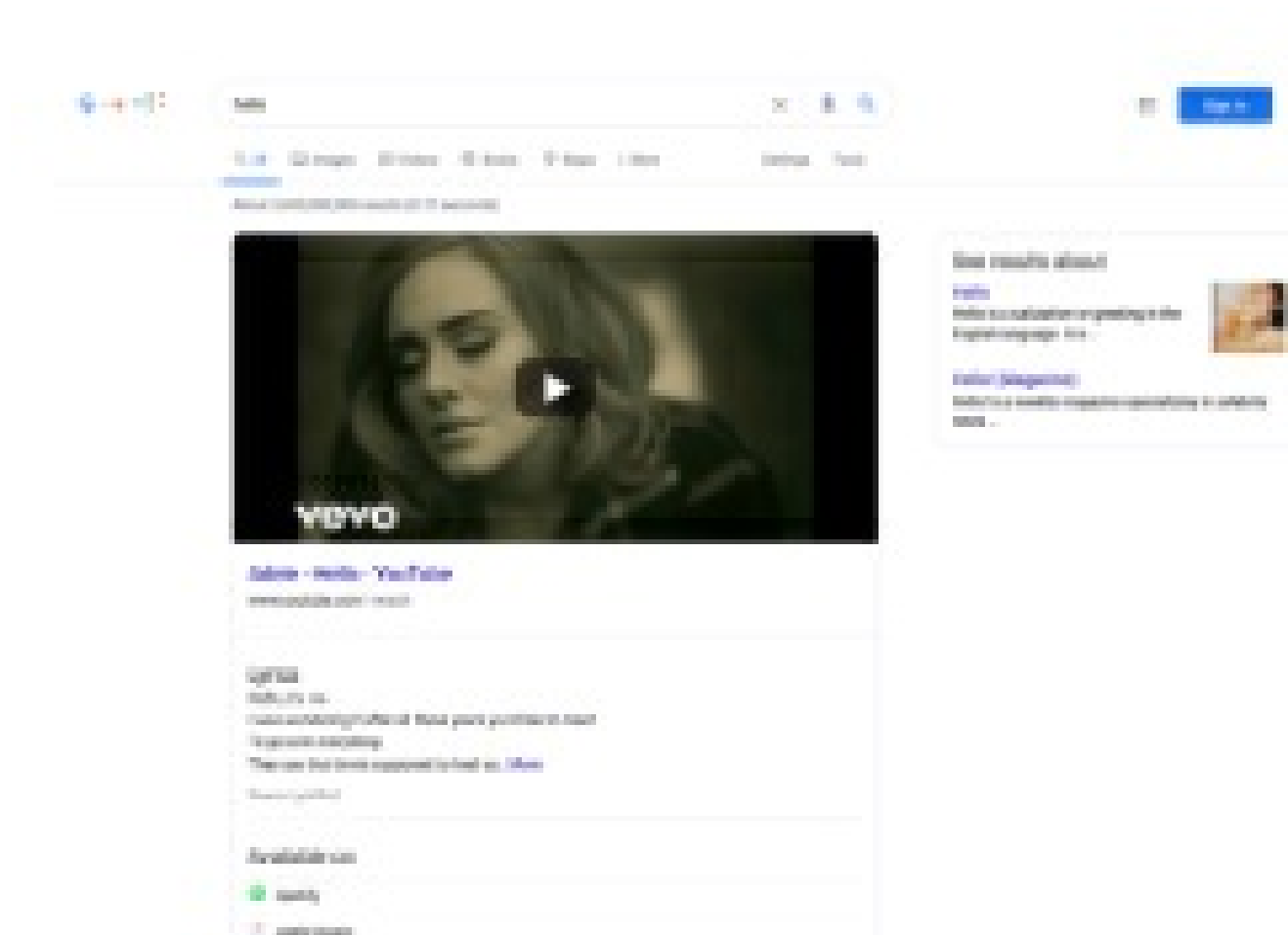
**Google Data Center**
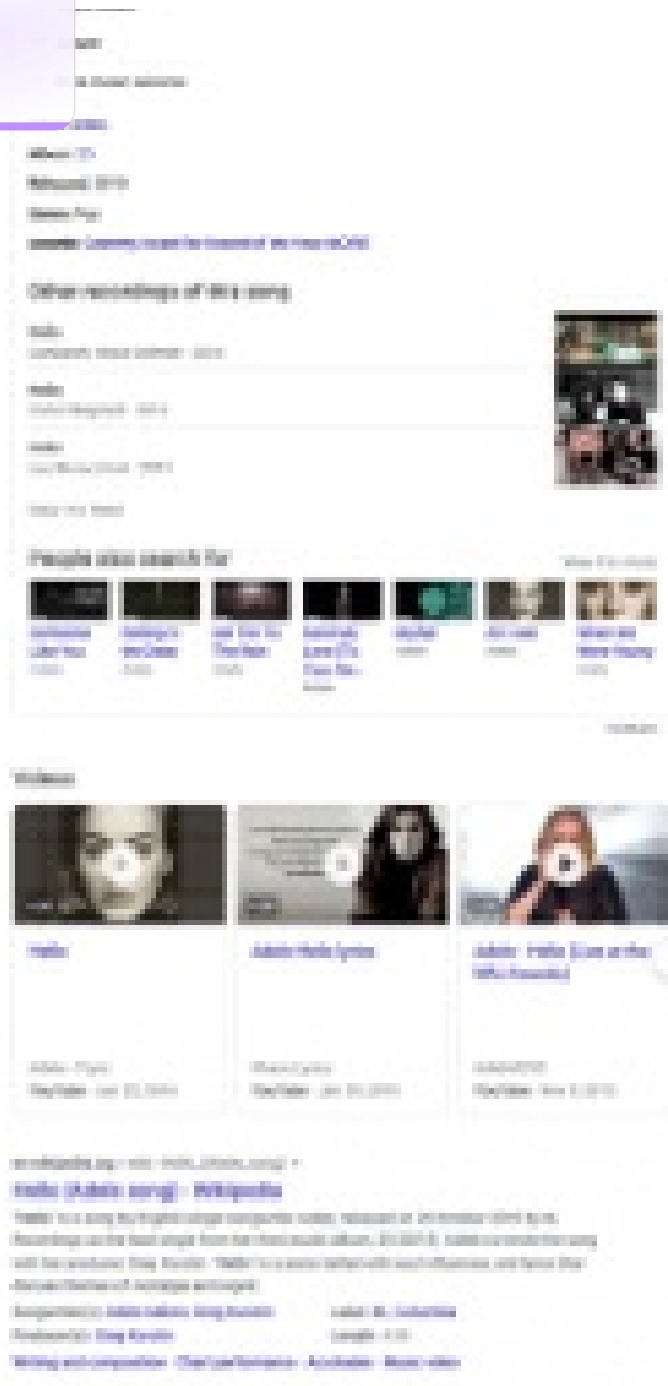
> ⑦ **DID YOU KNOW**

Buildings that are dedicated to housing computers are called **data centers**.

Google's homepage is more complex now, too. For example, Google is able to suggest search terms as you type — a feature that wasn't even possible in the 2000s. Search results aren't just text; they now include images, videos, and more.

Google Homepage 2020

Google search results including pictures 2020

Creating and running a website is no longer a task for just two friends, a computer, and some Legos bricks. As the Internet matured, so did its users and, as a result, companies that operate on the Internet grew as well. Although tech is more accessible now than ever before, in many ways, the industry itself has become less accessible. There are more rules, more best practices to be aware of, and a longer history of insider knowledge that can make industry seem impenetrable.

Our motivation for writing this course is to break down those barriers by introducing you to the world of web development and web engineering. By the end of this course, we hope that you'll be able to picture yourself in the tech industry and even consider it a viable career path.

This first unit will introduce you to the prerequisite knowledge you'll need to enjoy the rest of this course. First,

development in general before going into topics exclusive to web development.

[-] TERM TO KNOW

**Hyperlink**

Those pieces of text on a website that connect you from one page to another.

# 2. SOFTWARE ENGINEERING

Web development is a subset of **software development**— the work involved in creating programs that computers can execute. These programs can be as simple as the code used to drive a stopwatch and as complex as the applications on your computer. The types of tasks performed in software development are vast and encompasses contributions from diverse areas, one of which is **software engineering**.

```python
# Required argument is the title of the question
title_text = self.arguments[0]
textnodes, messages = self.state.inline_text(title_text, self.lineno)
title_node = nodes.title(title_text, '', *textnodes)
title_node.source, title_node.line = (
    self.state_machine.get_source_and_line(self.lineno))
title_node['classes'] += ['mcq-title']

node += title_node

body = nodes.container(text)
self.state.nested_parse(self.content, self.content_offset, body)

for listnode in body.traverse(nodes.enumerated_list):
    listnode['classes'] += ['mcq-choices']

    # Attach letters to each node; this will make it easier to
    # generate the answer key
    letter_iter = iter(LETTERS)
    for list_item in listnode.traverse(nodes.list_item):
        list_item.attributes['choice_letter'] = next(letter_iter)

node += body.children
node += messages
```

**Software engineers write code**

Software engineering is a unique discipline that combines features typically exclusive to mathematic, scientific, and creative fields. Like mathematicians, they use specialized languages to denote ideas. Like scientists, they must understand the rules that govern how code runs on computers. Like designers and

engineers create solutions by assembling individual components into systems and evaluate tradeoffs among alternatives.

The most important skill for a software engineer is problem solving — the ability to formulate problems, devise creative solutions for them, and communicate those solutions effectively.

The types of problems that software engineers encounter can vary, but the way they approach building solutions to those problems is similar:

- Develop a general solution
- Communicate the solution to a computer, in a way that the computer understands
- Use the computer to automate the execution of that solution

Don't worry if the steps above seem abstract right now. It's more important to keep these ideas in the back of your mind as you progress through the course. Although this course focuses on web development and engineering, *all* software engineers exercise the same skills and solve the same types of problems.

📄 **TERM TO KNOW**

**Software Development**
The work involved in creating programs that computers can execute. The tasks involved can include graphic design, copyediting, and software engineering.

**Software Engineering**
Software engineering is a discipline that combines features from mathematic, scientific, and creative fields. The work that software engineers do isn't just writing code. They also use specialized languages (code) to denote ideas like mathematicians, understand the rules of the system they're working with like scientists, and create solutions and evaluate tradeoffs like designers.

# 3. WEB DEVELOPMENT

**Web development** is a type of software development; it's the work involved in creating websites. The tasks involved in creating a website can cover many disciplines like graphic design, copyediting, and web engineering. **Web engineers** are responsible for writing code and programming websites, but, colloquially, many use the term *web engineer* and *web developer* interchangeably.

Like software engineers, web engineers write code to solve problems. However, there are certain qualities and requirements of web engineering that make it distinct from traditional software engineering. This is due to the rapidly evolving nature of the World Wide Web (or the web) and the unique constraints it imposes on websites.

The **World Wide Web (WWW)**, or **the web**, is an information system where websites and other resources are identified by **Uniform Resource Locators** (**URLs**, like **https://google.com**) and are accessible over the Internet. Early websites were simple, consisting of **static**, or unchanging, web pages. Modern websites are far more complex. Instead of static pages, they have dynamic web pages that can change and react to user input. Many of them — Facebook, Twitter, and Amazon, to name a few — are as complex and fully-featured as desktop or mobile applications. The complex behavior of these websites places unique demands on their usability, performance, security, and ability to grow and evolve.

The qualities and requirements that make web engineering distinct from software engineering are all related

It probably would be hard to ask someone to become interested in web engineering and appreciate its complexities without explaining how the web works first.

 **TERM TO KNOW**

**Web Development**

Web development is software development on the web. In other words, it's the work involved in creating websites.

**Web Engineers**

Web engineers are software engineers that are specifically responsible for writing code and programming websites.

**World Wide Web (WWW)**

An information system where websites and other resources are identified by URLs. Colloquially, the World Wide Web is known as, simply, the "web".

**Uniform Resource Locator (URL)**

Websites on the Internet are identified by their URL. They're unique addresses that you can use to access a website. For example, by typing **https://google.com** into your browser, you're able to go to Google's homepage.

# 4. HOW THE WEB WORKS
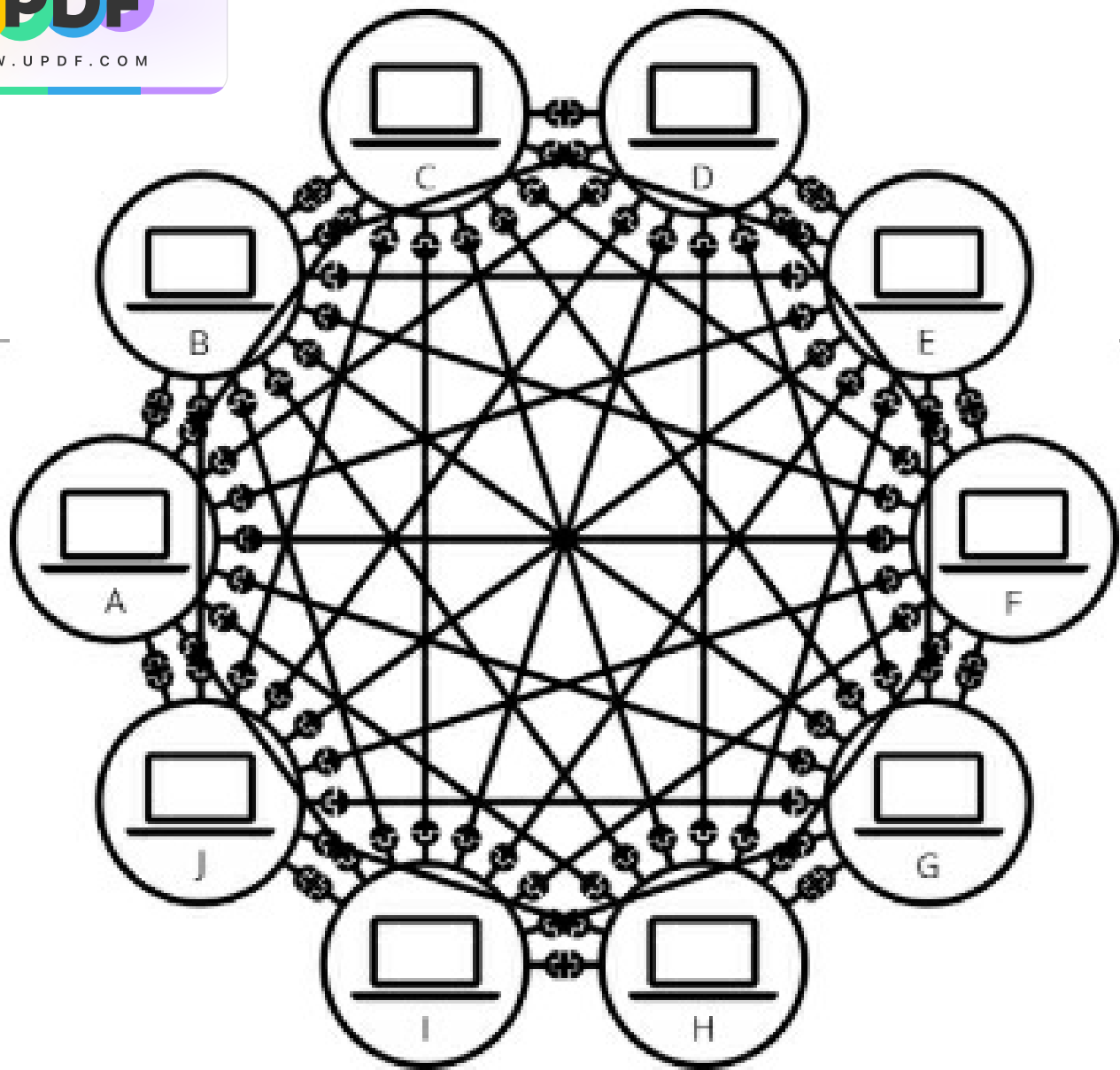
**A diagram of a computer network**

The **Internet** is the technological infrastructure that makes it possible for users to access information on the web. If the web is like snail mail — a system where we send documents and packages to each other's mailing addresses or P.O. boxes — the Internet is analogous to mail carriers like the United States Postal Service. In other words, it is the backbone of the web. At its simplest, the Internet is a very large **computer network** — a group of interconnected computers that can communicate all together.

All computers on the Internet use a particular style of communication called **HyperText Transfer Protocol (HTTP)**. HTTP defines a set of standards and best practices that specifies the communication protocols that computers on the Internet should adhere to.

**Communication protocols** are lists of instructions and rules that define how information ought to be transferred from one place to another. Communication protocols aren't exclusive to the tech industry. If you've ever sent a letter through the mail, you probably followed a set of rules dictated by your government's postal system. For instance, you had to put your letter in an envelope and write the recipient's address in the center of the envelope. In the context of the Internet, HTTP specifies how data should be transported from one

...other. This is how different computers, created by different manufacturers and running ...operating systems, are able to understand one another on the Internet. HTTP is like a common language — as long as everyone follows the HTTP standards, they'll all be able to talk to one another.



**Diagram of how clients and servers talk to each other.**

The protocol specified by HTTP is pretty simple. With HTTP, computers can either send requests or reply with responses. Computers that send requests are called **clients** and the ones that respond are called **servers**. To reword these statements in more technical terms, clients send **HTTP requests** to servers. Servers can listen to requests and send **HTTP responses** back to their clients. Clients make requests for information ("Show me the webpage at sophia.org") or for the server to perform an action ("Please enroll me in this course"). Servers typically respond with the information that the client requested ("Here's the code for the webpage at sophia.org") or a status message ("You've been successfully enrolled in this course") or both.

The term "server" might remind you of dining at a restaurant. When you go out and eat at a sit-down restaurant, you're usually given a menu. Then, a waiter will come to your table and ask for your order. They pass your order along to the kitchen and when your food is ready, they serve it to you at your table. In this analogy, you are the client and the waiter is the server. When you order food from the waiter, you're making a request. The waiter accepts your request and, sometime later, responds with your food.

The scenario above is not a perfect analogy though. At restaurants, the client is (usually) always right. Most restaurants want to provide good customer service to their clients by making sure their needs are met and they have a positive experience. As a result, waiters rarely decline to serve a customer. This is not how clients and servers work. On the Internet, the client is at the whim of the server. In other words, clients can't tell the server what to do; they can only ask. Clients can request data from servers all they want but servers can always reject them. After all, it's "HTTP *requests*", not "HTTP *demands*".

This is how things like user authentication work. Let's say you want to login to your email account to check your messages. You enter your username and password into a form and click "Sign In". When you do that, your web browser will send a request to your email server and ask, "Please authenticate this user. Their

their password is XXXXX". The server will check the username and password to ~~ensure they're correct.~~ If the given password doesn't match the one on file, the server will reject the request and, as a result, your login attempt will fail.

⭐ **BIG IDEA**

Clients send HTTP requests to servers. Servers can listen for HTTP requests and issue HTTP responses. A basic "conversation" between a client and server might look like this:

- **Client**: I would like to view the resource at Sophia.org

- **Server**: Ok, let me see if that resource exists… I found it! Here is the resource at Sophia.org; it's a webpage, by the way.

- **(Client receives the webpage and displays it to the user)**

Now you know how the web works. The web works because it's powered by the Internet, a technology that can connect many computers in a large network. Computers are able to talk to one another because they all adhere to standard communication protocols. The most widely used communication protocol on the Internet is HTTP. The nature of the Internet and HTTP are the technologies that make web engineering unique compared to other forms of software engineering. Web engineers get to tackle the unique challenge of orchestrating the transport of data securely and precisely using a system that, at the most basic level, is pretty simple. The Internet we're using today is, by and large, the same Internet we were using 20 years ago! Yet, we've made a huge amount of technological progress on the Internet since then, thanks to web engineers who were able to build complex software on top of a simple system.

📄 **TERM TO KNOW**

**Internet**
   The technological backbone of the World Wide Web.

**Computer Network**
   A group of interconnected computers that can communicate with one another.

**HyperText Transfer Protocol (HTTP)**
   The name of a communication protocol that's often used to communicate with computers on the web.

**Communication protocol**
   A set of instructions and rules that define how information ought to be transferred from one place to another.

**Clients and HTTP requests**
   Clients can send HTTP requests to servers. Servers don't always respond to every request, though. Servers are able to reject requests as well.

**Servers and HTTP responses**
   Servers can take HTTP requests clients and reply by sending HTTP responses back to their clients.

📄 **TERMS TO KNOW**

**Clients and HTTP requests**
   Clients can send HTTP requests to servers. Servers don't always respond to every request, though. Servers are able to reject requests as well.

**Communication protocol**

A set of instructions and rules that define how information ought to be transferred from one place to another.

**Computer network**

A group of interconnected computers that can communicate with one another.

**HyperText Transfer Protocol (HTTP)**

The name of a communication protocol that's often used to communicate with computers on the web.

**Hyperlink**

Those pieces of text on a website that connect you from one page to another.

**Internet**

The technological backbone of the World Wide Web.

**Servers and HTTP responses**

Servers can take HTTP requests clients and reply by sending HTTP responses back to their clients.

**Software Development**

The work involved in creating programs that computers can execute. The tasks involved can include graphic design, copyediting, and software engineering.

**Software Engineering**

Software engineering is a discipline that combines features from mathematic, scientific, and creative fields. The work that software engineers do isn't just writing code. They also use specialized languages (code) to denote ideas like mathematicians, understand the rules of the system they're working with like scientists, and create solutions and evaluate tradeoffs like designers.

**Uniform Resource Locator (URL)**

Websites on the Internet are identified by their URL. They're unique addresses that you can use to access a website. For example, by typing **https://google.com** into your browser, you're able to go to Google's homepage.

**Web Development**

Web development is software development on the web. In other words, it's the work involved in creating websites.

**Web Engineers**

Web engineers are software engineers that are specifically responsible for writing code and programming websites.

**World Wide Web (WWW)**

An information system where websites and other resources are identified by URLs. Colloquially, the World Wide Web is known as, simply, the "web".