Week 4

Links & Images (In Depth)

UMI202 • Web Design & Programming

Dr. Yakup Bakış

İstanbul Aydın Üniversitesi

Learning Objectives

What you should be able to do by the end of today

From Text Markup → Practical Navigation & Media
Goal: build clean, readable pages with working links & images
Today: URLs, paths, anchors, alt text, formats, sizing, responsive basics

- Explain what an <a> link actually does (requesting a resource via URL)
- Use absolute vs relative URLs correctly (and avoid common path mistakes)
- Create safe links that open in a new tab (target + basic safety)
- Link to sections on the same page using id + #fragment
- Use practical links: mailto: and tel:
- Add images with meaningful alt text and correct file paths
- Choose the right image format (JPG/PNG/SVG) and control size without distortion

Outcome

Mini project: Build a simple “Profile Card” page with photo + links.

Week 4 Quick Bridge

From text markup to links & images

- • Last time: structure & text markup (headings, paragraphs, emphasis)
- • Today: connect pages and add media
- • Links connect documents; images add meaning and visual context

Big idea

A website is not one page. It is a network of pages and assets connected by correct paths.

Common student problem

“It works on my computer” but breaks when folder structure changes. We fix this using relative paths and consistent structure.

Hyperlinks Recap

<a href=""> — what it really does

- A link points to a URL (address of a resource)
- <a> is an anchor element (a hyperlink)
- href = the destination address (URL or file path)

```
<a href="https://example.com">Visit
Example</a>

<a href="about.html">About</a>
```

- Clicking sends a request to that URL
- Clickable text should describe the destination

- The server responds with a page/file (HTML, image, PDF, etc.)

- • The browser navigates to it (or downloads it)

Key terms

URL, resource, request, response, navigation.

Absolute vs Relative URLs

External vs internal links

Absolute URL

Full address including protocol and domain.
Example: https://w3schools.com

Relative URL

Path from the current file to the target. path inside your site
Example: pages/about.html

```
<html>
<body>

<!-- Absolute -->
<a
href="https://w3schools.com">W3Schools</a
>

<!-- Relative (same site) -->
<a href="pages/about.html">About</a>
<a href="../index.html">Home</a>

</body>
</html>
```

Linking Within Your Own Site

Folders & paths & Structure

- Organize files in folders: index.html, /pages, /img, /css
- Same-folder: href="about.html«
- Child folder: href="pages/about.html"

- “./” means current folder

- “../” means go up one folder

- Case sensitivity matters on many servers (Image.jpg ≠ image.jpg)

```
Project/
  index.html
  pages/
    about.html
  img/
    me.jpg
```

```
<!-- from index.html -->
<a href="pages/about.html">About</a>

<!-- from pages/about.html -->
<a href="../index.html">Home</a>
```

target="_blank"

Open in a new tab (when to use)

- Opens the link in a new tab/window
- Use it for external sites or references students may want to keep open
- Do NOT overuse it (let users control navigation)

```
<a href="https://example.com"
target="_blank">
  External Resource
</a>
```

Teacher tip

Tell students: “External references can open in new tab. Internal navigation usually stays in the same tab.”

Safe Linking (Basic)

Why rel="noopener" matters

- When you open a new tab, the new page can sometimes access the old page
- Basic safe habit: add rel="noopener noreferrer" with target="_blank"
- You don’t need deep security today — just build the habit

```
<a href="https://example.com"
   target="_blank"
   rel="noopener noreferrer">
  External Resource
</a>
```

Rule of thumb

If target="_blank" → also add rel="noopener noreferrer".

Linking to Sections, Jump Links (Anchors):

id + #fragment (page anchors)

- Use anchors to jump within the same page

- You link to an element’s id using #id

Add id: <h2 id="contact">Contact</h2>

Link: Go to Contact

- Great for long pages: FAQ, documentation, table of contents

From another page: about.html#team

Useful for long pages and navigation menus

```
<!-- Link -->
<a href="#contact">Go to Contact</a>

<!-- Target section -->
<h2 id="contact">Contact</h2>
<p>Email: ...</p>
```

Email & Phone Links

mailto: and tel:

- mailto: opens the user’s default email app
- tel: starts a phone call on mobile devices
- Useful for contact pages and portfolios

```
<a href="mailto:yakup.bakis@iau.edu.tr">
  Email me
</a>

<a href="tel:+905001112233">
  Call us
</a>
```

Note

On desktop, tel: may not work unless a calling app is installed.

Link States (Concept)

normal / visited / hover / active

- Links can look different depending on user interaction
- Browsers apply default styles to links
- normal: default

- visited: user opened it before

- hover: mouse over the link

- active: when clicking

```
/* Concept only */
a { }
a:visited { }
a:hover { }
a:active { }
```

Why it matters

Better usability: users understand where they are and what they can click.

Good Link Text

Avoid “Click here”

- Link text should describe the destination
- Good for accessibility (screen readers) and usability
- Bad: “Click here” / “More” (no meaning)

Bad examples

Click here
More
Read this

Good examples

Download Week 4 Slides
View Course Syllabus (PDF)
Contact the Instructor

Debugging Links

404, wrong path, missing file , Typos

- Check:
- Most link bugs are path bugs
- Check: relative path (../) count
  - Folder path (../, pages/)
- Check: file name, folder name, extension (.html)

- Check: case sensitivity
- Is the file actually there?
- Use browser dev tools: Console + Network

```
Common errors:
- href="about" (missing .html)
- href="Pages/about.html" (wrong case)
- href="/pages/about.html" (root vs
relative)
```

Quick habit

After writing links, click every link once and fix immediately.

Images Basics

<img src="" alt=""> and why alt matters

- <img> embeds an image file into the page
- src = image path or URL to image (where the file is)
- alt = text alternative (for accessibility and when image fails)
- Always include alt — it’s professional

```
<img src="img/me.jpg"
     alt="Portrait photo of Yakup
Bakış">
```

Alt text rule

If the image is important content → describe it.
If purely decorative → alt="" (empty).

Image Paths

Relative paths + folder structure (Common Mistakes)

- Create an /img folder to keep images organized
- Use relative paths (img/photo.jpg) for portability
- Most common error: wrong folder level
- Remember: pages/about.html → ../img/photo.jpg

Correct: src="images/photo.jpg"
Wrong: src="C:\Users\..." (won’t work online)
Case sensitivity matters on servers
Keep images in /images folder

```
<!-- from index.html -->
<img src="img/logo.png" alt="Company
logo">

<!-- from pages/about.html -->
<img src="../img/me.jpg" alt="My
photo">
```

Debug tip

If image is broken: right click → “Open image in new tab” and read the URL.

Image Formats for Web

JPG vs PNG vs SVG (when to use)

JPG

Photos
Small file size
many colors
No transparency

PNG

Logos/icons
Transparency
sharp graphics
UI elements
Larger than JPG

SVG

Vector graphics
Scales perfectly on any screen
scalable without quality loss
Best for icons & logos
Often smaller than PNG for icons
Can be styled with CSS

GIF

simple animation
(limited colors)

- Rule: photos → JPG,
- UI graphics → PNG,
- icons/logos → SVG

- Avoid huge images: performance and user experience

Image Size Control

width / height + aspect ratio

- You can set width and height in HTML attributes or CSS
- But keep aspect ratio to avoid stretching
- Best practice: control size with CSS for responsive design

```
<img src="img/me.jpg"
     alt="My photo"
     width="300">
```

```
/* Basic responsive idea */
img { max-width: 100%; height: auto; }
```

Image as a Link

Wrap <img> inside <a>

- You can make an image clickable by putting it inside a link
- Common for logos that link to the home page
- Remember alt text still matters. Alt text should describe link destination

```
<a href="index.html">
  <img src="img/logo.png"
       alt="Company logo">
</a>
```

Usability

Clickable logo → users expect it to go to Home.

Figure & Caption

Clean semantics with <figure>+ <figcaption>

- <figure> groups an image and its caption

Use for images with captions

Clean semantics for content

- <figcaption> gives a description under the image

- This is clean and semantic HTML
- Better than random <p> under images

```
<figure>
  <img src="img/campus.jpg" alt="Campus
building">
  <figcaption>IAU Campus</figcaption>
</figure>
```

Responsive Images (Basic)

Make images fit the screen

- Goal: images fit inside the screen/container
- Problem: fixed-width images can overflow on mobile

- Solution: max-width: 100% keeps image inside its container

- height: auto keeps correct ratio

HTML attributes: quick sizing, limited control
CSS: best for consistent design
Example: img { max-width: 100%; height: auto; }
Use CSS for responsive behavior
Prevents overflow on small screens
Beginner-friendly responsive rule

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

Class tip

Use a browser window and resize it to show the effect live.

Image File Size & Performance

Why it matters

- Large images = slow pages (especially on mobile data)
- Users leave slow websites
- Resize images to display size
- Compress images appropriately (not today in depth)
- Choose the right format and size for the job

Rule of thumb

If an image is displayed at 400px wide, don’t upload a 4000px image.

Professional thinking

Web design is not only about looks — it is also about speed and user experience.

Common Image Problems

Broken src, stretching, huge files

- Broken image icon → wrong src path or missing file
- Distortion, Stretched image → wrong width/height ratio
- Blurry image → too small image scaled up
- Slow load page → huge file size
- Accessibility issue → missing alt

```
Checklist:
1) Is file in the right folder?
2) Is the name exactly the same?
3) Is the path correct (../)?
4) Is the extension correct
(.jpg/.png)?
```

Debug habit

Open the image URL directly to see what path the browser is trying.

Comments & Readable Code Style: Indentation, Clean Structure

- Indent nested tags consistently
- Comment when needed (explain WHY, not WHAT)
- Use simple file/folder names (avoid spaces)
- Readable code = fewer bugs

More HTML tags

- Tags Without Closing Tags
- They still have the 3 basic parts (opening/closing and content).
- do not require a formal </closingtag>
- Examples:
  - -- Image Tag
  -
 -- Line Break Tag
  - <input type="text" size="12" /> -- Input Field

5.3 Lists Overview

- Lists group related items
- Ordered and unordered types

Unordered Lists

- Bulleted items
- with <li> items
- Used for non-sequential data
- Each item uses
- Order is not important

- Unnumbered Lists:

```
<UL>
    <LI> apples </LI>
    <LI> bananas </LI>
    <LI> grapefruit </LI>
</UL>
```

- Unnumbered Lists with different pointer types:

```
<UL type="square">
    <LI> oranges </LI>
    <LI> peaches </LI>
    <LI> grapes </LI>
</UL>
```

type="square"
type="disc"
type="circle"

Ordered Lists

- Numbered sequence
- Shows steps or ranking
- with <li> items
- Used for steps or rankings
- Also uses

- Numbered Lists that starts with 4:

```
<OL start="4">
    <LI> oranges </LI>
    <LI> peaches </LI>
    <LI> grapes </LI>
</OL>
```

- Unnumbered Lists:

```
<UL>
    <LI> apples </LI>
    <LI> bananas </LI>
    <LI> grapefruit </LI>
</UL>
```

- Numbered Lists:

```
<OL>
    <LI> oranges </LI>
    <LI> peaches </LI>
    <LI> grapes </LI>
</OL>
```

- Numbered Lists with different ordering:

```
<OL type="a">
    <LI> oranges </LI>
    <LI> peaches </LI>
    <LI> grapes </LI>
</OL>
```

type="a": a, b, c
type="A": A, B, C
type="i": i, ii, iii
type="I": I, II, III

Put it all together so far

```
<HTML>
<HEAD>
<TITLE>The document title</TITLE>
</HEAD>

<BODY>
<H1>Main heading</H1>
<P>A paragraph.</P>
<P>Another paragraph.</P>
<UL> Things that I like </UL>
   <LI>A list item.</LI>
   <LI>Another list item.</LI>
</UL>
</BODY>
</HTML>
```

Nested Lists

- Lists inside lists
- Creates hierarchy
- Useful for menus and outlines
- Lists can exist inside other lists
- Creates hierarchy

5.4 div & span

- Generic containers with no semantic meaning
- Used for styling and grouping

div Element

- Block-level container
- Groups sections of content
- Used with CSS layout
- No semantic meaning by itself

span Element

- Inline container
- Used to style small text parts
- Does not break line
- Common with CSS classes

div vs span (Block vs Inline)

- = block element
- = inline element
- Both are non-semantic containers

Semantic vs Non-Semantic

- Prefer semantic tags when possible
- div/span only when needed
- Improves meaning and accessibility

When to Use div

- Layout grouping
- CSS styling hooks

When to Use span

- Styling small text parts
- Inline emphasis without semantics

Common Student Mistakes

- Using headings for size only
- Skipping heading levels
- Overusing div
- Forgetting list structure

Mini Exercise

- Create a page with headings
- Add 2 paragraphs
- Include ordered and unordered lists
- Use div and span for styling

Mini Practice

Build a “Profile Card” with Photo + Links

- Create a folder: Project/
- Add index.html and an img/ folder
- Add a profile photo into img/
- Add 3 links: external (new tab), mailto, and #contact anchor
- Add the photo with good alt text
- Bonus: make the photo clickable to open your portfolio link

Deliverable

A clean single page that demonstrates correct paths and accessible images.

Mini Practice 1: Build a “Profile Card” (Photo + Links)

- Create: name, short bio, image, 2 links (email + website)
- Use <figure> + <figcaption> for photo caption
- Use relative paths for image
- Use good link text

Mini Practice 2: Build a Clean “About Me” Page Using These Tags

- Requirements:
- 1) 1 internal link to a section (#id)
- 2) 1 external link with target=_blank
- 3) 1 image with correct alt
- 4) 1 list (skills or hobbies)
- 5) clean indentation

Week 4 Wrap-Up: Checklist + Next Week

- ✓ Absolute vs Relative URLs
- ✓ Internal links + folders
- ✓ Anchors (#id), mailto/tel
- ✓ img + alt + formats + sizing
- ✓ Responsive basics + semantics
- Next: Tables + Forms (intro)

Mini Practice: Starter Code

Copy and modify

```
<!doctype html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>My Profile Card</title>
</head>
<body>
  <h1>Your Name</h1>

  <a href="https://example.com" target="_blank" rel="noopener noreferrer">My Portfolio</a>
  | <a href="mailto:you@example.com">Email</a>
  | <a href="#contact">Contact Section</a>

  <p></p>

  <a href="https://example.com" target="_blank" rel="noopener noreferrer">
    <img src="img/me.jpg" alt="Your portrait photo" width="240">
  </a>

  <h2 id="contact">Contact</h2>
  <p>Phone: <a href="tel:+905001112233">+90 500 111 2233</a></p>
</body>
</html>
```

Quick Recap

What we learned today

- • <a href=""> links to a resource (URL) and triggers navigation
- • Absolute vs relative URLs — use relative for your own site
- • Folder structure is everything: ./ and ../
- • target="_blank" for external resources (with rel for safety)
- • Anchors: id + #fragment for page jumps
- • Images: <img src alt> with correct paths and meaningful alt text
- • Formats: JPG (photos), PNG (transparency), SVG (icons/logos)
- • Responsive basics: max-width: 100%, height: auto

Checkpoint (2 minutes)

Answer together

- • Q1: From pages/about.html, how do we link to img/logo.png?
- • Q2: Which is absolute? (a) pages/about.html (b) https://iau.edu.tr
- • Q3: What does alt text do?
- • Q4: When should we use target="_blank"?
- • Q5: Which format is best for a logo: JPG, PNG, or SVG? Why?

Tip

Let students answer. Correct gently. Make them explain the reasoning.

Homework

Submit next week

- • Create a small website folder with: index.html + pages/about.html + img/
- • Add at least 3 links: external (new tab), internal (about page), anchor link
- • Add at least 2 images with correct paths and alt text
- • Make the site readable: headings, paragraphs, and clean indentation
- • Zip the folder and submit on the course platform

Grading focus

Correct paths + meaningful link text + alt text.

Next Week

Week 5 preview: Tables & Forms

- • Tables: <table>, <tr>, <th>, <td> and structured tables (thead/tbody)
- • Forms: collecting user input (form, input, label, button)
- • Goal: build a simple contact form and display data structure

Bring your Week-4 folder to class — we will reuse it.