



LECTURE 1

Introduction to Internet and WWW

History of the Internet

- ◉ J.C.R. Licklider
 - > envisioned "Galactic Network" concept - a globally interconnected set of computers through which everyone could quickly access data and programs from any site (MIT, Aug 1962)
 - > 1st head of the computer research program at DARPA (Oct 1962)
- ◉ Lawrence G. Roberts
 - > develop the computer network concept & publish ARPANET (DARPA, 1967)
 - > ARPANET had been turned over to the Defense Communications Agency
- ◉ Ira Fuchs and Greydon Freeman (1981)
 - > devised BITNET, which linked academic mainframe computers for electronic mail

History of the Internet

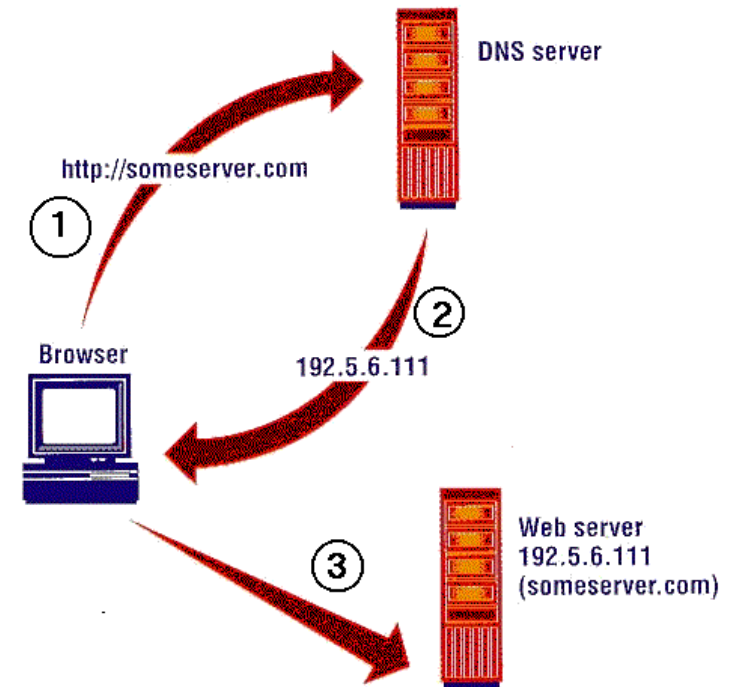
◎ CSNET- 1981

- > [National Science Foundation](#) (NSF) developed the [Computer Science Network](#) (CSNET) (1981) to extend the ARPANET networking benefits for computer science departments at academic & research institutions.
- > The [Internet protocol suite](#) (TCP/IP) was standardized, and consequently, the concept of a world-wide network of interconnected TCP/IP networks, called the Internet, was introduced. (1982)
- > Commercial [Internet service providers](#) (ISPs) began to emerge in the late 1980s and early 1990s.

How does the internet work?

What is the internet?

- A huge collection of computers connected by **TCP/IP** (Transmission Control Protocol/Internet Protocol) in a network
- IP addresses
 - > Set of four integers uniquely identifying each node
 - > Example: 128.135.197.76
- Since numbers are difficult to remember, the Internet evolved DNS addresses

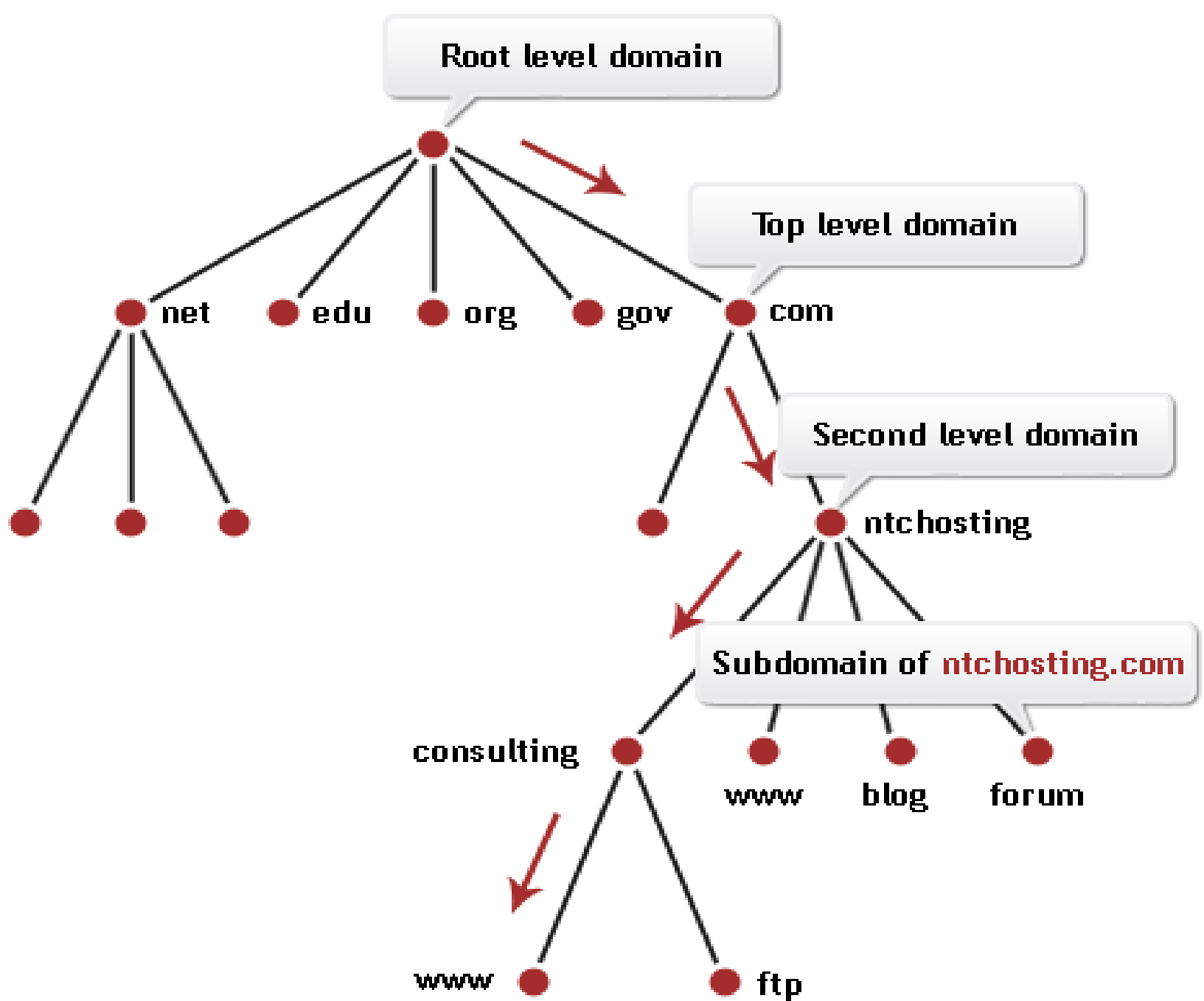


Internet Protocol (IP)

- ◉ Computers are identified by unique numeric addresses
- ◉ Form: 32-bit binary number
- ◉ Example : **191.57.126.0** to **191.57.126.255** has 256 IP addresses
- ◉ Written as four 8-bit numbers, separated by periods
- ◉ Organizations are assigned groups of IPs for their computers

Domain Name System (DNS)

- ◉ DNS translates domain names to network addresses. For example:
 - > altavista.com is 192.136.112.39
- ◉ Separate domain administrations:
 - > Defined types: COM, EDU, GOV, BIZ, TEL, NET, ORG, INFO, NAME, MOBI
 - > Countries: US, JP, FR, MY, RU, CH, UK, etc.
- ◉ Tree structured directory
- ◉ A DNS address (`ftmk.utem.edu.my`) consists of:
 - > Domain name for organizations (**`ftmk.utem.edu.my`**)
 - institutional site name (**`ftmk.utem`**)
 - top-level domain(tld) name (**`edu.my`**)
 - > host name for individual machines (**`ftmk`**)



Domain naming rules

- ◉ Max 255 characters per name
- ◉ From 2 to 5 labels per domain name
 - > faizal.uhost.co.tv has 4 labels
- ◉ Labels of up to 63 characters
- ◉ Allowable characters are A-Z, 0-9, and '-'
- ◉ Domain names are not case sensitive
 - > Other parts of a URL may be case sensitive
- ◉ Trademark owners get preference

World Wide Web (Web)

- ◉ Web allows computer users to locate and view multimedia-based documents on almost any subject over the Internet
- ◉ Web is an application to share and access Web documents on top of the Internet
 - > Other applications: email, FTP, newsgroups, instant messaging, etc.
- ◉ Founded by Tim Berners Lee of CERN, 1989
- ◉ The WWW is *not* the Internet
- ◉ Tim developed a technology for sharing information via hyperlinked text documents called HTML
- ◉ Tim also wrote communication protocols to form the backbone of the WWW. He wrote the Hypertext Transfer Protocol (HTTP) – a communication protocol used to send information over the Web
- ◉ Web documents (Web pages) are formatted in HyperText Markup Language(HTML)

Web browsers

- Client software that allows users to access the Web's rich content
 - Microsoft's Internet Explorer, Mozilla's Firefox, Apple's Safari & Opera Software's Opera
- People use web browsers to access the information available on the Web & to share or exchange the content with other users
- May include tools for e-mail, address book, news, Web authoring, etc.
- May run programs in Java, Javascript, ActiveX, or Shockwave
- Records data in Cookies, logs, cache



Web Servers

- ◉ A specialized software that responds to client requests (typically from a web browser) by providing resources such as HTML documents.
- ◉ E.g. Apache HTTP Server, Microsoft Internet Information Server (IIS) etc.
- ◉ Provides access to files
- ◉ Runs programs in CGI, Perl, Java, C, etc.
- ◉ May support relational database (Oracle, DB2, SQL Server, etc.)
- ◉ May provide access to legacy applications
- ◉ May log access requests



Uniform Resource Locator (URL)

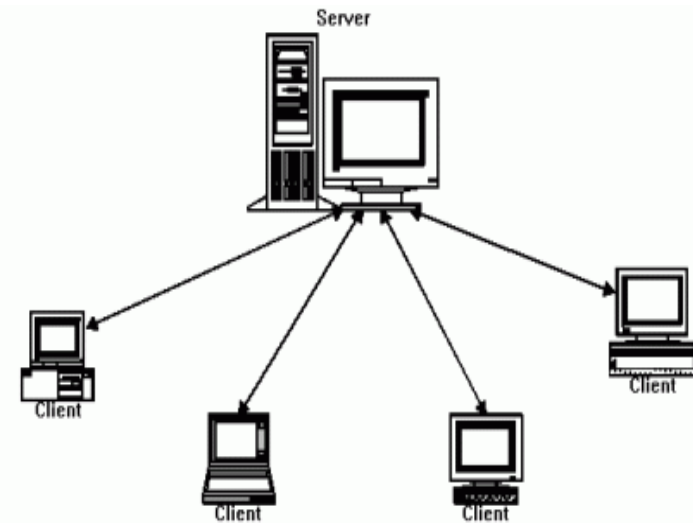
- ◉ All Web pages are addressed with URLs
- ◉ Format: *protocol:address*
 - > *protocol* may be
 - ftp, http, mailto, telnet, etc
 - > *address* specifies
 - A server name
 - A directory path (optional)
 - A filename
- ◉ Example:
 - > <http://www.eftmk.utem.edu.my/bitm2113/raja1.png>

MIME

- ◉ MIME stands for Multipurpose Internet Mail Extensions (MIME)
- ◉ Originally developed for email
- ◉ Used to specify to the browser the form of a file returned by the server (attached by the server to the beginning of the document)
- ◉ Form: type/subtype
 - > Examples: `text/plain`, `text/html`, `image/gif`, `image/jpeg`
- ◉ Server gets type from the requested file name's suffix
 - > `*.html` implies `text/html`
- ◉ Browser gets the type explicitly from the server
- ◉ Experimental types
 - > Subtype begins with x-, example, `video/x-msvideo`
 - > Experimental subtypes are added to MIME specification stored in user's Web server.
- ◉ Experimental types require the server to send a helper application or plug-in so the browser can deal with the file

HTTP - Hyper Text Transfer Protocol

- ◉ Transactions between client and server
 - > Client connects
 - > Client makes one or more Request
 - > Server Responds to Requests
 - > Client drops connection
- ◉ Http client request has three parts:
 - 1) Method, document URL, HTTP version
 - Most frequently used methods are:
 - > GET request a document or data
 - > HEAD request document attributes only
 - > POST send data to server
 - 2) Browser type, OS, and acceptable media
 - 3) Optional data



HTTP request example

```
GET /articles/news/today.asp HTTP/1.1
Accept: */*
Accept-Language: en-us
Connection: Keep-Alive
Host: localhost
Referer: http://localhost/links.asp
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows
          NT 5.0)
Accept-Encoding: gzip, deflate
```

The header of a request must be followed by a blank line, which is used to separate the header from the body of the request.

HTTP server response

- ⦿ Response has three parts:
- ⦿ 1) HTTP version, response code, message
- ⦿ 2) Header information
 - > Date and time
 - > Server type
 - > Last modified date and time
 - > Content type and length
- ⦿ 3) Body (optional)

Response Phase Form

Status line
Response header fields
blank line
Response body

Status line format:

HTTP version status code explanation

Example: HTTP/1.1 200 OK

Status code is a three-digit number; first digit specifies the general status

1 => Informational

2 => Success

3 => Redirection

4 => Client error

5 => Server error

Status code 404 is for?????

HTTP response example

- ⦿ The header field, **Content-type**, is required
- ⦿ An example of a complete response header:

```
HTTP/1.1 200 OK
Date: Mon, 27 Jun 2002 17:22:47 GMT
Server: Apache/1.3.22 (Unix) (Red-Hat/Linux)
Last-modified: Wed, 26 Jun 2002 18:12:29 GMT
Accept-ranges: bytes
Content-length: 75
Connection: close
Content-type: text/html
```

```
<HTML>
```

```
<BODY>
```

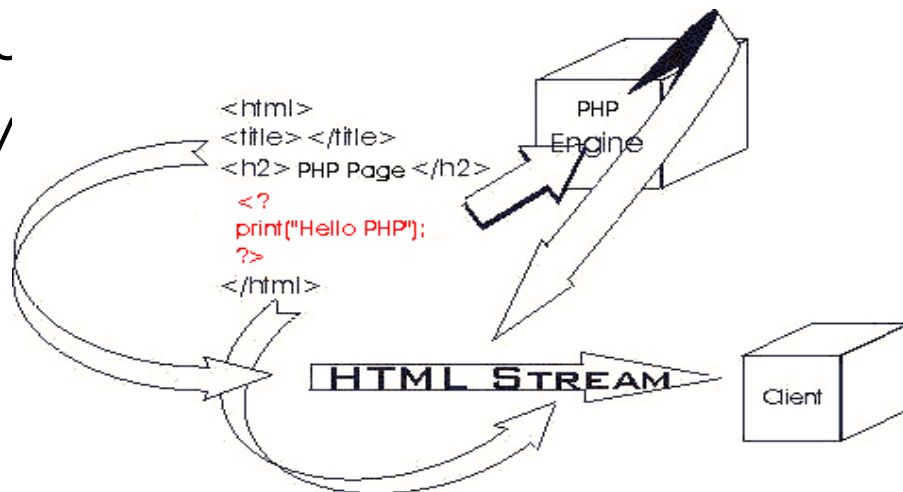
```
....
```

Scriptings

- ◉ Client-side scripting
 - > Validates user input
 - > Accesses the browser
 - > Enhances Web pages with ActiveX® controls, applets, etc.
 - > Manipulates browser documents
- ◉ Client-side validation
 - > Reduces number of requests that need to be passed to server
- ◉ Client-side scripting limitations
 - > Browser dependency
 - > Viewable to users through View Source command
- ◉ Example of Client-side scripting – JavaScript, VBScripts

Scriptings

- ◉ Server-side scripts
 - > Provides programmers greater flexibility
 - > Generates custom responses for clients
 - > Contains greater programmatic capabilities than client-side equivalents
 - > Has access to server-side software that extend server fu
 - > Example of serv JSP, CGI/Perl



Frontend vs. Backend Developers

- ◉ **Frontend developers** build how a website *looks*.
- ◉ **Backend developers** build how a website *works*
- ◉ Let's say that you wanted to build a WordPress website for your business.
 - > The frontend developer would create the theme: the images, style, and presentation.
 - > The backend developer may work on managing the database, as well as the site's users, security, and site performance issues.

Frontend Developers

- ◉ A front-end web developer is mainly in charge of
 - > the user interface
 - > the style of the website.
- ◉ The most commonly used languages a front-end Web developer uses are:



- ◉ These three languages are essential to any aspiring front-end web developer and are very important in defining the actual design of a website.

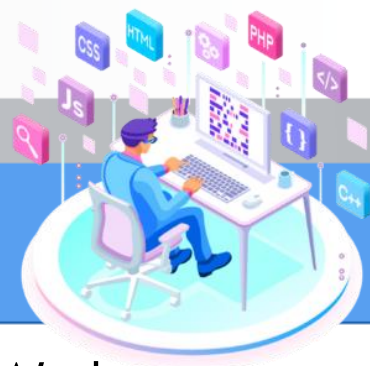
Backend Developers

- ◉ A back-end web developer is mainly in charge of
 - > How things work
 - > Logic and data
- ◉ Need to be proficient in programming languages that render on the server-side of a website or application.
- ◉ The most popular backend programming languages are PHP, Ruby, Python, Node.js, and Java.
- ◉ Also need to be proficient in working with databases like MySQL, Oracle, and SQL Server.



Full Stack Developers

- ◉ A full-stack web developer's role in a website's development lies in both the front-end and Back-End categories.
- ◉ "Stack," means layer.
- ◉ A full-stack web developer has expertise in all layers of a website's development.
- ◉ This includes, but not limited to: the server, client and hosting, a form of data structuring or modelling, user interface and experience, as well as the needs of the actual business.

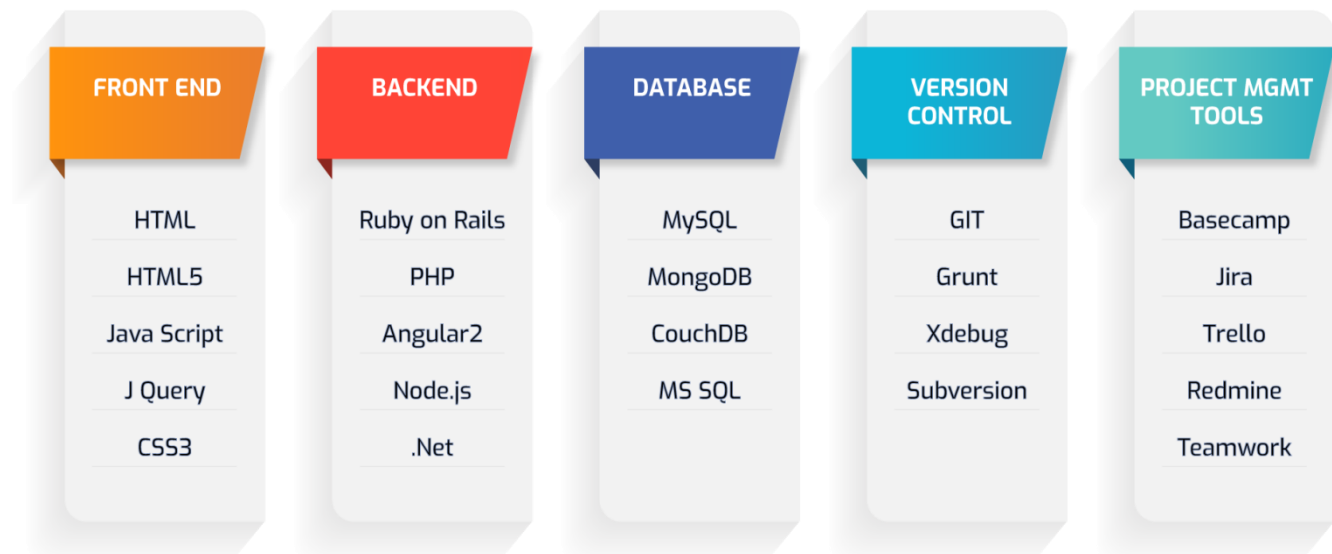


Full Stack Developers

- ◉ key roles and responsibilities of a Full Stack Web Developer:
 - > Should be well acquainted with system infrastructure including hardware and operating systems.
 - > Create and design front end by using HTML, CSS and JavaScript.
 - > Build interactive UIs with the help of JavaScript frameworks, such as, ReactJS, AngularJS, VueJs, Ember, etc.
 - > Code server-side APIs and back end by using programming languages like Python, Ruby, Java, PHP and frameworks like NodeJS.
 - > Manage and operate Relational Database Management Systems like MySQL, MongoDB, SQL, etc.
 - > Must be well versed with Project Management tools and client co-ordination.

Full Stack Developers

- **Full Stack Web Developer Technologies** involves developing and designing the following three layers:
 - > Front-End Layer
 - > Back-End Layer
 - > Database Layer



Web Development Tools and Editors

- ◉ Before you even start learning Web Development, one must choose and download the code editor of your choice.
- ◉ The 5 Best Code Editors For Web Development in are:



Web Development Tools and Editors

◉ Example web design software:

- > Adobe Dreamweaver CC
- > Wix
- > Wordpress
- > Weebly
- > Webflow
- > Bluefish
- > Mobirise



Version Control System (VCS)

- ◉ VCS tracks the history of changes as people and teams collaborate on projects together. As the project evolves, teams can run tests, fix bugs, and contribute new code with the confidence that any version can be recovered at any time. Developers can review project history to find out:
 - > Which changes were made?
 - > Who made the changes?
 - > When were the changes made?
 - > Why were changes needed?

Distributed Version Control System (DVCS)

- ◉ Git is an example of a DVCS commonly used for open source and commercial software development.
- ◉ DVCS allow full access to every file, branch, and iteration of a project, and allows every user access to a full and self-contained history of all changes.
- ◉ Unlike once popular centralized version control systems, DVCSs like Git don't need a constant connection to a central repository. Developers can work anywhere and collaborate asynchronously from any time zone.
- ◉ Without version control, team members are subject to redundant tasks, slower timelines, and multiple copies of a single project.

Git VS GitHub

- Both Git and GitHub give programmers valuable version-control functionality so that they can build ongoing coding projects without being afraid of messing everything up.
- GitHub just takes things a little bit further than Git, offering more functionality and resources, as well as a place online to store and collaborate on projects.

In Simple Terms

Git

is a version control system that lets you manage and keep track of your source code history

Git
Hub

is a cloud-based hosting service that lets you manage Git repositories

Git VS GitHub

Git vs. GitHub comparision

GIT	GITHUB
Installed locally	Hosted in the cloud
First released in 2005	Company launched in 2008
Maintained by The Linux Foundation	Purchased in 2018 by Microsoft
Focused on version control and code sharing	Focused on centralized source code hosting
Primarily a command-line tool	Administered through the web
Provides a desktop interface named Git Gui	Desktop interface named GitHub Desktop
No user management features	Built-in user management
Minimal external tool configuration features	Active marketplace for tool integration
Competes with Mercurial, Subversion, IBM, Rational Team Concert and ClearCase	Competes with Atlassian Bitbucket and GitLab
Open source licensed	Includes a free tier and pay-for-use tiers

