

Style
Your Ideas.
Design Your
World!

Learn • Code • Create

Make
Websites
Beautiful!



for

Kids



A Fun and Easy Guide to Learn CSS
and Design Amazing Websites



Easy Lessons
for Beginners



Real Examples
& Practice



Fun Projects
to Build



Quizzes &
Challenges



Step-by-Step
Learning



Perfect for
8-15 Years

```
1 h1 {  
2   color: #2d9cdb;  
3   font-size: 36px;  
4   text-align: center;  
5   font-family: 'Poppins', sans-serif;  
6 }  
7 .box {  
8   background-color: #ffef96;  
9   border: 2px solid #f39c12;  
10  padding: 20px;  
11  border-radius: 10px;  
12 }  
p {  
  color: #333;  
  line-height: 1.6;  
}
```



Learn Today,
Design
Tomorrow!



Be Creative.
Be Confident.
Be a Web
Designer!



10
Modules

50
Lessons

150+
Topics

10+
Projects

?
Quizzes
& More!



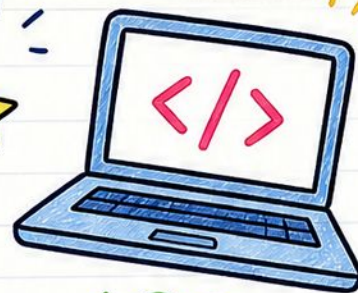
www.codescompiler.com



Let's make web pages beautiful! ♥

MODULE 1

Introduction to CSS

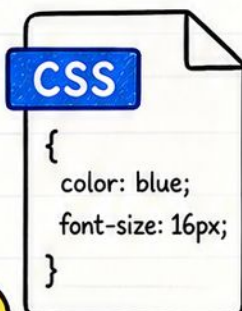


Lesson 1: What is CSS?



1 What Does CSS Mean?

CSS stands for **Cascading Style Sheets**. CSS is used to style and design the look and feel of a website.



It adds colors, fonts, spacing, layouts and more!



2 Why CSS is Important



Makes Websites Beautiful
CSS adds colors, fonts, images, spacing and animations.



Responsive Design
CSS helps websites look good on all devices - mobile, tablet, and desktop.



Saves Time & Effort
Change the style in one place, and it updates everywhere!



Better User Experience
A well-designed website is easy to read and fun to use.

3 HTML vs CSS

HTML builds the **structure**. CSS adds the **style**.



HTML (Structure)

Tells the browser what content to show.

```
<h1>Hello World!</h1>
<p>This is a paragraph.</p>
<button>Click Me</button>
```

HTML is like the skeleton of a house.



Result

CSS (Style)

Tells the browser how the content should look.

```
h1 { color: blue; font-size: 32px; }
p { color: gray; line-height: 1.6; }
button { background: green; color: white; padding: 10px 16px; border: none; border-radius: 6px; }
```

CSS is like the interior design and colors.



Together, HTML and CSS create amazing websites!



Remember:
HTML = Structure
CSS = Style
Perfect Together! ♥



Lesson 2: How Websites Get Their Style

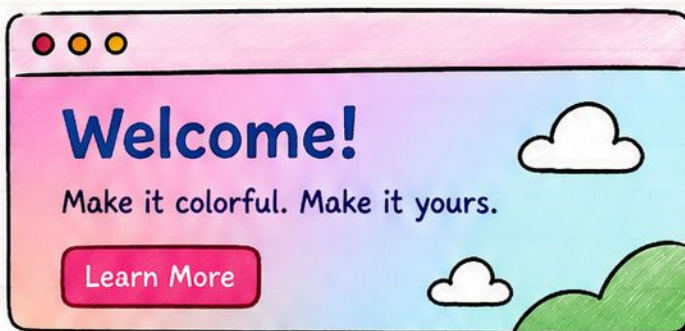
CSS (Cascading Style Sheets) is the magic behind the colors, fonts, and layouts you see on websites!

1

Colors and Designs

CSS helps websites look beautiful with colors, backgrounds, borders, and shapes.

- ★ Set text color, background color, and more.
- ★ Add borders, shadows, and rounded corners.
- ★ Create patterns and gradients to make pages stand out.



CSS EXAMPLE

```
body {  
  background-color: #e0f7fa;  
}  
h1 {  
  color: #1565c0;  
}  
.box {  
  background: linear-gradient(45deg,  
    #ff9a9e, #fad0c4);  
  border-radius: 12px;  
}
```



2

Fonts and Layouts

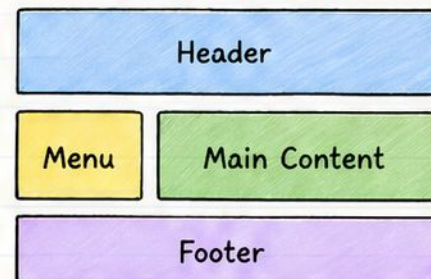
CSS controls how text looks and how things are arranged on the page.

- ★ Choose cool fonts and sizes.
- ★ Make text bold, italic, or underlined.
- ★ Adjust spacing, line height, and alignment.
- ★ Use layout tools to organize content into rows and columns.

Aa Bb Cc

Good fonts + smart layout
= easy to read and
fun to explore! ♥

SIMPLE LAYOUT



3

Real-World Examples

Every website you visit uses CSS to look the way it does!

- ★ Social media sites
- ★ Online stores
- ★ News websites
- ★ Blogs and portfolios
- ★ Even this colorful note! 😊



Facebook
Clean layout
and blue
theme



Amazon
Organized
products and
easy shopping



Netflix
Dark theme,
big images,
bold text



Spotify
Great fonts,
colors, and
user experience



News Sites
Clear layout
and easy to
read



★ CSS is what makes websites not just useful,
but also **beautiful** and **fun** to use! ♥

www.codescompiler.com





Lesson 3: Adding CSS to HTML



CSS

makes HTML
look beautiful
and organized!

In this lesson, we will learn 3 ways
to add CSS to HTML.

Good Design
Makes a
Big Difference!

1

Inline CSS

We use the **style** attribute inside an
HTML element to apply CSS.

Example:

```
<p style="color: blue;  
font-size: 20px;  
font-weight: bold;">  
Hello, Inline CSS!  
</p>
```

Key Points

- CSS is written directly in the HTML tag.
- Affects only that specific element.
- Good for quick and small changes.



2

Internal CSS

We use the **<style>** tag inside the **<head>**
section of HTML to write CSS.

Example:

```
<head>  
  <style>  
    p {  
      color: green;  
      font-size: 20px;  
    }  
  </style>  
</head>  
<body>  
  <p>Hello, Internal CSS!</p>  
</body>
```

Key Points

- CSS is written inside the **<style>** tag in **<head>**.
- Affects all elements in that page.
- Useful for single page styling.



Great for
one-page
websites!

3

External CSS

We write CSS in a separate **.css** file and link
it to the HTML using the **<link>** tag.

Example:

```
<head>  
  <link rel="stylesheet" href="styles.css">  
</head>  
<body>  
  <p>Hello, External CSS!</p>  
</body>
```

styles.css

```
p {  
  color: navy;  
  font-size: 20px;  
  font-weight: bold;  
}
```

Key Points

- CSS is written in a separate file.
- Affects all pages linked to it.
- Best for big websites (clean & reusable).



www.codescompiler.com

Keep Coding,
Keep Learning! ❤️

LESSON 4:

CSS
{ }

YOUR FIRST CSS RULE

 CSS helps us tell the browser how to style HTML elements.

Let's break down a CSS rule! ←

THE 3 PARTS OF A CSS RULE

1 SELECTORS

The selector tells the browser which HTML element(s) you want to style.

Example:
p

This selects all <p> (paragraph) elements.

2 PROPERTIES

The property is what you want to change about the selected element.

Example:
color

This changes the text color.

3 VALUES

The value is how you want to change the property.

Example:
blue

This sets the color to blue.

PUTTING IT ALL TOGETHER!

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

↑ ↑ ↑
Selector Property Value

This rule says:

"Make the text inside all <p> elements blue."

HTML

<p>Hello, world!</p>

BROWSER

Hello, world!

TIPS!

- ★ Every CSS rule ends with a semicolon (;).
- ★ You can write many rules to style your page!

Practice more,
Experiment, and
Have fun with CSS!



MODULE 1:

Introduction to CSS



Lesson 5: Mini Project

KEY POINTS

1 Style a Simple HTML Page

Use CSS to change how HTML elements look and make your page neat and attractive.



CSS makes HTML look awesome!



2 Add Colors

Use colors to set the background, text, borders, and more. Colors make your page lively!

Background



#E6F4FF

Text



#222222

Border



#4CAF50

Button



#FF9800



3 Add Fonts

Change fonts to make your text easy to read and give your page its own style and personality.

Serif Font

Hello!

(Times New Roman)

Sans-Serif Font

Hello!

(Arial)

Google Font

Hello!

(Poppins)



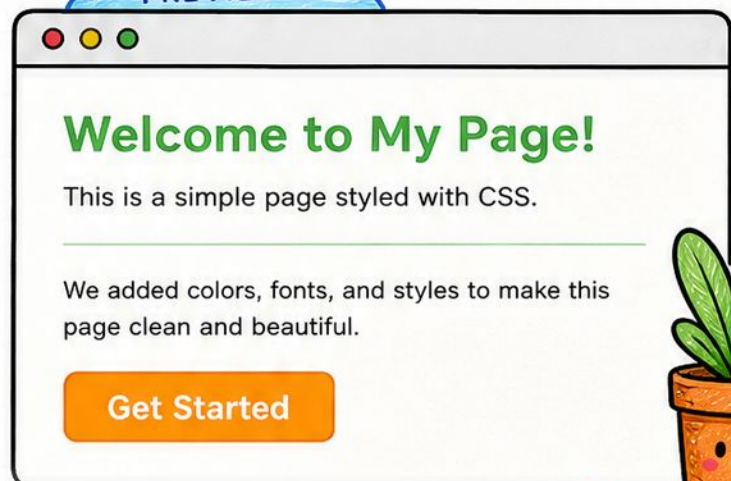
This is how CSS styles your page!



EXAMPLE CSS

```
body {
  background-color: #E6F4FF;
  font-family: 'Poppins', sans-serif;
  color: #222222;
  margin: 0;
  padding: 20px;
}
h1 {
  color: #4CAF50;
}
button {
  background-color: #FF9800;
  color: white;
  padding: 10px 20px;
  border: none;
  border-radius: 6px;
  font-weight: bold;
}
```

PREVIEW



REMEMBER: Good design makes your page look better, but good code makes it work better!



Practice today,
Build amazing
things tomorrow!



MODULE 2



Colors and Backgrounds



Lesson 1:

Let's explore the world of colors!

Working with Colors

In this lesson, we will learn about:



1

Color Names

Colors have names that help us talk about them. Here are some basic color names:



Red



Orange



Yellow



Green



Blue



Purple



Pink



Brown



Black



White

2

HEX Colors

HEX (Hexadecimal) colors are a way to show colors using 6 letters and numbers that start with #.

Example:

#FF5733 is a bright orange color.

Some Popular HEX Colors

Color	Preview	HEX Code
Red		#FF0000
Orange		#FFA500
Yellow		#FFFF00
Green		#00FF00
Blue		#0000FF
Purple		#800080
Pink		#FFC0CB
Black		#000000
White		#FFFFFF



3

RGB Colors

RGB stands for Red, Green, Blue. Colors are made by mixing these 3 colors in different amounts.

Each color uses a value from 0 to 255.

Some Popular RGB Colors

Color	Preview	RGB Code	How it's made
Red		(255, 0, 0)	Full Red
Green		(0, 255, 0)	Full Green
Blue		(0, 0, 255)	Full Blue
Yellow		(255, 255, 0)	Red + Green
Cyan		(0, 255, 255)	Green + Blue
Magenta		(255, 0, 255)	Red + Blue
White		(255, 255, 255)	All Together
Black		(0, 0, 0)	No Color





Module 2: Colors and Backgrounds



Lesson 2:

Text Colors



Text colors help our words look **beautiful**, **easy to read**, and more **interesting**.
Let's learn how to use them!



1 Changing Font Color

We can change the color of text to make it stand out or match the style we want.

Example:

Learning is fun!
Learning is fun!
Learning is fun!
Learning is fun!

How to do it?

Select the text →
Choose a color →
Click Apply!



2 Highlighting Important Text

Highlighting helps people notice key points quickly.
Great for titles, reminders, or important ideas!

Example:

The **Sun** is **very hot** and gives us light and energy.

Why highlight?

- ✓ Grabs attention
- ✓ Improves reading
- ✓ Makes it memorable!

3 Color Combinations

Good color combinations make text easy to read and look great together!

Great Combinations:

A a

Navy Blue + White

A a

Dark Green + Light Yellow

A a

Purple + White

A a

Black + Light Blue

A a

Dark Red + Cream

Tip:

Use light background with dark text or dark background with light text for better readability.



Use colors **smartly**, keep it **simple**, and let your words **shine**!

www.codescompiler.com



Lesson 3: Background Colors

Colors make
our pages
beautiful and
easy to read!

Module 2: Colors and Backgrounds

KEY POINTS

1 PAGE BACKGROUNDS

The page background is the color that fills the entire web page.

It sets the mood of the whole page!

```
body {  
  background-color: #E6F7FF;  
}
```

My Web Page

This is my page with a light blue background.

2 SECTION BACKGROUNDS

You can add background colors to parts (sections) of a page to highlight important areas.

Great for grouping content and improving readability!

```
.top {  
  background-color: #FFE8B3;  
}  
  
.bottom {  
  background-color: #D1F7C4;  
}
```

Top Section

This area has a light yellow background.

Bottom Section

This area has a light green background.

3 MULTIPLE BACKGROUND COLORS

You can use different colors for different sections to make your page more colorful and attractive.

Use colors that look good together!

```
<header style="background-color: #FFB6C1;">Header</header>  
<main style="background-color: #E6F7FF;">Main Content</main>  
<footer style="background-color: #DCD6FF;">Footer</footer>
```

Header

Main Content

Footer

Remember:

Good use of background colors makes your webpage beautiful, clear and fun!



Module 2: Colors and Backgrounds



Lesson 4: Background Images



We can make web pages more fun and beautiful by adding **background images**!

KEY POINTS

- ✓ Adding Images
- ✓ Repeating Backgrounds
- ✓ Background Position



1

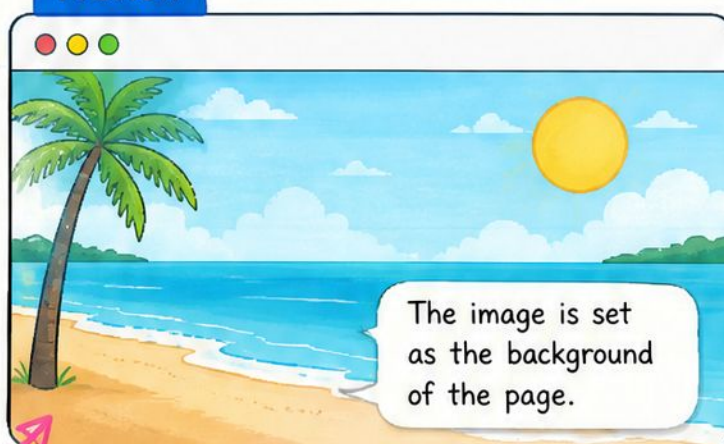
Adding Images

We use the **background-image** property to add an image behind the content.

Example:

```
body {  
  background-image: url("beach.jpg");  
  background-size: cover;  
}
```

Preview



2

Repeating Backgrounds

By default, background images repeat horizontally and vertically. We can change how the image repeats using the **background-repeat** property.

Examples:



Think!



What happens if we use **no-repeat**? Try it and see!



3

Background Position

We can control where the background image appears using the **background-position** property.

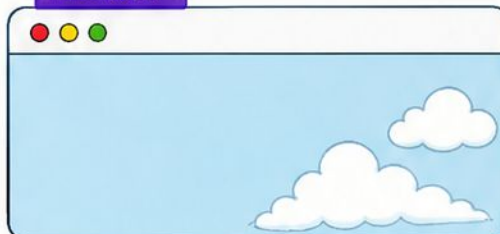
Examples:

top left	top center	top right
center left	center	center right
bottom left	bottom center	bottom right

Example:

```
body {  
  background-image: url("clouds.png");  
  background-repeat: no-repeat;  
  background-position: right bottom;  
}
```

Preview



www.codescompiler.com



Lesson 5:

Mini Project

In this mini project, you will use colors and backgrounds to create amazing designs!

1 Colorful Birthday Invitation

Design a bright and cheerful birthday invitation using colors, shapes, and fun backgrounds!



2 Rainbow Webpage

Create a colorful webpage with a rainbow theme. Use backgrounds, buttons, and text to make it eye-catching!



3 School Poster Design

Design a school poster on any topic you like. Use colors, borders, and backgrounds to make your message stand out!





Lesson 1:

GOOD DESIGN
STARTS WITH
GREAT TEXT!

Font Basics

The right font helps your message
look great and easy to read!



KEY POINTS



1 Font Families

2 Serif vs Sans-Serif

3 Choosing Fonts

1 FONT FAMILIES

Fonts are grouped into families.
Each family has a similar style.
Examples:



Aa

Arial
(Sans-Serif)

Aa

Times New Roman
(Serif)

Aa

Pacifico
(Script)

Aa

Impact
(Display)

★ Different families create different feelings
and are used for different purposes.

2 SERIF VS SANS-SERIF

The main difference is the little feet
(serifs) at the ends of letters.

SERIF

Has small lines
or strokes.

Aa

Example:
Times New Roman

★ Feels traditional,
formal and easy to
read in print.

SANS-SERIF

No small lines
or strokes.

Aa

Example:
Arial

★ Feels modern,
clean and easy to
read on screens.

3 CHOOSING FONTS

Pick fonts that match your message and audience. ★

CONSIDER PURPOSE



Use playful fonts
for fun projects.
Use clean fonts
for serious or
professional work.

THINK ABOUT READABILITY



Choose fonts that
are easy to read.
Avoid fonts that
are too fancy for
long text.

LIMIT YOUR FONTS

2

Use 1 or 2 fonts
in a design.
Too many fonts
can look messy.

CONSIDER YOUR AUDIENCE



Think about the
age, interests and
preferences of
your audience.

MAINTAIN CONSISTENCY



Use the same fonts
across your project
for a professional
and consistent look.

QUICK TIP



Great fonts don't just look good,
they help people understand
and enjoy your message!

GOOD FONTS
MAKE GREAT
DESIGNS! ♥



www.codescompiler.com

MODULE 3:

Fonts and Text Styling

Good text makes your ideas easy to read! ♥

Lesson 2: Text Formatting

Text formatting helps us change the look of text to make it clear, beautiful, and more effective.

1 Font Size

Font size changes how big or small the text is. Larger text is easier to see and shows what's important!

This is small text.

This is medium text.

This is large text!

2 Font Weight

Font weight shows how thick or thin the text is. It helps highlight important words or information!

- This is light text.
- This is normal text.
- **This is bold text.**
- **This is extra bold text.**

3 Font Style

Font style changes the shape of the text. It can look normal, slanted, or have beautiful curves!

- Normal (Regular) This is normal.
- *Italic* This is italic.
- ***Bold Italic*** This is bold italic.
- Underline This is underlined.

TIP:

Use the right size, weight, and style to make your text easy to read and fun to see!

Great formatting makes your message shine!



Lesson 3: Text Alignment

Text alignment is how text is arranged horizontally between the margins.

1 Left Alignment

The text begins at the **left margin**. Each line starts from the left and the right edge is uneven.

- Learning to code is fun!
- It helps us create games, apps, and websites.
- Keep practicing every day.
- You will become a great coder!

Text starts from the left.



2 Center Alignment

The text is **centered** between the margins. Both left and right edges are uneven.

- Learning to code is fun!
- It helps us create games, apps, and websites.
- Keep practicing every day.
- You will become a great coder!

Text is centered in the middle.



3 Right Alignment

The text ends at the **right margin**. Each line starts from the left and the left edge is uneven.

- Learning to code is fun!
- It helps us create games, apps, and websites.
- Keep practicing every day.
- You will become a great coder!

Text ends at the right.



Why is it important?

Good text alignment makes your content **easy to read**, looks **neat**, and helps others **understand better**.





TEXT EFFECTS



Text effects make our words look more creative, fun, and eye-catching!



1 TEXT SHADOW

Adds a **shadow** behind the text to make it stand out and look more 3D.

Great for headings and titles!

EXAMPLE

HELLO!

```
text-shadow: 4px 4px 6px gray;
```

2 LETTER SPACING

Adds **space** between the letters in a word to improve readability and style.

Use more space for a clean and modern look!

EXAMPLE

NORMAL

CREATIVE

WIDE LETTER SPACING

C R E A T I V E

```
letter-spacing: 4px;
```

3 TEXT TRANSFORM

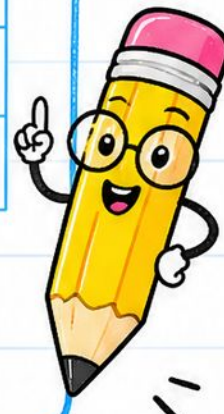
Changes the **case** of the text to uppercase, lowercase, or capitalize each word.

ABC

EXAMPLE

UPPERCASE	HELLO WORLD!
lowercase	hello world!
Capitalize	Hello World!

```
text-transform: uppercase;  
text-transform: lowercase;  
text-transform: capitalize;
```



Lesson 5: Mini Project

In this mini project, you will create
3 awesome designs using fonts and text styling!

Let's
Get
Creative!

1

My Story Page

Create a page that shares a little about you. Use different fonts, colors, and text styles to make your story fun and unique!

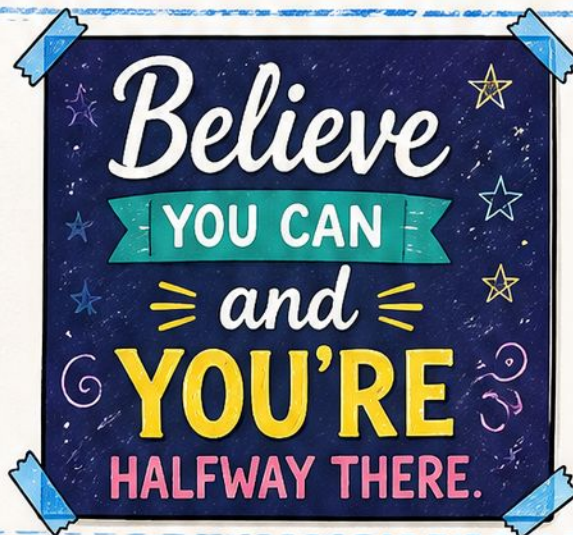
My Story
Hi! I'm Alex 😊
I love coding, drawing
and exploring new ideas.
Dream big, work hard,
and have fun! ★



2

Favorite Quote Poster

Design a poster featuring your favorite quote. Use creative fonts, colors, and effects to make it stand out!



Tips:

- ✓ Choose a strong font for impact.
- ✓ Play with sizes and colors.
- ✓ Add decorations to make it pop!

3

Motivational Web Banner

Create a web banner that inspires and motivates others. Use bold text and attractive styling to grab attention!



www.codescompiler.com

MODULE 4

Borders
and Boxes

Lesson 1: Understanding the Box Model



Every element on a webpage is like a box. The **Box Model** shows how space is added around the content inside that box.

Think of it
like a gift box
with layers
around it!



MARGIN

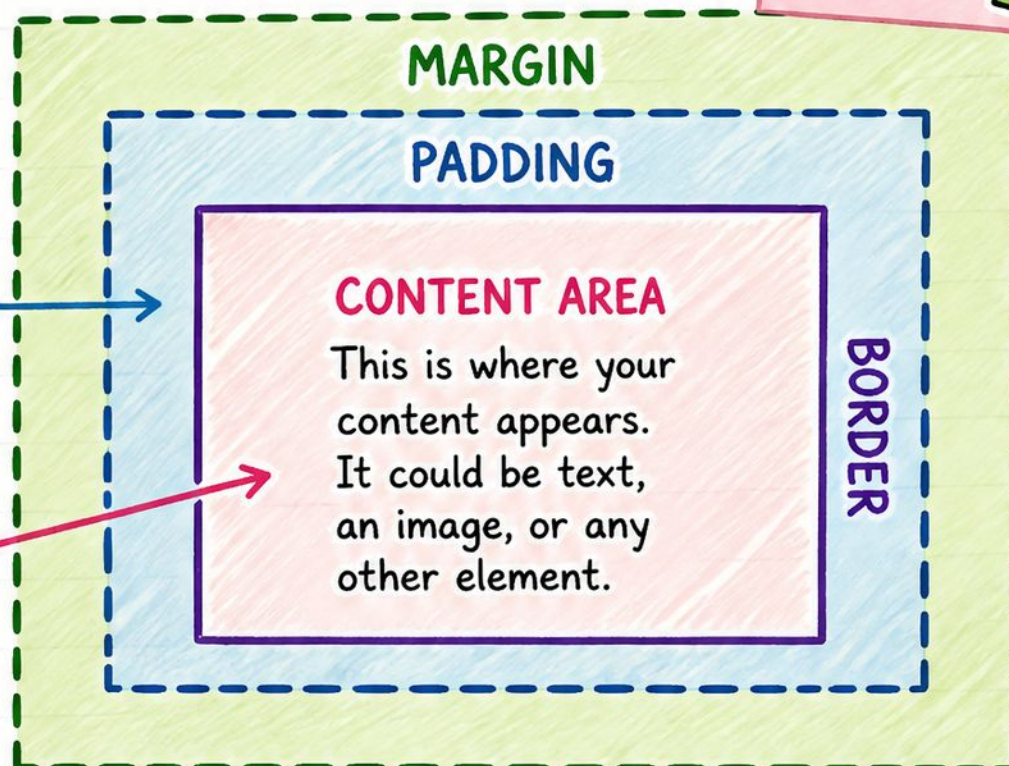
The space outside the box. It separates this box from other elements.

PADDING

The space inside the box, between the content and the border.

CONTENT AREA

The actual content of the box, like text, images, or anything you put inside.



1 CONTENT AREA

This is the main part of the box. It holds the content that users see and read.

Example:

A paragraph of text or an image.



2 PADDING

This is the space between the content and the border. It gives the content some breathing room.

Example:

Adding space inside a gift box so the gift doesn't touch the sides.



3 MARGIN

This is the space outside the border. It separates this box from other boxes on the page.

Example:

Leaving space around a gift box so it's not too close to the other gifts.



Remember: **Margin** is outside, **Padding** is inside, and **Content** is what people see!

Lesson 2: Borders



Borders help us highlight and organize content. We can change how a border looks, how thick it is, and what color it has!

1

Border Styles

Border style decides the kind of line used for the border.



Solid



Dashed



Dotted



Double



Groove

2

Border Width

Border width changes how thick or thin the border line is.



Very Thin



Thin



Medium



Thick



Very Thick

3

Border Colors

Border color makes your content look brighter and more fun!



Red



Orange



Yellow



Green



Blue



Purple



Black

Try different styles, widths, and colors to make your work awesome!



Module 4: Borders and Boxes



Lesson 3: Rounded Corners



Small changes, big impact!

KEY POINTS

1 BORDER RADIUS

Border radius lets us round the corners of a box. You can control how round the corners are!

EXAMPLE

```
.box {  
  background: #4CAF50;  
  color: white;  
  padding: 20px;  
  border-radius: 20px;  
}
```

Try changing 20px to see the difference!

SEE THE MAGIC!

Before



After



2 CIRCULAR SHAPES

Make circles or ovals using border radius! When the radius is 50% (or half of width/height), you get a perfect circle.

EXAMPLE

```
.circle {  
  width: 100px;  
  height: 100px;  
  background: #2196F3;  
  border-radius: 50%;  
}
```

FROM SQUARE TO CIRCLE!

0%



25%



50%



50% makes it a perfect circle!

3 MODERN DESIGNS

Rounded corners make designs look soft, clean, and modern! Used in buttons, cards, images, and more.

EXAMPLE

```
.card {  
  background: #fff;  
  padding: 20px;  
  border-radius: 16px;  
  box-shadow: 0 4px 12px  
    rgba(0,0,0,0.15);  
}
```

LOOKS AWESOME!



Hello, Coder!

I love building cool projects with code.

Learn More

TIP:

You can set different radius for each corner too!

```
border-radius: 20px 10px  
              30px 40px;
```

WHY IT MATTERS?

- ♥ Improves user experience
- ♥ Looks friendly and approachable
- ♥ Makes your projects stand out!



MODULE 4

LESSON 4: SHADOWS

Shadows
make designs
look awesome!

KEY POINTS

1 BOX SHADOWS

Box shadows add depth to boxes and make them stand out from the background.

Without Shadow



With Box Shadow



2 GLOW EFFECTS

Glow effects add a soft light around elements to make them shine and grab attention.

Normal



With Glow



3 CARD DESIGNS

Combine shadows and glow effects to create beautiful cards for any occasion!



www.codescompiler.com

KEEP
LEARNING
&
KEEP
CREATING!

MODULE 4

BORDERS AND BOXES

Lesson 5: Mini Project

In this mini project, you will create:

1

Stylish Information Card

Design a colorful card to show important information in a stylish way.



FUN FACT

Honey never spoils!

Archaeologists have found pots of honey in ancient tombs that are over **3000** years old!



2

Profile Box

Create your own profile box with your picture, name, hobbies and more.



MY PROFILE

Name : Aarav Sharma
Age : 11
Hobby : Drawing
Favorite Color : Blue
Dream : Space Explorer

3

Fun Message Board

Make a message board where friends can write fun messages and notes.



LEAVE A MESSAGE!

Great work!
Keep it up! 😊
- Riya

You are awesome!
- Kabir

Super creative!
- Mehak



MODULE 5: WORKING WITH IMAGES



Lesson 1: Styling Images



We can make images look better on a web page by changing their size, making them responsive, and aligning them.



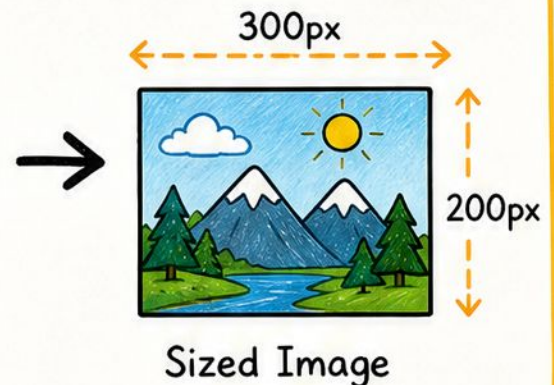
1 IMAGE SIZES

We can set the size of an image using the **width** and **height** properties.

```
img {  
  width: 300px;  
  height: 200px;  
}
```



Original Image



Sized Image

2 RESPONSIVE IMAGES

Responsive images adjust their size to look great on **all** devices.

```
img {  
  max-width: 100%;  
  height: auto;  
}
```

Same image, perfect on every screen!



Desktop



Tablet



Mobile

3 ALIGNMENT

We can **align** images with text or other elements on the page.

Align Left



The image is aligned to the left. Text flows on the right side.

Align Right



The image is aligned to the right. Text flows on the left side.

Center



The image is in the center.



TIP!

Always choose the right size and alignment to make your web pages clean and beautiful!





Module 5: Working with Images



Lesson 2: Image Borders



We can make pictures more interesting by adding **borders**, **rounding the corners**, and using **decorative effects**.

Borders and effects make our images look more fun and creative!

1

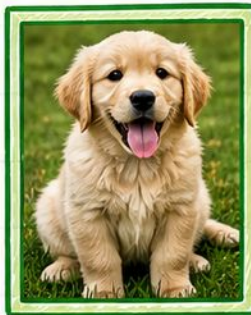
Frames Around Images

Frames are borders that go around the image. They can be simple or fancy!

Why use frames?

- ★ They highlight the image.
- ★ They make it look neat and eye-catching.

Examples



Simple Frame



Wavy Frame



Double Frame

2

Rounded Images

We can round the corners of an image to make it look soft and friendly.

Why use rounded images?

- ★ They look modern and smooth.
- ★ They are gentle on the eyes.

Examples



Slightly Rounded



More Rounded



Fully Rounded (Circle)

3

Decorative Effects

Decorative effects add style and creativity to images.

Popular Effects:

- ★ Shadow - adds depth
- ★ Glow - makes it shine
- ★ Sketch - gives a drawn look
- ★ Sepia - gives a warm, vintage look

Examples



Shadow



Glow



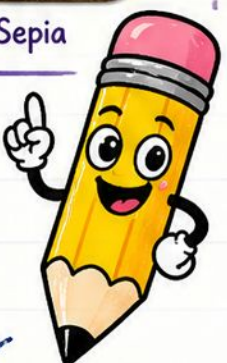
Sketch



Sepia

QUICK TIP!

Choose the right border or effect based on the **purpose** of your image and the **feeling** you want to create.



MODULE 5: WORKING WITH IMAGES



LESSON 3: IMAGE HOVER EFFECTS



Hover effects make images more
interactive and fun!

Hover means
when we move
the mouse over
an image!

KEY POINTS

1 ZOOM EFFECT

The image becomes
larger when you
hover over it.



Looks closer!

BEFORE



HOVER



2 OPACITY EFFECT

The image becomes
lighter or darker
(more transparent)
on hover.



See-through
Magic!

BEFORE



HOVER



3 COLOR CHANGES

The image color
changes when you
hover over it.

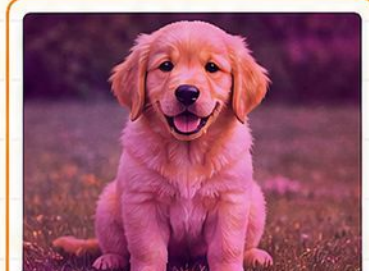


New look,
New vibe!

BEFORE



HOVER



Why use hover effects?

- ✓ Make your website attractive
- ✓ Improve user experience
- ✓ Show creativity with images

TIP:

Use hover effects
softly. Too much
can be distracting!



Try these
effects and
make your
images
awesome!

MODULE 5

Working
with Images



Lesson 4: Background Pictures

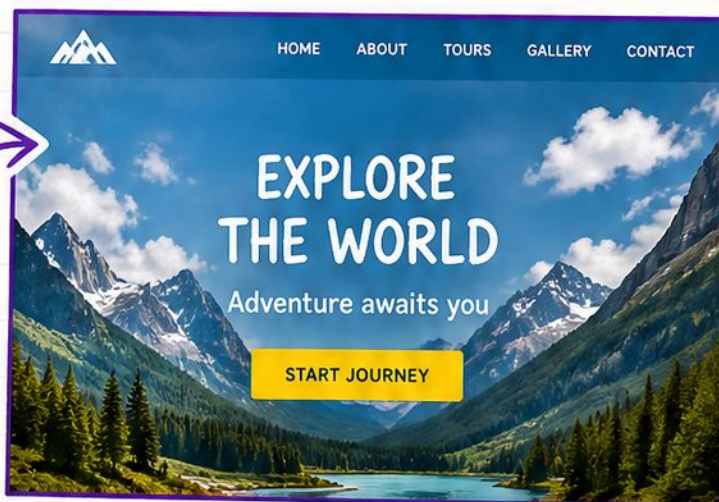


Background pictures help make our web pages look beautiful and engaging!

In web design, we use background pictures in different ways. Let's learn three important types!

1 Hero Images

Hero images are large background pictures used on the top of a webpage to grab attention. They make a strong first impression!



Think!

Hero images are big, bold and beautiful!

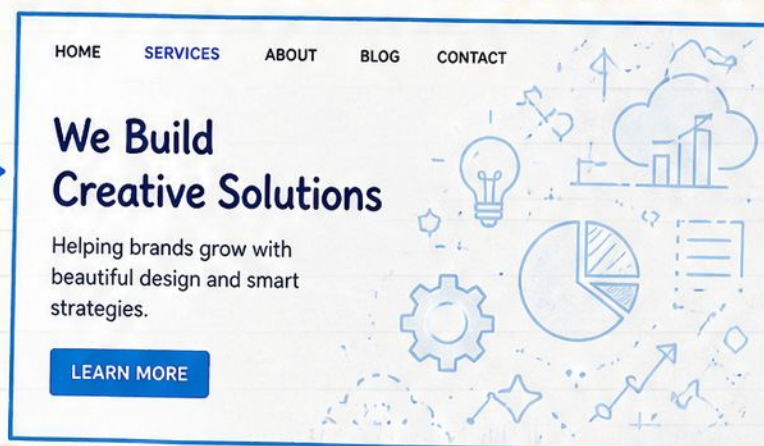
2 Cover Images

Cover images cover the entire section or page. They are used to set the mood and give more information about the content.



3 Positioning Images

These images are placed in the background at a specific position. They are usually small and support the design without distracting.



Remember!

Choose the right background picture to make your website look great and easy to read!

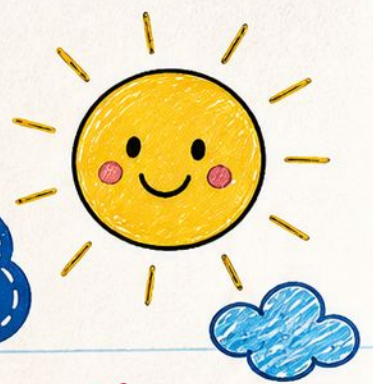
In Short:

- ★ Hero Images – Grab attention at the top.
- ★ Cover Images – Cover the area and set the mood.
- ★ Positioning Images – Placed at specific spots to support design.





MODULE 5



Working with Images



Lesson 5: Mini Project



Key points:

1

Photo Gallery

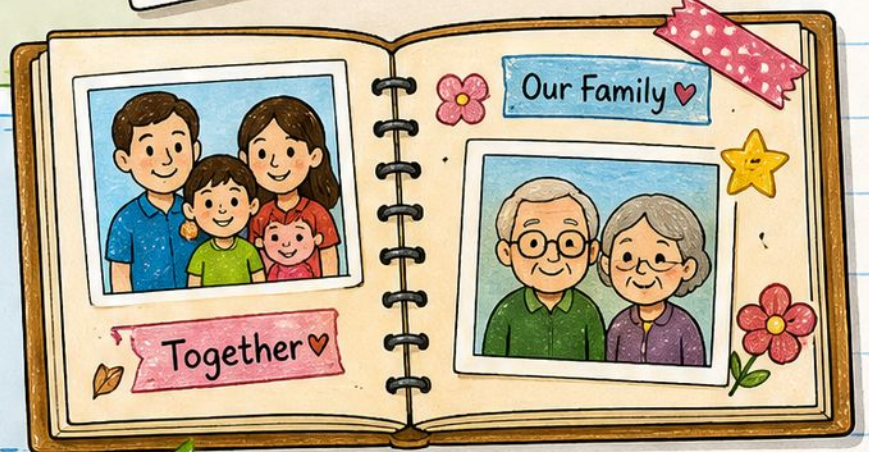
Create a collection of your favorite photos. Arrange and display them beautifully on a page.



2

Family Album

Make a special album for your family memories. Add photos, names, and sweet captions.



3

Travel Memories Page

Design a page to share your travel adventures. Add photos, places, and fun moments.



Use your creativity and make your project unique and amazing!



Module 6: Layout Design

A good layout helps users find information easily!

Lesson 1: Understanding Layouts

Every webpage is made up of different sections working together. 😊

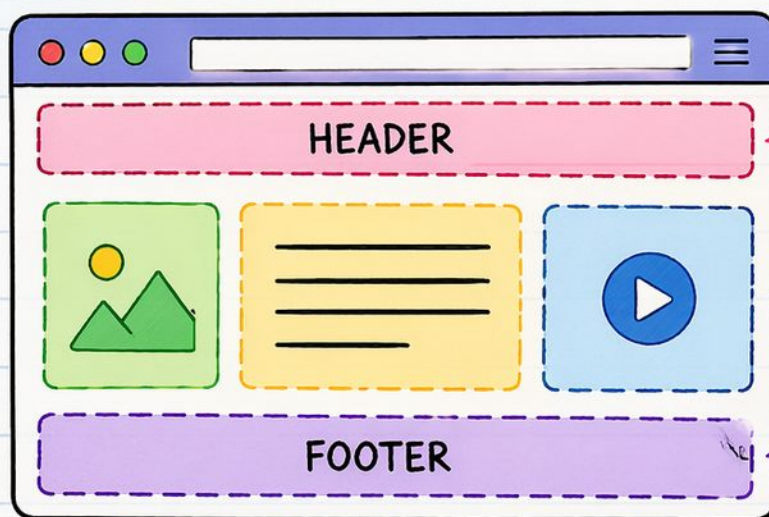
Key Points

1 Webpage Sections

A webpage is divided into sections. Each section has a specific purpose and helps organize information.



★ Think of sections as different parts of a building. Each part has its own job!



2 Header and Footer

The header is at the top of the page. It usually contains the logo, website name, and navigation menu.

The footer is at the bottom. It may include contact info, links, copyright, or social media icons.

HEADER
(Top)

🌐 MyWebsite Home About Services Contact

📌 Helps users navigate and understand the website quickly. 😊

FOOTER
(Bottom)

✉ info@mywebsite.com | Privacy Policy | 📘 🐦 📷

© 2024 MyWebsite. All rights reserved.

3 Content Areas

The content area is the main part of the webpage. It shows the important information like text, images, videos, or forms. This is where users focus most of their attention.

Examples of Content:



Articles and Text



Images and Graphics



Videos and Animations



Forms and Surveys

Remember:

A well-structured layout makes websites easy to use and visually appealing! ❤️



Good Layout

- ✓ Easy to read
- ✓ Easy to navigate
- ✓ Looks good
- ✓ Works on all devices



www.codescompiler.com

MODULE 6 LAYOUT DESIGN



Lesson 2: DISPLAY PROPERTY

The **display** property decides how an HTML element is shown on a web page and how it behaves with other elements.

THERE ARE 3 MAIN TYPES OF DISPLAY:

1

BLOCK ELEMENTS

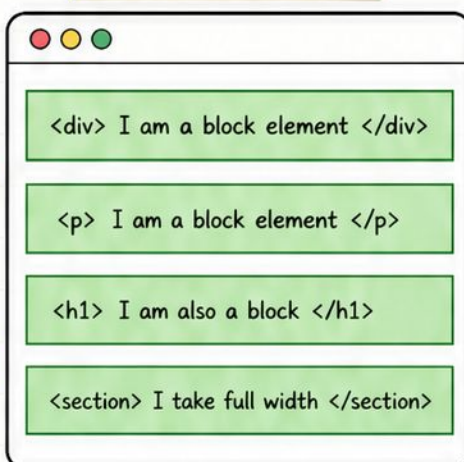
Block elements take up the full width available. They always start on a new line and push other elements below them.

Examples:

div, p, h1-h6, form, header, footer, section

Each element starts on a new line

HOW THEY APPEAR



CSS EXAMPLE

```
div {  
  display: block;  
  background: #d4f5d4;  
  padding: 10px;  
  margin: 10px 0;  
  border: 2px solid #4caf50;  
}
```

★ Key Point: Block elements are great for creating sections and layouts.



2

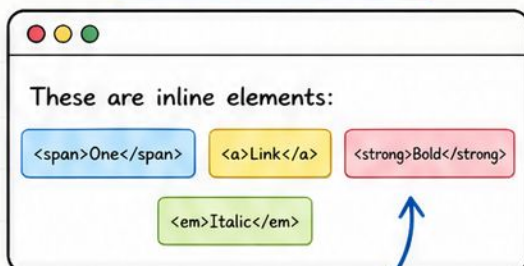
INLINE ELEMENTS

Inline elements do NOT start on a new line. They only take up as much width as needed and sit next to other inline elements.

Examples:

span, a, strong, em, img, label, small

HOW THEY APPEAR



They sit next to each other on the same line.

CSS EXAMPLE

```
span {  
  display: inline;  
  color: #0d47a1;  
  padding: 2px 6px;  
}
```

★ Key Point: Inline elements are perfect for text and small parts inside content.



3

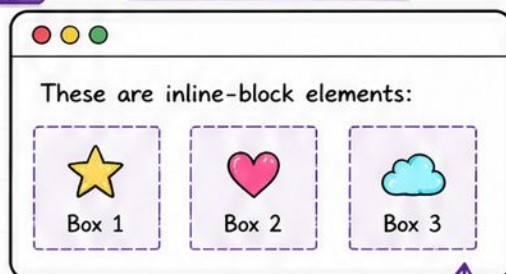
INLINE-BLOCK ELEMENTS

These elements behave like inline elements (sit in a line) but you can set width, height, margin, and padding like block elements.

Examples:

button, input, textarea, img, select, custom elements

HOW THEY APPEAR



They sit in a line, but we can set width, height, margin and padding.

CSS EXAMPLE

```
.box {  
  display: inline-block;  
  width: #00px;  
  height: 60px;  
  margin: 10px;  
  padding: 10px;  
  background: #ede7f6;  
  border: 2px dashed #7e57c2;  
  text-align: center;  
}
```

★ Key Point: Inline-block gives you the best of both worlds! Great for buttons, menus, and cards.



QUICK TIP

You can change how any element is displayed using the **display** property!



Understanding display helps you build clean, beautiful, and well-structured web pages!



Lesson 3: Positioning Elements

Module 6:
Layout
Design

What is Positioning?

Positioning tells the browser how to place an element on a webpage using different rules.



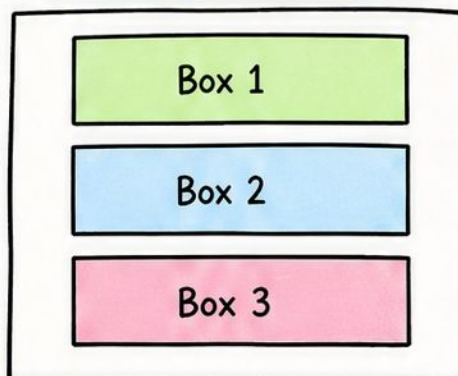
1

STATIC POSITION

This is the default position. Elements appear in the normal flow of the page, one after another.

```
.box {  
  position: static;  
}
```

EXAMPLE



Boxes are placed in order, following the normal flow.

No top, right, bottom, left offsets apply.



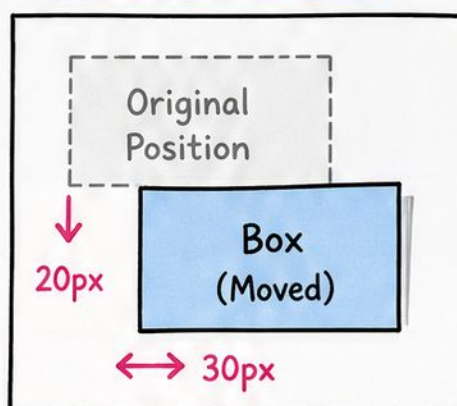
2

RELATIVE POSITION

The element is moved relative to where it would normally be. It still takes up space in the page.

```
.box {  
  position: relative;  
  top: 20px;  
  left: 30px;  
}
```

EXAMPLE



The box is moved 20px down and 30px to the right from its original spot.

Space is still reserved for it.



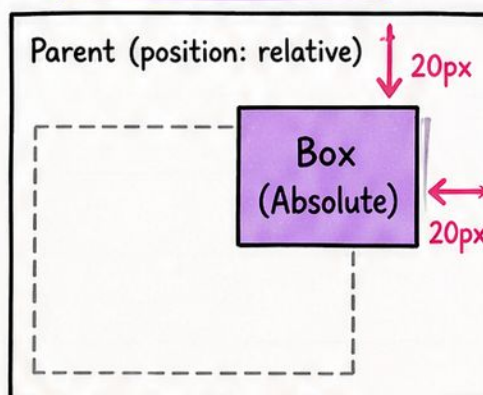
3

ABSOLUTE POSITION

The element is removed from the normal flow and positioned using top, right, bottom, left relative to its nearest positioned ancestor (parent).

```
.box {  
  position: absolute;  
  top: 20px;  
  right: 20px;  
}
```

EXAMPLE



The box is placed 20px from the top and 20px from the right of its parent.

It does NOT take space in the flow.



QUICK SUMMARY

- ♥ **Static** : Default position, follows normal flow.
- ♥ **Relative** : Moved from its normal spot, space is reserved.
- ♥ **Absolute** : Removed from flow, positioned using offsets.

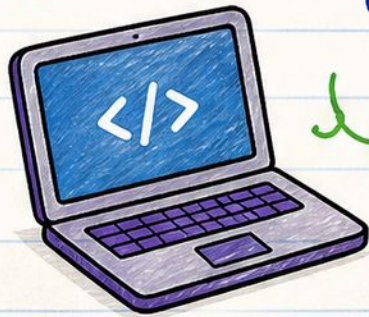
Try them out and see the magic!



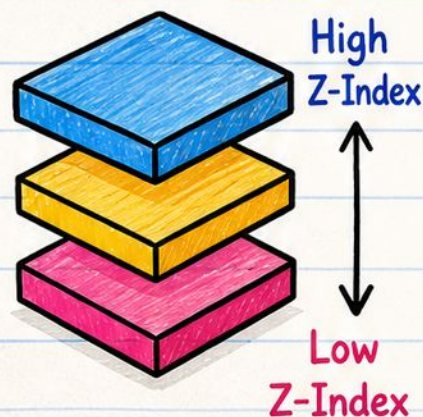
www.codescompiler.com



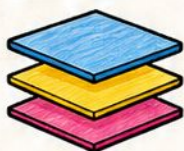
Z-Index and Layering



Decide which elements appear on top when things overlap!



KEY POINTS



Stacking Elements



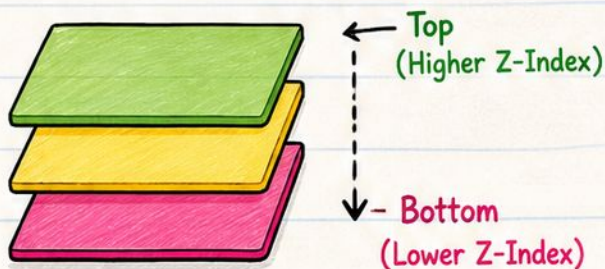
Overlapping Objects



Visual Effects

1 Stacking Elements

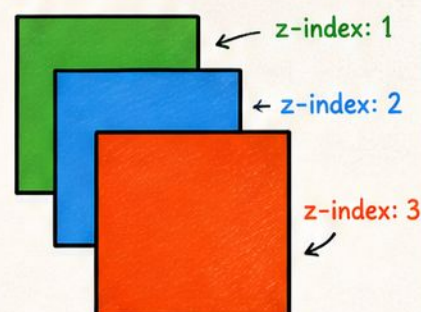
Every element is like a card in a stack. The Z-index controls which card appears on top.



EXAMPLE

```
.box1 {  
  position: absolute;  
  top: 20px; left: 20px;  
  width: 120px; height: 120px;  
  background: #4CAF50;  
  z-index: 1;  
}  
.box2 {  
  position: absolute;  
  top: 50px; left: 50px;  
  width: 120px; height: 120px;  
  background: #2196F3;  
  z-index: 2; /* on top */  
}  
.box3 {  
  position: absolute;  
  top: 80px; left: 80px;  
  width: 120px; height: 120px;  
  background: #FF9800;  
  z-index: 3; /* on top of all */  
}
```

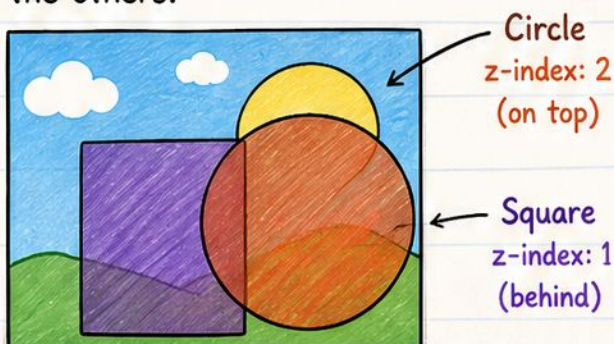
Result:



Higher z-index means it appears on top!

2 Overlapping Objects

When elements overlap, the one with higher z-index will cover the others.



TIPS

- ★ Default z-index is 0.
- ★ Works best on positioned elements (position: relative, absolute, fixed, sticky).
- ★ Use meaningful values and keep it simple!

3 Visual Effects

Using z-index carefully can create cool effects and improve user experience!

Modal Popup



Tooltip



Sticky Header



Great layering makes your designs clear, beautiful and interactive!

Think in Layers!



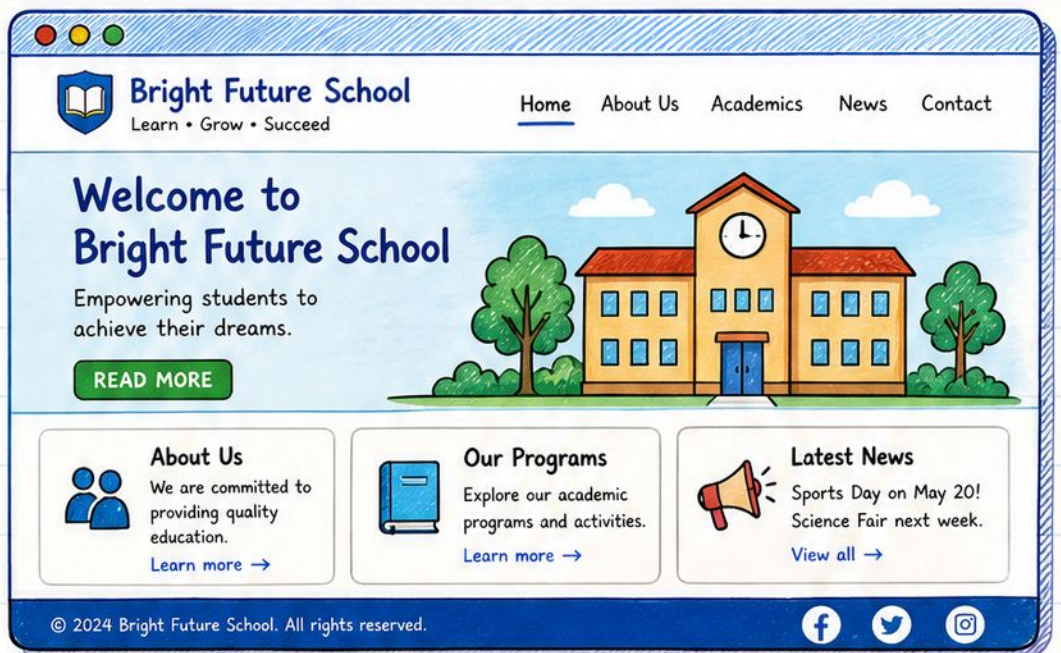
Lesson 5: Mini Project

KEY POINTS:

In this mini project, you will design and plan different layouts. Choose one or try them all!

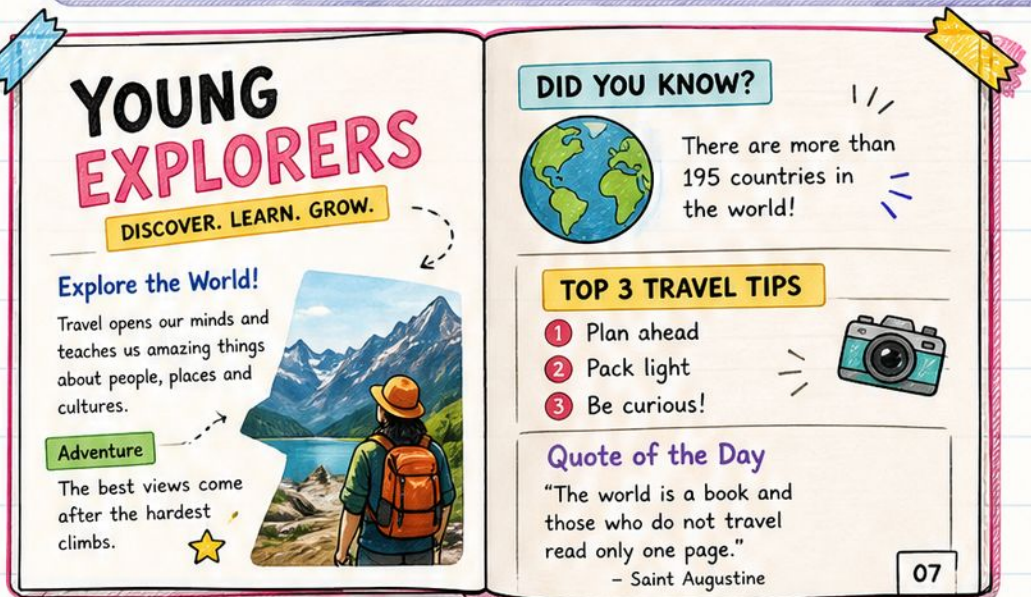
1 School Website Layout

Design a homepage for a school website. Include the header, menu, main image, about section, news and footer.



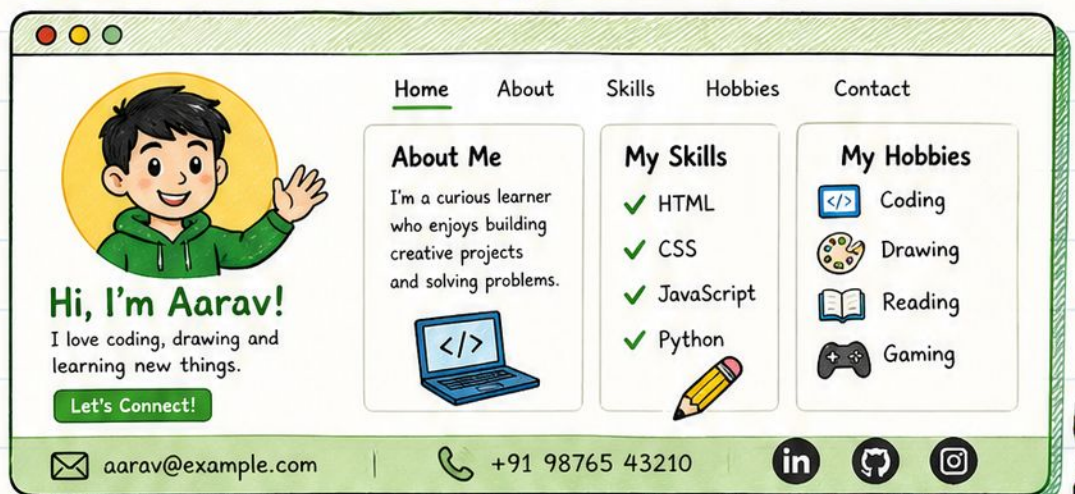
2 Magazine Page

Create a magazine page on any topic you like. Add a catchy title, images, articles and page number.



3 Personal Homepage

Design your own personal homepage. Introduce yourself, show your hobbies, skills and contact info.



FLEXBOX
MAKES
LAYOUTS
EASY!

Lesson 1: What is Flexbox?

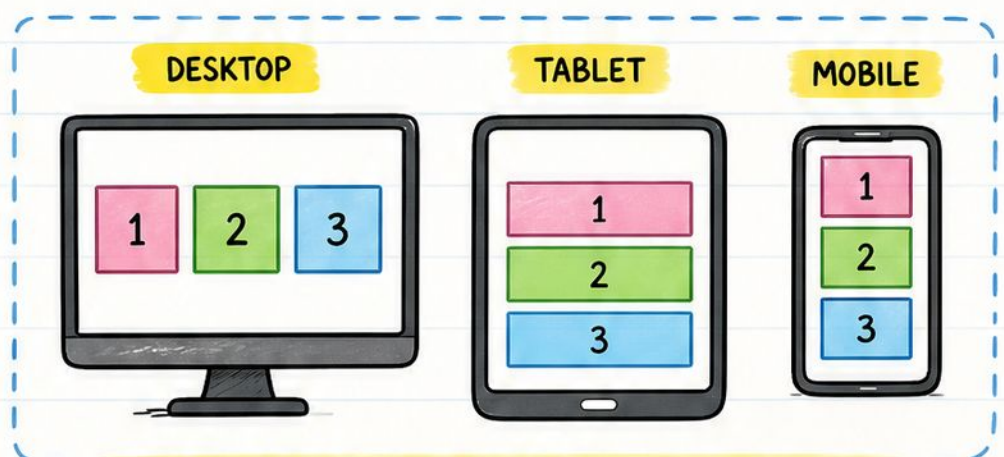
Flexbox (Flexible Box Layout) is a CSS tool that helps us **arrange items** in a container, even when the **size** of the items or the **screen** changes.



1 Flexible Layouts

Flexbox helps create layouts that can adapt to different screen sizes and devices. Items can stretch, shrink, or wrap automatically.

Perfect for responsive web design! ❤️



Flexbox adjusts items beautifully on any screen size!

2 Why Use Flexbox

- ✓ Easy alignment of items
- ✓ Space distribution in a container
- ✓ Works in one direction (row or column)
- ✓ Less code, more power!

WITHOUT FLEXBOX ☹️



- ✗ Hard to align
- ✗ Inconsistent spacing
- ✗ Much more code

WITH FLEXBOX 😊



- ✓ Easy to align
- ✓ Even spacing
- ✓ Clean and simple code

3 Flex Containers

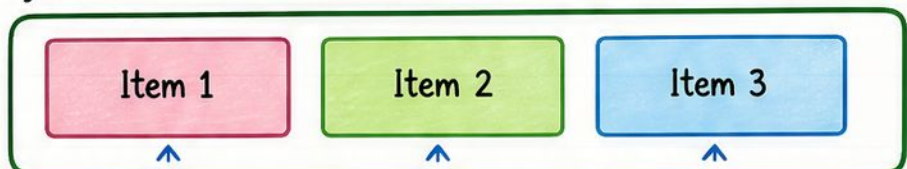
Flexbox works inside a parent element called a **Flex Container**.

All direct children become **Flex Items**.



```
.container {
  display: flex;
}
```

This is the
Flex Container



These are Flex Items (direct children)



★ Think of Flexbox as a smart way to organize boxes inside a bigger box! 😊

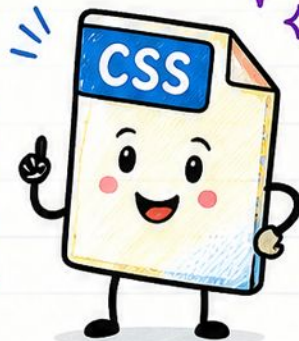
Remember:

Flexbox = Flexible
Layouts = Happy Users! ❤️



MODULE 7: FLEXBOX MADE EASY

Lesson 2: FLEX DIRECTION



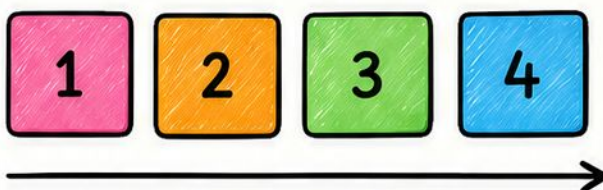
Flex Direction decides the direction in which items are placed in a flex container!

KEY POINTS

1 ROWS

This is the default direction. Items are placed from left to right in a row.

flex-direction: row;



Left to Right

CSS

```
.container {  
  display: flex;  
  flex-direction: row;  
}
```

2 COLUMNS

Items are placed from top to bottom in a column.

flex-direction: column;



Top to Bottom

CSS

```
.container {  
  display: flex;  
  flex-direction: column;  
}
```

3 REVERSING LAYOUTS

You can reverse the direction! Items will appear in the opposite order.

flex-direction: row-reverse;



Right to Left

CSS

```
.container {  
  display: flex;  
  flex-direction: row-reverse;  
}
```

flex-direction: column-reverse;



Bottom to Top

CSS

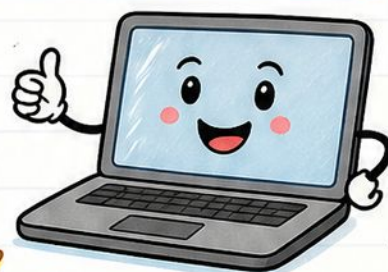
```
.container {  
  display: flex;  
  flex-direction: column-reverse;  
}
```

QUICK TIP

Think of flex-direction as the arrow that tells items where to go! 😊

REMEMBER!

Flexbox gives you superpowers to control layouts easily!



www.codescompiler.com



Module 7: Flexbox Made Easy

Flexbox helps us arrange items beautifully!



Lesson 3: Alignment



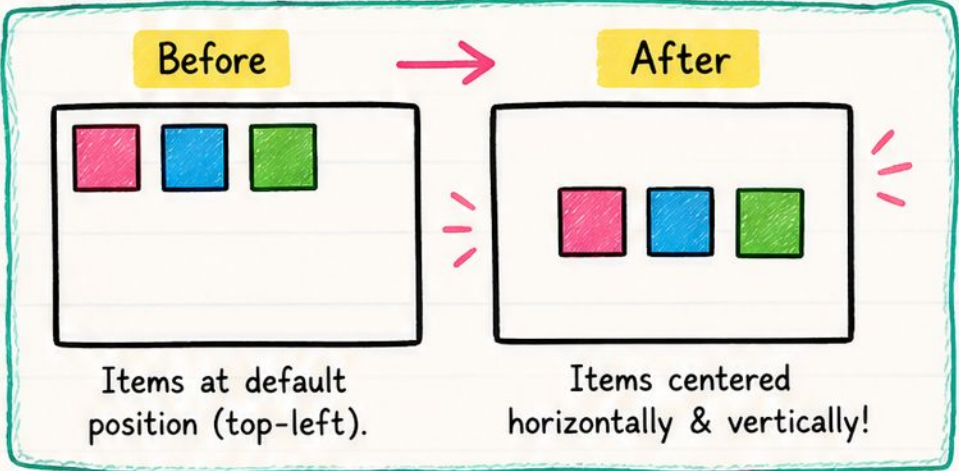
Alignment in Flexbox helps us control where items are placed along the **main axis (horizontal)** and **cross axis (vertical)**.



1 Centering Content

Center items both horizontally and vertically in a flex container.

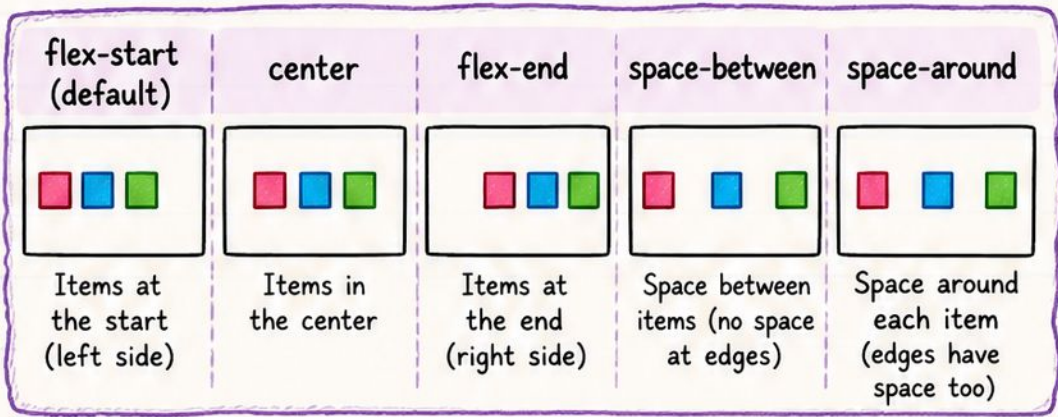
Use these properties:
justify-content: center;
align-items: center;



2 Horizontal Alignment

Align items along the **main axis** (left to right).

Use **justify-content**.

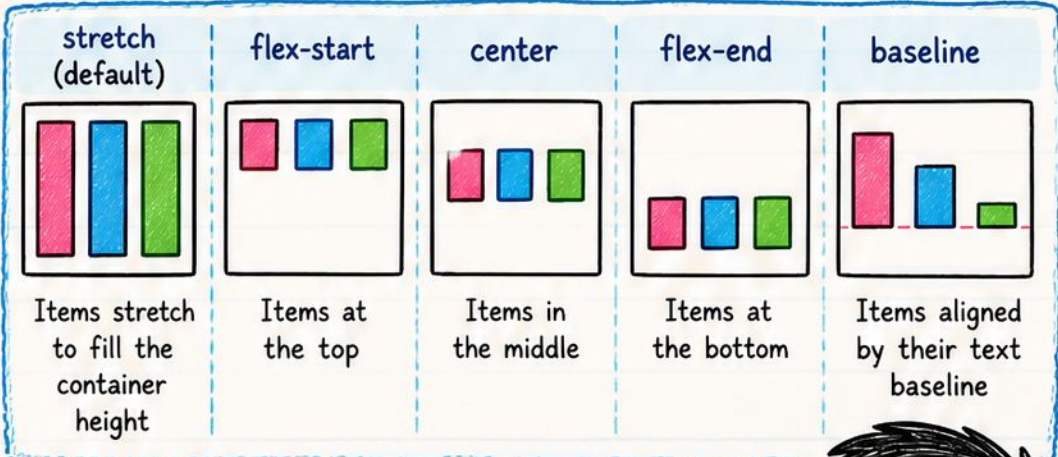


3 Vertical Alignment

Align items along the **cross axis** (top to bottom).

Use **align-items**.

Think:
Rows = horizontal
Columns = vertical



Pro Tip! Remember: **justify-content** works on the main axis (→) and **align-items** works on the cross axis (↓).





Module 7: Flexbox Made Easy



Lesson 4: Flex Wrapping

Let's make
our items wrap
like magic!



What is Flex Wrapping?

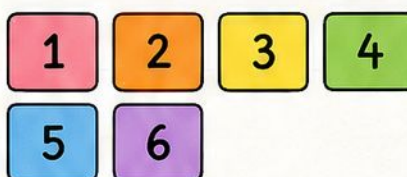
Flex wrapping allows items inside a flex container to move onto the next line when there isn't enough space.

No Wrapping (flex-wrap: nowrap)



All items stay in one line and may overflow the container.

Wrapping (flex-wrap: wrap)



Items move to the next line when space runs out.

Think of it like friends lining up in a photo. When there's no more room, they move to the next row!



Key Points

1 Multiple Rows

Flex wrapping lets items flow onto multiple rows automatically.



Looks neat and organized, even with many items!

2 Responsive Design

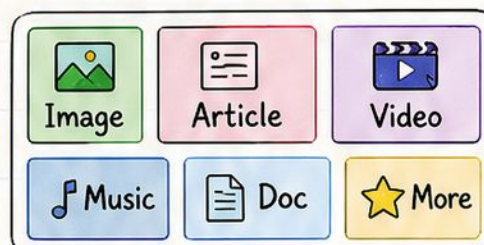
When the screen size changes, items wrap to fit, making your layout look great on any device.



One layout, works everywhere!

3 Organizing Content

Wrapping helps you keep content tidy and easy to read, without clutter.



Perfect for cards, galleries, menus, and more!

Quick Code

```
.container {  
  display: flex;  
  flex-wrap: wrap; /* Magic! */  
  gap: 16px; /* Space between items */  
}
```

This makes
items wrap
to the next
line!

Try It Out!

Resize your browser window and see how the items wrap automatically!



Keep it flexible, keep it awesome!

www.codescompiler.com



Let's Build Something Amazing!

Module 7: Flexbox Made Easy

FLEXBOX makes layout **EASY!**

Lesson 5: Mini Project

In this mini project, we'll build three awesome UI sections using Flexbox!

1 Navigation Menu

A clean and simple navigation bar to help users move around the website easily.

🚢 Oceanic [Home](#) [About](#) [Products](#) [Services](#) [Contact](#) [Sign Up](#)

★ Using Flexbox, the items are aligned in a row, spaced evenly, and look great on any screen!

2 Product Card Layout

A responsive row of product cards to showcase items beautifully.



Wireless Headphones
\$59.99

🛒 Add to Cart



Smart Watch
\$89.99

🛒 Add to Cart



Travel Backpack
\$49.99

🛒 Add to Cart



Flexbox helps the cards stay in a neat row and adjust nicely on different screen sizes!

3 Team Member Section

A row of team members to introduce the people behind the project.



Arjun Singh
Frontend Dev



Priya Sharma
UI/UX Designer



Rahul Verma
Backend Dev



Neha Kapoor
Project Manager



Flexbox keeps everyone in a row, evenly spaced and looking awesome together!

www.codescompiler.com

Teamwork makes the dream work!



Lesson 1: Introduction to Animations



1 Why Animations Matter



- ★ Animations make websites more engaging and fun.
- ★ They guide users' attention to important things.
- ★ They improve user experience and make interactions smooth.
- ★ They show feedback, changes and actions in a cool way!



2 CSS Animation Basics

CSS animations let you change CSS properties from one state to another over a period of time.

Key Ingredients

- ✓ **@keyframes** – Define the animation steps.
- ✓ **animation-name** – The name of your animation.
- ✓ **animation-duration** – How long the animation takes.
- ✓ **animation-iteration-count** – How many times it repeats.
- ✓ **animation-timing-function** – The speed curve of the animation.
- ✓ **animation-fill-mode** – What happens before and after.

Example:

```
@keyframes bounce {
  0% { transform: translateY(0); }
  50% { transform: translateY(-20px); }
  100% { transform: translateY(0); }
}

.box {
  width: 80px;
  height: 80px;
  background: #4CAF50;
  animation-name: bounce;
  animation-duration: 1s;
  animation-iteration-count: infinite;
  animation-timing-function: ease-in-out;
  border-radius: 10px;
}
```



3 Fun Examples

1. Hover Button



The button grows or changes color on hover.

2. Loading Spinner



Spinners show that something is loading.

3. Fade In Text

Hello!

Text appears smoothly with a fade in effect.

4. Moving Icons



Icons can move, spin or slide across the screen.

Lesson 2:

Transitions

Transitions help
us make changes
smooth and fun!

Key points to remember:

1 Smooth Changes

Transitions allow elements to change from one state to another smoothly over a period of time.



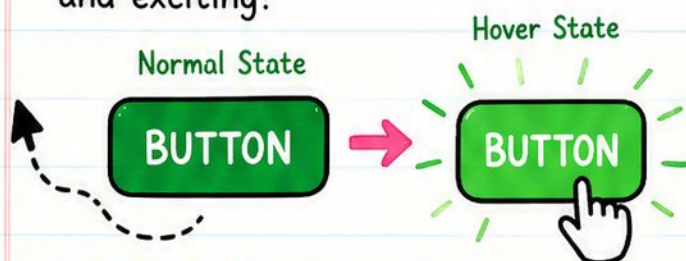
EXAMPLE

```
.box {  
  width: 100px;  
  height: 100px;  
  background: skyblue;  
  transition: all 0.5s ease;  
}  
  
.box:hover {  
  width: 200px;  
  background: dodgerblue;  
}
```

The box
changes
smoothly
when the
state
changes!

2 Hover Effects

Hover effects happen when we move our mouse over an element. Transitions make the effect smooth and exciting!



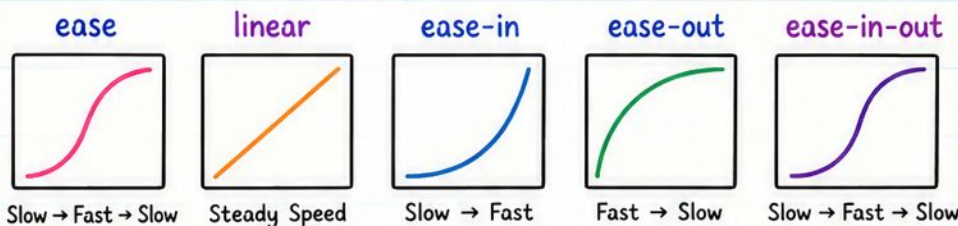
EXAMPLE

```
.btn {  
  padding: 10px 20px;  
  background: green;  
  color: white;  
  border: none;  
  border-radius: 6px;  
  transition: background 0.3s ease,  
              color 0.3s ease;  
}  
  
.btn:hover {  
  background: limegreen;  
  color: black;  
}
```

When you
hover over
the button,
the colors
change
smoothly!

3 Timing Functions

Timing functions control how the transition speeds up, slows down, or moves at a steady pace.



EXAMPLE

```
.box {  
  transition: all 1s ease-in-out;  
}
```

Other timing options:

- ease (default)
- linear
- ease-in
- ease-out
- ease-in-out
- cubic-bezier(n, n, n, n)

Transitions make websites feel **alive** and help guide your users' attention!

www.codescompiler.com



Module 8:

CSS Animations and Effects



Let's
Animate!



Lesson 3: Keyframe Animations

Keyframe animations let us control how an element changes at different points in time.



KEY POINTS

1 Creating Animations

We use `@keyframes` to define the animation steps. Then we apply the animation to an element.



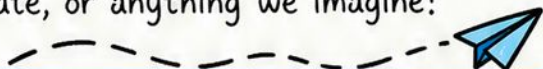
EXAMPLE

```
@keyframes fadeIn {  
  0% { opacity: 0; }  
  50% { opacity: 0.5; }  
  100% { opacity: 1; }  
}  
  
.box {  
  animation: fadeIn 2s ease-in-out;  
}
```

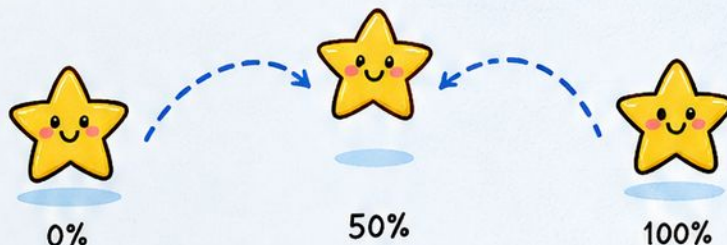
Define
steps at
different
points! 😊

2 Moving Objects

We can move elements from one position to another, change size, rotate, or anything we imagine!



EXAMPLE: MOVE



3 Repeating Effects

Animations can repeat! We can make them loop forever or play a certain number of times.



EXAMPLE: REPEAT

```
.heart {  
  animation: beat 1s infinite;  
}
```

infinite
= forever
loop! ❤️



QUICK TIP

Use small steps in `@keyframes` to create smooth and natural animations!



Animate your ideas
and bring your web
pages to life! ✨



Module 8:
CSS Animations
and Effects

Lesson 4:

Transformations

Change
the shape,
size or
position!



CSS transformations let us move, resize,
and rotate elements on a web page.



Key Points

Rotate

Scale

Translate



1

ROTATE



Rotates an element around
a fixed point (by default,
the center).

```
.box {  
  transform: rotate(45deg);  
}
```

BEFORE



AFTER (rotate 45deg)



Positive values rotate clockwise.
Negative values rotate counter-clockwise.

2

SCALE



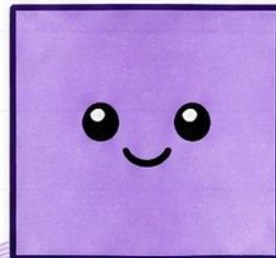
Changes the size of an element.
1 = normal size, > 1 = bigger,
< 1 = smaller.

```
.box {  
  transform: scale(1.5);  
}
```

BEFORE



AFTER (scale 1.5)



You can also use `scaleX()`, `scaleY()`,
or `scale(x, y)` for more control!

3

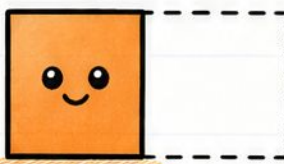
TRANSLATE



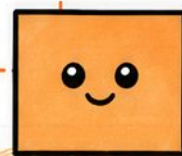
Moves an element from
its current position.
Use pixels (px), %, or other
length units.

```
.box {  
  transform: translate(60px, 30px);  
}
```

BEFORE



AFTER (translate 60px, 30px)

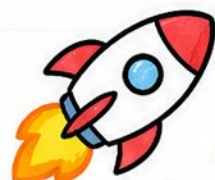


First value = move left/right (X axis)
Second value = move up/down (Y axis)

</>



www.codescompiler.com





MODULE 8



CSS ANIMATIONS AND EFFECTS



Lesson 5: Mini Project

KEY POINTS:

1

Animated Button

Create a button that changes color, grows, and glows when we hover over it!



2

Moving Rocket

Animate a rocket that moves across the sky with a smooth motion!



3

Dancing Character

Make a character dance using fun and cool animations!



Have Fun
Coding!



www.codescompiler.com



Module 9:

Responsive Web Design



Lesson 1:

What is Responsive Design?

Websites that look great on **every device!**



Responsive design means a website automatically adjusts its layout, images and content to fit any screen size.



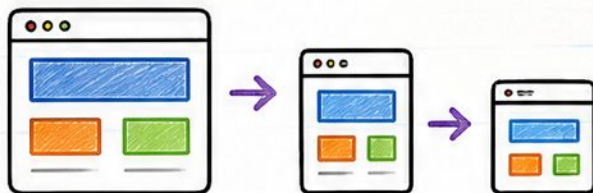
It gives a better look and feel on all devices.



Key Points

1

Different Screen Sizes



Screens come in many sizes—large, medium, small and extra small. Responsive design makes sure the website fits perfectly everywhere.

2

Mobile Devices



Phones are small but powerful! Responsive design makes websites easy to read, tap and navigate on mobile devices.

3

Tablets



Tablets have bigger screens than phones. Responsive design adjusts the layout to use the space beautifully.

4

Computers



On big screens, the website shows more content in a well-organized way for the best experience.



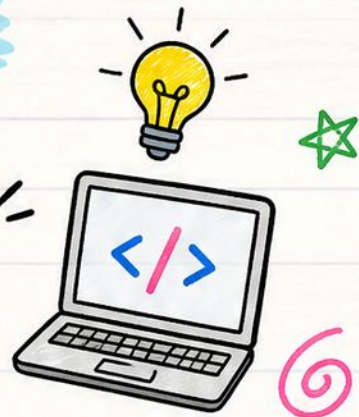
In Short:

Responsive design = One website, all devices, happy users!



Let's make websites that look great on any screen!

Lesson 2: CSS UNITS



CSS units tell the browser how big or small something should be. Different units help our website look good on different devices!



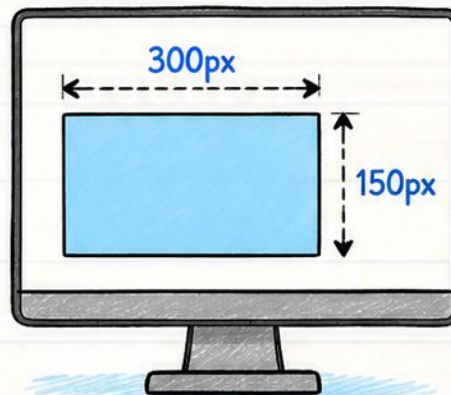
KEY POINTS

1 PIXELS (px)

Pixels are fixed units. They don't change size, no matter what screen or device is used.

GOOD TO KNOW!

Great for things that need a specific size, like icons or borders.



EXAMPLE

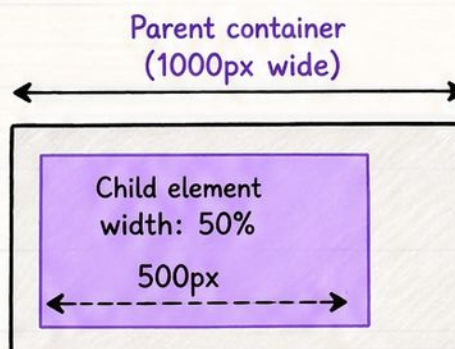
```
.box {
  width: 300px;
  height: 150px;
  background: #87cefa;
}
```

2 PERCENTAGES (%)

Percentages are relative units. They size things based on their parent element.

GOOD TO KNOW!

Super useful for flexible layouts that adjust to different screen sizes.



★ 50% of 1000px = 500px

EXAMPLE

```
.container {
  width: 1000px;
}
.box {
  width: 50%;
  background: #d8b4fe;
}
```

3 VIEWPORT UNITS (vw, vh, vmin, vmax)

Viewport units are based on the size of the browser window (viewport). They change when the screen size changes.

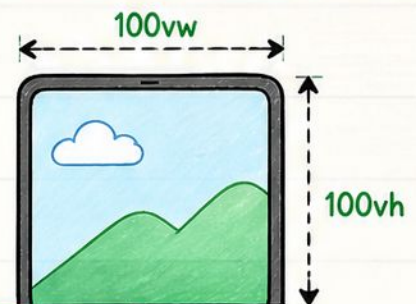
GOOD TO KNOW!

Perfect for full-screen sections and truly responsive designs!



Most Used Viewport Units

1vw	1% of viewport width (wide)
1vh	1% of viewport height (tall)
1vmin	1% of the smaller of width or height
1vmax	1% of the larger of width or height



EXAMPLE

```
.hero {
  width: 100vw;
  height: 100vh;
  background: #a7f3d0;
}
```



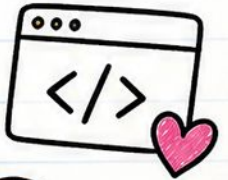
Use the right unit for the right job and your website will shine on every device! ✨



REMEMBER

Responsive websites use flexible units to adapt to any screen size!





Lesson 3: Media Queries



Media Queries help our websites look great on all devices by applying different styles based on the screen size and type.



KEY POINTS

1

Screen Detection

Media Queries detect the characteristics of a device, like **width**, **height**, and **orientation**. This helps CSS know which styles to apply.



EXAMPLE

```
@media (max-width: 600px) {  
  body {  
    background-color: lightblue;  
  }  
}
```

Checks if screen width is 600px or less!



2

Device-Specific Styling

We can apply different CSS styles for different devices like **phones**, **tablets**, **laptops**, and **desktops**.



PHONE
< 600px



TABLET
600px - 1024px



LAPTOP
1024px - 1440px



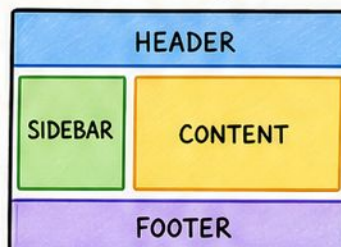
DESKTOP
> 1440px

3

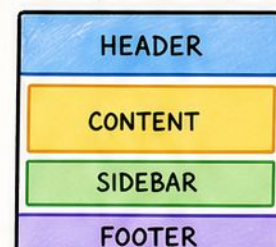
Layout Adjustments

Media Queries help us change the **layout** for better readability and **user experience** on different screen sizes.

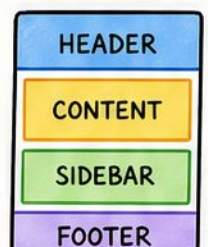
DESKTOP
(> 1024px)



TABLET
(600px - 1024px)



PHONE
(< 600px)

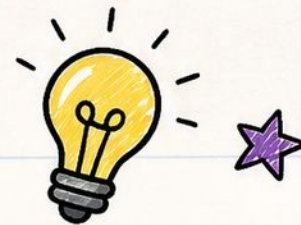


One website,
many devices,
one goal:
Great Experience!



Think Responsive,
Build Awesome!





Lesson 4: Responsive Images and Text

What is it?

Making sure images and text look great on all devices and screen sizes! ❤️

KEY POINTS

1

Flexible Images

Images resize to fit the screen. They never overflow or break the layout.

Same image, different sizes!



Laptop



Tablet



Mobile

2

Adaptive Fonts

Text adjusts its size and spacing so it's easy to read on any device.

Big screen



Small screen



3

Better User Experience

Happy users stay longer!
A responsive design makes websites easy, beautiful, and fun to use.



Remember! ★

A responsive website looks good everywhere and works for everyone!



www.codescompiler.com



Module 9:

Responsive Web Design



BUILD ONCE,
WORKS EVERYWHERE!



Lesson 5: Mini Project



KEY POINTS



1

Mobile-Friendly Website

A mobile-friendly website looks great and works smoothly on all devices, especially phones and tablets.



2

Responsive Portfolio

A responsive portfolio adapts to every screen size and beautifully showcases your work.



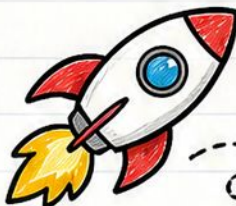
3

Responsive Blog Page

A responsive blog page adjusts layout for any device, making reading easy and enjoyable.



RESPONSIVE DESIGN
= BETTER EXPERIENCE
FOR EVERYONE! ♥



Think Responsive,
Think Future!



DESIGNING A COMPLETE WEBSITE

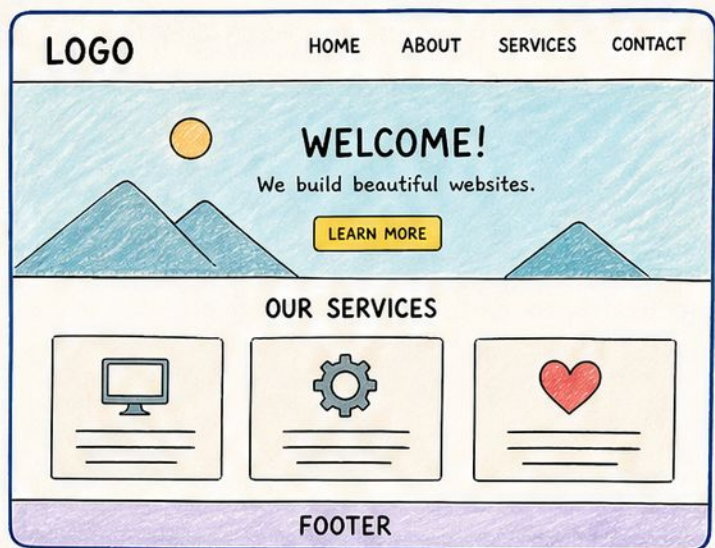
A great website starts with a great plan!
Let's design it step by step.

1 PLANNING THE DESIGN



Plan before you design!
A good plan helps your website look neat, clear, and easy to use.

- Decide the purpose of your website.
- Sketch the layout (header, main, footer).
- Think about what pages you need.



2 CHOOSING COLORS



Colors set the mood of your website. Choose colors that look good together and are easy to read.

- Use 2-3 main colors.
- Make sure text and background have good contrast.
- Use a consistent color scheme across all pages.

COLOR PALETTE EXAMPLES

Calm & Professional



Fun & Friendly



Modern & Bold



3 ORGANIZING CONTENT

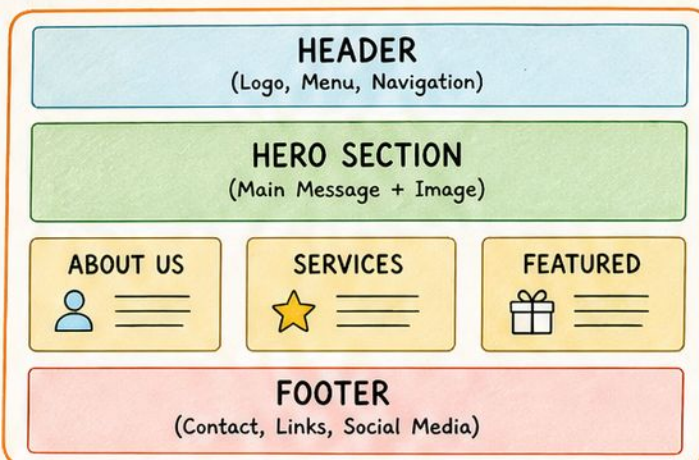


Keep it:
✓ Clear
✓ Simple
✓ Useful

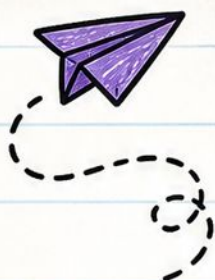
Good content organization helps visitors find what they need quickly.

- Put important information at the top.
- Use headings, short text, and images.
- Keep the flow simple:

Header → Main → Footer



A well-planned design + the right colors + organized content = an amazing website!



Lesson 2: Personal Portfolio Website



A great portfolio tells your story
and shows your skills to the world!



1

About Me Section

Introduce yourself! Share your name, a short bio, your interests and what you love to do.

→ Make it friendly, simple and engaging.



Hi, I'm Alex!

I'm a web developer who loves creating beautiful and useful websites.

I enjoy coding, drawing and learning new things.

[More About Me](#)

2

Skills Section

Show your skills and tools you are good at.

→ Use icons, progress bars or badges to make it look awesome!



My Skills

	HTML	90%		Bootstrap	80%
	CSS	85%		React	70%
	JavaScript	75%		Git & GitHub	85%

3

Contact Area

Let visitors get in touch with you easily.

→ Add a contact form, email, and your social media links.



Get in Touch

[Send Message](#)

Let's Connect!

hello@alexdev.com

www.alexdev.com

@alex.dev

linkedin.com/in/alexdev

github.com/alexdev



Keep building,
keep improving!



Lesson 3:

Kids Learning Website

In this lesson, we will design an awesome Kids Learning Website using HTML & CSS. We will focus on these key parts:

1 Home Page Design

The home page is the first thing users see. It should look friendly, colorful and easy to understand.

Includes:

- ✓ A welcoming banner
- ✓ Short description
- ✓ Call-to-action button
- ✓ Fun illustrations



2 Course Cards

Course cards show different topics in a clean and attractive way. They help users explore and choose what to learn.

Includes:

- ✓ Course image / icon
- ✓ Course title
- ✓ Short description
- ✓ "Learn More" button



3 Navigation Menu

The navigation menu helps users move around the website easily. It should be simple, clear and easy to use.

Includes:

- ✓ Logo / Brand name
- ✓ Menu links (Home, Courses, About, Contact)
- ✓ Hover effect for interactivity
- ✓ Mobile-friendly menu



CSS Magic!

We will use CSS to add colors, fonts, spacing, shadows, hover effects and make the website responsive for all devices!

How It All Comes Together



What we learn:

- ✓ Create a beautiful home page
- ✓ Design course cards with CSS
- ✓ Build a clean and responsive navigation menu
- ✓ Make the website kids-friendly and engaging

Let's build
something
AWESOME!

Lesson 4:

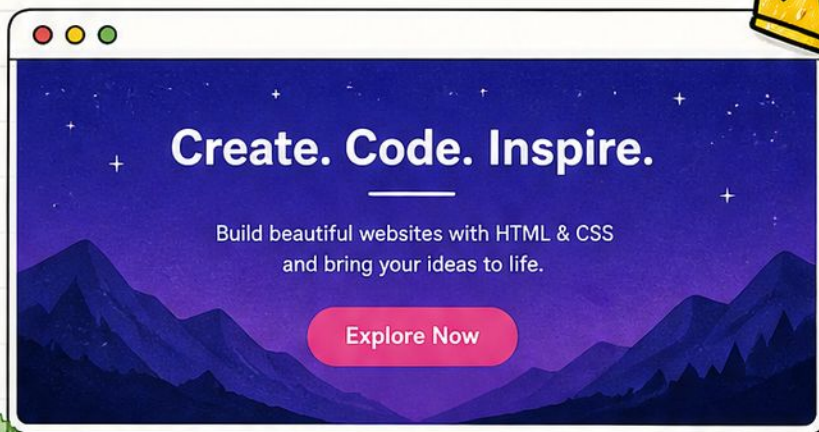
Creative Landing Page

In this lesson, we will design a stunning landing page using CSS creativity and cool effects!

1 Hero Banner

The hero banner is the big first section that grabs attention!

- ✓ Show a catchy headline
- ✓ Add a short description
- ✓ Use background image or gradient
- ✓ Make it visually appealing



```
.hero {  
  height: 100vh;  
  background: linear-gradient(135deg, #6a11cb, #2575fc);  
  color: white;  
  display: flex;  
  align-items: center;  
  justify-content: center;  
  text-align: center;  
}
```

First impressions matter!

2 Buttons and Effects

Buttons guide the user and make your page interactive!

- ✓ Use vibrant colors
- ✓ Add hover effects
- ✓ Change colors, scale or add shadows
- ✓ Make buttons stand out

Example Buttons

Get Started

Learn More

Join Now

```
.btn {  
  padding: 12px 28px;  
  border: none;  
  border-radius: 30px;  
  background: #2575fc;  
  color: white;  
  font-size: 16px;  
  font-weight: bold;  
  cursor: pointer;  
  transition: all 0.3s ease;  
}  
.btn:hover {  
  background: #6a11cb;  
  transform: translateY(-3px) scale(1.05);  
  box-shadow: 0 8px 20px rgba(0,0,0,0.2);  
}
```

Small effects make a big IMPACT! ✨

3 Call-to-Action Section

The CTA section encourages users to take action!

- ✓ Use a strong message
- ✓ Add a clear button
- ✓ Keep it simple and focused

Ready to Start Your Journey?

Join thousands of learners and start building your dream projects today!

Start Now

```
.cta {  
  background: #fff6e0;  
  text-align: center;  
  padding: 80px 20px;  
}  
.cta h2 { color: #333; }  
.cta p { color: #555; margin-bottom: 20px; }
```

A strong CTA turns visitors into happy users! 😊

Module 10: Final CSS Projects

Lesson 5: Final Capstone Project

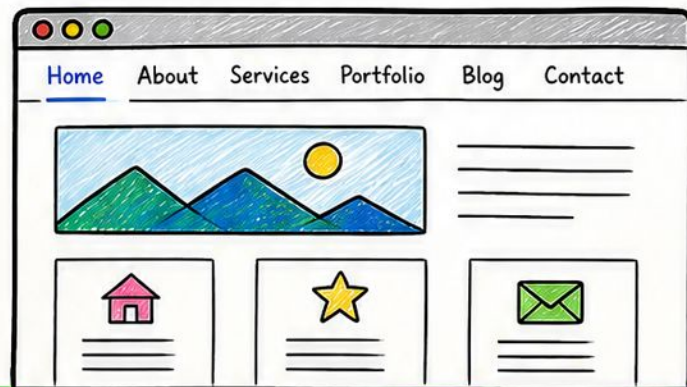


This is your final project! Build a complete website, make it look amazing on all devices, and showcase your creativity to the world! 😊

KEY POINTS

1 Complete Multi-Page Website

Create a fully functional website with multiple pages like Home, About, Services, Portfolio, Blog, and Contact.



2 Responsive Design

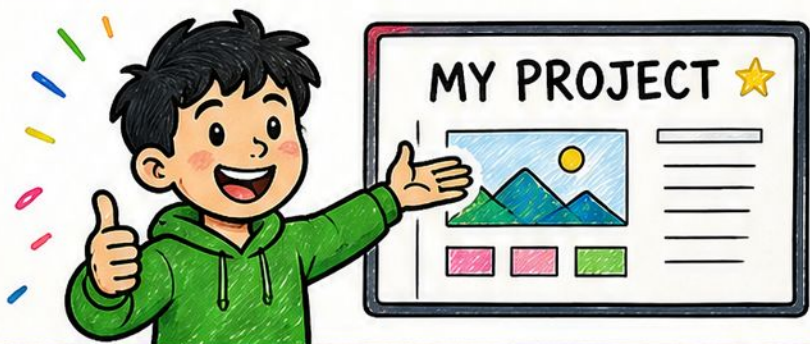
Make your website look perfect on all devices – desktops, tablets, and smartphones.



One Website,
All Screens!

3 Project Showcase

Present your project with confidence! Share your work, explain your ideas, and be proud of what you've built.



TIPS

- ✓ Plan your pages
- ✓ Use clean & organized code
- ✓ Test on different devices
- ✓ Add your creative touch!

Code with
passion,
create with
purpose!

YOU DID IT! ★

You've learned,
built, and grown.
The web is yours!

