

HTML | CSS

JAN 2023

5

thursday

(005-360) Week 01

[describe meaning in human and machine readable way]
names to meaning show
like content tag

J	Monday	30	2	9	16	23
A	Tuesday	31	3	10	17	24
N	Wednesday	4	11	18	25	
2		5	12	19	26	
0		6	13	20	27	
2		7	14	21	28	
3	Sunday	1	8	15	22	29
	Week	53/5	1	2	3	4

1 Semantic HTML

Tags tell both browser and developers what kind of content is inside.

- using tags that convey meaning about the content instead of just being containers.

Example - <header>, <nav>, <main>
<article>, <section>, <aside>
<footer>, <time>, <mark>.

Why matters -

Accessibility - Screen Reader know "this is nav" vs "just a div".

SEO - better Page Structure

Code clarity - developer instantly know the role of a tag.

Maintainability - easier to debug. extend style.

without

with

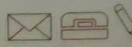
<div class="header"></div>

<header></header>

• self-descriptive

Questions

- What, why use - meaningful structure, accessibility
- diff <div>, <section> - div → generic, section → implies group related content
- When to use <article> - <article> is standalone and <section> - <section> is just grouping
- Accessibility Benefits - Screen Readers can announce role.
- SEO impact - Google crawlers understand



Example - Portfolio website <header> → intro links section
Project → footer

Blog → article post → <aside>

② Form attributes

action - Where to send data

method - GET / POST

Monday 13
Tuesday 14
Wednesday 15
Thursday 16
Friday 17
Saturday 18
Sunday 19

E

B

ENCODING 22 Use for file upload

2 9 16 23

target 24 open in same/new window

4 11 18 25

Autocomplete - on/off

5 6 7 8 9

③ Form Elements

<input> - text, password, email, no.

<textarea> - multi line text

<select> + <option> - dropdowns

<button> - Submit, reset, form actions

JAN 2023

(006-359) Week 01

[Typically Submit data to Server
Using GET or POST]

[User can collect data
value, age, email, username etc]

2. ① Forms Validation

• Collect user Input

• HTML provides built-in validation attributes

Required - mandatory input

type - "email", type = "number", etc.

pattern - regex for custom validation

min, max, minlength, maxlength.

 → new line

 → two line break

<label> → define a label for an input element

<button type="submit"> Submit </button>

→ Inner Text
shown on button

3. CSS Flexbox

Flexible box layout

Display: flex

• Used for 1D layout (row or column)

• Parent (container) Properties

main-axis - defined by display → flex, justify-content → main axis
flex-direction align-items → cross axis

cross-axis - perpendicular to main-axis flex-wrap → wrap items

• Child (item) properties

flex-grow, flex-shrink, flex-basis.

(Flexible like user area
row or column)

4. CSS Grid

1fr-1fr → fractional unit. • Used for 2D layout

Parent display - Grid, gap → space b/w items.

grid-template-columns, grid-template-rows.

• Child → grid-column, grid-row

JAN 2023

7

saturday

(007-358) Week 01

| | | | | | |
|---|-----------|------|----|----|----|
| N | Tuesday | 31 | 3 | 16 | 23 |
| | Wednesday | 4 | 10 | 17 | 24 |
| 2 | Thursday | 5 | 11 | 18 | 25 |
| 0 | Friday | 6 | 12 | 19 | 26 |
| 2 | Saturday | 7 | 13 | 20 | 27 |
| 3 | Sunday | 1 | 8 | 14 | 21 |
| | Week | 53/5 | 1 | 2 | 3 |
| | | | | | 4 |

Properties: top, left, right, bottom, z-index

5. Positioning :- Static → default (elements in normal flow)
relative → positioned relative to itself

Floating
Absolute → " " to nearest positioned ancestor
Fixed → relative to viewpoint, stay on scroll
Sticky → Switches b/w relative and fixed
(Stick when crossing threshold)

6. Responsive Design : • Goal - make layout adapt to screen sizes.
making web pages look good and work properly on all devices.

- Mobile first → start with small screen
- Use media queries

@media → conditional rule → apply only if condition true
(used for better design in desktop, tablet, mobile)

max-width → applies styles up to that width

min-width → applies style from that width and larger

• Uses flexible layouts, Percentage, relative units, media queries.

7. CSS Units :-

px → absolute pixels.

it defines lengths, sizes em → relative to parent font size and spacing for elements like rem → relative to root font size

width, height, padding, margin, font-size, vw, vh → " " viewport height / width rem for consistent typography vw/vh for full-screen layouts

8. CSS Specificity and Cascade

Cascade → order of rules matters.

Specificity order → inline styles → style = "", ID = #id, classes, attributes, Pseudo classes → .class



element → div p

important → overrides all, but avoid in production.

JAN 2023

monday

9

(009-356) Week 02

SEMANTIC HTML

- difference b/w main and body
`<body>` → everything `<main>` → main unique content
- Can use multiple header & footer
 - Yes each article and section have own header, footer, but main must be unique.
 - When use no `<div>` instead of semantic.
 - When no semantic meaning applies.
- `<aside>` and `<section>` difference.
`<aside>` → tangential / secondary content
`<section>` → part of main flow.
- Can we nest `<section>` inside `<article>`?
Yes, blog article with subsections `<section>` for comments, related posts.
- Why need semantic when CSS can do styling.
CSS controls presentation, Semantic HTML controls meaning. Both uses for modern, accessible and SEO friendly web apps.

① Form - Validation

validation ensure that user input is correct and secure.

• Types of validation

- ① HTML5 Validation (Client-side, built-in)
 - required, minlength, maxlength, pattern, type=email/no-email, min, max
 - Browser automatically validates before submission.

• addEventlistener → add event so when form submitted this function runs.

JAN 2023

10

event → give us control what happens to form submission
event.preventDefault → stop form from Submission if Validation fails
then need to highlight or style
 → inline container element

tuesday (010-355) Week 02

Monday 30 2 9 16 23
Tuesday 31 10 17 24
Wednesday 1 11 18 25
Thursday 2 12 19 26
Friday 3 13 20 27
Saturday 4 14 21 28
Sunday 5 15 22 29

Week

53/5 1 2 3 4

② JavaScript Validation (client-side, custom)

full control, can validate in real-time or

on form submit.

9.30

③ Server-Side Validation (Secure, final check)

- Always validate again on server (never trust only client)
- En - check SQL injection, XSS, invalid inputs.

12.00

④ Advanced Validation Concepts

RegEx validation - pattern like email, phone no, pass

Real-time " " - validate as user type

Custom error message - Using JS or setCustomValidity

Accessibility (A11y) - aria-invalid, aria-describedby.
accessibility

⑤ React Form-Validation React uses Controlled Components (form Input controlled by state)

- useState → Used to handle component state

(Value that change during user interaction)

- const → constant (Once we assign a value to const variable, cannot change the reference to something else).

Var, let → can be reassigned but const cannot.

- email → to store user input
- error → to store validation error message

Setemail / Seterror → update function to change the state.

if (!email.includes("@")) {
 show

setError ("Invalid email");
}

else {
 clear error show success Alert.

`type = "email"` → Specific email input
`value = &email` → make it a controlled component
~~`onchange = &e`~~ ⇒ `setEmail (e, target, value)` JAN 2023
update state when user type

| | | | | | |
|-----------|---|----|----|----|---|
| Monday | 6 | 13 | 20 | 27 | F |
| Tuesday | 7 | 14 | 21 | 28 | |
| Wednesday | 1 | 8 | 15 | 22 | B |
| Thursday | 2 | 9 | 16 | 23 | |
| Friday | 3 | 10 | 17 | 24 | |
| Saturday | 4 | 11 | 18 | 25 | |
| Sunday | 5 | 12 | 19 | 26 | |
| Week | 5 | 6 | 7 | 8 | 9 |

Update when user type

Questions

wednesday
(011-354) Week 02

11

- diff b/w GET or POST in forms

Get : append data to URL, less secure, length limit.
POST : Send data in body, more secure, no length limit.

- Form Input Type

Text, email, password, number, radio, file, date, range, color.

- What is novalidate in forms?

Disables HTML5 Validation.

- How To Handle real-time form validation in React.

use state hooks with `onchange` and `Validate` inputs dynamically.

ARIA - Accessible Rich Internet Applications

CSS FLEXBOX

Container Properties

- `display: flex` → active flexbox
- `flex-direction: row / column / row-reverse / column-reverse`
- `justify-content: flex-start / flex-end / center / space-between / space-around`
- `align-items: flex-start / center / stretch / baseline`
- `flex-wrap: nowrap / wrap / wrap-reverse`
- `align-content: control row when wrapping`

Item Properties

- `flex-grow`
- `flex-shrink` (how much item grow)
- `flex-basis (initial size)` shrinks
- `order (reorder flex items)`
- `align-self (override align-items for one item)`

FEBRUARY

MARCH

APRIL

Alenbon used for 1D layouts (navbar, list button)
Grid used for 2D " Full page layout, dashboards)

JAN 2023

12

thursday

(012-353) Week 02

CSS GRID

| | | | | | | |
|---|-----------|------|---|----|----|----|
| J | Monday | 30 | 2 | 9 | 16 | 23 |
| A | Tuesday | 31 | 3 | 10 | 17 | 24 |
| N | Wednesday | | 4 | 11 | 18 | 25 |
| 2 | Thursday | | 5 | 12 | 19 | 26 |
| 0 | Friday | | 6 | 13 | 20 | 27 |
| 2 | Saturday | | 7 | 14 | 21 | 28 |
| 3 | Sunday | | 8 | 15 | 22 | 29 |
| | Week | 53/5 | 1 | 2 | 3 | 4 |

Container Properties display: grid

- grid-template-columns: repeat(2, 1fr)
- grid-template-rows
- gap - 10px
- grid-auto-rows | grid-auto-columns
- justify-items & align-items (inside each cell)
- justify-content & align-content

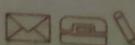
Item Properties

- grid-column: 1/3 (span column)
- grid-row: 2/4
- grid-area

Questions (Alenbon)

- diff b/w justify-content and align-items
justify-content aligns along the main-axis
align-items aligns along cross-axis
- when use flex-wrap
allow items to move onto new lines when they don't fit in row/column.
- diff b/w flex-grow, flex-shrink, flex-basis.
grow - expand space
shrink - shrink space
basis - initial size before growing/shrinking

- How to make an item span across multiple columns
grid-column: 1/3, grid-column: span 2.
- What is fr in Grid
fractional unit divide space proportionally.



SrcSet → short for Source Set

Purpose → provide multiple versions of an image so
browsers can choose the most appropriate image acc. to size and resolution

JAN 2023

| | | | | |
|-----------|---|----|----|----|
| Monday | 6 | 13 | 20 | 27 |
| Tuesday | 7 | 14 | 21 | 28 |
| Wednesday | 1 | 8 | 15 | 22 |
| Thursday | 2 | 9 | 16 | 23 |
| Friday | 3 | 10 | 17 | 24 |
| Saturday | 4 | 11 | 18 | 25 |
| Sunday | 5 | 12 | 19 | 26 |
| Week | 5 | 6 | 7 | 8 |
| | 9 | | | |

F
E
B

Agar ek folder m se ek html or css file hi tum leni
hो तो <link rel="stylesheet" href="style.css" type="text/css"/>

Responsive Design ↑

intentional has this
'liked'

Core Concepts

* Fluid layouts

Containers

width : 80% }

use %, em, rem, vw, vh instead fixed px.

* Viewport Meta Tag

Ensures Proper Scaling on mobile.

<meta name="viewport" content="....." />

* media Queries

CSS rule applied conditionally, based on device features. like (width)

* mobile-first Approach

write style for mobile first → then use min-width to add rule for larger devices.

• why? → better performance, Progressive enhancement

Questions

• Responsive design?

• diff b/w Responsive - flexible, adjust dynamically
adaptive - fixed layout, specific break down.
fluid design - use % units, scale proportionally

• Why mobile first is recommended

• What media queries? Syntax.

Conditional CSS Rule applied

@media (max-width : 768px) {

body { background : lightblue; } }

• How test Responsiveness

Resize browser manually, DevTools Device Simulator.
Online Tools.

• diff

min-width → mobile first (style apply above that size)

max-width - desktop first (below that size)



FEBRUARY

MARCH

APRIL

JAN 2023

14

saturday

(014-351) Week 02

CSS UNITS

| | | | | | | |
|---|-----------|------|---|----|----|----|
| J | Monday | 30 | 2 | 9 | 16 | 23 |
| A | Tuesday | 31 | 3 | 10 | 17 | 24 |
| N | Wednesday | | 4 | 11 | 18 | 25 |
| — | Thursday | | 5 | 12 | 19 | 26 |
| 2 | Friday | | 6 | 13 | 20 | 27 |
| 0 | Saturday | | 7 | 14 | 21 | 28 |
| 2 | Sunday | 1 | 8 | 15 | 22 | 29 |
| 3 | Week | 53/5 | 1 | 2 | 3 | 4 |

• Intro

Categorized As

- 9.00 Absolute Unit - (fixed sizes → do not scale with parent or screen)
- 9.30 Relative Units - (Scale relative to something → parent, root, viewport)
- 10.00

• Common CSS Units

- 11.00 • px (Pixels), Absolute unit
absolute unit.
12.00 Represent single dot on screen.
12.30 Does not scale with parent or viewport.
1.00 Use for borders, line adjustments, when precision
1.30 is needed.
- 2.00
- 2.30 • em (Relative to Parent Font Size)
3.00 relative unit based on font-size of parent element.
 $1em = 100\% \text{ of parent font size}$.
if parent has font-size: 16px, 1em = 16px.
Use for: Padding, margin, relative to parent
font-size.

15

sunday

(015-350) Week 02

rem (Relative to Root Font Size)

Relative to the root (`<html>`) font size.

$1rem = 100\% \text{ of root font size}$

does not compound like em.

use for font-sizes (keep consistency
across entire project).

$$1rem = 20px$$



| | | | | | | |
|-----------|---|----|----|----|---|-----------|
| Tuesday | 1 | 8 | 15 | 22 | . | B |
| Wednesday | 2 | 9 | 16 | 23 | . | Wednesday |
| Thursday | 3 | 10 | 17 | 24 | . | Thursday |
| Friday | 4 | 11 | 18 | 25 | . | Friday |
| Saturday | 5 | 12 | 19 | 26 | . | Saturday |
| Sunday | | | | | 3 | Sunday |
| Week | 5 | 6 | 7 | 8 | 9 | |

- **vh (viewport height)** ← Relative to
 $1vh = 1\%$ of browser window's height.
 Use for - full-screen sections, hero banners.
 $1vh = 8px$
- **vw (viewport width)** ← Relative to
 $1vw = 1\%$ of browser window's width.
 Use for : Responsive typography, fluid layouts.
 $1vw = 12px$

Questions

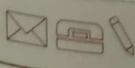
- Why prefer rem over em?
- To avoid unexpected compounding.
- easier to maintain global consistency.
- diff b/w em and rem
- When to use vh and vw
- What is the problem using only px.
 not responsive
- Breaks accessibility
- em of responsive typography using units?

Font Size: clamp(1.5rem, 5vw, 3rem);
 → minimum 1.5rem, scales with viewport, max 3rem.

CSS Specificity & Cascade

The Cascade word cascade refers to how styles are applied when multiple rules target the same element.

CSS follow three key principles.



- If two rule have the same specificity then the rule defined later in stylesheet (closer to element) wins.

| | | | | | | |
|----------|---|----|---|----|----|----|
| JAN 2023 | • avoid !important it makes CSS harder to maintain, breaks the normal Cascade tuesday and causes debugging headaches. | 30 | 2 | 9 | 16 | 23 |
| A | Tuesday | 31 | 3 | 10 | 17 | 24 |
| N | Wednesday | | 4 | 11 | 18 | 25 |
| | Thursday | | 5 | 12 | 19 | 26 |
| O | Friday | | 6 | 13 | 20 | 27 |
| Z | Saturday | | 7 | 14 | 21 | 28 |
| 3 | Sunday | 1 | 8 | 15 | 22 | 29 |
| Week | 53/5 | 1 | 2 | 3 | 4 | |

17

(017-348) Week 03

Source Order - two selector have same specificity. who come later win.
 Specificity - mathematical weight system determine which Selector is stronger.
 Importance - !Important override specificity.

Importance > Specificity > Source order

Specificity

is an Ranking System.

each selector have numeric value.

Formula

4 part value : (a, b, c, d)

a - Inline Styles (style = "")

b - IDs (#id)

c - classes (.class), attributes ([type = "text"]), pseudo-classes

d - Elements (div, h1) and pseudo-elements (:before, :after)

Weight

Inline style

(1, 0, 0, 0)

ID Selector

(0, 1, 0, 0)

Class, attribute, pseudo

(0, 0, 0, 1)

Element, Pseudo-element

(0, 0, 0, 0)

Universal (*)

* If two conflicting rules both have !important then specificity decides.

Inheritance Some CSS properties are inherited (like font-size, color) while others not (like margin, border)

You can force inheritance using : inherit



Stop inheritance using : initial or unset