Complete HTML Learning Guide

**This comprehensive guide covers all HTML concepts from basic to advanced, combining theoretical explanations with practical code examples. Each section builds upon the previous one, creating a logical learning progression.**

# 1. HTML Headings

## Theory & Concepts

# HTML Headings - Overview  
  
## Purpose  
HTML headings create hierarchical structure in web pages.  
They range from <h1> (most important) to <h6> (least important).  
Headings help organize content and improve accessibility.  
  
## Key Concepts  
- <h1> is the main heading (only one per page)  
- <h2> through <h6> are subheadings  
- Headings create document outline  
- Screen readers use headings for navigation  
  
## Best Practices  
- Use only one <h1> per page  
- Follow logical hierarchy (h1 → h2 → h3)  
- Don't skip heading levels  
- Use headings for structure, not styling

## Practice Exercise

**Problem:**

Book  
 Chapter 1  
 Section 1  
 Section 2  
 Chapter 2  
 Section 1  
 Diagram 1  
 Chapter 3  
 Section 1  
 Section 2

## Solution

**HTML Code:**

<h1>Book</h1>  
<h2>Chapter 1</h2>  
<h3>Section 1</h3>  
<h3>Section 2</h3>  
<h2>Chapter 2</h2>  
<h3>Section 1</h3>  
<h4>Diagram 1</h4>  
<h2>Chapter 3</h2>  
<h3>Section 1</h3>  
<h3>Section 2</h3>

# 2. HTML Paragraphs

## Theory & Concepts

# HTML Paragraph Element - Summary  
  
## Overview  
The <p> element structures text on web pages.  
It uses opening <p> and closing </p> tags.  
Without it, all text runs together on one line.  
  
## Purpose  
Paragraph tags separate blocks of text.  
They make content clearer and more readable.  
Each paragraph appears on its own line.  
  
## Accessibility  
Screen readers recognize paragraph starts.  
This helps visually impaired users navigate content.  
They can understand the structure better.  
  
## Understanding Lorem Ipsum  
Lorem Ipsum is standard placeholder text.  
It simulates real content without meaning.  
Used for centuries in publishing and design.  
Generate it at lipsum.com or try fun alternatives.  
  
## Key Points  
- <p> element is essential for text structure.  
- Improves readability and accessibility.  
- Screen readers can navigate paragraphs.  
- Lorem Ipsum is useful for design.

## Practice Exercise

**Problem:**

First paragraph. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna  
aliqua. Arcu cursus vitae congue mauris. In nisl nisi scelerisque eu ultrices vitae auctor eu augue. Nisi est sit amet  
facilisis magna. Diam sit amet nisl suscipit adipiscing bibendum est ultricies integer. Quis ipsum suspendisse ultrices  
gravida dictum fusce ut. Euismod elementum nisi quis eleifend. Habitant morbi tristique senectus et. Amet nisl suscipit  
adipiscing bibendum est ultricies integer. Viverra orci sagittis eu volutpat odio facilisis mauris sit. Nisi quis  
eleifend quam adipiscing. Neque convallis a cras semper auctor neque vitae. Magna fermentum iaculis eu non. Vivamus arcu  
felis bibendum ut tristique et. Justo nec ultrices dui sapien eget mi. In vitae turpis massa sed elementum tempus. Eu  
facilisis sed odio morbi quis commodo. Sagittis aliquam malesuada bibendum arcu vitae elementum curabitur vitae.  
  
Second paragraph. Suscipit adipiscing bibendum est ultricies. Tortor aliquam nulla facilisi cras fermentum. Eget aliquet nibh praesent  
tristique magna. In hac habitasse platea dictumst vestibulum. Ornare quam viverra orci sagittis eu. Sit amet est  
placerat in. Proin fermentum leo vel orci porta non pulvinar neque laoreet. Turpis in eu mi bibendum neque egestas  
congue. Enim eu turpis egestas pretium aenean pharetra magna ac placerat. Ultrices sagittis orci a scelerisque purus  
semper eget duis at. Egestas egestas fringilla phasellus faucibus scelerisque eleifend donec pretium. Condimentum  
lacinia quis vel eros donec ac odio.  
  
Third paragraph. Nisl purus in mollis nunc sed id semper risus. Ipsum a arcu cursus vitae congue mauris rhoncus aenean. Ridiculus mus  
mauris vitae ultricies leo integer malesuada nunc. In tellus integer feugiat scelerisque. Lectus mauris ultrices eros in  
cursus turpis massa. Sollicitudin ac orci phasellus egestas. Massa massa ultricies mi quis hendrerit dolor. Quam  
elementum pulvinar etiam non quam lacus suspendisse faucibus interdum. Iaculis nunc sed augue lacus viverra. Id ornare  
arcu odio ut sem nulla pharetra. Amet luctus venenatis lectus magna fringilla urna porttitor. Eu nisl nunc mi ipsum  
faucibus vitae aliquet nec ullamcorper. Nunc mattis enim ut tellus elementum sagittis. Mauris augue neque gravida in  
fermentum et sollicitudin. Pellentesque habitant morbi tristique senectus. Tristique senectus et netus et. Turpis  
egestas sed tempus urna et pharetra pharetra. Feugiat vivamus at augue eget arcu dictum varius duis at. Lacus sed  
viverra tellus in hac habitasse platea dictumst vestibulum. Nisl condimentum id venenatis a condimentum vitae sapien.

## Solution

**HTML Code:**

<p>First paragraph. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna  
aliqua. Arcu cursus vitae congue mauris. In nisl nisi scelerisque eu ultrices vitae auctor eu augue. Nisi est sit amet  
facilisis magna. Diam sit amet nisl suscipit adipiscing bibendum est ultricies integer. Quis ipsum suspendisse ultrices  
gravida dictum fusce ut. Euismod elementum nisi quis eleifend. Habitant morbi tristique senectus et. Amet nisl suscipit  
adipiscing bibendum est ultricies integer. Viverra orci sagittis eu volutpat odio facilisis mauris sit. Nisi quis  
eleifend quam adipiscing. Neque convallis a cras semper auctor neque vitae. Magna fermentum iaculis eu non. Vivamus arcu  
felis bibendum ut tristique et. Justo nec ultrices dui sapien eget mi. In vitae turpis massa sed elementum tempus. Eu  
facilisis sed odio morbi quis commodo. Sagittis aliquam malesuada bibendum arcu vitae elementum curabitur vitae.</p>  
  
<p>Second paragraph. Suscipit adipiscing bibendum est ultricies. Tortor aliquam nulla facilisi cras fermentum. Eget aliquet nibh praesent  
tristique magna. In hac habitasse platea dictumst vestibulum. Ornare quam viverra orci sagittis eu. Sit amet est  
placerat in. Proin fermentum leo vel orci porta non pulvinar neque laoreet. Turpis in eu mi bibendum neque egestas  
congue. Enim eu turpis egestas pretium aenean pharetra magna ac placerat. Ultrices sagittis orci a scelerisque purus  
semper eget duis at. Egestas egestas fringilla phasellus faucibus scelerisque eleifend donec pretium. Condimentum  
lacinia quis vel eros donec ac odio.</p>  
  
<p>Third paragraph. Nisl purus in mollis nunc sed id semper risus. Ipsum a arcu cursus vitae congue mauris rhoncus aenean. Ridiculus mus  
mauris vitae ultricies leo integer malesuada nunc. In tellus integer feugiat scelerisque. Lectus mauris ultrices eros in  
cursus turpis massa. Sollicitudin ac orci phasellus egestas. Massa massa ultricies mi quis hendrerit dolor. Quam  
elementum pulvinar etiam non quam lacus suspendisse faucibus interdum. Iaculis nunc sed augue lacus viverra. Id ornare  
arcu odio ut sem nulla pharetra. Amet luctus venenatis lectus magna fringilla urna porttitor. Eu nisl nunc mi ipsum  
faucibus vitae aliquet nec ullamcorper. Nunc mattis enim ut tellus elementum sagittis. Mauris augue neque gravida in  
fermentum et sollicitudin. Pellentesque habitant morbi tristique senectus. Tristique senectus et netus et. Turpis  
egestas sed tempus urna et pharetra pharetra. Feugiat vivamus at augue eget arcu dictum varius duis at. Lacus sed  
viverra tellus in hac habitasse platea dictumst vestibulum. Nisl condimentum id venenatis a condimentum vitae sapien.</p>

# 3. Self-Closing Tags

## Theory & Concepts

# Self Closing Tags - Summary  
  
## Overview  
Void elements don't contain any content.  
They don't need closing tags.  
Examples: <hr /> and <br />.  
  
## Syntax  
Use <tagname /> format.  
Forward slash before closing angle bracket.  
Example: <hr /> and <br />.  
  
## Important Note  
Forward slash is different from closing tag.  
It appears just before closing angle bracket.  
Not like </p> which is a closing tag.  
  
## Horizontal Rule (<hr />)  
Creates a horizontal line on the page.  
Visually separates sections of content.  
Place between paragraphs or sections.  
  
## Break Element (<br />)  
Inserts a line break within paragraph.  
Useful for addresses or poems.  
Preserves formatting and meaning.  
  
## Important Rules  
Don't use <br /> for new paragraphs.  
Use <p> tags for paragraphs instead.  
This ensures accessibility and semantic correctness.  
  
## HTML5 Compatibility  
Can write as <hr> or <br>.  
But including slash (/) is recommended.  
Makes code clearer and more readable.  
  
## Benefits  
Improves structure and accessibility.  
Especially helpful for screen readers.  
Makes your HTML more professional.  
  
## Best Practices  
Use <br /> for line breaks in paragraphs.  
Use <hr /> to separate sections.  
Compare code with solutions using diffchecker.com.

## Practice Exercise

**Problem:**

William Blake   
  
17 south molton street  
London  
W1K 5QT  
UK  
  
  
William Blake (28 November 1757 – 12 August 1827) was an English poet, painter, and printmaker. Largely unrecognised  
during his life, Blake is now considered a seminal figure in the history of the poetry and visual art of the Romantic  
Age. What he called his "prophetic works" were said by 20th-century critic Northrop Frye to form "what is in proportion  
to its merits the least read body of poetry in the English language".[2] His visual artistry led 21st-century critic  
Jonathan Jones to proclaim him "far and away the greatest artist Britain has ever produced".[3] In 2002, Blake was  
placed at number 38 in the BBC's poll of the 100 Greatest Britons.[4] While he lived in London his entire life, except  
for three years spent in Felpham,[5] he produced a diverse and symbolically rich collection of works, which embraced the  
imagination as "the body of God"[6] or "human existence itself".[7]  
  
Although Blake was considered mad by contemporaries for his idiosyncratic views, he is held in high regard by later  
critics for his expressiveness and creativity, and for the philosophical and mystical undercurrents within his work. His  
paintings and poetry have been characterised as part of the Romantic movement and as "Pre-Romantic".[8] In fact, he has  
been said to be "a key early proponent of both Romanticism and Nationalism".[9] A committed Christian who was hostile to  
the Church of England (indeed, to almost all forms of organised religion), Blake was influenced by the ideals and  
ambitions of the French and American revolutions.[10][11] Though later he rejected many of these political beliefs, he  
maintained an amiable relationship with the political activist Thomas Paine; he was also influenced by thinkers such as  
Emanuel Swedenborg.[12] Despite these known influences, the singularity of Blake's work makes him difficult to classify.  
The 19th-century scholar William Michael Rossetti characterised him as a "glorious luminary",[13] and "a man not  
forestalled by predecessors, nor to be classed with contemporaries, nor to be replaced by known or readily surmisable  
successors".[14]

## Solution

**HTML Code:**

<h1>William Blake</h1>  
  
<p>  
17 south molton street<br />  
London<br />  
W1K 5QT<br />  
UK<br />  
</p>  
  
<hr />  
  
<p>William Blake (28 November 1757 – 12 August 1827) was an English poet, painter, and printmaker. Largely unrecognised  
during his life, Blake is now considered a seminal figure in the history of the poetry and visual art of the Romantic  
Age. What he called his "prophetic works" were said by 20th-century critic Northrop Frye to form "what is in proportion  
to its merits the least read body of poetry in the English language".[2] His visual artistry led 21st-century critic  
Jonathan Jones to proclaim him "far and away the greatest artist Britain has ever produced".[3] In 2002, Blake was  
placed at number 38 in the BBC's poll of the 100 Greatest Britons.[4] While he lived in London his entire life, except  
for three years spent in Felpham,[5] he produced a diverse and symbolically rich collection of works, which embraced the  
imagination as "the body of God"[6] or "human existence itself".[7]</p>  
  
<p>Although Blake was considered mad by contemporaries for his idiosyncratic views, he is held in high regard by later  
critics for his expressiveness and creativity, and for the philosophical and mystical undercurrents within his work. His  
paintings and poetry have been characterised as part of the Romantic movement and as "Pre-Romantic".[8] In fact, he has  
been said to be "a key early proponent of both Romanticism and Nationalism".[9] A committed Christian who was hostile to  
the Church of England (indeed, to almost all forms of organised religion), Blake was influenced by the ideals and  
ambitions of the French and American revolutions.[10][11] Though later he rejected many of these political beliefs, he  
maintained an amiable relationship with the political activist Thomas Paine; he was also influenced by thinkers such as  
Emanuel Swedenborg.[12] Despite these known influences, the singularity of Blake's work makes him difficult to classify.  
The 19th-century scholar William Michael Rossetti characterised him as a "glorious luminary",[13] and "a man not  
forestalled by predecessors, nor to be classed with contemporaries, nor to be replaced by known or readily surmisable  
successors".[14]</p>

# 4. Project: Movie Ranking

## Theory & Concepts

# Favorite Movie Project Summary  
  
## Overview  
Build a simple HTML website for favorite movies.  
Track and recommend movies you enjoyed.  
Use various HTML elements to structure content.  
  
## Requirements  
- Use <h1> for website title.  
- Use <h2> for subtitle.  
- Add <hr> to separate sections.  
- List at least three movies with <h3>.  
- Add <p> describing why you liked each movie.  
  
## Customization  
You can add more HTML elements.  
Make it your own unique project.  
Express your personal movie preferences.  
  
## Benefits  
Helps you remember great movies.  
Share recommendations with others.  
Practice HTML structure skills.  
  
## Example Solution  
Title and subtitle at the top.  
Three movies with titles and descriptions.  
Horizontal rule for visual separation.  
  
## Next Steps  
Complete the project with your movies.  
Consider sharing a screenshot.  
Inspire others with your recommendations.

## Practice Exercise

**Problem:**

<!-- Write your code below -->

## Solution

**HTML Code:**

<h1>The Best Movies According to Angela</h1>  
<h2>My top 3 movies of all-time.</h2>  
<hr />  
<h3>Spirited Away</h3>  
<p>This is my favourite anime. I love the beautiful images.</p>  
<h3>Ex Machina</h3>  
<p>Really cool sci-fi movie.</p>  
<h3>Drive</h3>  
<p>Super beautiful film. Really artistic.</p>

# 5. HTML Lists

## Theory & Concepts

# HTML List Elements - Summary  
  
## Overview  
Lists organize information in structured format.  
Used across websites for various purposes.  
Examples: recipes, navigation menus, content organization.  
  
## Types of Lists  
  
### Unordered Lists (<ul>)  
Used when order doesn't matter.  
Display bullet points by default.  
Perfect for shopping lists or feature lists.  
Structure: <ul><li>item</li></ul>  
  
### Ordered Lists (<ol>)  
Used when order is important.  
Display numbered items automatically (1, 2, 3...).  
Ideal for step-by-step instructions or rankings.  
Structure: <ol><li>item</li></ol>  
  
## List Items (<li>)  
Both list types use <li> tags.  
Each item can contain text, images, or other HTML.  
Multiple items can be nested within a list.  
  
## Real-World Applications  
Recipe websites (ingredients as unordered, instructions as ordered).  
News websites (article lists, rankings).  
Navigation menus and feature lists.  
Any content needing structured presentation.  
  
## Practical Exercise  
Create a cinnamon roll recipe website.  
Dough ingredients: unordered list (order doesn't matter).  
Filling ingredients: unordered list (order doesn't matter).  
Instructions: ordered list (steps must be sequential).  
  
## Coding Best Practices  
Choose appropriate list type based on content.  
Use proper indentation for readability.  
Consider whether order matters when selecting type.  
Lists can be nested and combined with other HTML.  
  
## Key Takeaways  
- <ul> for unordered lists with bullet points.  
- <ol> for ordered lists with numbers.  
- <li> for individual list items.  
- Lists improve content organization and user experience.  
- Proper HTML structure enhances accessibility and SEO.

## Practice Exercise

**Problem:**

Angela's Cinnamon Roll Recipe  
  
Ingredients  
  
For the dough:  
¾ cup warm milk  
2 ¼ teaspoons yeast   
¼ cup granulated sugar  
1 egg plus 1 egg yolk  
¼ cup butter  
3 cups bread flour  
  
For the filling:  
2/3 cup dark brown sugar   
1 ½ tablespoons ground cinnamon  
¼ cup butter  
  
Instructions  
  
Mix the milk with the yeast, sugar, eggs.  
Melt the butter and add to the mixture.  
Add in the flour and mix until combined into a dough.  
Knead the dough for 10 minuites.   
Transfer the dough into a large bowl and cover with plastic wrap. Leave it somewhere to rise for 2 hours.  
After the dough has doubled in size, roll it out into a large rectangle.   
Melt the butter for the filling and mix in the sugar and cinnamon.  
Spread the filling onto the dough then roll the dough into a swiss roll.   
Cut the roll into 3cm sections and place flat into a baking tray.  
Pre-heat the oven to 350F or 180C, then bake the rolls for 20-25min until lightly brown.

## Solution

**HTML Code:**

<h1>Angela's Recipe for the Best Cinnamon Rolls </h1>  
  
<h2>Ingredients</h2>  
  
<h3>For the dough:</h3>  
<ul>  
 <li>¾ cup warm milk</li>  
 <li>2 ¼ teaspoons yeast </li>  
 <li>¼ cup granulated sugar</li>  
 <li>1 egg plus 1 egg yolk</li>  
 <li>¼ cup butter</li>  
 <li>3 cups bread flour</li>  
</ul>  
  
<h3>For the filling:</h3>  
<ul>  
 <li>2/3 cup dark brown sugar </li>  
 <li>1 ½ tablespoons ground cinnamon</li>  
 <li>¼ cup butter</li>  
</ul>  
  
  
<h2>Instructions</h2>  
  
<ol>  
 <li>Mix the milk with the yeast, sugar, eggs.</li>  
 <li>Melt the butter and add to the mixture.</li>  
 <li>Add in the flour and mix until combined into a dough.</li>  
 <li>Knead the dough for 10 minuites.</li>  
 <li>Transfer the dough into a large bowl and cover with plastic wrap. Leave it somewhere to rise for 2 hours.</li>  
 <li>After the dough has doubled in size, roll it out into a large rectangle.</li>  
 <li>Melt the butter for the filling and mix in the sugar and cinnamon.</li>  
 <li>Spread the filling onto the dough then roll the dough into a swiss roll.</li>  
 <li>Cut the roll into 3cm sections and place flat into a baking tray.</li>  
 <li>Preheat the oven to 350F or 180C, then bake the rolls for 20-25min until lightly brown.</li>  
</ol>

# 6. Nesting Lists

## Theory & Concepts

# NESTING AND INDENTATION - HTML LISTS  
  
## Introduction  
Builds upon basic ordered and unordered lists.  
Introduces nesting elements inside others.  
Learn complex nested lists and proper indentation.  
  
## Key Concepts  
  
### Nested Lists  
Lists can be embedded inside other list items.  
Creates hierarchical structure with visual indentation.  
Allows complex, multi-level list organization.  
  
### Code Structure  
Inside a list item (<li>), embed entire list.  
Write text first, then embed nested list.  
Results in "nested list" structure.  
  
### Indentation Importance  
Proper indentation makes code easier to read.  
Helps identify list hierarchy at a glance.  
Visual clues help understand structure quickly.  
Essential for debugging and error identification.  
  
## Complex Nested List Example  
Structure:  
- Unordered list with items: A, B, C.  
- Under item B: Ordered list with B1, B2, B3.  
- Under item B2: Unordered list with B2a, B2b, B2c.  
- Inside B2a: Unordered list with B2aa, B2ab.  
- Under item B3: Ordered list with B31, B32.  
  
## Closing Tags Rule  
When embedding list inside list item:  
Closing </li> tag does NOT go after text.  
Closing </li> tag goes AFTER nested list.  
Crucial for proper HTML structure.  
  
## Visual Studio Code Features  
Automatically indents nested elements as you write.  
Re-indents everything when you save (Cmd+S, Ctrl+S).  
Helps keep code clean and readable.  
Links opening tags to matching closing tags.  
  
## Debugging with Nesting  
Indentation helps diagnose and fix errors.  
Check indentation and matching tags for issues.  
Missing tags become visually apparent.  
Helps identify structural problems in HTML.  
  
## Best Practices  
Always use proper indentation for nested lists.  
Take advantage of VS Code's auto-indentation.  
Use indentation to verify tag matching.  
Preview code frequently while writing.  
  
## Coding Exercise  
Create complex nested list with:  
Main unordered list (A, B, C).  
Nested ordered list under B (B1, B2, B3).  
Further nested unordered list under B2.  
Additional nesting under B2a.  
Ordered list under B3 (B31, B32).  
  
## Key Takeaways  
Nested lists create hierarchical structure in HTML.  
Proper indentation is crucial for readability.  
VS Code automatically handles indentation.  
Indentation helps identify errors and missing tags.  
Complex lists need careful closing tag placement.  
  
## Next Lesson  
Will cover anchor elements and hyperlinks.

## Practice Exercise

**Problem:**

<!-- Write your code below -->

## Solution

**HTML Code:**

<ul>  
 <li>A</li>  
 <li>  
 B  
 <ol>  
 <li>B1</li>  
 <li>B2  
 <ul>  
 <li>B2a  
 <ul>  
 <li>B2aa</li>  
 <li>B2ab</li>  
 </ul>  
 </li>  
 <li>B2b</li>  
 <li>B2c</li>  
 </ul>  
 </li>  
 <li>B3  
 <ol>  
 <li>B31</li>  
 <li>B32</li>  
 </ol>  
 </li>  
 </ol>  
 </li>  
 <li>C</li>  
</ul>

# 7. Anchor Elements

## Theory & Concepts

# ANCHOR ELEMENTS - COMPLETE SUMMARY  
  
## Introduction to Anchor Elements  
Anchor elements (<a></a>) create hyperlinks in HTML.  
Basic structure: <a>link text</a>.  
Without attributes, they display text but aren't functional.  
  
## HTML Attributes  
Attributes provide extra information about HTML elements.  
Placed inside opening tag after element name.  
Structure: element\_name attribute\_name="value".  
Multiple attributes separated by spaces.  
  
## The HREF Attribute  
href = "Hypertext Reference".  
Specifies URL that hyperlink points to.  
Required to make anchor elements functional.  
Syntax: <a href="https://example.com">Link Text</a>.  
  
## Global Attributes  
Can be applied to any HTML element.  
Example: draggable attribute.  
draggable="true" - allows element to be dragged.  
draggable="false" - prevents dragging (default).  
  
## Practical Exercise: Favorite Websites List  
Create ordered list (<ol>) with 5 favorite websites.  
Each list item (<li>) contains anchor tag.  
Structure:  
<ol>  
 <li><a href="URL">Website Name</a></li>  
 <li><a href="URL">Website Name</a></li>  
</ol>  
  
## Ordered List Attributes  
start attribute: specifies starting number.  
Syntax: <ol start="5"> (starts from 5).  
Useful for customizing list numbering.  
  
## Key Syntax Rules  
Always use quotation marks around attribute values.  
URLs in href must be complete (include https://).  
Anchor text should be descriptive.  
Proper nesting: <ol><li><a></a></li></ol>.  
  
## Common Mistakes to Avoid  
Forgetting the href attribute.  
Missing quotation marks around URLs.  
Incorrect nesting of elements.  
Using incomplete URLs.  
  
## Best Practices  
Use descriptive link text.  
Ensure all links are functional.  
Test links after creation.  
Use proper HTML structure and indentation.  
  
## Next Steps  
This lesson builds foundation for complex HTML elements.  
Understanding attributes is crucial for HTML development.  
Practice creating various types of links.  
Explore additional anchor element attributes.  
  
## Key Takeaways  
Anchor elements create hyperlinks with <a> tag.  
href attribute is essential for functional links.  
Attributes provide additional functionality to HTML elements.  
Global attributes work on any element.  
Proper HTML structure is crucial for functional websites.

## Practice Exercise

**Problem:**

<h1>My top 5 Favourite Websites</h1>  
<!-- Write your code below -->

## Solution

**HTML Code:**

<h1>My top 5 Favourite Websites</h1>  
<ol>  
 <li><a href="https://www.producthunt.com/">Product Hunt</a></li>  
 <li><a href="https://smashthewalls.com/">Smash the Walls</a></li>  
 <li><a href="https://www.nytimes.com/games/wordle">Wordle</a></li>  
 <li><a href="https://hackertyper.com/">Hacker Typer</a></li>  
 <li><a href="https://stellarium-web.org/">Stellarium</a></li>  
</ol>  
  
<!-- Extra challenge Solution-->  
<ol start="5">  
 <li><a href="https://www.producthunt.com/">Product Hunt</a></li>  
 <li><a href="https://smashthewalls.com/">Smash the Walls</a></li>  
 <li><a href="https://www.nytimes.com/games/wordle">Wordle</a></li>  
 <li><a href="https://hackertyper.com/">Hacker Typer</a></li>  
 <li><a href="https://stellarium-web.org/">Stellarium</a></li>  
</ol>

# 8. Image Elements

## Theory & Concepts

# Image Elements Summary  
  
## Overview  
<img> adds images to websites.  
Structure: <img src="url" />.  
  
## Key Components  
- src: image source/location  
- Self-closing tag (void element)  
- alt: alternative text for accessibility  
  
## Important Features  
Self-closing tag, no closing tag needed.  
src is required, contains image URL.  
alt is highly recommended for accessibility.  
  
## Accessibility  
Screen readers use alt text for images.  
Alt text helps visually impaired users.  
Essential for accessible websites.  
  
## Image Types  
JPEG, PNG, GIF all work the same.  
GIFs animate automatically when used.  
Any image format works if URL is correct.  
  
## Example Usage  
<img src="https://picsum.photos/200/200" alt="Random placeholder image" />  
  
## Best Practices  
Always include alt text when possible.  
Use descriptive alt text for images.  
For decorative images, use alt="".  
Alt text should be meaningful and contextual.  
  
## Lesson Example  
Create webpage with h1 and image.  
Show cat or dog person with proper image and alt.

## Practice Exercise

**Problem:**

<!-- Kitten image URL -->  
https://raw.githubusercontent.com/appbrewery/webdev/main/kitten.jpeg  
  
  
<!-- Puppy image URL -->  
https://raw.githubusercontent.com/appbrewery/webdev/main/puppy.gif

## Solution

**HTML Code:**

<h1>I am a Cat Person</h1>  
<img src="https://raw.githubusercontent.com/appbrewery/webdev/main/kitten.jpeg" alt="kitten held in hand" />  
  
<h1>I am a Dog Person</h1>  
<img src="https://raw.githubusercontent.com/appbrewery/webdev/main/puppy.gif" alt="puppy digging in the sand" />

# 9. Birthday Invite Project

## Theory & Concepts

# Birthday Invite Project Summary  
  
## Project Overview  
Create a retro 90s style birthday invitation website.  
Combine all learned HTML elements in a project.  
Announce birthday details, date, images, and info.  
  
## Key Components  
- Main heading (h1) for "My Birthday"  
- Subheadings (h2, h3) for sections  
- Image with src and alt for accessibility  
- Unordered list for what to bring  
- Anchor tag linking to Google Maps  
  
## Technical Elements  
- HTML heading tags (h1, h2, h3)  
- Image tag with src and alt  
- Unordered list (ul) with list items (li)  
- Anchor tag (a) with href for navigation  
  
## Guidelines  
- At least one image, one list, one anchor tag  
- Customize based on your style  
- Focus on practice, not strict solutions  
  
## Learning Outcomes  
- Apply HTML in real-world scenario  
- Understand accessibility with alt text  
- Link to external resources (Google Maps)  
- Develop personal coding style and creativity  
  
## Project Result  
Shows integration of HTML concepts in a user-friendly invite.

## Practice Exercise

**Problem:**

<!-- Example image URL -->  
https://raw.githubusercontent.com/appbrewery/webdev/main/birthday-cake3.4.jpeg  
  
<!-- Example Google Maps Link -->  
https://www.google.com/maps/@35.7040744,139.5577317,3a,75y,289.6h,87.01t,0.72r/data=!3m6!1e1!3m4!1sgT28ssf0BB2LxZ63JNcL1w!2e0!7i13312!8i6656

## Solution

**HTML Code:**

<h1>It's My Birthday!</h1>  
<h2>On the 12th May</h2>  
  
<img src="https://raw.githubusercontent.com/appbrewery/webdev/main/birthday-cake3.4.jpeg"  
 alt="purple birthday cake with candles" />  
  
<h3>What to bring:</h3>  
<ul>  
 <li>Baloons (I love baloons)</li>  
 <li>Cake (I'm really good at eating)</li>  
 <li>An appetite (There will be lots of food)</li>  
</ul>  
  
<h3>This is where you need to go:</h3>  
<a  
 href="https://www.google.com/maps/@35.7040744,139.5577317,3a,75y,289.6h,87.01t,0.72r/data=!3m6!1e1!3m4!1sgT28ssf0BB2LxZ63JNcL1w!2e0!7i13312!8i6656">Google  
 map link</a>

# Conclusion & Next Steps

Congratulations! You've completed the comprehensive HTML learning journey. This guide has covered:  
  
• Basic HTML structure and headings  
• Text formatting with paragraphs  
• Self-closing tags for breaks and rules  
• Creating organized content with lists  
• Building complex nested structures  
• Adding interactivity with links  
• Including images with accessibility  
• Combining all concepts in practical projects  
  
## What You've Learned  
- Semantic HTML structure  
- Accessibility best practices  
- Code organization and readability  
- Real-world project implementation  
  
## Next Steps  
- Practice building your own websites  
- Explore CSS for styling  
- Learn JavaScript for interactivity  
- Build a portfolio of projects  
- Join web development communities  
  
## Resources for Continued Learning  
- MDN Web Docs (developer.mozilla.org)  
- W3Schools HTML Tutorial  
- FreeCodeCamp HTML Course  
- HTML5 Semantic Elements  
- Web Accessibility Guidelines (WCAG)  
  
Remember: The best way to learn is by doing. Keep coding, keep building, and keep learning!