



Google Cloud Platform for Everyone



Day 2: Website Hosting Workshop

What you have learnt



- 01 History of Google Cloud
- 02 IaaS, PaaS and SaaS
- 03 Cloud Digital Leaders learning path
- 04 Writing command lines in a terminal
- 05 Host a Minecraft server on Google Cloud VM instance



Github Page

Link and Details of workshop



Event Page

<https://bit.ly/GCPEDay02>





Google Developer Student Clubs

Speakers



Kyung Kit
Year 1
Mechanical
Engineering
@kkit_03

Dominic Chong
Year 1
Software
Engineering
@dmc_ry



```
Chong = filterByOrg ? study.lead_organization === filterByOrg : true  
let status = filterByStatus ? study.status === filterByStatus : true  
if (matchStatus) {  
    return true;  
}  
function filterStudies({ studies, filterByOrg, filterByStatus }) {  
    return studies.filter(study => (filterByOrg ? study.lead_organization === filterByOrg : true) && (filterByStatus ? study.status === filterByStatus : true))  
}
```



Join at
slido.com
#4699 426

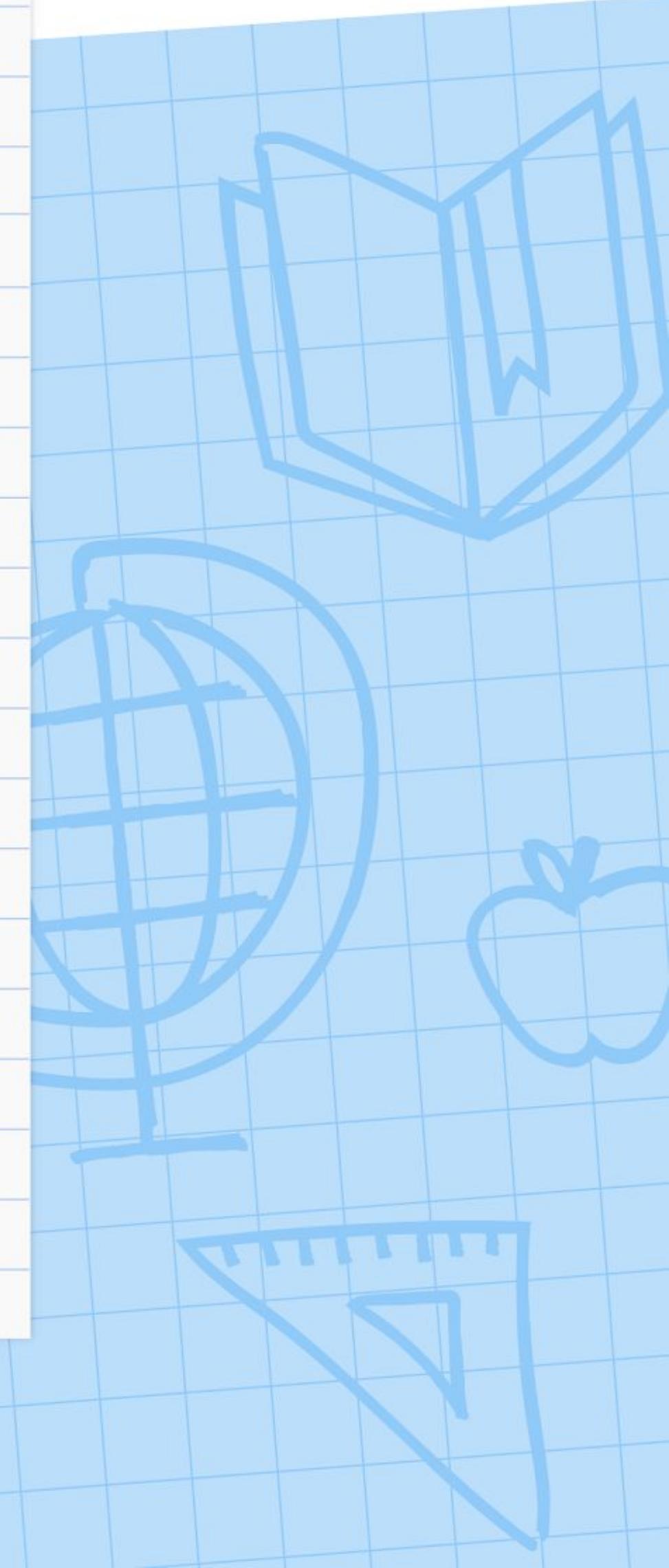




Google Developer Student Clubs

Introduction to HTML

```
filterByOrg = study.lead_organization === filterByOrg .  
filterStatus = filterByStatus ? study.status === filterByStatus : true  
      : !filterStatus) {  
  
function filterStudies({ studies, filterByOrg = "", filterByStatus = false }) {  
  const filteredStudies = studies.filter(study =>  
    filterByOrg || filterStatus || true);  
  return filteredStudies;
```



Introduction to HTML

What is HTML?

- HTML - HyperText Markup Language
- HTML **provides structure** to the content appearing on a website, such as images, text or videos
- Right-click on any page on the internet, choose “Inspect” and you will see HTML in a panel of your screen

Introduction to HTML

Preparing for HTML

- We can let web browsers know that we are using HTML by starting our document with a *document type declaration*
- It tells the browser what [type](#) of document to expect, along with what [version](#) of HTML is being used in the document

```
<!DOCTYPE html>
```



Introduction to HTML

The `<html>` tag

- To create HTML structure and content, we must add opening and closing `<html>` tags after declaring `<!DOCTYPE html>`
- Anything between the opening and closing tags will be interpreted as HTML code

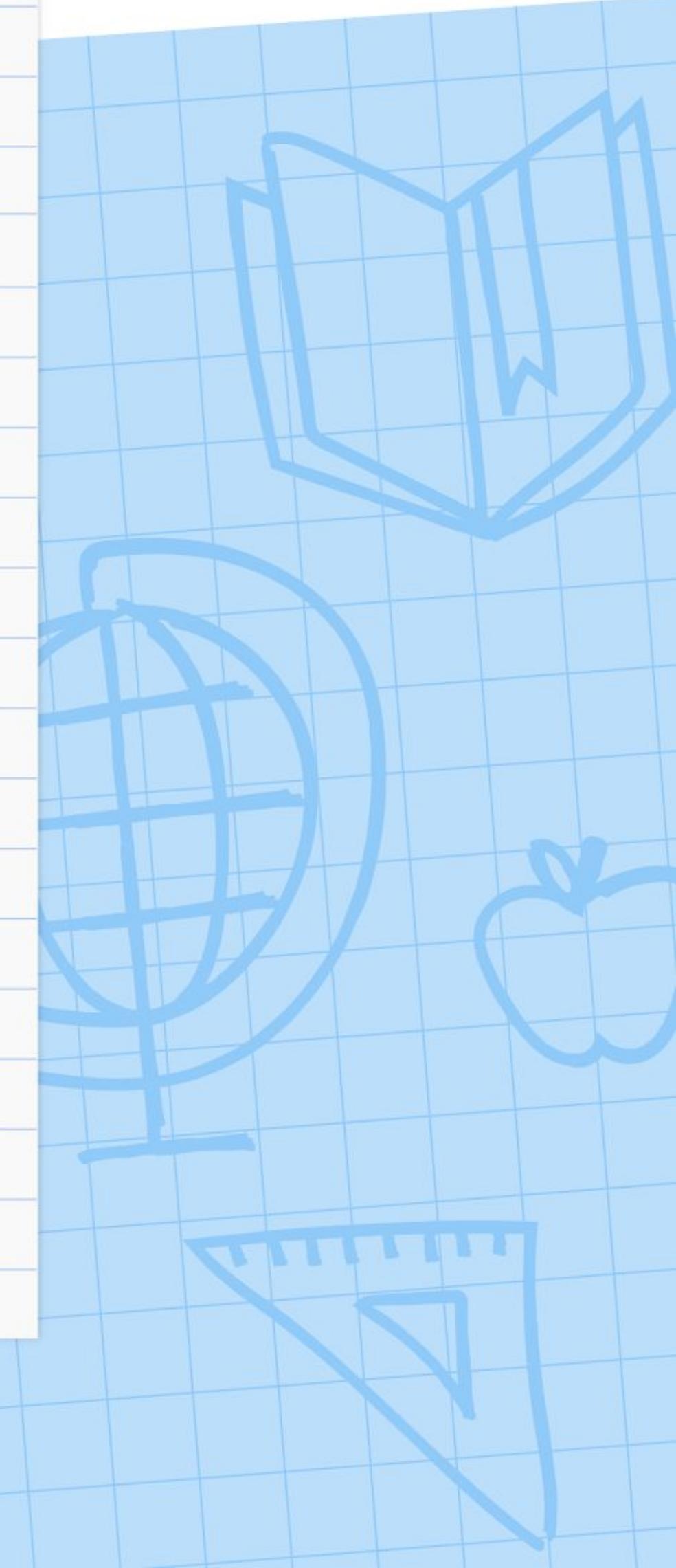




Google Developer Student Clubs

```
<!DOCTYPE html>  
<html>  
  
</html>
```

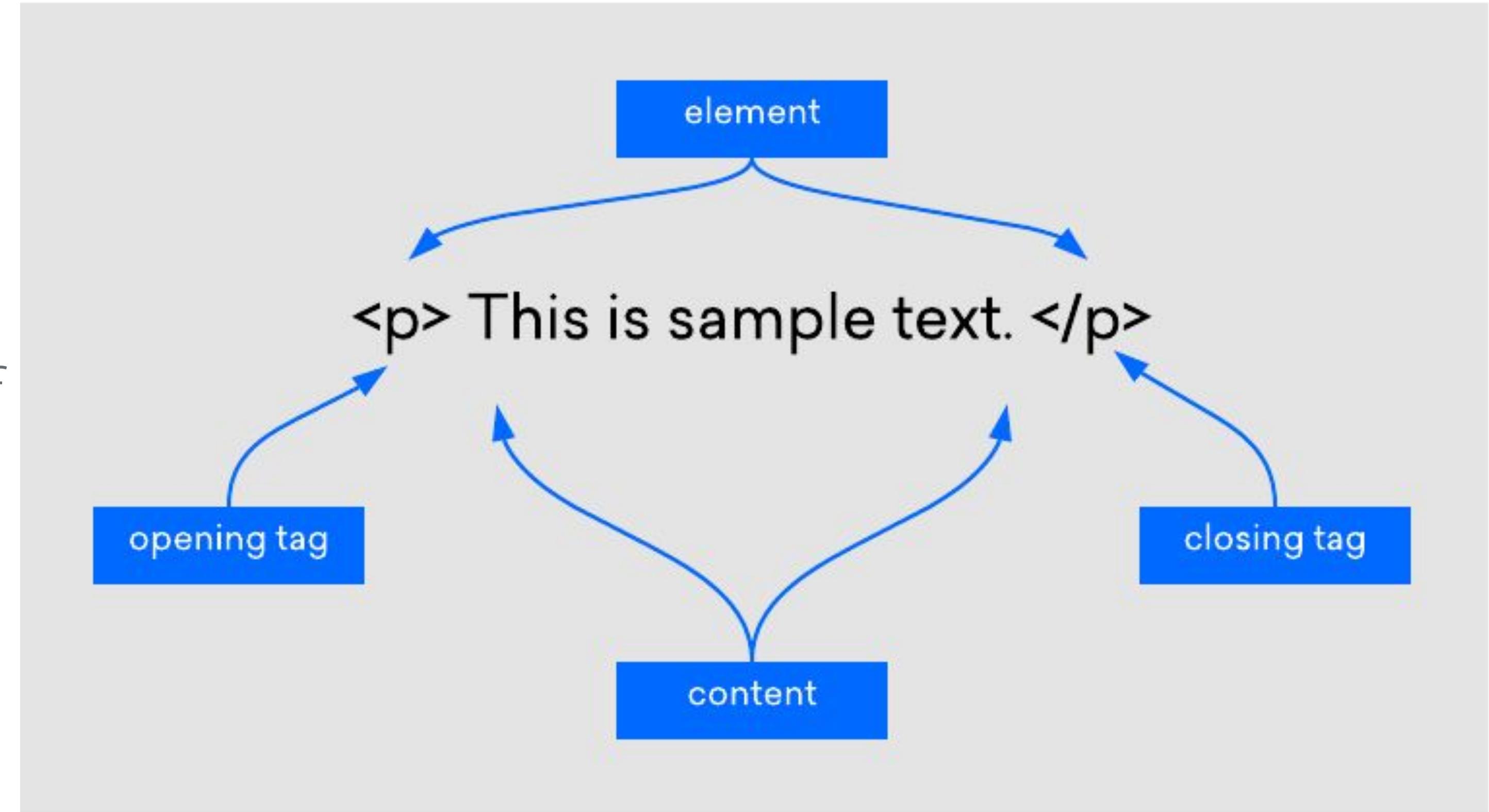
```
    study -> filterByOrg ? study.team_organization === filterByOrg .  
    study -> filterByStatus ? study.status === filterByStatus : true  
    study -> !filterStatus || study.status === filterStatus }  
  
function filterStudies({ studies, filterByOrg = null, filterByStatus = null }) {  
  return studies.filter(study => study -> filterByOrg ? study.team_organization === filterByOrg .  
    study -> filterByStatus ? study.status === filterByStatus : true  
    study -> !filterStatus || study.status === filterStatus )
```



Introduction to HTML

HTML Anatomy

- HTML consists of **elements**
- A tag and the content between it is called an **HTML element**
- The paragraph is made up of an *opening tag*, the *content*, and a *closing tag*



Introduction to HTML

The Head and Page Titles

- We give the browser some information about the page by adding a `<head>` element
- The `<head>` element contains **information** about the page **that is not displayed** directly on the web page
- A browser's tab displays the title specified in the `<title>` tag
- The `<title>` tag is always inside of the `<head>`



```
<!DOCTYPE html>  
  
<html>  
  <head>  
    <title>GDSC Workshop</title>  
  </head>
```

If we were to open a file containing the HTML code in the example above, the browser would display the words **GDSC Workshop** in the tab's title.

Introduction to HTML

The Body and HTML Structure

- One of the key HTML elements used to build a webpage is the *body* element
- HTML is organized as a collection of *family tree relationships*
- When an element is *contained* inside another element, it is considered the *child* of that element

```
<body>  
  <p>Welcome to our workshop!</p>  
</body>
```

Welcome to our workshop!



Introduction to HTML

Headings

- Headings are typically used to **capture a reader's attention** and **describe content**
- There are six different headings in HTML
- They are ordered from largest to smallest in size



Introduction to HTML

Headings

```
<body>
  <h1>Google Developer Student Clubs</h1>
  <h2>Google Developer Student Clubs</h2>
  <h3>Google Developer Student Clubs</h3>
  <h4>Google Developer Student Clubs</h4>
  <h5>Google Developer Student Clubs</h5>
  <h6>Google Developer Student Clubs</h6>
</body>
```

Google Developer Student Clubs

Introduction to HTML

Divs

- <div> is short for “division” – it divides a page into sections
- These sections are very useful for grouping elements in your HTML together
- <div>s are very useful when we want to apply custom styles to our HTML elements

```
<div id="intro">  
  <h1>Introduction</h1>  
</div>
```



Introduction to HTML

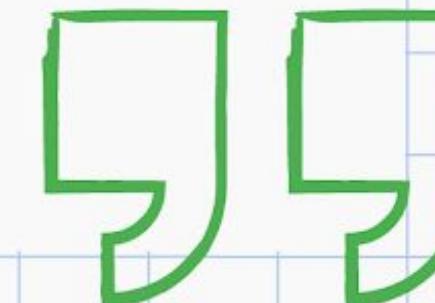
Attributes

- If we want to **expand an element's tag**, we can do so by using an attribute
- Attributes are content added to the **opening tag** of an element
- It can be used in several different ways, from **providing information** to **changing styling**
- Attributes are made up of two parts – *name, value*



Attributes

```
<div id="intro">  
  <h1>Introduction</h1>  
</div>
```



Introduction to HTML

Styling Text

- You can also [style text](#) using HTML tags
- The `` tag renders as *italic* emphasis
- The `` tag renders as **bold** emphasis

```
<p><strong>Google Developer Student Clubs</strong> is an  
<em>amazing</em> community of students representing hundreds  
of colleges and universities across the globe.</p>
```

Google Developer Student Clubs is an *amazing* community of students representing hundreds of colleges and universities across the globe.

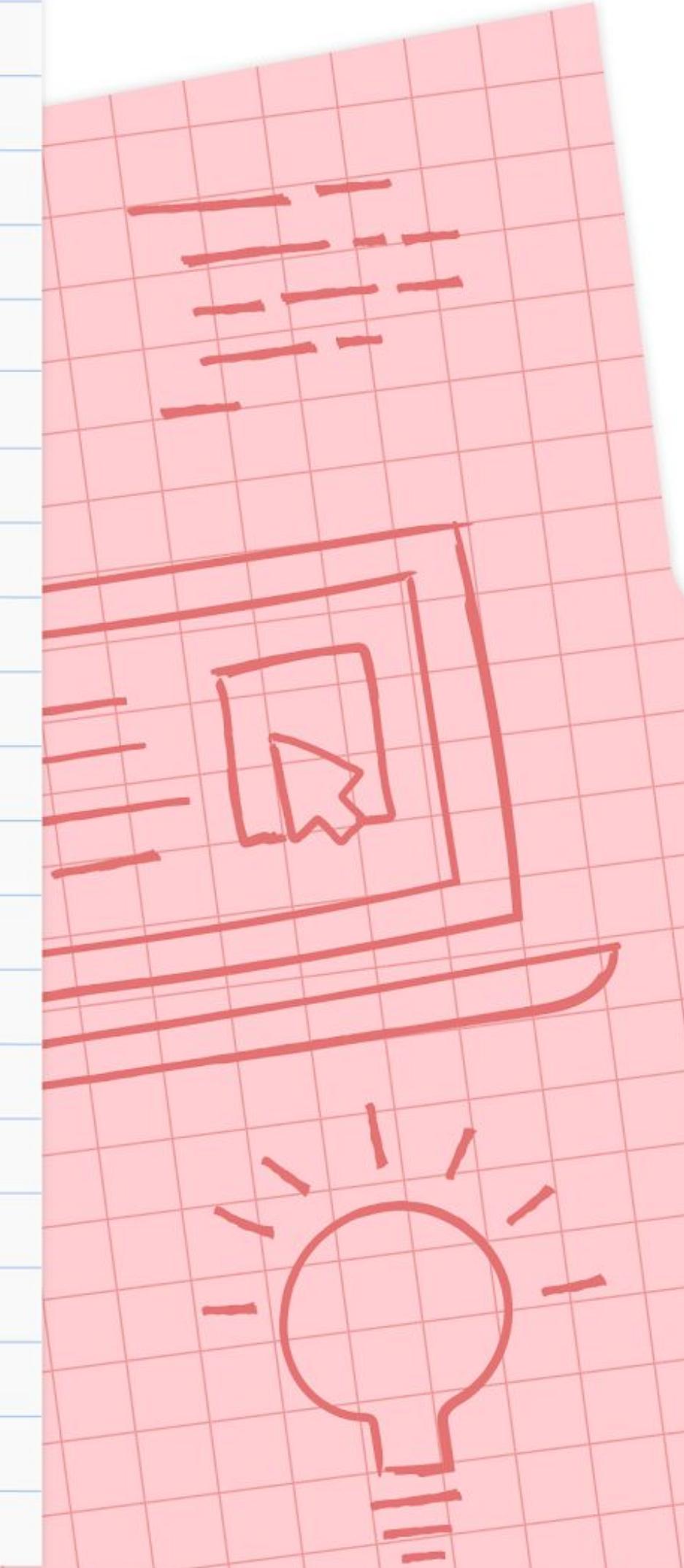
Introduction to HTML

Line Breaks

- You can use HTML's line break element `
` to modify the spacing in the browser
- The line break element is **unique** because it is **only** composed of a starting tag

```
<p>University Malaya, <br> 50603 Kuala Lumpur, <br> Federal  
Territory of Kuala Lumpur</p>
```

University Malaya,
50603 Kuala Lumpur,
Federal Territory of Kuala Lumpur



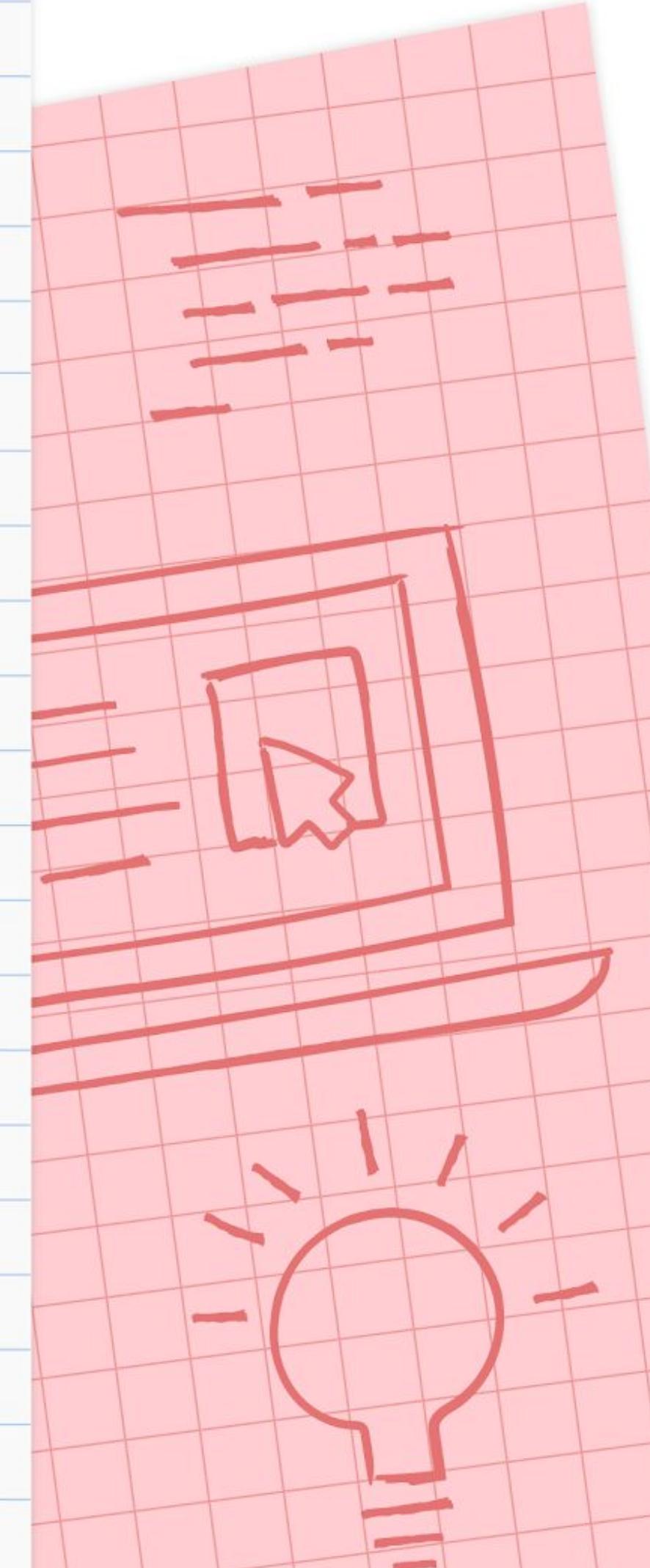
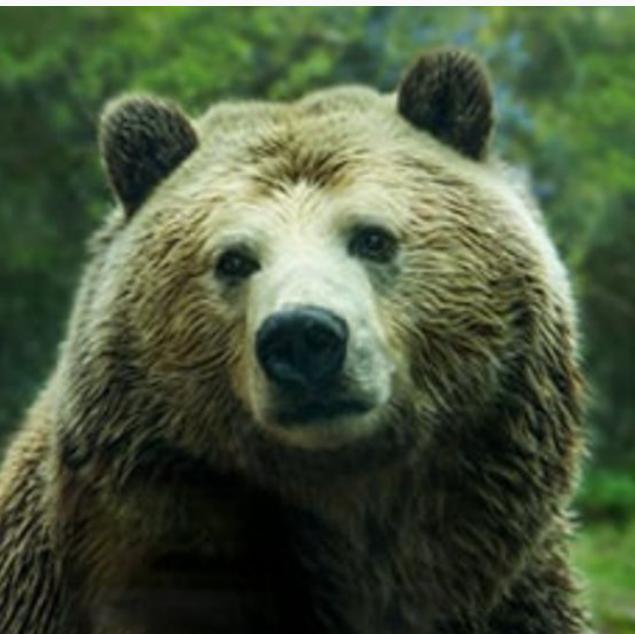
Introduction to HTML

Images

- The `` tag allows you to add an image to a web page
- The `` tag is a self-closing tag
- The `` tag has a required attribute called `src`
- The attribute must be set to the URL of the image

```

```

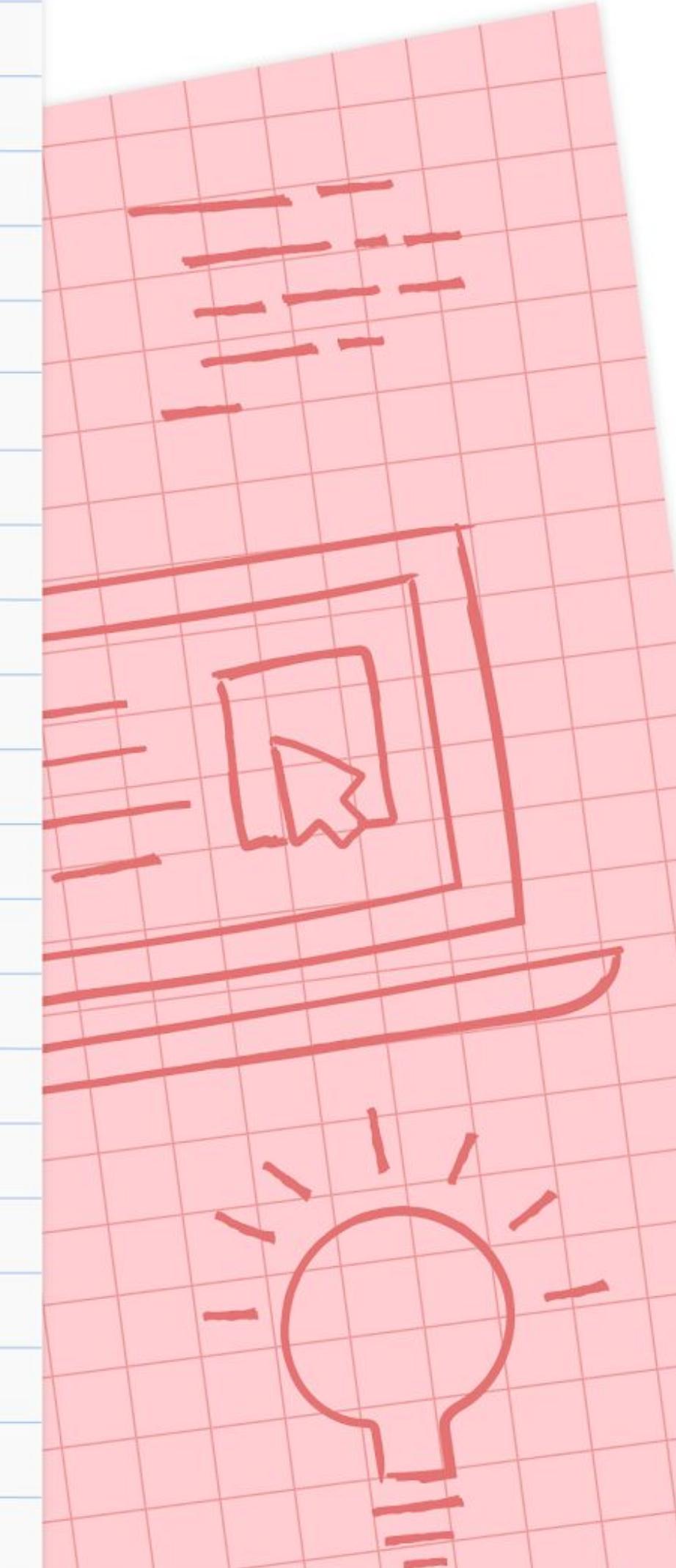
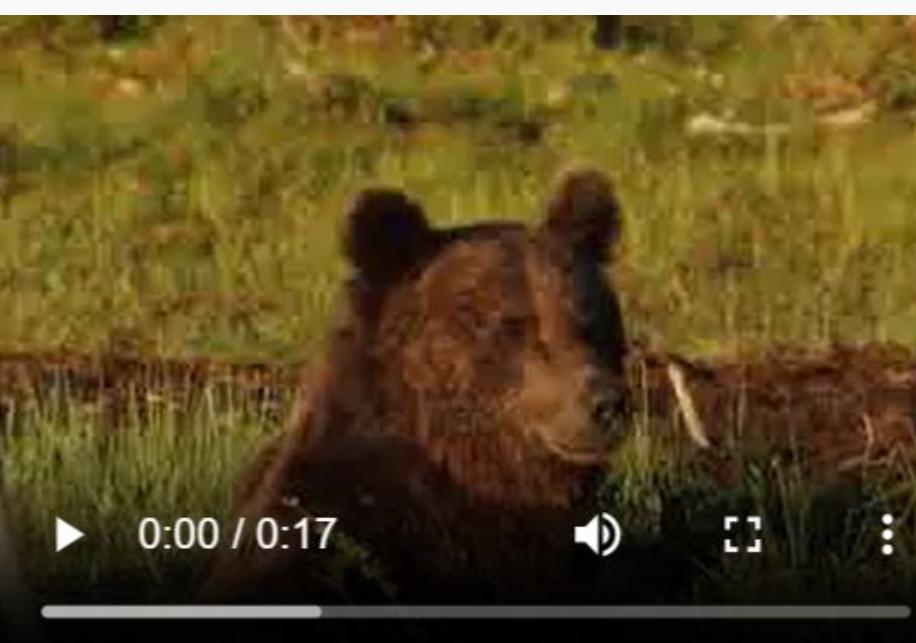


Introduction to HTML

Videos

- The `<video>` element requires a `src` attribute with a link to the video source
- The `controls` attribute instructs the browser to include basic video controls such as pausing and playing

```
<video src="vid_brown-bear.mp4" height="240"  
width="320" controls>Video not supported</video>
```



HTML Document Standards

Linking to Other Web Pages

- You can **add links** to a web page by adding an anchor element `<a>` and including the text of the link in between the opening and closing tags with the `href` attribute

```
<a href="https://en.wikipedia.org/wiki/  
Brown_bear">Learn More</a>
```

[Learn More](#)



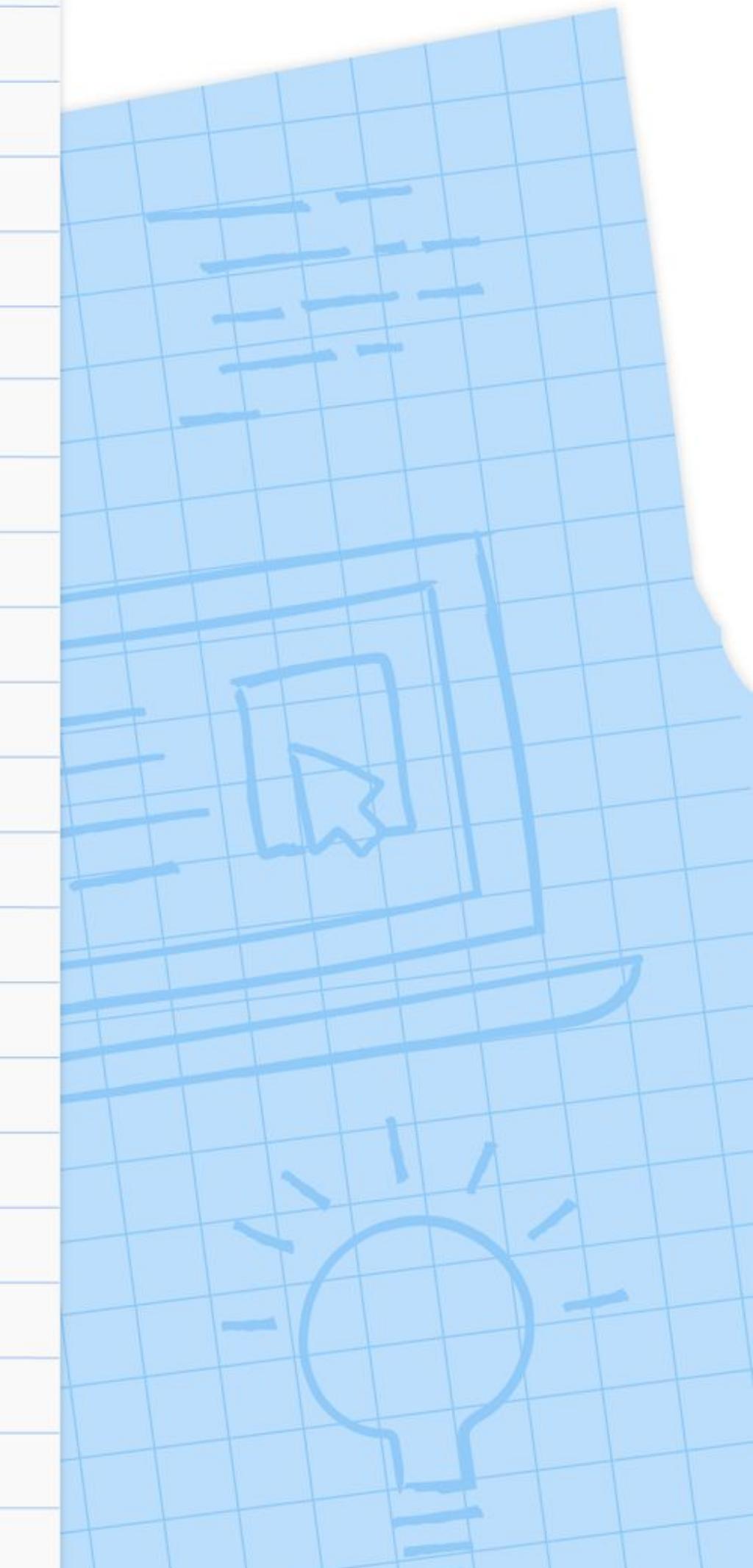
HTML Document Standards

Opening Links in a New Window

- The *target* attribute specifies how a link should open
- For a link to **open in a new window**, the *target* attribute requires a value of `_blank`

```
<a href="https://en.wikipedia.org/wiki/Brown_bear"  
target=_blank>Learn More</a>
```

[Learn More](#)



HTML Document Standards

Comments

- HTML files also allow us to add comments to our code
- Comments begin with `<!--` and end with `-->`
- Any **characters in between** will be **ignored** by your browser.

```
<!-- This is a comment -->
```



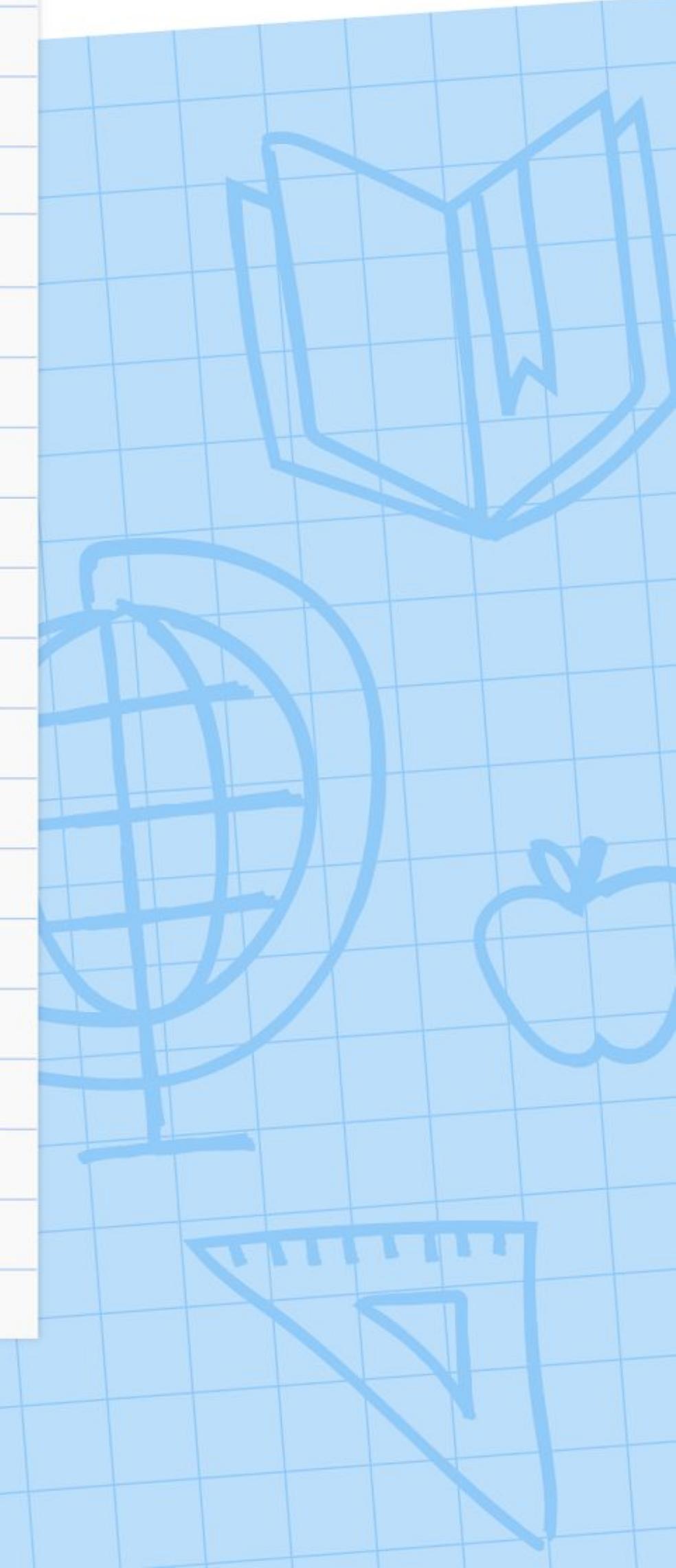


Google Developer Student Clubs

Introduction to CSS

```
const filterByOrg = study => study.lead_organization === filterByOrg;
const filterStatus = filterByStatus ? study.status === filterByStatus : true;
const filterPatchStatus = patchStatus ? study.patch_status === patchStatus : true;
```

```
function filterStudies({ studies, filterByOrg = null, filterByStatus = null, filterPatchStatus = null }) {
  const filteredStudies = studies.filter(study => {
    if (filterByOrg) {
      return filterByOrg(study);
    }
    if (filterStatus) {
      return filterStatus(study);
    }
    if (filterPatchStatus) {
      return filterPatchStatus(study);
    }
    return true;
  });
  return filteredStudies;
}
```



Setup and Syntax

Intro to CSS

- While HTML is the fundamental structure of every web page, it can be visually unappealing on its own
- **CSS** is a language web developers use to **style the HTML content** on a web page



Setup and Syntax

Inline Styles

- To style an HTML element, you can add the *style* attribute directly to the opening tag
- If you'd like to add more than one style with **inline styles**, simply keep adding to the style attribute

```
<p style='color: red; font-size: 25px;'> Introduction to CSS!</p>
```

Introduction to CSS!

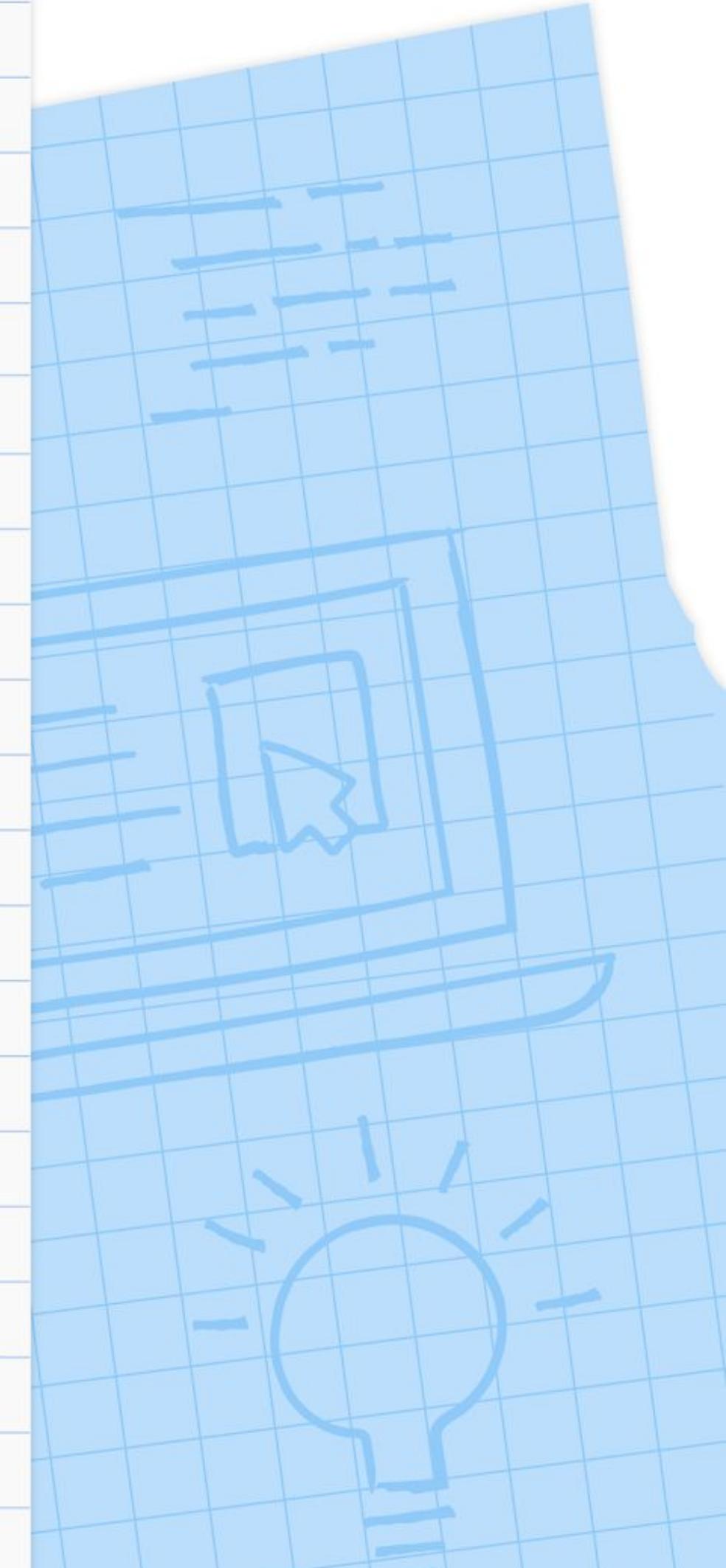


Setup and Syntax

Internal Stylesheet

- HTML allows you to write CSS code in its own dedicated section with a `<style>` element **nested inside** of the `<head>` element

```
<head>
  <style>
    p {
      color: red;
      font-size: 20px;
    }
  </style>
</head>
```



Setup and Syntax

External Stylesheet

- Developers avoid mixing code by storing HTML and CSS code in separate files
- With an **external stylesheet**, you can write all the CSS code needed to style a page **without sacrificing the readability and maintainability** of your HTML file



Setup and Syntax

Linking the CSS File

- You can use the `<link>` element to link HTML and CSS files together
- The `<link>` element must be placed within the head of the HTML file
- `href` - the value of this must be the address
- `rel` - this attribute describes the relationship between the HTML file and the CSS file

```
<head>
  <link href='style.css' rel='stylesheet'>
  <title>Coding is fun!</title>
</head>
```





Google Developer Student Clubs

Selectors

```
const filterByOrg = study.lead_organization === filterByOrg ? true : false
const filterByStatus = filterByStatus ? study.status === filterByStatus : true
const matchStatus = filterByStatus ? filterByStatus : matchStatus

function filterStudies({ studies, filterByOrg, filterByStatus, matchStatus }) {
  return studies.filter(study => filterByOrg || study.lead_organization === filterByOrg || !filterByOrg)
}
```



Selectors

Type

- A selector is used to target the specific HTML element(s) to be styled by the declaration
- The **type** selector matches the type of the element in the HTML document

```
p {  
    color: blue;  
}
```



Selectors

Universal

- The **universal** selector selects all elements of any type

```
* {  
  color: magenta;  
}
```

In the code to the left, every text element on the page will have its font changed to *magenta*

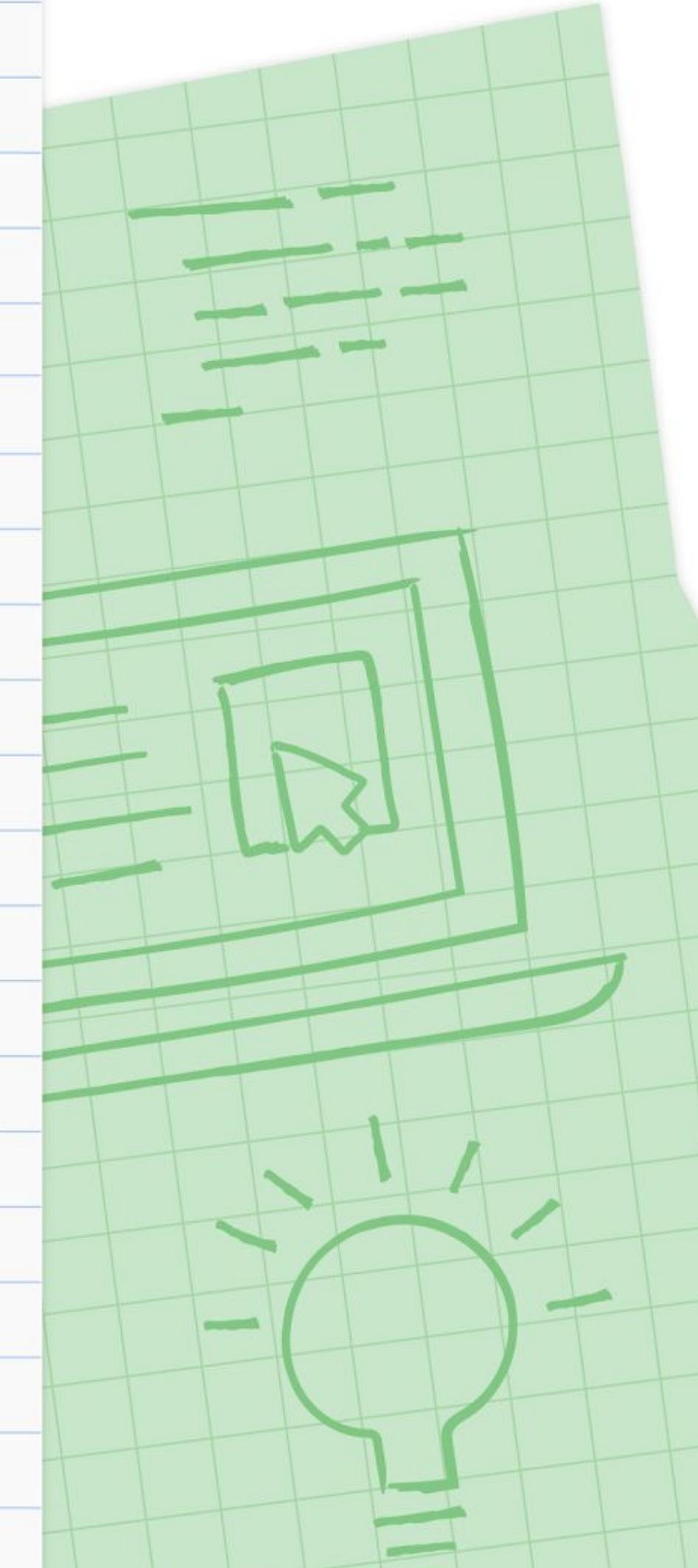


Selectors

Class

- When working with HTML and CSS, a *class* attribute is one of the **most common** ways to select an element
- The paragraph element below has a *class* attribute within the opening tag of the <p> element
- The *class* attribute is set to 'type'

```
<p class='apple'>This is a class with the value "apple"</p>
```



- To select this element using CSS, we can create a ruleset with a class selector of `.apple`
- To select an HTML element by its class using CSS, a period `(.)` must be prepended to the class's name

```
<p class='apple'>This is a class with the value "apple"</p>
```

```
.apple {  
}
```

Selectors

ID

- If an HTML element needs to be [styled uniquely](#), we can give it an ID using the *id* attribute, in contrast to class which accepts multiple values
- To select an element's ID with CSS, we prepend the id name with a **number sign (#)**
- The id name is **large-title**

```
#large-title {  
}
```



Selectors

Specificity

- Specificity is the order by which the browser decides which CSS styles will be displayed
- IDs are the **most specific** selector in CSS, followed by **classes**, and finally, type **[ID > class > type]**
- A best practice in CSS is to **style elements** while **using the lowest degree of specificity** so that if an element needs a new style, it is easy to override



Website Hosting



Website Hosting

Definition

Website Hosting is an online service that allows you to publish your website files onto the internet.

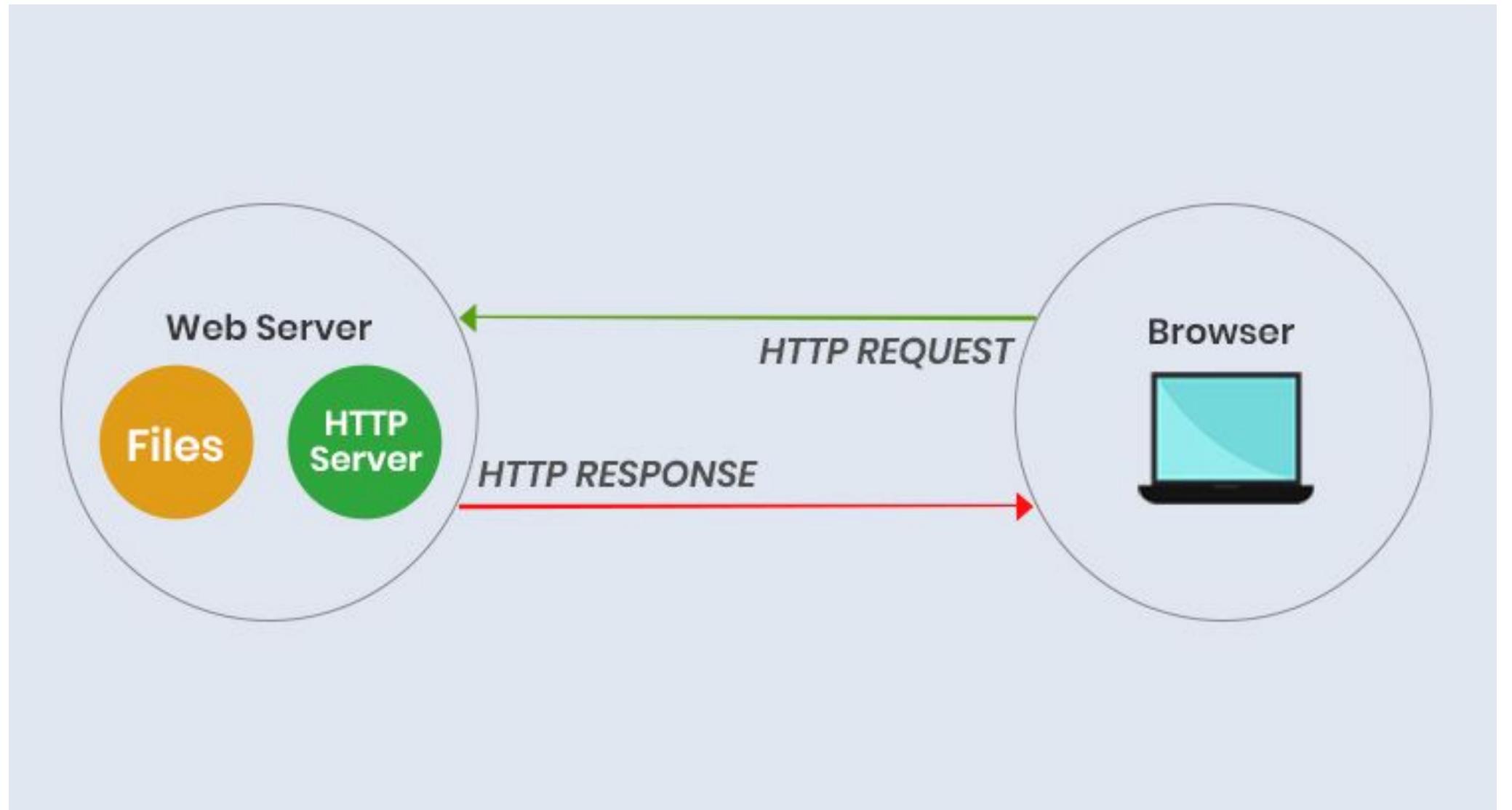
Anyone having access to Internet can access the website.



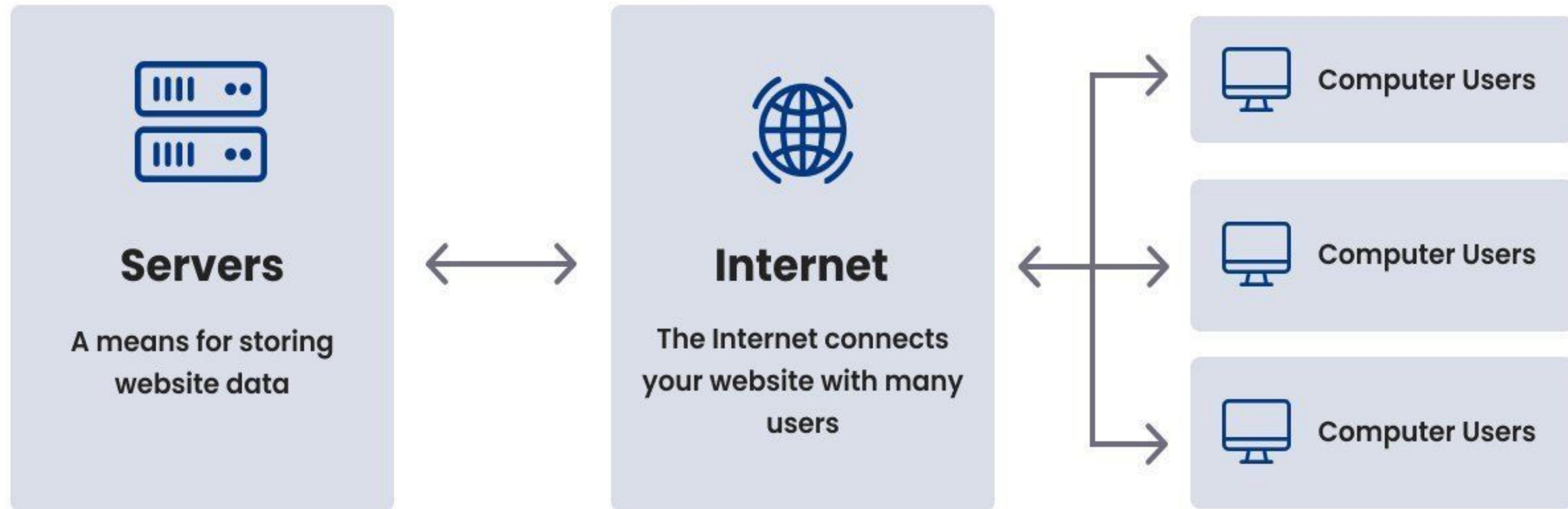
Server

A computer program or device that provides a service to another computer program and its user, also known as the client

- Can be physical or virtual server



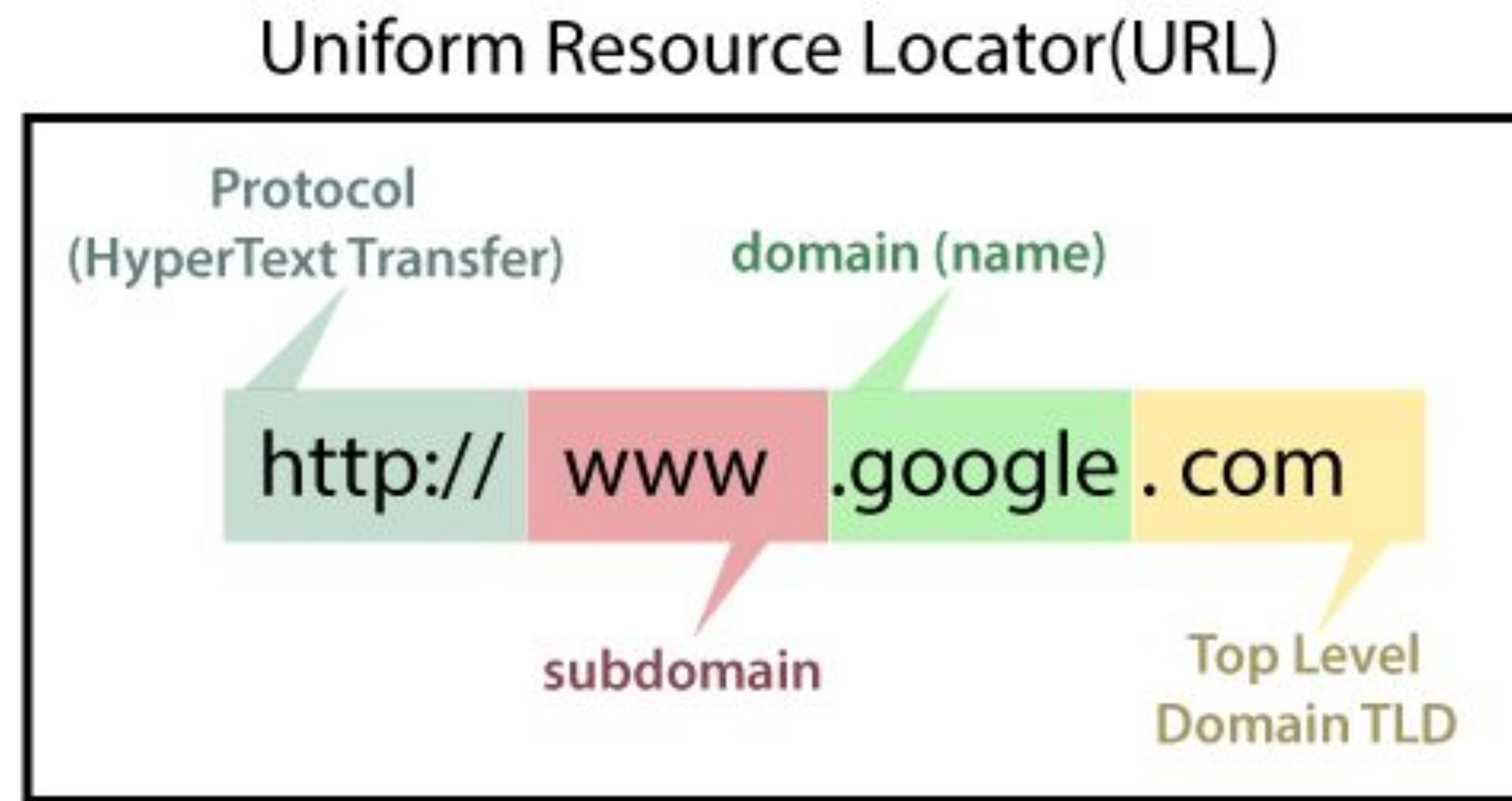
How does web hosting work?



Domain Name

A domain name is a unique name used to access the website

- It shows the address of a website on the Internet

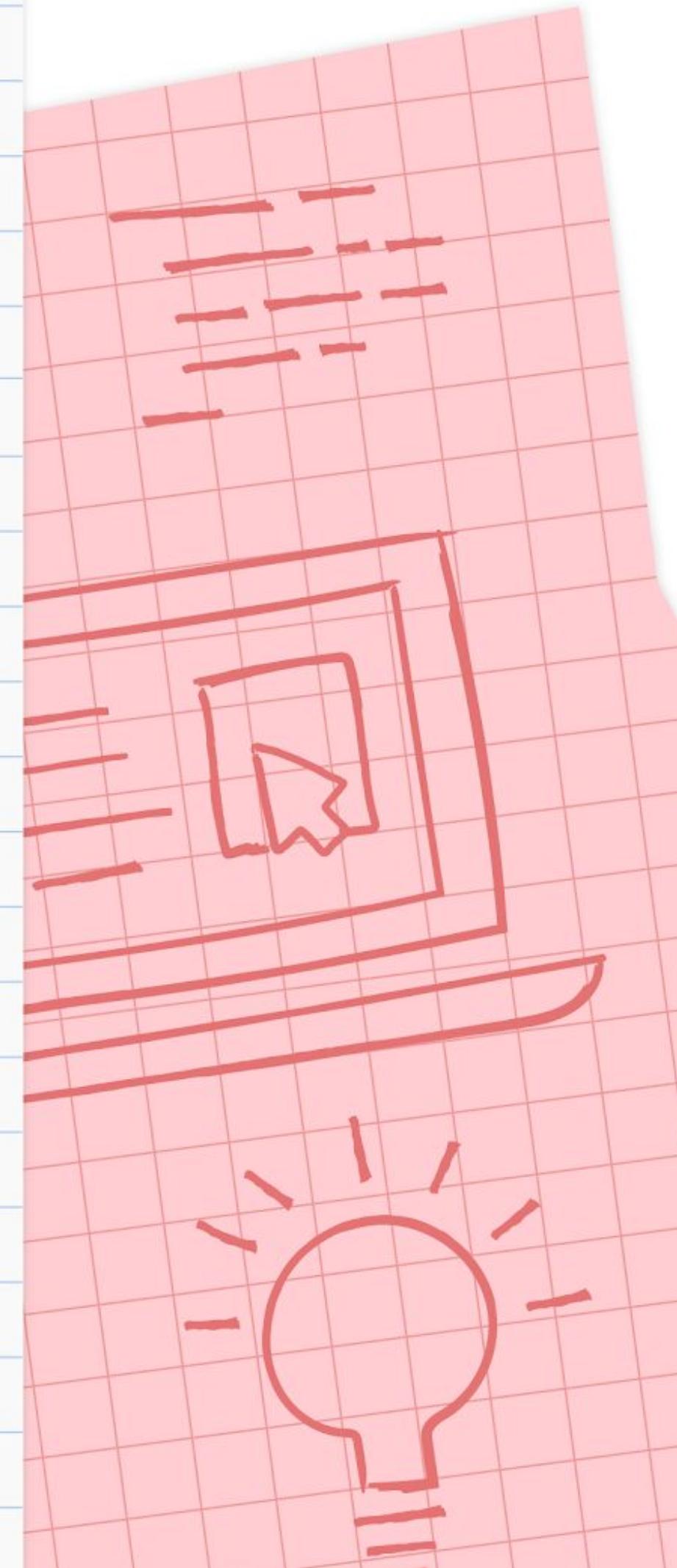


Hosting Service Provider

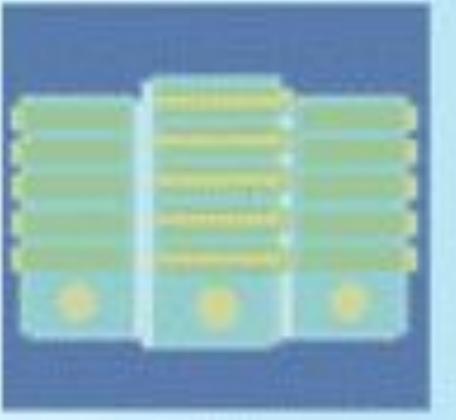
Steps to Consider

1. What type of website are you building? eCommerce, blog, portfolio, etc.
2. Based on the type website, what is the bandwidth needed to run your site?
3. Can you create email addresses for your domain?
4. What type of hosting options are available?
5. Do they provide SSL Certificates?

Types of Website Hosting



VPS Hosting



Shared Hosting



Web Hosting

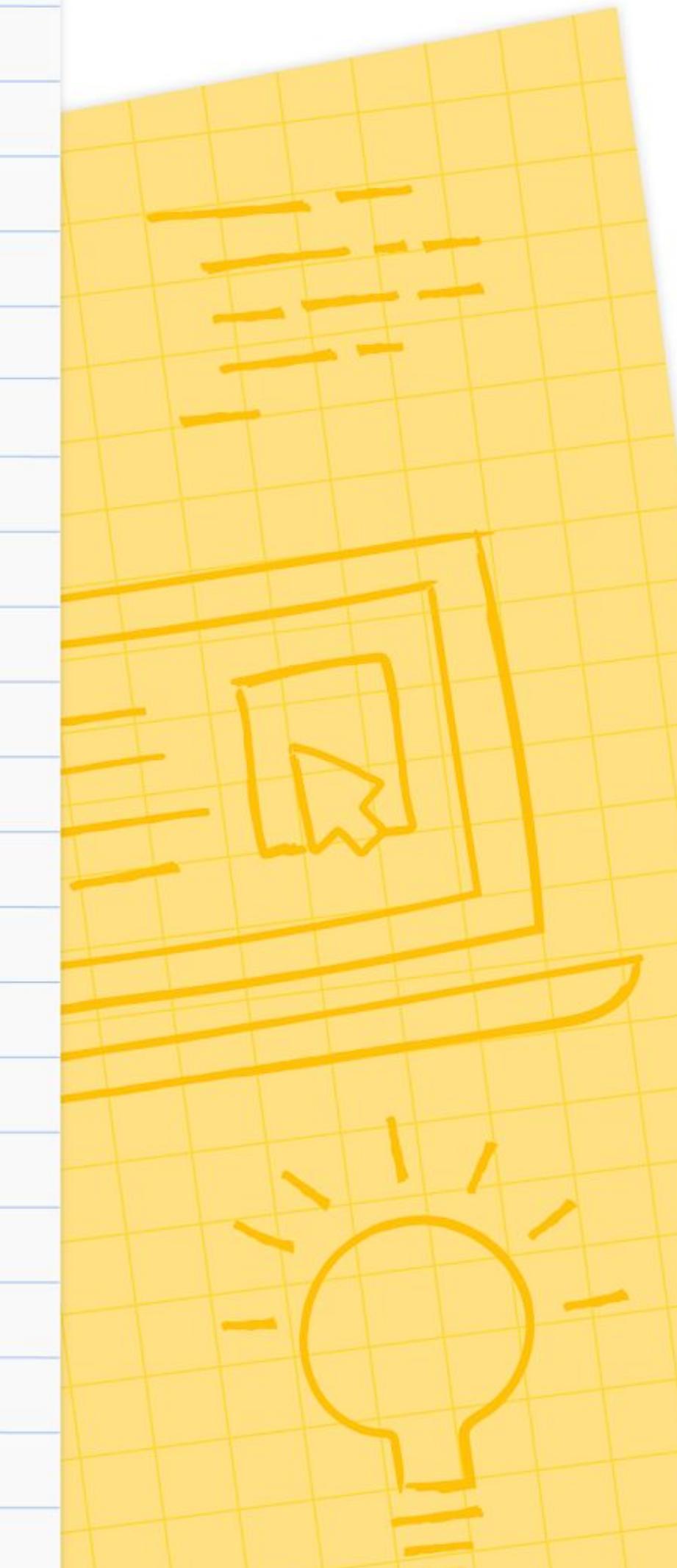


Word-press hosting

**Dedicated
Server Hosting**

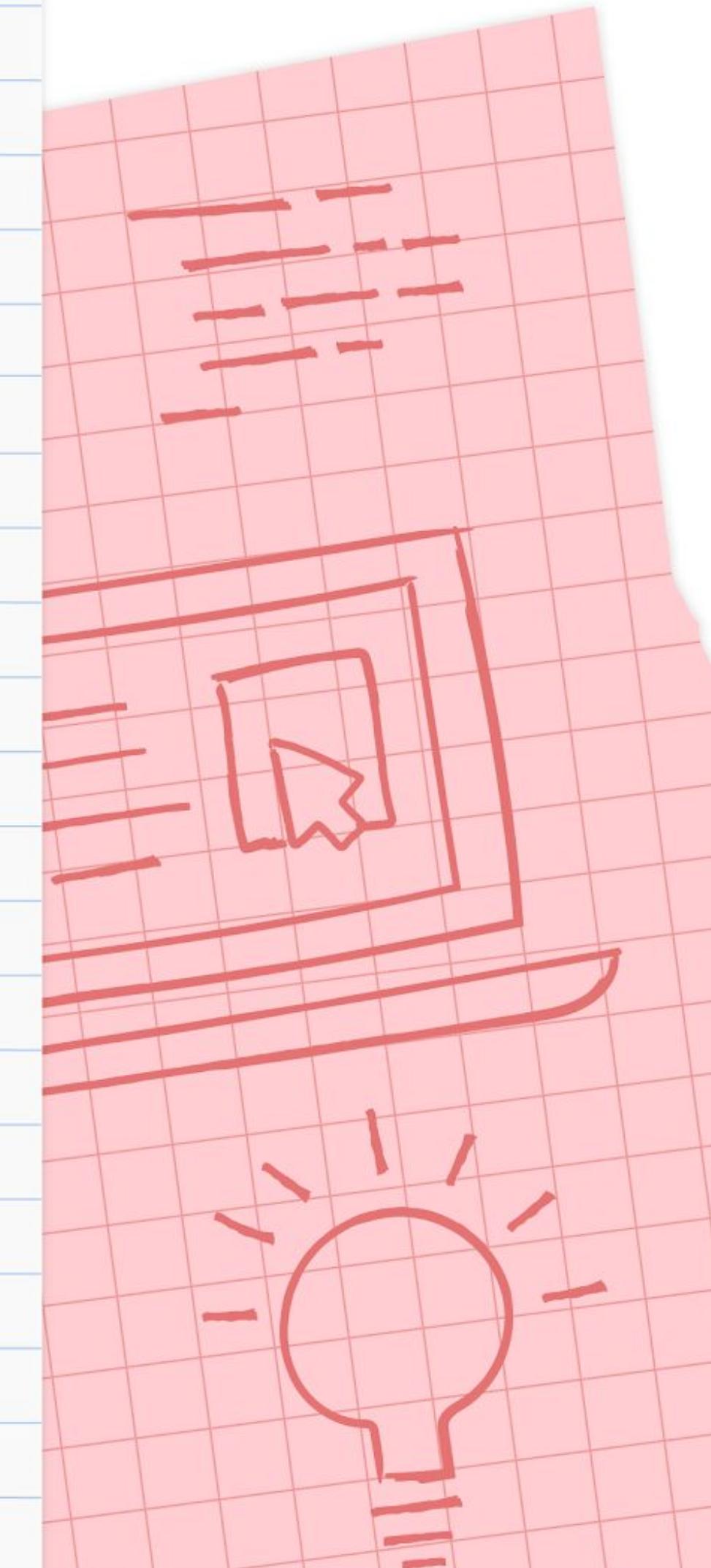
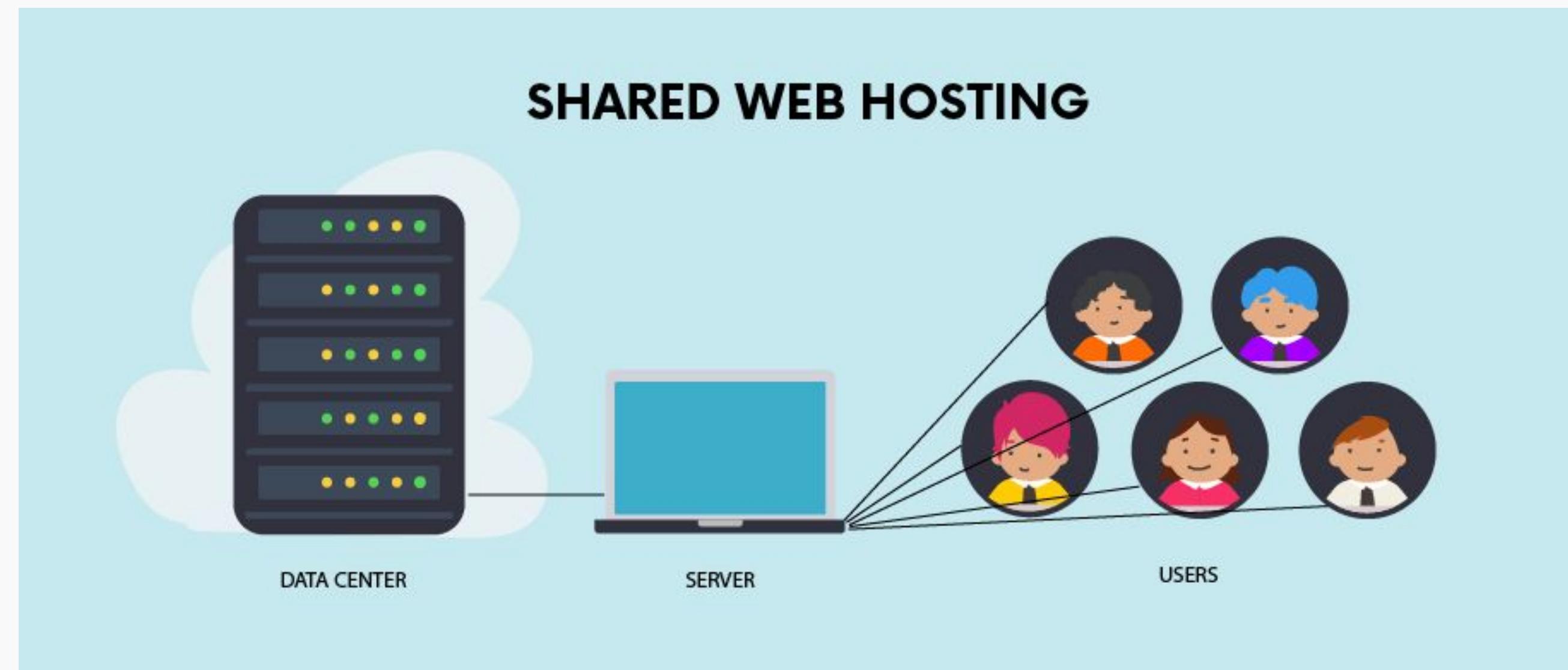


Cloud Hosting



1. Shared Hosting

Multiple users share the same server resources, including memory, processing power, and storage space.



Shared Hosting

Allows multiple websites to utilize a single server

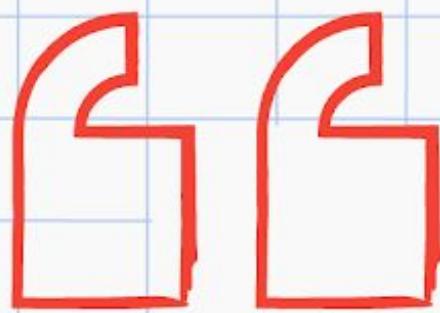
Characteristics

- Cost-effective, ideal for small-scale websites
- Technical expertise is not required
- Pre-configured server options
- No need to take care of maintenance and server administration

Disadvantage

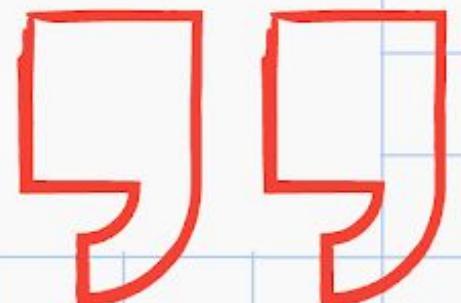
- Minimal access to server configuration
- Increased traffic on other websites can affect your website's speed





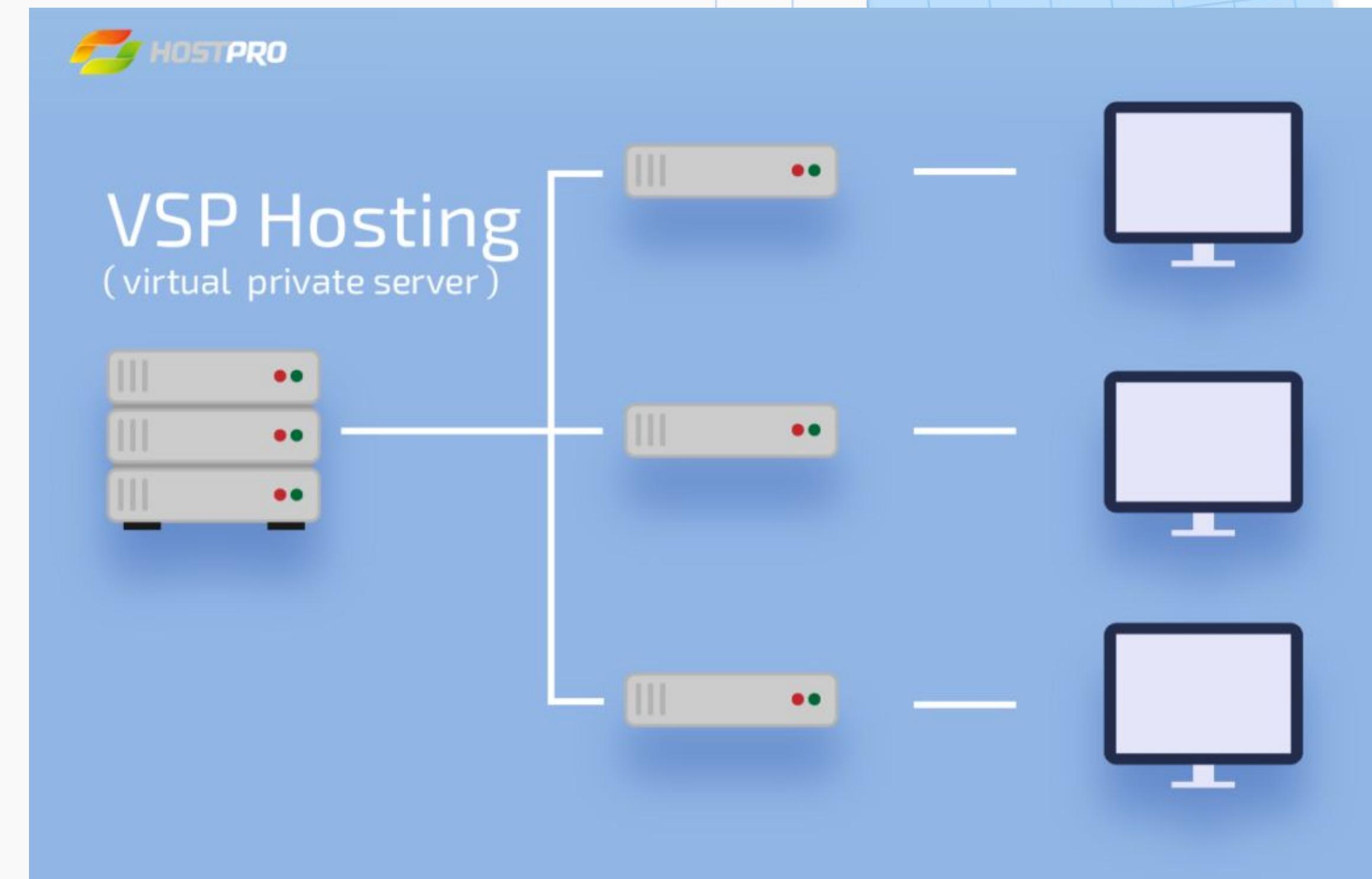
Shared Hosting

Examples: BlueHost, Hostinger



2. Virtual Private Server (VPS) Hosting

Provides virtualized server resources on a physical server that is shared with other users.



Virtual Private Server (VPS) Hosting

Characteristics

- Dedicated server space
- Increased traffic on other websites has no impact on your site's performance
- High customizability

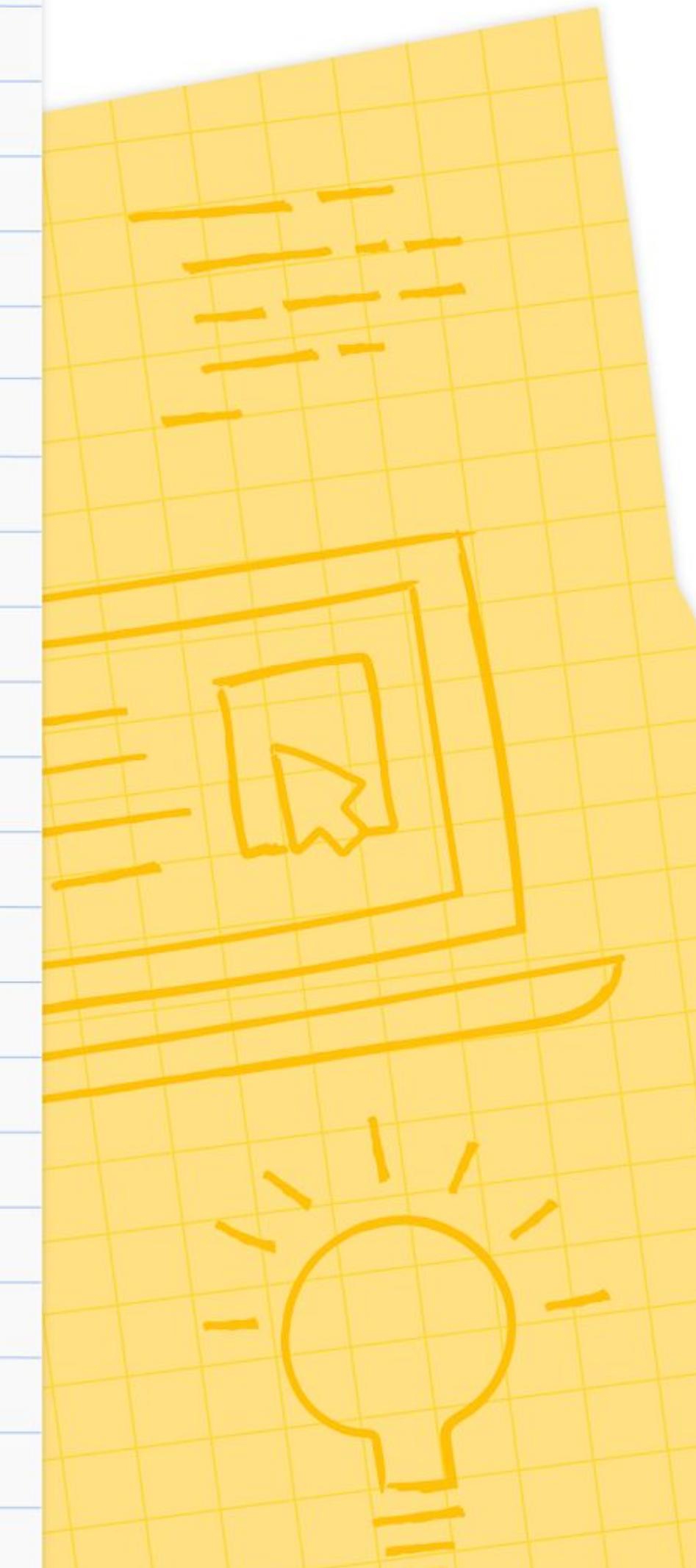
Disadvantages

- Users need technical expertise to manage it
- Even though it's relatively affordable, some users may have to hire a developer to manage the virtual server, increasing the overall costs



3. WordPress Hosting

Provides servers and resources specifically optimized for WordPress websites.



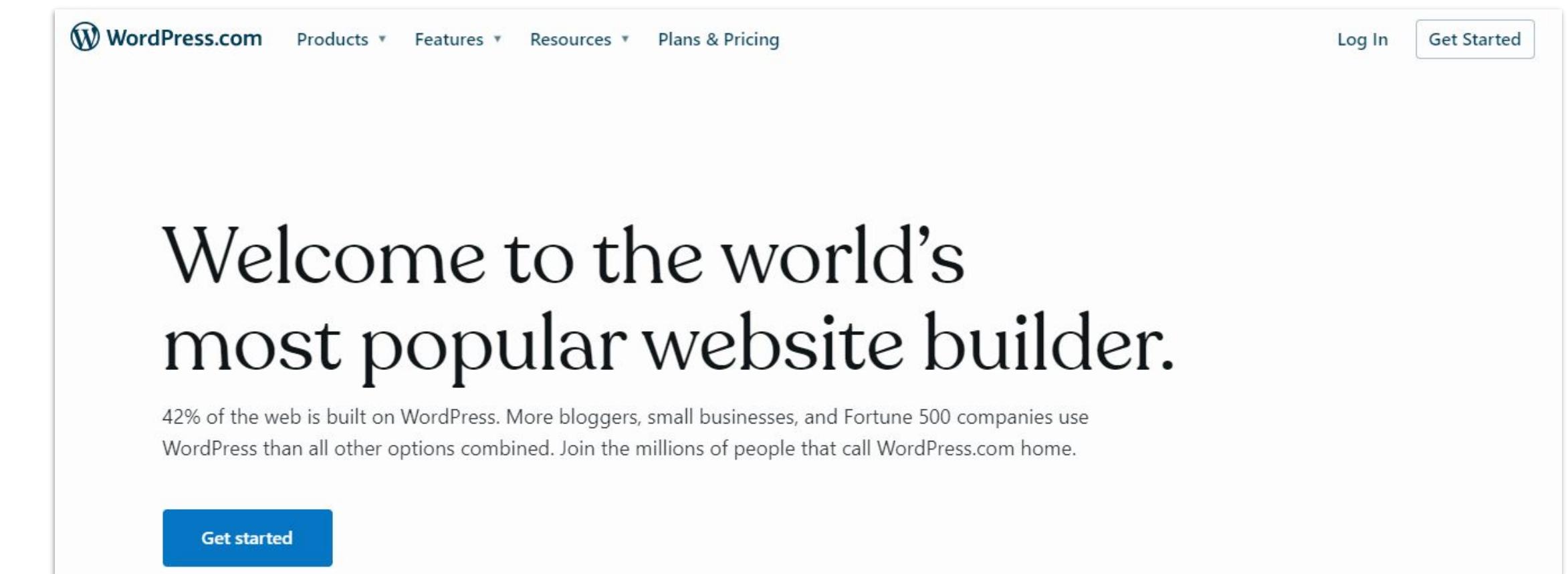
WordPress Hosting

Characteristics

- Low cost and beginner-friendly
- Optimized performance for WordPress sites
- Customer support team trained in WordPress issues
- Pre-installed WordPress plugins and themes

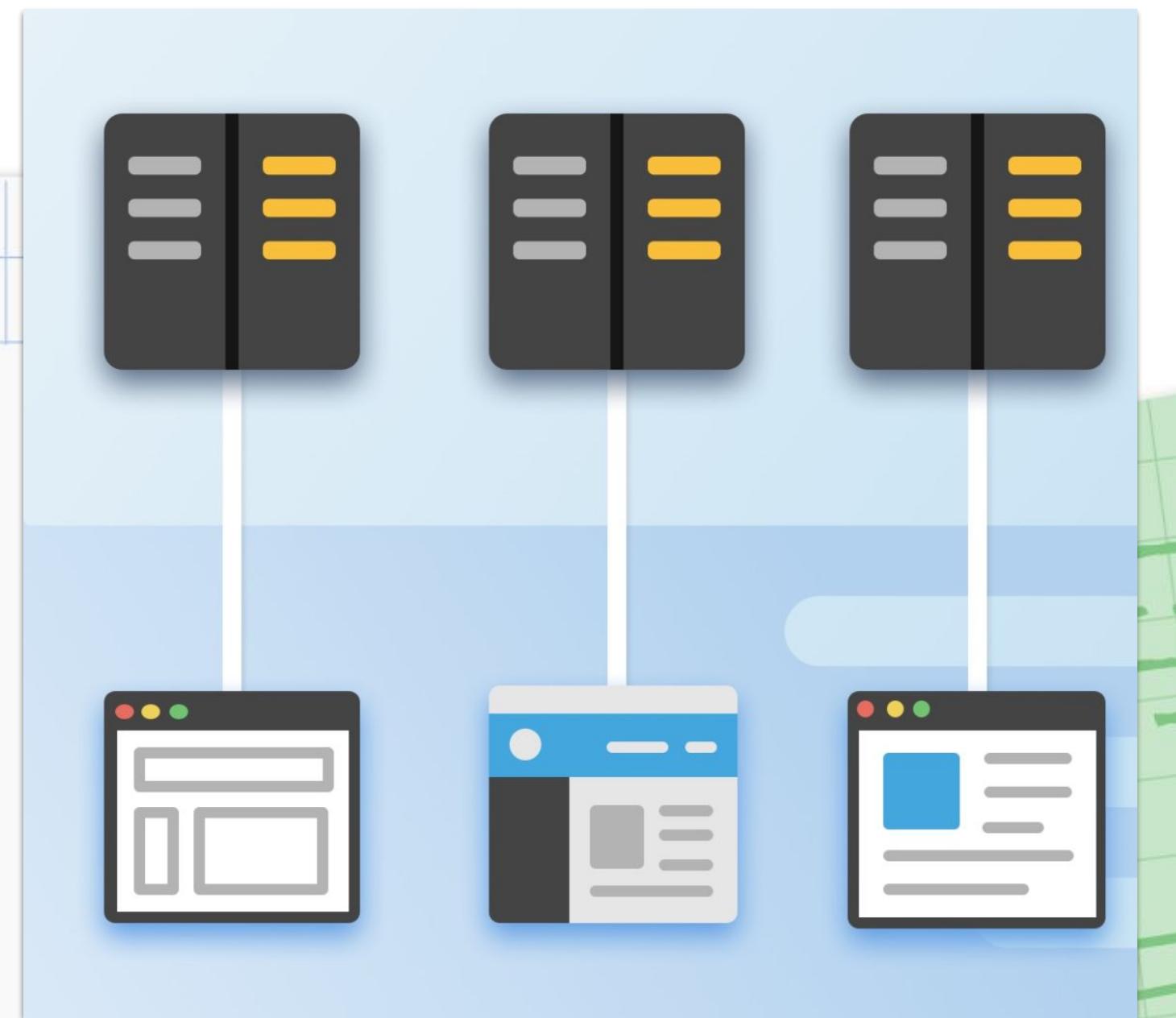
Example: WordPress

But not an ideal type of web hosting for non-WordPress websites



4. Dedicated Hosting

Internet hosting in which a physical server is dedicated to a single client or user.



Dedicated Hosting

Characteristics

- Complete control over the server's configuration
- High reliability

Disadvantage

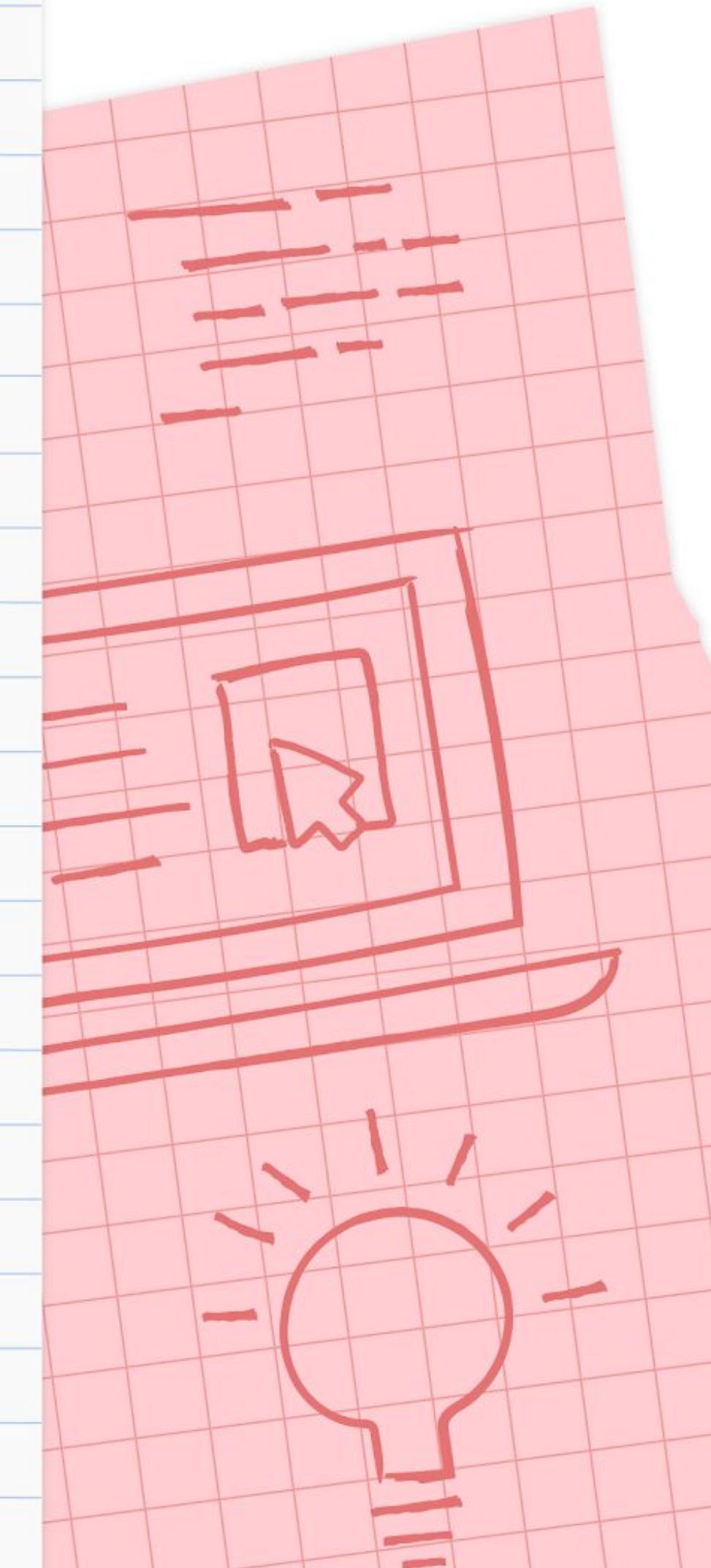
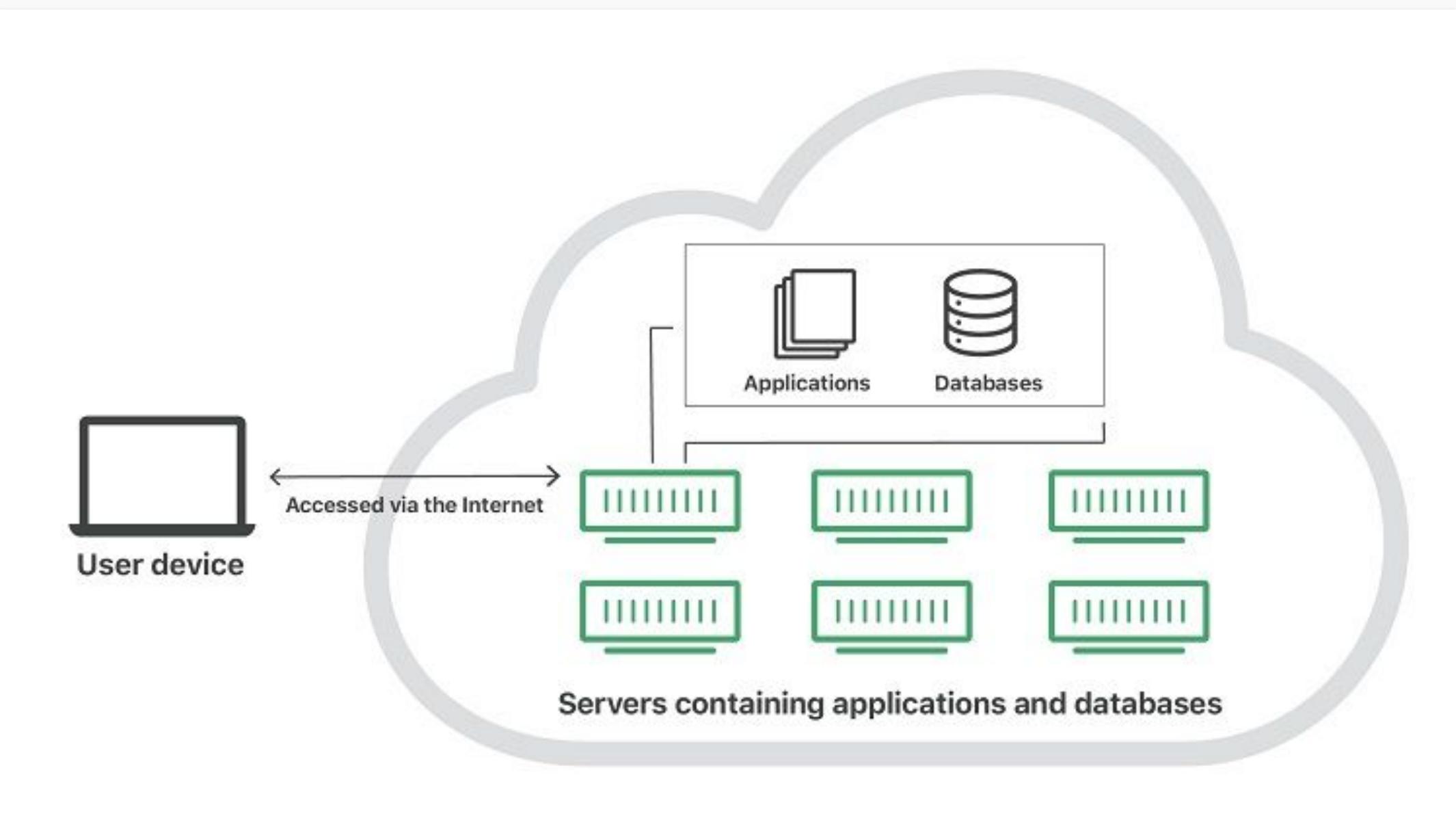
- High cost, more oriented towards large businesses
- Technical and server management knowledge is required

Examples: Dedicated server rented or purchased physically



5. Cloud Hosting

Applications and websites accessible using cloud resources



Cloud Hosting

Characteristics

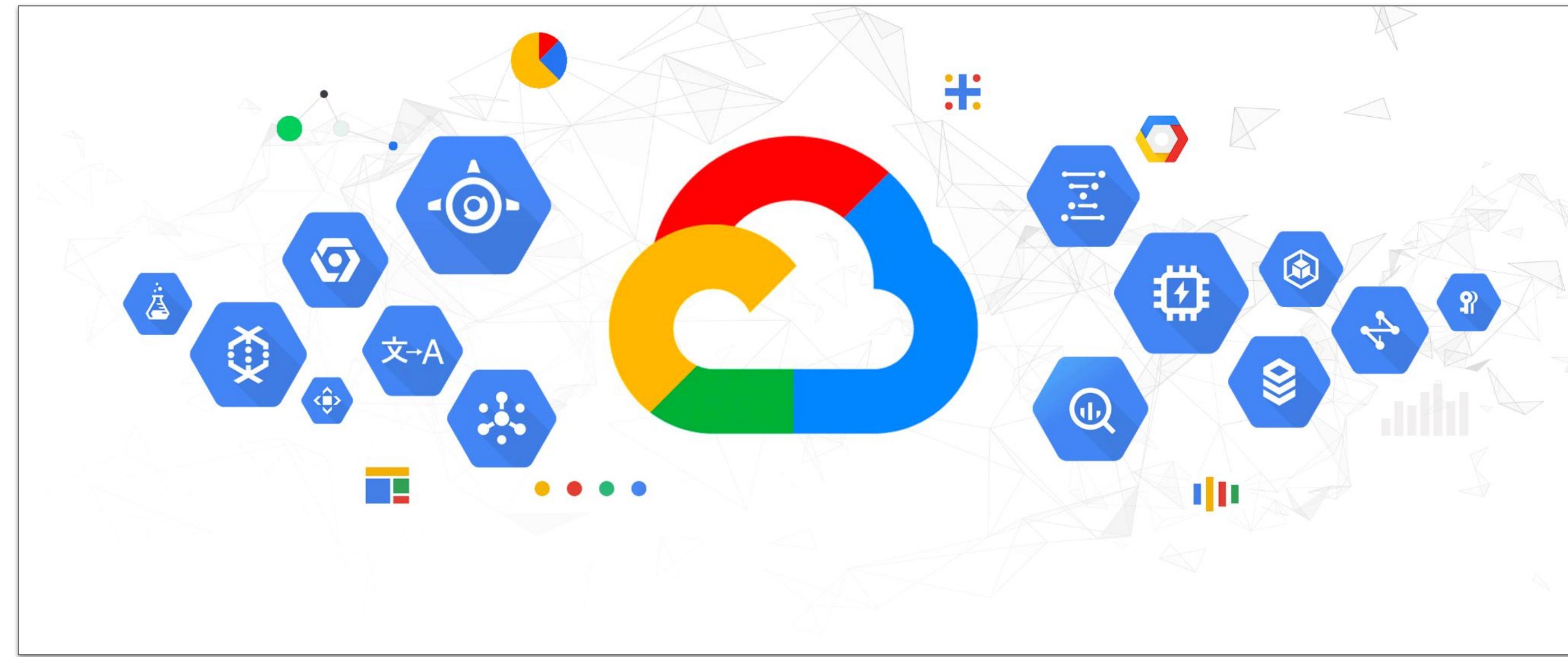
- Reduced likelihood of downtime and hardware failure
- Scalability = The website is not limited to the resources of a single server

Disadvantage

- Root access is not always provided (Access to do anything to the server)
- More expensive than VPS and shared hosting

Examples: Google Cloud, Amazon Web Services (AWS)





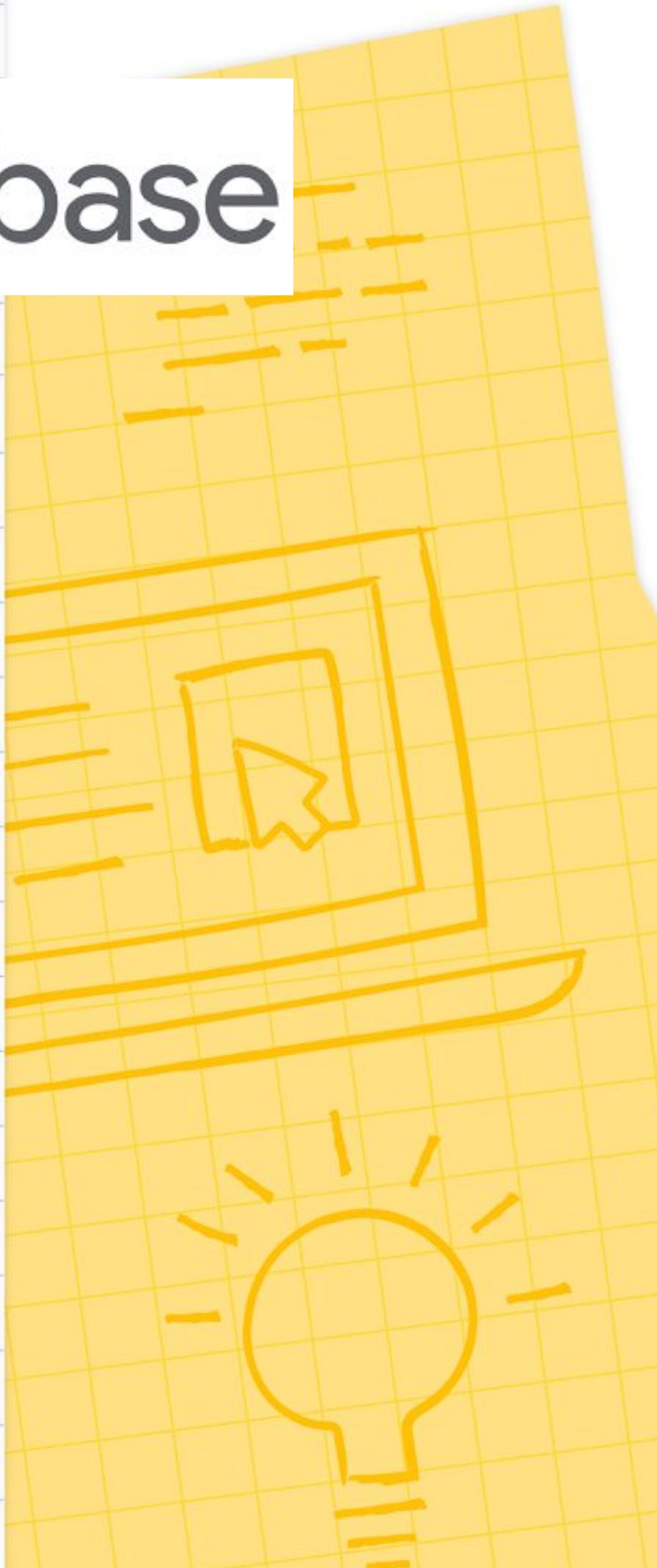
Google Cloud as a cloud hosting service

Installing Firebase

1. Download Firebase CLI from
<https://firebase.google.com/docs/cli>
2. When installing, make sure that you know where you save it.



Firebase





Google Developer Student Clubs



```
filterByOrg = filterByOrg ? study.lead_organization === filterByOrg : true  
filterStatus = filterByStatus ? study.status === filterByStatus : true  
if (filterByOrg & filterStatus) {  
    return studies.filter(study =>  
        study.lead_organization === filterByOrg && study.status === filterByStatus  
    )  
}  
  
function filterStudies({ studies, filterByOrg, filterByStatus }){  
    if (!studies || !filterByOrg || !filterByStatus) {  
        return studies  
    }  
    let filteredStudies = studies  
    if (filterByOrg) {  
        filteredStudies = filteredStudies.filter(study =>  
            study.lead_organization === filterByOrg  
        )  
    }  
    if (filterByStatus) {  
        filteredStudies = filteredStudies.filter(study =>  
            study.status === filterByStatus  
        )  
    }  
    return filteredStudies  
}
```



Hands On Session

```
function filterStudies({ studies, filterByOrg = false, filterByCategory = false }) {
  const filteredStudies = studies.filter(study => {
    if (filterByCategory) {
      return study.categories.some(category => category === filterByCategory);
    }
    if (filterByOrg) {
      return study.organizations.some(organization => organization === filterByOrg);
    }
    return true;
  });
  return filteredStudies;
}
```

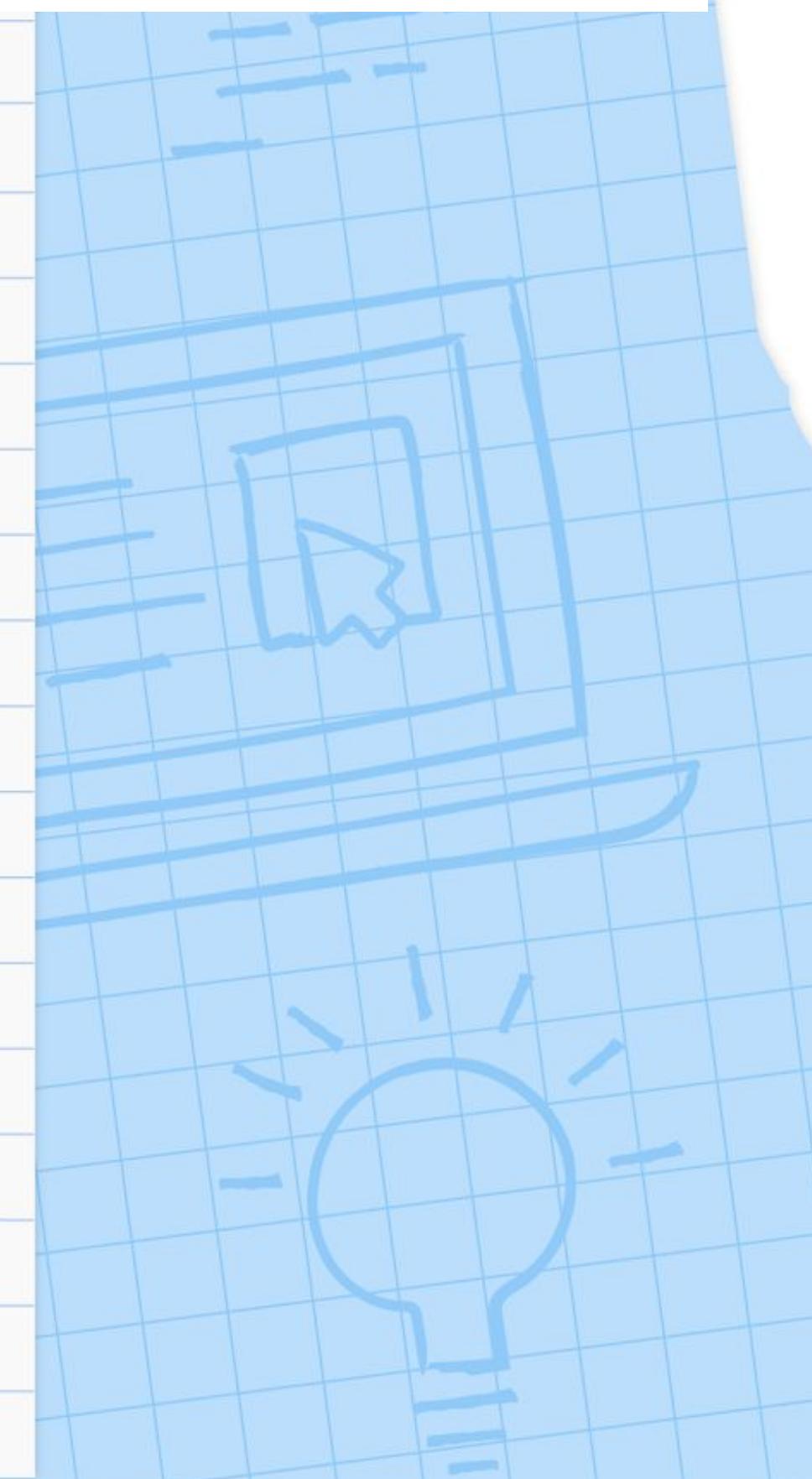
Firebase

A platform that is a part of Google Cloud

- Used to host websites and for building apps
- One account is only needed to sign in for both Google Cloud platform and Firebase



Firebase

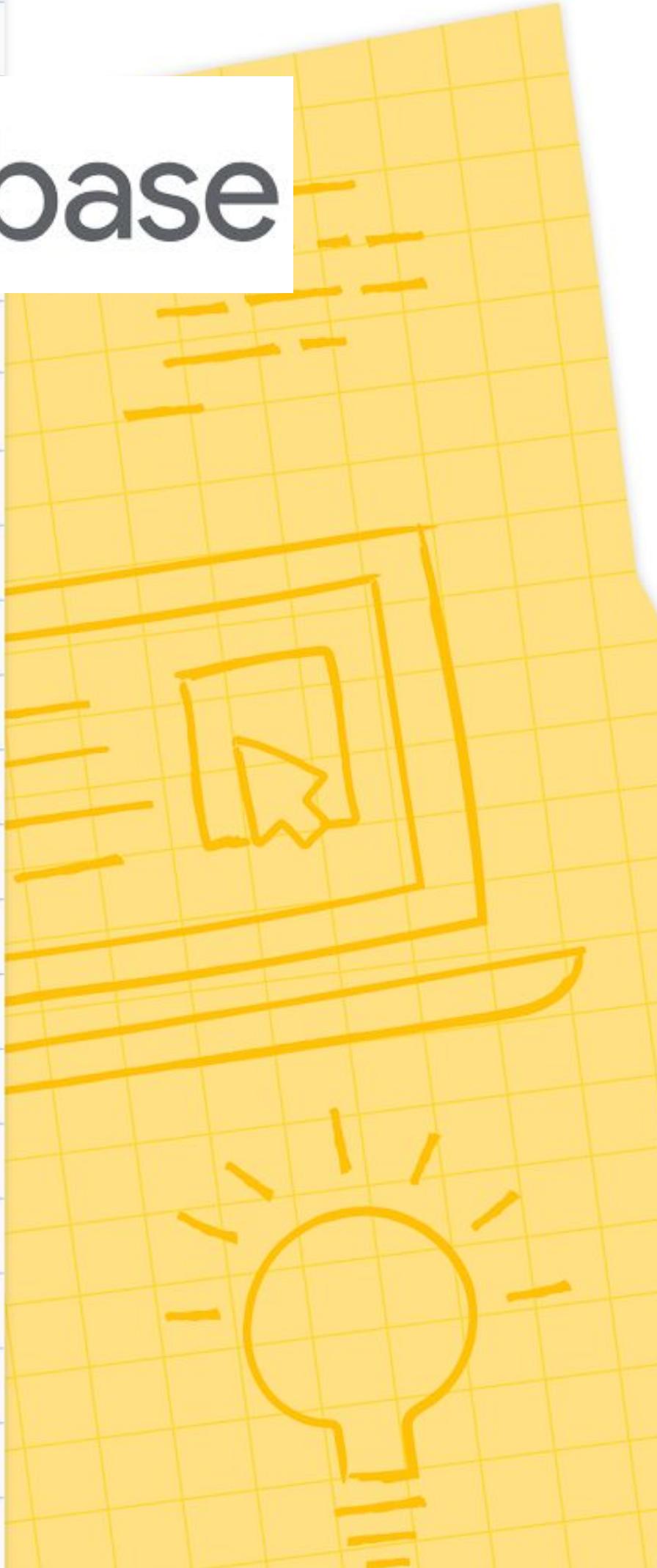


Installing Firebase

1. Download Firebase CLI from
<https://firebase.google.com/docs/cli>
2. When installing, make sure that you know where you save it.



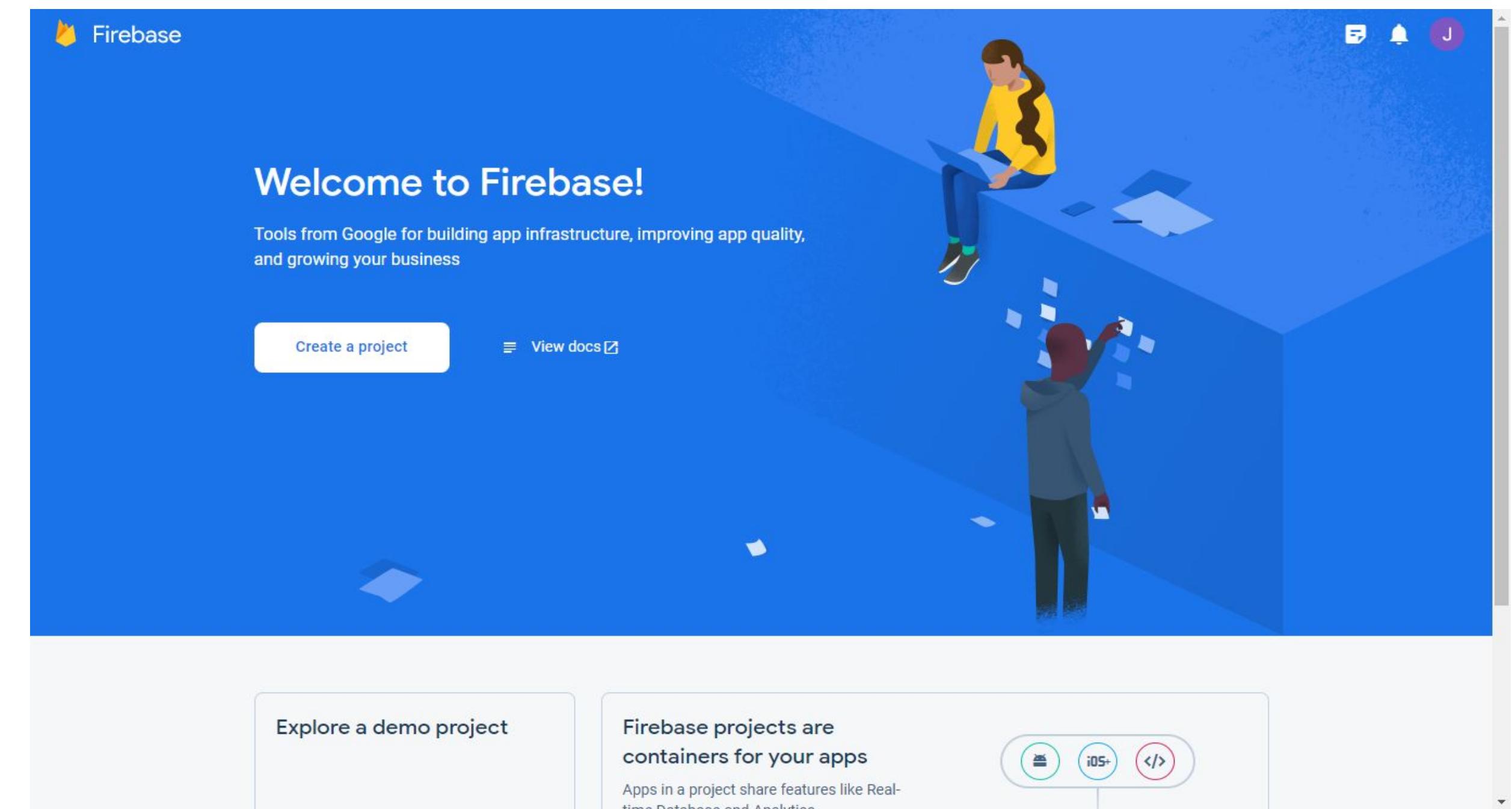
Firebase



Login to Firebase

Firebase Website

1. Log in to Firebase console using the same account used to login to Google Cloud.
2. Link to Firebase website:
<https://firebase.google.com/>
3. Create a new project with an unique project name.



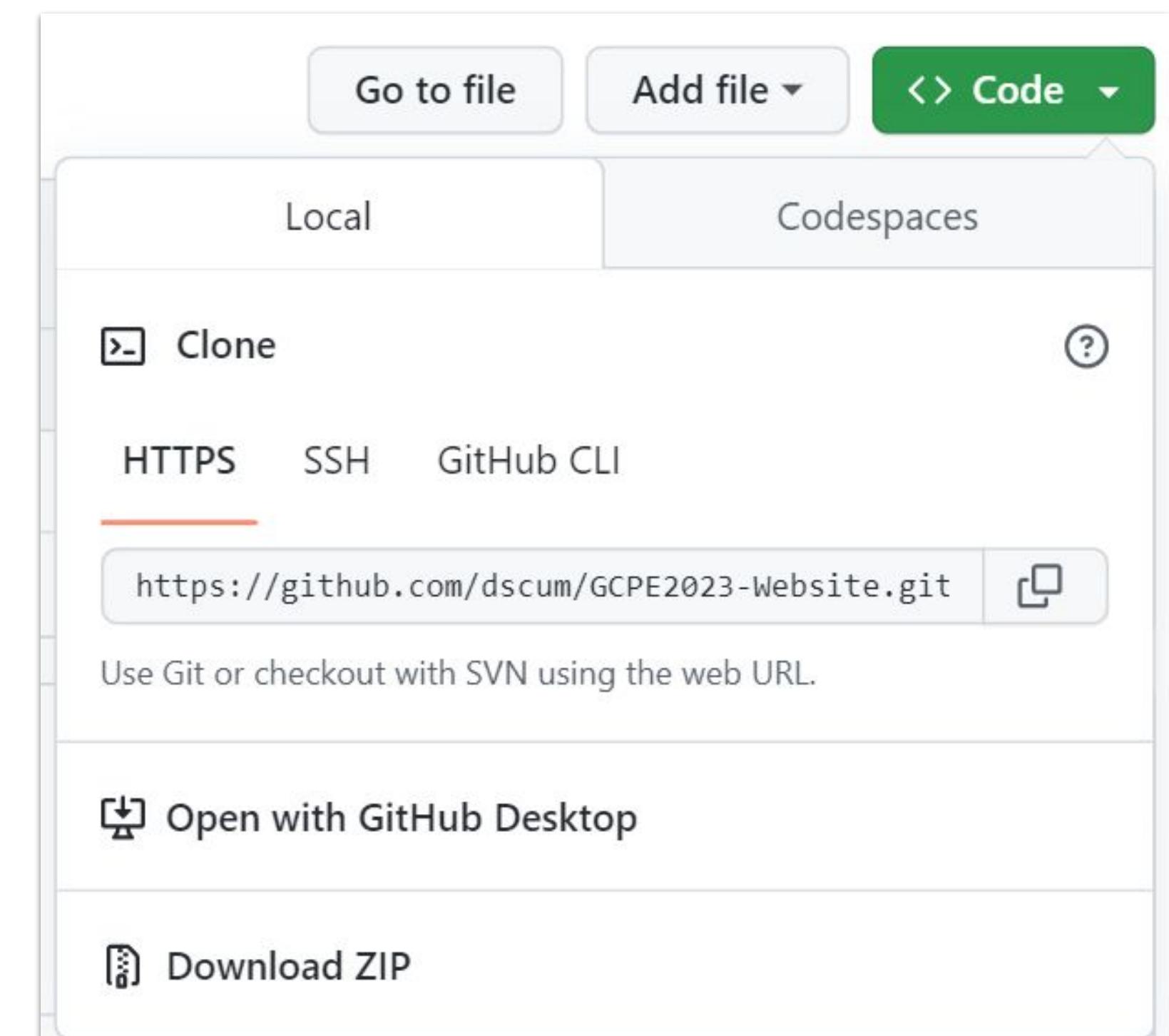
Download .zip file

Github repository

1. Go to Github repository link:

<https://github.com/dscum/GCPE2023-Website>

2. Click the green button of '**Code**' to download the .zip file of the code.
3. Track where the .zip file is downloaded and extract all the files.
4. Copy the directory of '**finished**' folder inside '**gdscUM-cloud-23**' folder.



Steps to host website in Firebase

Firebase CLI(Command Line Interface)

1. Command - **firebase login**

```
> firebase login
i  Firebase optionally collects CLI and Emulator Suite usage and error reporting information to help improve our products. Data is collected in accordance with Google's privacy policy (https://policies.google.com/privacy) and is not used to identify you.

? Allow Firebase to collect CLI and Emulator Suite usage and error reporting information? Yes
i  To change your data collection preference at any time, run 'firebase logout' and log in again.

Visit this URL on this device to log in:
https://accounts.google.com/o/oauth2/auth?client\_id=563584335869-fgrhgmd47bqnekij5i8b5pr03ho849e6.apps.googleusercontent.com&scope=email%20openid%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloudplatformprojects.readonly%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Ffirebase%20https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-platform&response\_type=code&state=577746562&redirect\_uri=http%3A%2F%2Flocalhost%3A9005

Waiting for authentication...
```

Steps to host website in Firebase

Firebase CLI

2. Next, copy and paste directory of folder '**finished**' in **gdscUM-cloud-23** for command - **cd (directory folder finished)**

```
> cd "D:\gdsc-UM\GCPE2023-Website-main\GCPE2023-Website-main\finished"
```

Steps to host website in Firebase

Firebase CLI

3. After that, use command - **dir**

```
Directory of D:\gdsc-UM\GCPE2023-Website-main\GCPE2023-Website-main\finished

18/03/2023  10:38 PM    <DIR>        .
18/03/2023  10:15 PM    <DIR>        ..
18/03/2023  10:15 PM    <DIR>        public
```



Steps to host website in Firebase

Firebase CLI

4. firebase init hosting

```
> firebase init hosting

#####
##      ##      ##      ##      ##      ##      ##
#####      ##      #####      ##      ##      ##      ##
##      ##      ##      ##      ##      ##      ##      ##
##      ##      ##      ##      ##      ##      ##      ##
##      ##      ##      ##      ##      ##      ##      ##

You're about to initialize a Firebase project in this directory:

D:\gdsc-UM\GCPE2023-Website-main\GCPE2023-Website-main\finished

Before we get started, keep in mind:

  * You are initializing within an existing Firebase project directory

? Are you ready to proceed? (Y/n) |
```



Steps to host website in Firebase

Firebase CLI

5. firebase deploy

```
> firebase deploy

=== Deploying to 'cloudproject938943'...

i  deploying hosting
i  hosting[cloudproject938943]: beginning deploy...
i  hosting[cloudproject938943]: found 10 files in public
+ hosting[cloudproject938943]: file upload complete
i  hosting[cloudproject938943]: finalizing version...
+ hosting[cloudproject938943]: version finalized
i  hosting[cloudproject938943]: releasing new version...
+ hosting[cloudproject938943]: release complete

+ Deploy complete!

Project Console: https://console.firebaseio.google.com/project/cloudproject938943/overview
Hosting URL: https://cloudproject938943.web.app
```



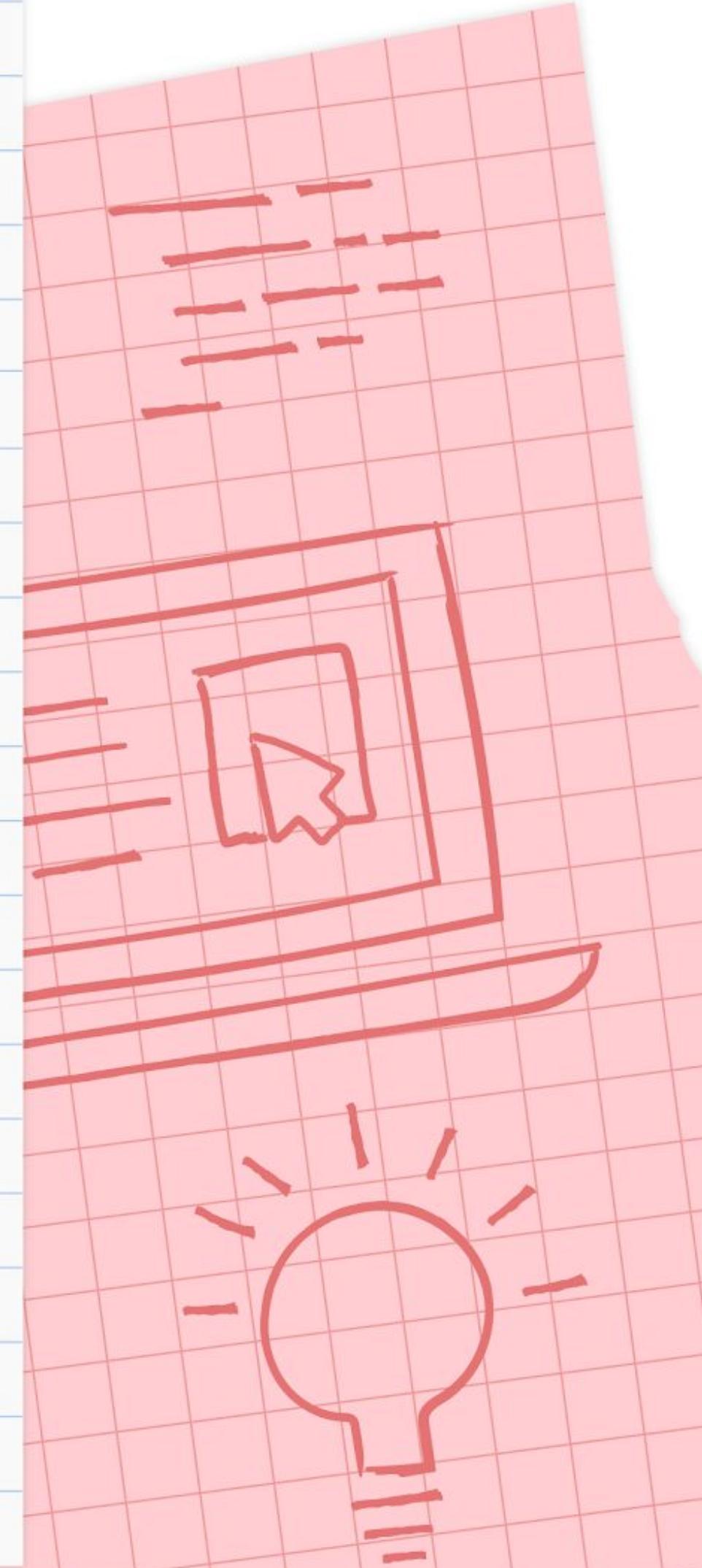
QnA



Join at
slido.com
#4699 426



Quiz Time





Prize Giving Ceremony

Day 2: Website Hosting

Attendance

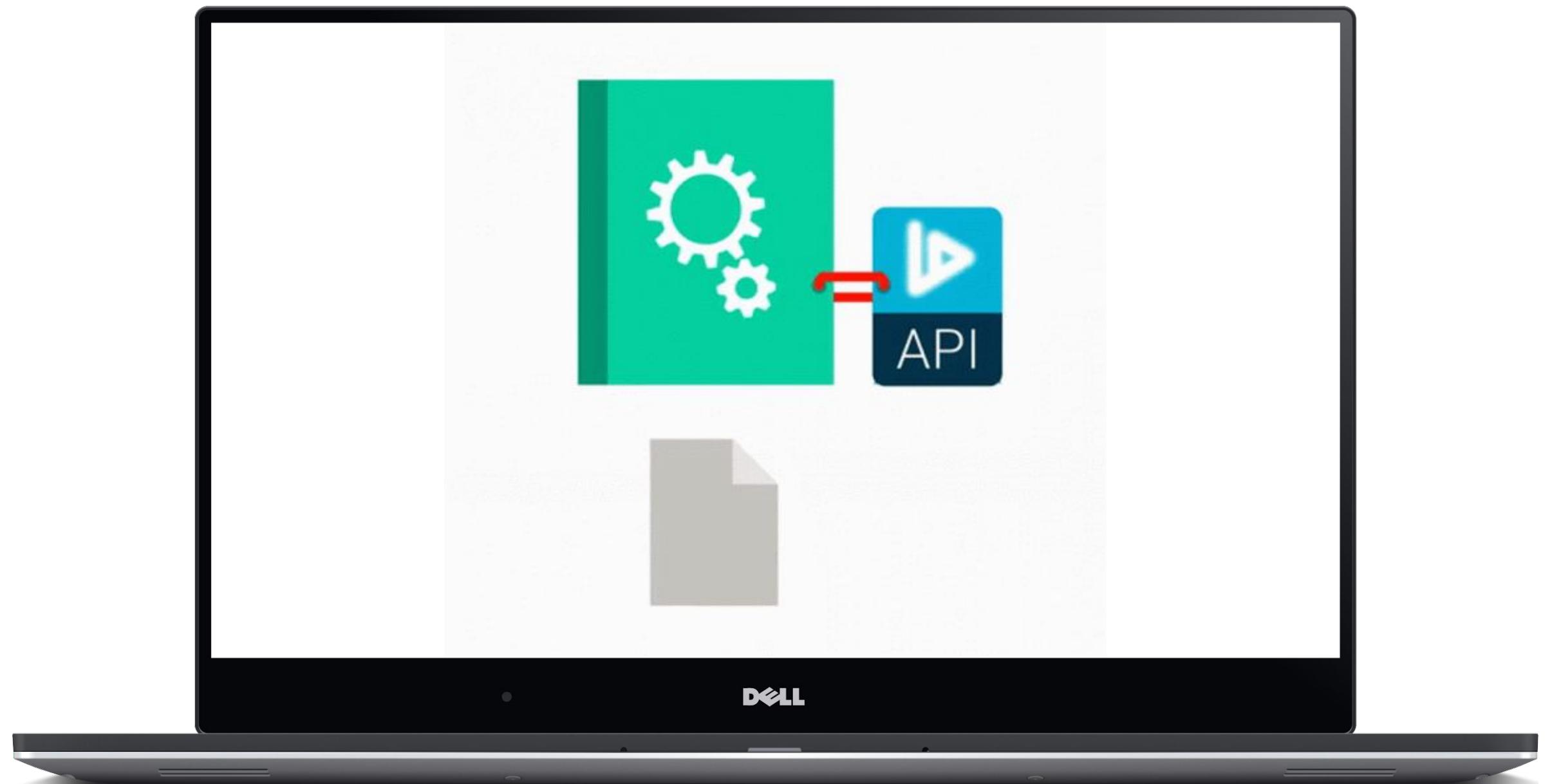
<https://bit.ly/GCPEattendance>



Next Workshop

@ Dewan Perdana 1, KPS

- 01** Intro to API
- 02** Google sheet API and Google Drive API
- 03** NodeJS to send request





Google Developer Student Clubs

Thank you



```
filterByOrg = filterByOrg ? study leadOrganization === filterByOrg : true  
filterStatus = filterByStatus ? study status === filterByStatus : true  
if (matchOrg & matchStatus) {  
    return study  
}  
  
function filterStudies({ studies, filterByOrg = false, filterByStatus = false }) {  
    const filteredStudies = studies.filter(study =>  
        filterByOrg && filterByStatus ? study leadOrganization === filterByOrg && study status === filterByStatus : true  
        filterByOrg ? study leadOrganization === filterByOrg : true  
        filterByStatus ? study status === filterByStatus : true  
        if (matchOrg & matchStatus) {  
            return study  
}  
    )  
    return filteredStudies  
}
```

