WEB PROGRMMING WITH HTML, CSS, JS

LESSON 3

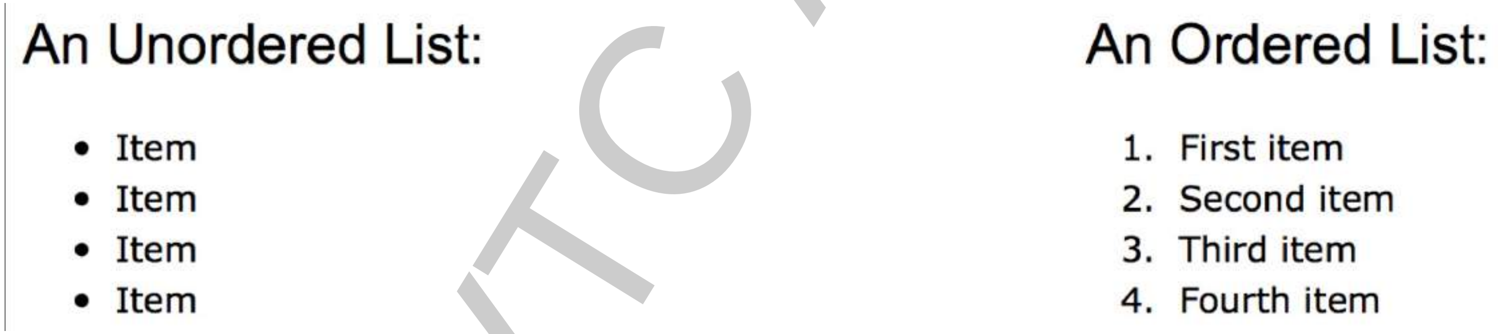
Organizing Content of Web Page

CONTENT

1. HTML list elements
2. Horizontal list with CSS
3. Figure and image element
4. Organizational elements: section, nav, article, aside, header, footer
5. CSS cascade and inheritance
6. Summary

Introduction to List Elements

- HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements. List items can contain new list, and other HTML elements, like images and links, etc.
- Types of list:
    - <ul> – An unordered list. This will list items using plain bullets.
    - <ol> – An ordered list. This will list items in ascending or descending order.
    - <dl> – A definition list. This will arrange your items like a dictionary.



An Unordered List:

- Item
- Item
- Item
- Item

An Ordered List:

1. First item
2. Second item
3. Third item
4. Fourth item

HTML Unordered Lists

- An unordered list is a collection of related items that have no special order or sequence.
- An unordered list starts with the <ul> tag.
- Each list item starts with the <li> tag.
- The list items will be marked with bullets (small black circles) by default:

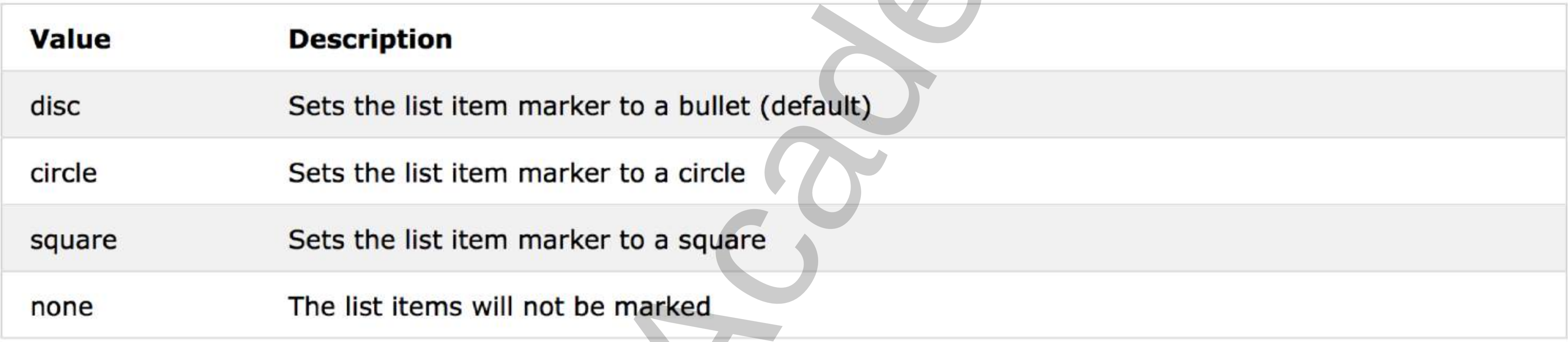
```
<ul>
  <li>Java</li>
  <li>C#</li>
  <li>Swift</li>
</ul>
```

→

- Java
- C#
- Swift

HTML Unordered Lists

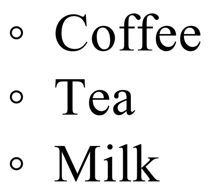
- The type attribute is used to define the style of the list item marker:



Value	Description
disc	Sets the list item marker to a bullet (default)
circle	Sets the list item marker to a circle
square	Sets the list item marker to a square
none	The list items will not be marked

- Example:

```
<ul type="circle">
  <li>Coffee</li>
  <li>Tea</li>
  <li>Milk</li>
</ul>
```



- ◦ Coffee
- ◦ Tea
- ◦ Milk

HTML Ordered Lists

- If you are required to put your items in a numbered list instead of bulleted, then HTML ordered list will be used. This list is created by using <ol> tag.
- The numbering starts at one and is incremented by one for each successive ordered list element tagged with <li>.
- The list items will be marked with numbers by default:

```
<ol>
  <li>Java</li>
  <li>C#</li>
  <li>Swift</li>
</ol>
```

1. Java
2. C#
3. Swift

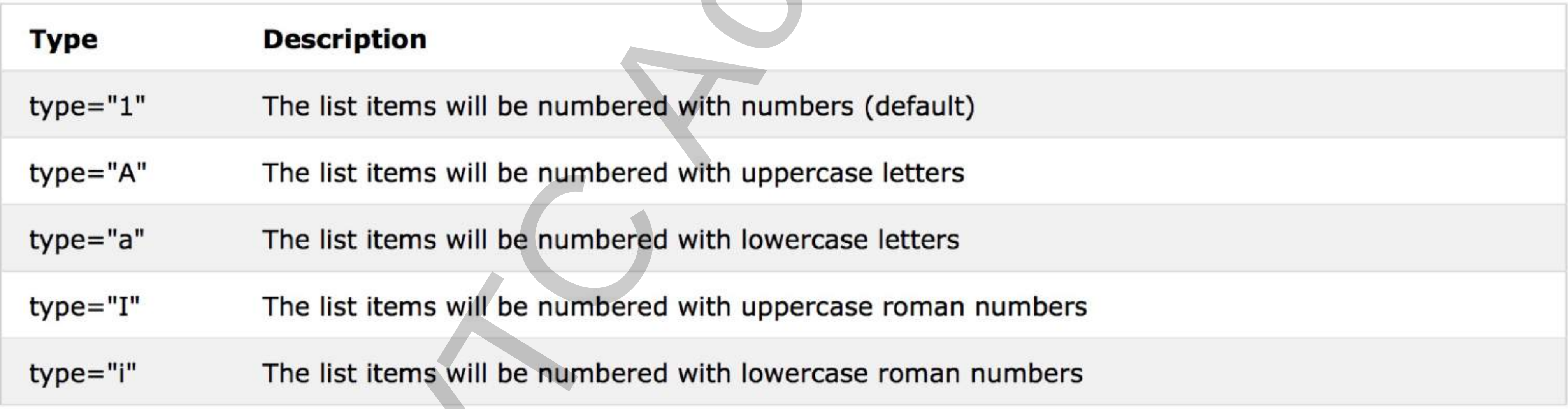
HTML Ordered Lists

- By default, an ordered list will start counting from 1. If you want to start counting from a specified number, you can use the start attribute:

```
<ol start="10">...</ol>
```

```
10. Java
11. …
```

- You can use type attribute for <ol> tag to specify the type of numbering:

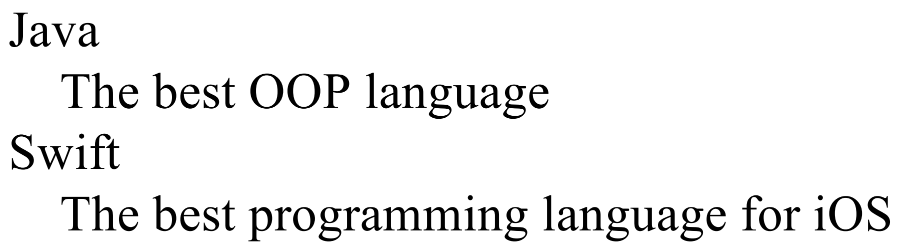


Type	Description
type="1"	The list items will be numbered with numbers (default)
type="A"	The list items will be numbered with uppercase letters
type="a"	The list items will be numbered with lowercase letters
type="I"	The list items will be numbered with uppercase roman numbers
type="i"	The list items will be numbered with lowercase roman numbers

HTML Description Lists

- HTML also supports description lists. A description list is a list of terms, with a description of each term.
- The <dl> tag defines the description list, the <dt> tag defines the term (name), and the <dd> tag describes each term:

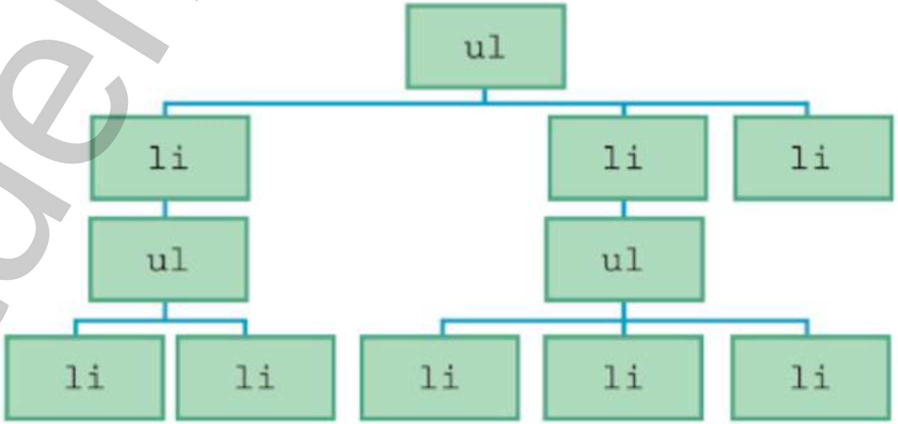
```
<dl>
  <dt>Java</dt>
  <dd>The best OOP language</dd>
  <dt>Swift</dt>
  <dd>The best programming language for iOS</dd>
</dl>
```



```
Java
   The best OOP language
Swift
   The best programming language for iOS
```

Nested HTML Lists

- Lists can be nested within other lists.
- List outside is called parent list and list inside is called child list.
- You must insert the nested list inside an <li> element of the parent list.



```
<ul>
  <li>Coffee</li>
  <li>Tea
    <ul>
      <li>Black tea</li>
      <li>Green tea</li>
    </ul>
  </li>
  <li>Milk</li>
</ul>
```

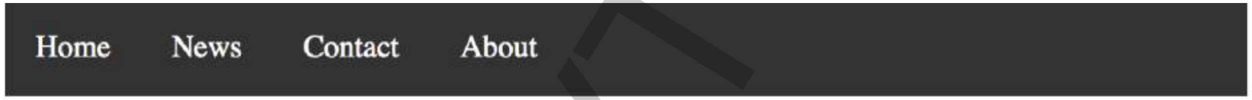
```
• Coffee
• Tea
   ◦ Black tea
   ◦ Green tea
• Milk
```

Horizontal List with CSS

- By default, HTML list is displayed in vertical but it can be styled in many different ways with CSS.
- Horizontal list with CSS example:

```
<ul>
  <li><a href="#home">Home</a></li>
  <li><a href="#news">News</a></li>
  <li><a href="#contact">Contact</a></li>
  <li><a href="#about">About</a></li>
</ul>
```

```
ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
    overflow: hidden;
    background-color: black;
}
li {
    float: left;
}
li a {
    display: block;
    color: white;
    text-align: center;
    padding: 16px;
    text-decoration: none;
}
```



HTML <figure> Element

- In traditional printed material like books and magazines, an image, chart, or code example would be accompanied by a caption. Before HTML5, we didn’t have a way of semantically marking up this sort of content directly in our HTML, instead resorting to CSS class names.
- HTML5 introduced the <figure> and <figcaption> elements to solve this problem.
- The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.

HTML <figure> Element

- In traditional printed material like books and magazines, an image, chart, or code example would be accompanied by a caption.
- Before HTML5, we didn’t have a way of semantically marking up this sort of content directly in our HTML, instead resorting to CSS class names.
- HTML5 introduced the <figure> and <figcaption> elements to solve this problem.

A cheeky macaque, Lower Kintaganban River, Borneo. Original by Richard Clark

HTML <figure> Element

- While the content of the <figure> element is related to the main flow, its position is independent of the main flow, and if removed it should not affect the flow of the document. The <figcaption> element is used to add a caption for the <figure> element.

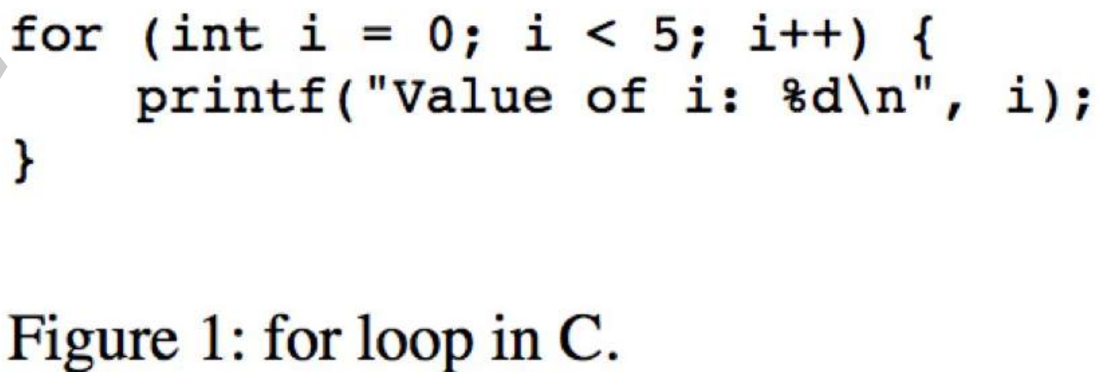
```
<figure>
  <img src="img_pulpit.jpg" alt="The
  Pulpit Rock" width="304"
   height="228">
  <figcaption>Figure 1: A view of the
  pulpit rock in Norway.</figcaption>
</figure>
```

Figure 1: A view of the pulpit rock in Norway.

HTML <figure> Element with Code

- You can explain the code using <figcaption>:

```
<figure>
  <pre>
for (int i = 0; i < 5; i++) {
    printf("Value of i: %d\n", i);
}
  </pre>
  <figcaption>Figure 1: for loop in C.</figcaption>
</figure>
```



```
for (int i = 0; i < 5; i++) {
    printf("Value of i: %d\n", i);
}
```

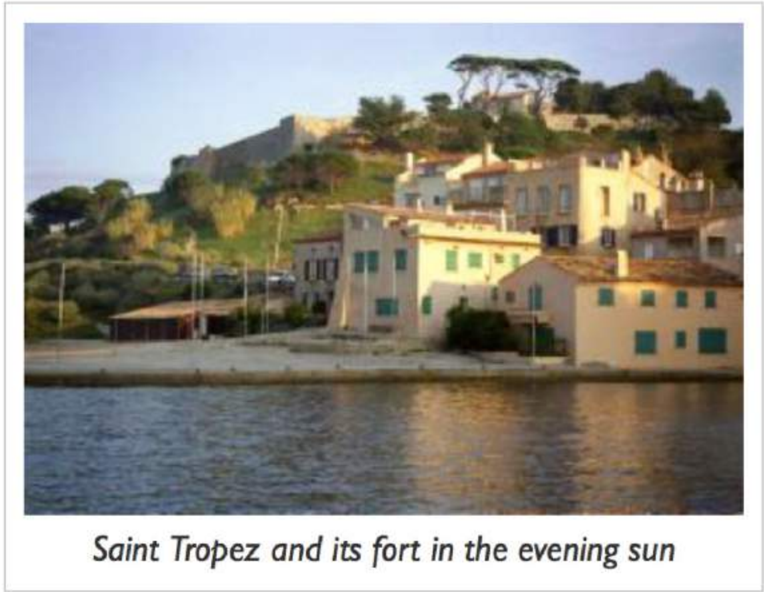
Figure 1: for loop in C.

Customize <figure> Element with CSS

- You can add border of figure, change font style of caption or scale image in <figure> element as following:

```
<figure>
  <img src="img_pulpit.jpg" alt="The Pulpit Rock" width="304" height="228">
  <figcaption>Figure 1: A view of the pulpit rock in Norway.</figcaption>
</figure>
```

```
figure {
  float: right;
  width: 30%;
  text-align: center;
  font-style: italic;
  font-size: smaller;
  text-indent: 0;
  border: thin silver solid;
  margin: 0.5em;
  padding: 0.5em;
}
```



HTML <img> Element

- In HTML, images are defined with the <img> tag:

```
<img src="url">
```

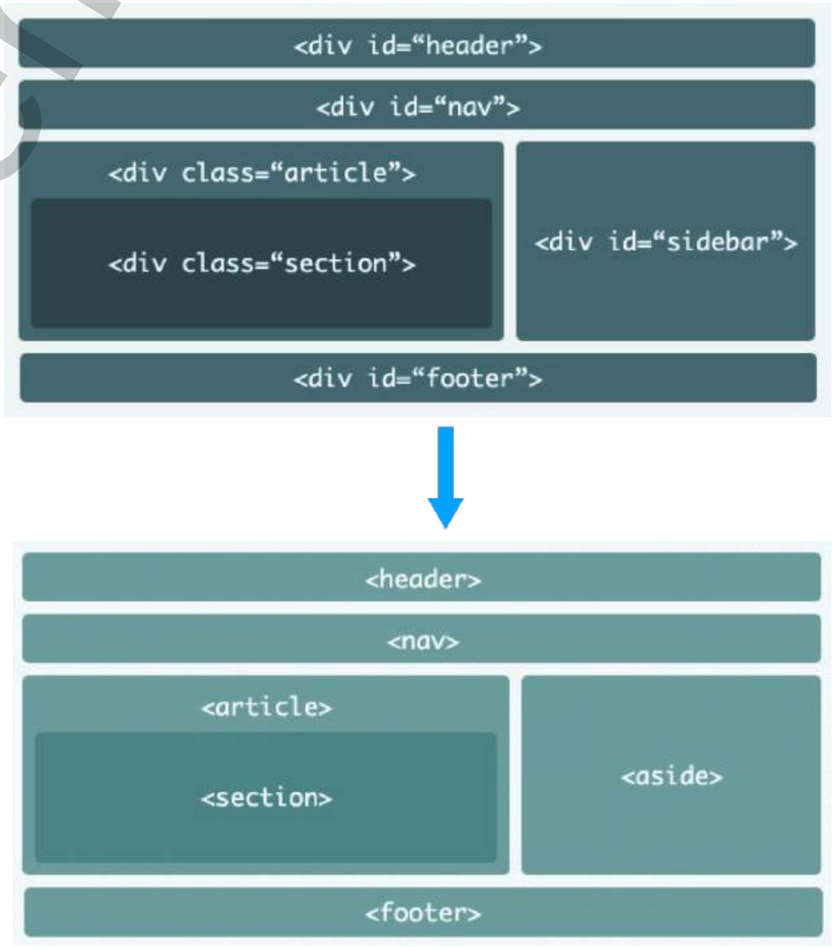
  - The <img> tag is empty, it contains attributes only, does not have a closing tag.
  - The src attribute specifies the URL (web address) of the image.

- Example:

```
<img src="pulpitrock.jpg"
width= "500" height="500">
```

Organizational Elements

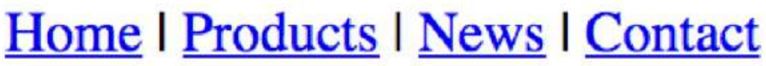
- Many web sites contain HTML code like: <div id="nav">, <div class="header">, <div id="footer"> to indicate navigation, header, and footer.
- **HTML5** offers new semantic elements to define different parts of a web page:
  - <nav>
  - <section>
  - <article>
  - <aside>
  - <header>
  - <footer>



HTML <nav> Element

- The <nav> tag defines a set of navigation links.
- Notice that NOT all links of a document should be inside a <nav> element. The <nav> element is intended only for major block of navigation links.
- Browsers, such as screen readers for disabled users, can use this element to determine whether to omit the initial rendering of this content.

```
<nav>
  <a href="/home/">Home</a> |
  <a href="/news/">News</a> |
  <a href="/products/">Products</a> |
  <a href=“/contact/">Contact</a>
</nav>
```



Home | Products | News | Contact

HTML <section> Element

- The <section> tag defines sections in a document, such as chapters, headers, footers, or any other sections of the document.
- According to W3C's HTML5 documentation: "A section is a thematic grouping of content, typically with a heading”
- A home page could normally be split into sections for introduction, content, and contact information.
- Example:

```
<section>
  <h1>WWF</h1>
  <p>The World Wide Fund for Nature (WWF) is....</p>
</section>
```

HTML <article> Element

- The <article> tag specifies independent, self-contained content.
- An article should make sense on its own and it should be possible to distribute it independently from the rest of the site.
- Potential sources for the <article> element:
  - Forum, blog post
  - News story
  - Comment
- Example:

```
<article>
  <h1>Google Chrome</h1>
  <p>Google Chrome is a free, open-source web browser developed by Google,
  released in 2008.</p>
</article>
```

Nesting <article> in <section> or Vice Versa

- The <article> element specifies independent, self-contained content.
- The <section> element defines section in a document.
- On the Internet, you will find HTML pages with <section> elements containing <article> elements, and <article> elements containing <section> elements.
- You will also find pages with <section> elements containing <section> elements, and <article> elements containing <article> elements.
- Example for a newspaper: The sport <article> in the sport section, may have a technical section in each <article>.

HTML <aside> Element

- The <aside> tag defines some content aside from the content it is placed in.
- The aside content should be related to the surrounding content.
- Example:

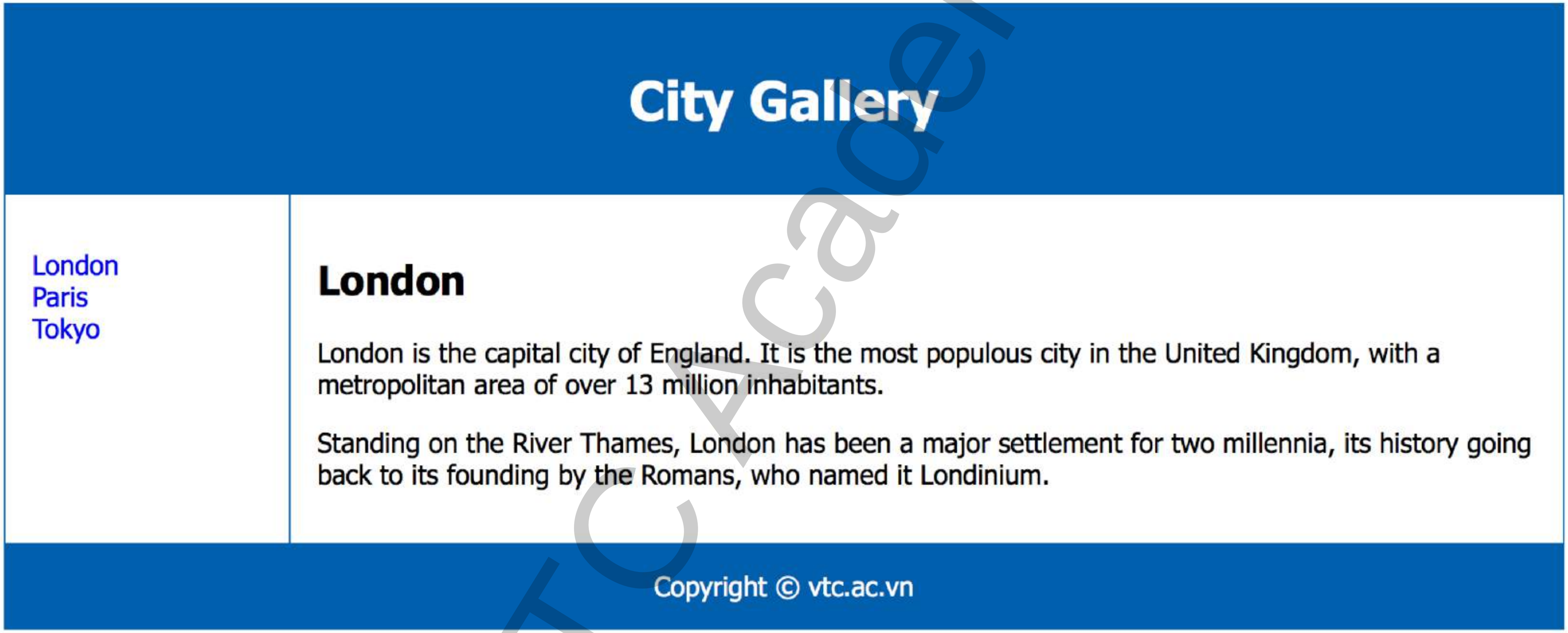
```
<p>My family and I visited The Epcot center this summer.</p>

<aside>
  <h4>Epcot Center</h4>
  <p>The Epcot Center is a theme park in Disney World, Florida.</p>
</aside>
```

HTML <footer> Element

- The <footer> element specifies a footer for a document or section.
- A <footer> element should contain information about its containing element.
- A footer typically contains the author of the document, copyright information, links to terms of use, contact information, etc.
- You may have several <footer> elements in one document.
- Example:

```
<footer>
  <p>Posted by: Hege Refsnes</p>
  <p>Contact information: <a
  href="mailto:abc@google.com">abc@google.com</a>.</p>
</footer>
```

Completed Page with Layout Elements



City Gallery

London
Paris
Tokyo

London

London is the capital city of England. It is the most populous city in the United Kingdom, with a metropolitan area of over 13 million inhabitants.

Standing on the River Thames, London has been a major settlement for two millennia, its history going back to its founding by the Romans, who named it Londinium.

Copyright © vtc.ac.vn

Completed Page with Layout Elements

```
<div class="container">
   <header>
      <h1>City Gallery</h1>
   </header>
   <nav>
     <ul>
       <li><a href="#">London</a></li>
       <li><a href="#">Paris</a></li>
       <li><a href="#">Tokyo</a></li>
     </ul>
   </nav>
   <article>
     <h1>London</h1>
     <p>London is the capital city of England. It is the most populous city in
     the United Kingdom, with a metropolitan area of over 13 million inhabitants.</p>
     <p>Standing on the River Thames, London has been a major settlement for two
     millennia, its history going back to its founding by the Romans, who named it
     Londinium.</p>
   </article>
   <footer>Copyright &copy; VTC Academy</footer>
</div>
```

Completed Page with Layout Elements

```
div.container {
    width: 100%;
    border: 1px solid #0060ae;
}
header, footer {
    padding: 1em;
    color: white;
    background-color: #0060ae;
    clear: left;
    text-align: center;
}
nav {
    float: left;
    max-width: 160px;
    margin: 0;
    padding: 1em;
}
```

```
nav ul {
    list-style-type: none;
    padding: 0;
}
nav ul a {
    text-decoration: none;
}
article {
    margin-left: 170px;
    border-left: 1px solid #0060ae;
    padding: 1em;
    overflow: hidden;
}
```

CSS Cascade

- CSS is an acronym for *Cascading Style Sheets*, which indicates that the notion of the cascade is important. At its most basic level it indicates that the order of CSS rules matter, but it's more complex than that.
- What selectors win out in the cascade depends on three factors (these are listed in order of weight — earlier ones will overrule later ones):
  - Importance
  - Specificity
  - Specificity Hierarchy
  - How to Calculate Specificity?

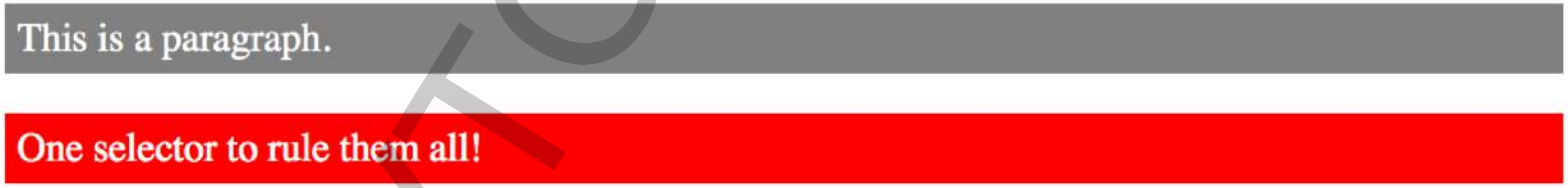
CSS Cascade - Importance

- In CSS, there is a special piece of syntax you can use to make sure that a certain declaration will *always* win over all others: !important.

```
<p class="better">This is a paragraph.</p>
<p class="better" id="winning">One selector to rule them all!</p>
```

```
#winning {
    background-color: red;
    border: 1px solid black;
}
.better {
    background-color: gray;
    border: none !important;
}
```

```
p {
    background-color: blue;
    color: white;
    padding: 5px;
}
```



This is a paragraph.

One selector to rule them all!

CSS Cascade - Specificity

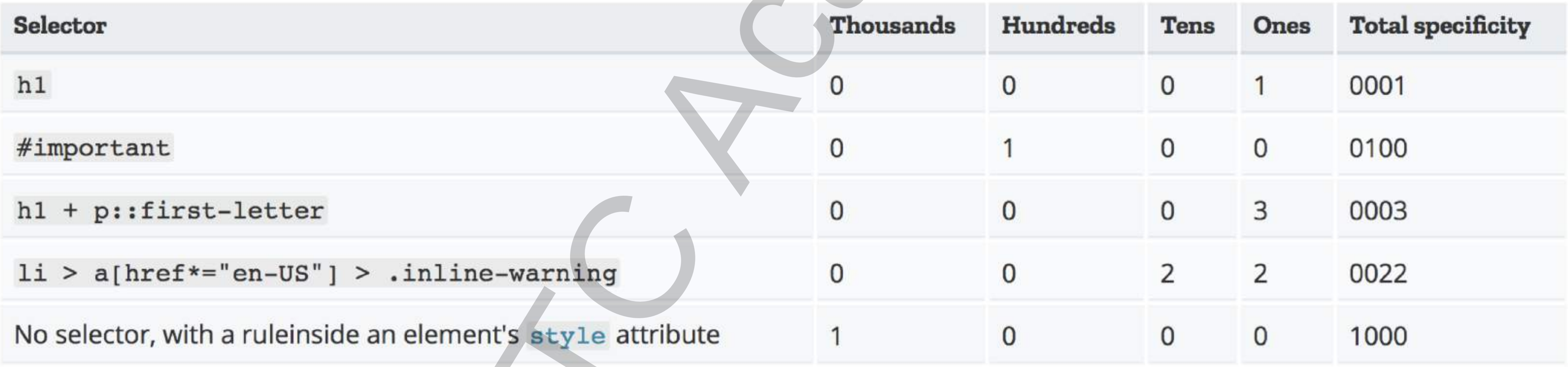
- If there are two or more conflicting CSS rules that point to the same element, the browser follows some rules to determine which one is most specific and therefore wins out.
- Think of specificity as a score/rank that determines which style declarations are ultimately applied to an element.
- The universal selector (*) has low specificity, while ID selectors are highly specific!
- **Note:** Specificity is a common reason why your CSS-rules don't apply to some elements, although you think they should.

CSS Cascade - Specificity Hierarchy

- Every selector has its place in the specificity hierarchy. There are four categories which define the specificity level of a selector:
  - **Inline styles** - An inline style is attached directly to the element to be styled. Example: <h1 style="color: #ffffff;">.
  - **IDs** - An ID is a unique identifier for the page elements, such as #navbar.
  - **Classes, attributes and pseudo-classes** - This category includes .classes, [attributes] and pseudo-classes such as :hover, :focus etc.
  - **Elements and pseudo-elements** - This category includes element names and pseudo-elements, such as h1, div, :before and :after.

How to Calculate Specificity?

- Start at 0, add 1000 for style attribute, add 100 for each ID, add 10 for each attribute, class or pseudo-class, add 1 for each element name or pseudo-element.
- Consider these three code fragments:



Selector	Thousands	Hundreds	Tens	Ones	Total specificity
`h1`	0	0	0	1	0001
`#important`	0	1	0	0	0100
`h1 + p::first-letter`	0	0	0	3	0003
`li > a[href*="en-US"] > .inline-warning`	0	0	2	2	0022
No selector, with a ruleinside an element's `style` attribute	1	0	0	0	1000

CSS Inheritance

- CSS inheritance is the last piece we need to investigate to get all the information and understand what style is applied to an element.
- The idea is that some property values applied to an element will be inherited by that element's children, and some won't.
- Example 1:
  - For example, it makes sense for font-family and color to be inherited, as that makes it easy for you to set a site-wide base font by applying a font-family to the <html> element; you can then override the fonts on individual elements where needed.
  - It would be really annoying to have to set the base font separately on *every* element.

CSS Inheritance

- Example 2:
  - As another example, it makes sense for margin, padding, border, and background-image to NOT be inherited.
  - Imagine the styling/layout mess that would occur if you set these properties on a container element and had them inherited by every single child element, and then had to *unset* them all on each individual element!

Specifying Inheritance

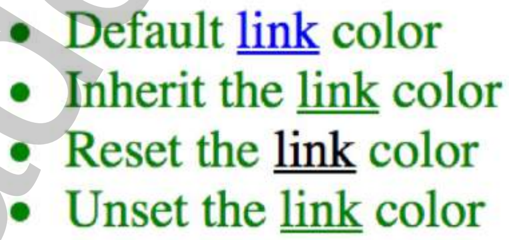
- inherit
  - Sets the property value applied to a selected element to be the same as that of its parent element.
- initial
  - Sets the property value applied to a selected element to be the same as the value set for that element in the browser's default style sheet. If no value is set by the browser's default style sheet and the property is naturally inherited, then the property value is set to inherit instead.
- unset
  - Resets the property to its natural value, which means that if the property is naturally inherited it acts like inherit, otherwise it acts like initial.

Specifying Inheritance

- revert
  - Reverts the property to the value it would have had if the current origin had not applied any styles to it. In other words, the property's value is set to the user stylesheet's value for the property (if one is set), otherwise the property's value is taken from the user-agent's default stylesheet.

CSS Inheritance Example

```
body {
    color: green;
}
.inherit a {
    color: inherit;
}
.initial a {
    color: initial;
}
.unset a {
    color: unset;
}

<ul>
    <li>Default <a href="#">link</a> color</li>
    <li class="inherit">Inherit the <a href="#">link</a> color</li>
    <li class="initial">Reset the <a href="#">link</a> color</li>
    <li class="unset">Unset the <a href="#">link</a> color</li>
</ul>
```



- Default link color
- Inherit the link color
- Reset the link color
- Unset the link color

Summary

- HTML offers web authors three ways for specifying lists of information. All lists must contain one or more list elements.
- There are two main kinds of list: ordered list and unordered list
- The <figure> tag specifies self-contained content, like illustrations, diagrams, photos, code listings, etc.
- **HTML5** offers new semantic elements to define different parts of a web page: header, nav, section, footer,…
- In CSS, there is a special piece of syntax you can use to make sure that a certain declaration will *always* win over all others: !important.
- CSS inheritance is the last piece we need to investigate to get all the information and understand what style is applied to an element.

References

- Books:
  - Web Design with HTML, CSS, JavaScript and jQuery Set: https://www.amazon.com/Web-Design-HTML-JavaScript-jQuery/dp/1118907442/
  - HTML5 and CSS3 All-in-One For Dummies: https://www.amazon.com/HTML5-CSS3-All-One-Dummies/dp/1118289382/
- Tutorials:
  - https://www.w3schools.com/html/default.asp
  - https://www.w3schools.com/css/default.asp
  - https://www.w3schools.com/js/default.asp

VTC Academy
Think Ahead, Beyond Boundaries

Thank you!