INFX 1606
INTRODUCTION TO
WEBSITE CREATION

DALHOUSIE
UNIVERSITY
FACULTY OF
COMPUTER SCIENCE

HTML Elements:
Containers and Including Media

Raghav V. Sampangi
Instructor, Faculty of Computer Science, Dalhousie University

raghav@cs.dal.ca

Outline

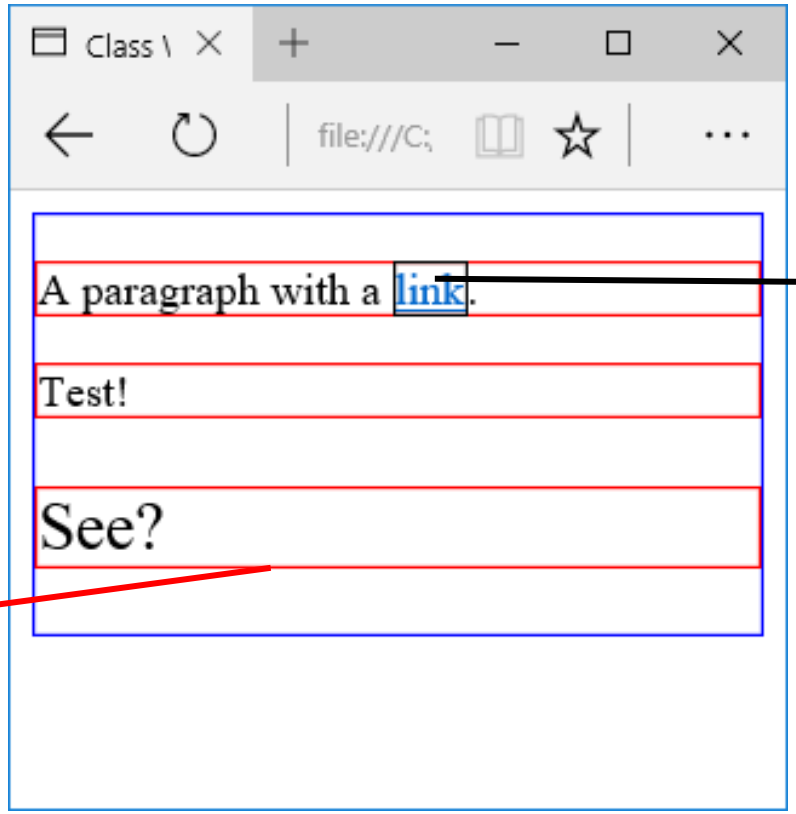
Containers of HTML elements

Including images in HTML markup

Block & Inline Elements

A **block-level** element starts on a new line, and takes up the full width of its **parent container**.

E.g. <p>, <h1>, <h2>,…<h6>, and so on.



An **inline** element exists within a block element, and only uses as much width as necessary.

E.g. <a>, <img>, and so on.

Generic Containers:
Semantically Neutral HTML Elements

Semantically Neutral?

We discussed semantic HTML elements – they help in conveying meaning not only to the user but also to user agents (e.g. screen readers)

Why do we need semantically neutral elements?

- They are semantically neutral in that they have no assigned meaning
- They act as a transparent envelope around content that helps you define the content's structure the way you want it to appear.
- These help in grouping content, add extra information around content (e.g. styling), and make content interactive (e.g. using JavaScript)
- Examples: ** ** and ****

Division:

 helps us define sections or divisions in an HTML document

This is a **block level container**, i.e. it is useful in grouping other block level elements (e.g. headings, paragraphs, etc.)

After grouping various elements in a group, we can apply separate styles to them using CSS

What other HTML elements do something similar?

Some semantic HTML elements in HTML5:
<header>, <footer>, <section>, <article>, <aside>

Division:

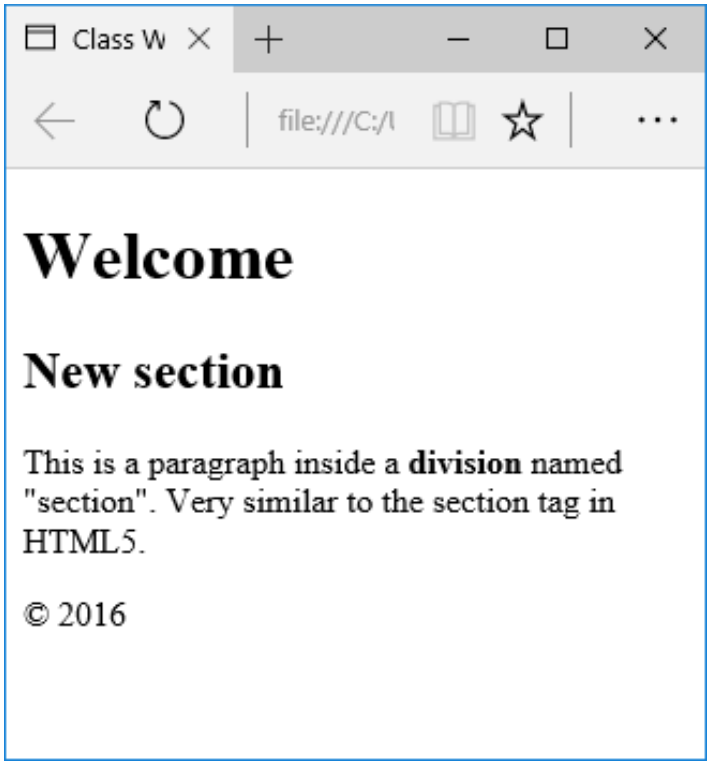
Example:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Class Work</title>
        <meta charset="UTF-8">
    </head>
    <body>

        <div id="header">
            <h1>Welcome</h1>
        </div>

        <div id="section">
            <h2>New section</h2>
            <p>This is a paragraph inside a
                <strong>division</strong> named "section". Very
                similar to the section tag in HTML5.</p>
        </div>

        <div id="footer">
            <p>&copy; 2016</p>
        </div>
    </body>
</html>
```



Division:

Remember to include the ***id*** attribute when you create divisions.

That way, you will be able to apply styles or introduce additional functionality as and when required.

Span:

This is an **inline level generic container**, i.e. it is used to wrap or group inline elements that appear within paragraphs (e.g. hypertext links, highlighted words or phrases and short quotations)

Example:

Suppose you are writing a piece of news, and one line inside a paragraph describes an alert or information to be given to the public. You could simply use a span element within the paragraph having an ***id*** and apply a different style to that element.

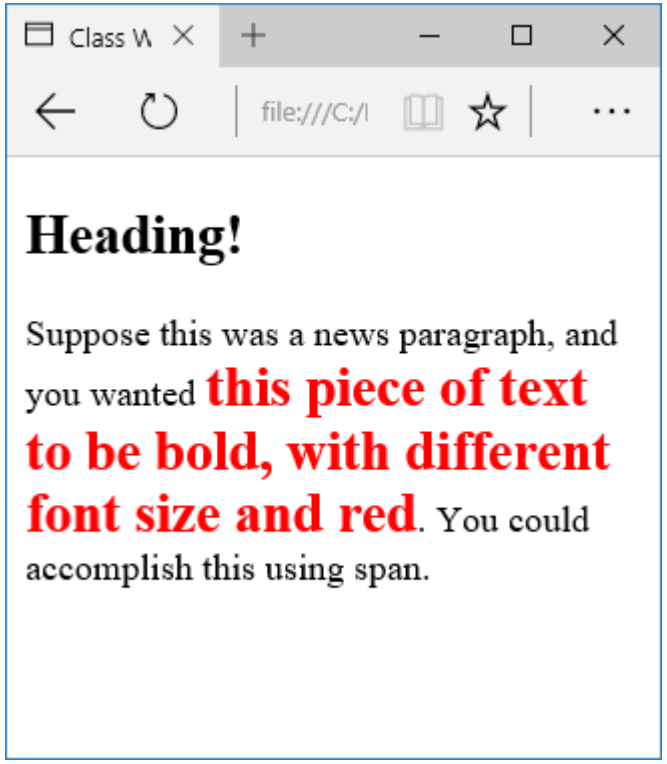
Span:

Example:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>Class Work</title>
        <meta charset="UTF-8">
    </head>
    <body>

        <div id="section">
            <h2>Heading!</h2>
            <p>Suppose this was a news paragraph,
and you wanted <span id="alertinfo" style="font-
size:150%; font-weight:bold; color:red;">this piece
of text to be bold, with different font size and
red</span>. You could accomplish this using
span.</p>
        </div>

    </body>
</html>
```



Key Ideas

- Generic containers can be used to define structural elements and for some inline formatting of content on web pages.
- elements are used to define structure of various elements on the web page.
- elements help in applying semantically neutral styling to some sections of content.

HTML Elements:
Media (Images, Video, etc.)

HTML Elements: Images

You might want to include images on web pages for various reasons: to include logos, photographs, charts, or any other illustrations.

Some of the commonly used image file formats:

- JPEG (.jpg): Lossy image compression standard proposed by Joint Photographic Experts Group

- GIF (.gif): Graphics Interchange Format; also supports animations

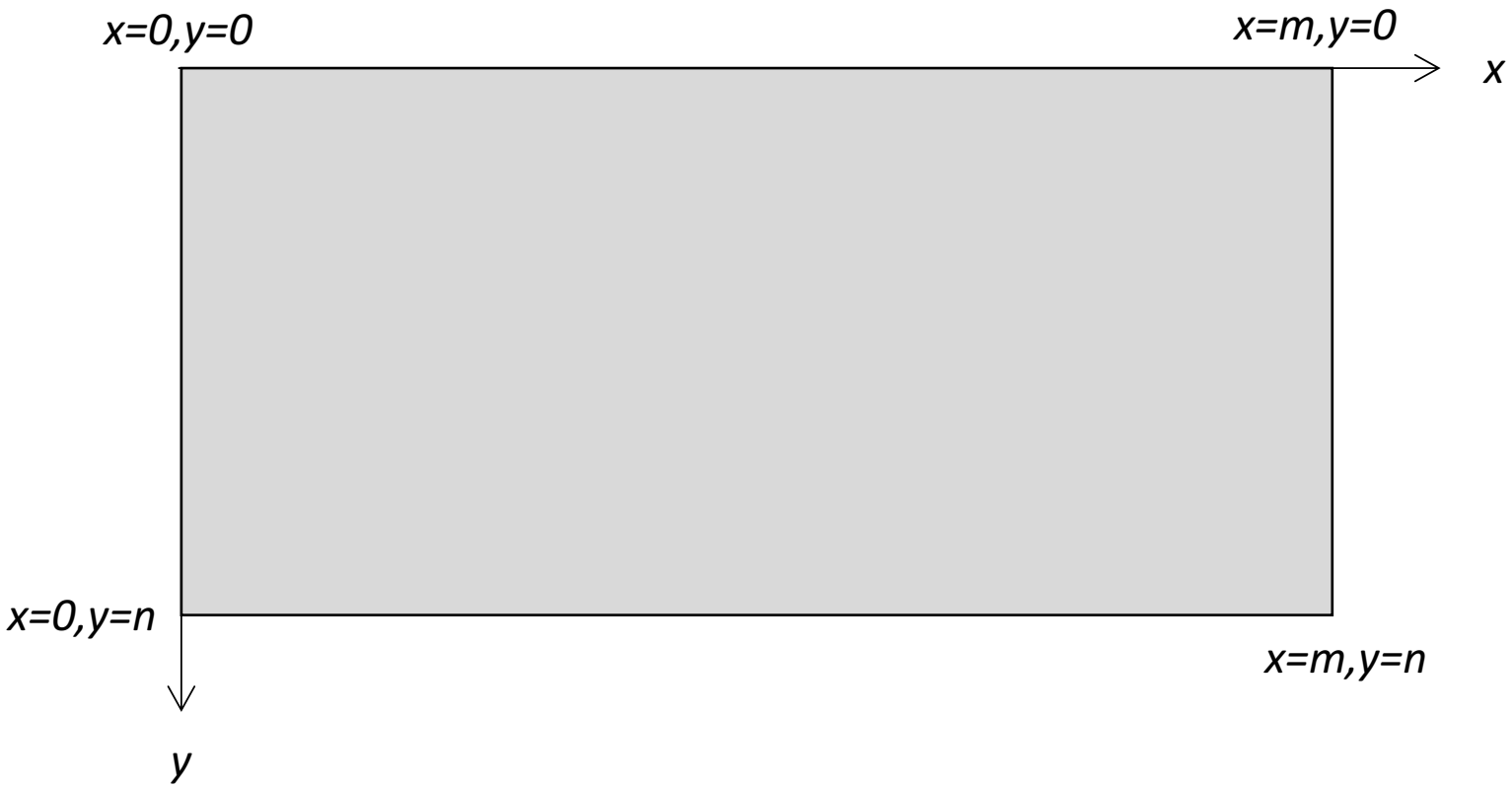
- PNG (.png): Portable Network Graphics, supports lossless compression of image data. *Designed for transferring images over the Internet.*

- SVG (.svg): Scalable Vector Graphics, XML-based 2D image format. Supports interactivity and animation.

HTML Elements: Images

Characteristics of an image:

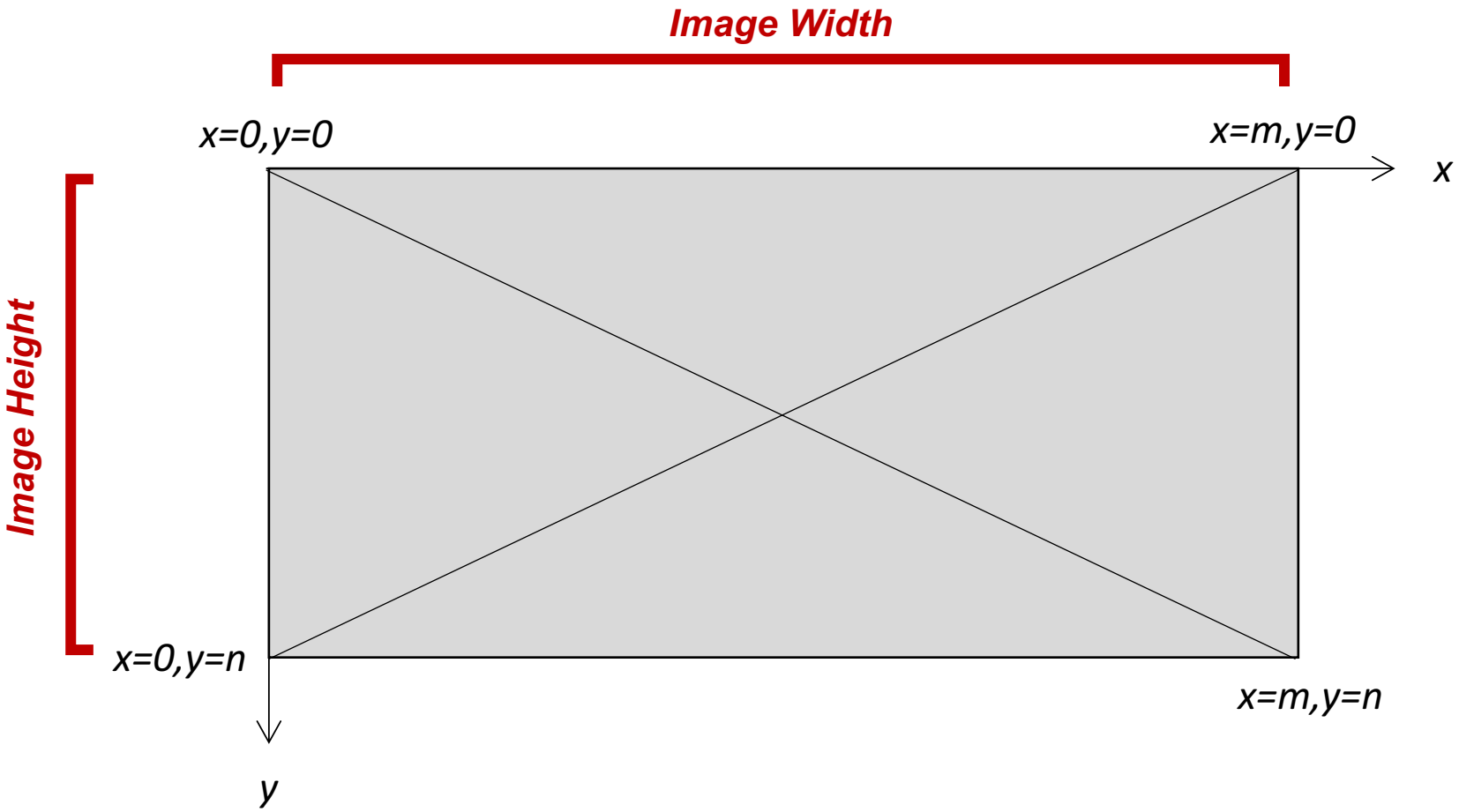
- 2-dimensional



HTML Elements: Images

Characteristics of an image:

- 2-dimensional
- Has width and height, expressed in pixels (px)



HTML Elements: Images

Characteristics of an image:

- 2-dimensional
- Has width and height, expressed in pixels (px)

Pixel:
Smallest controllable element of a picture

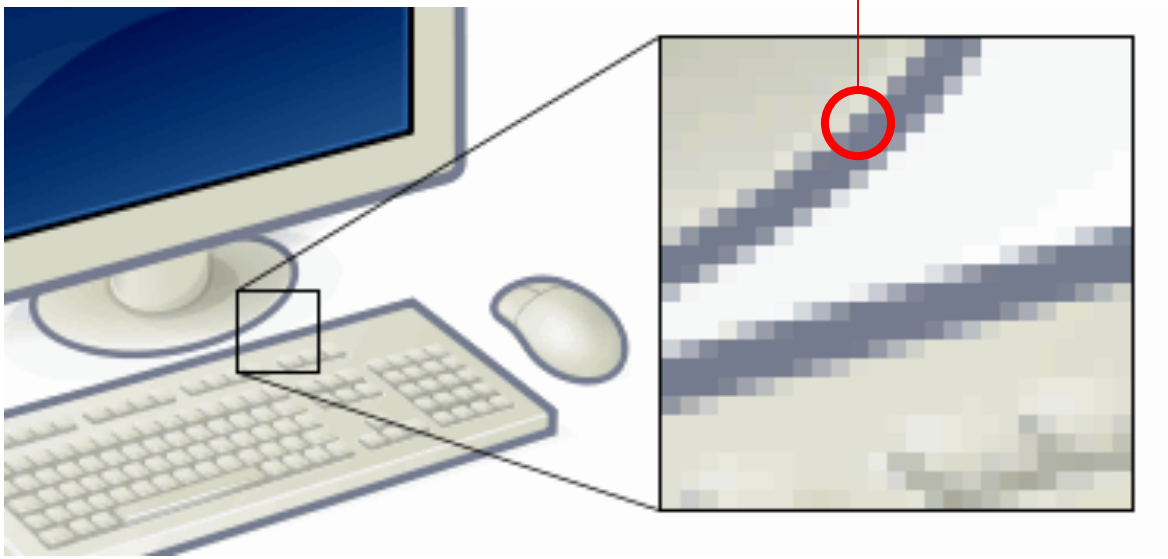
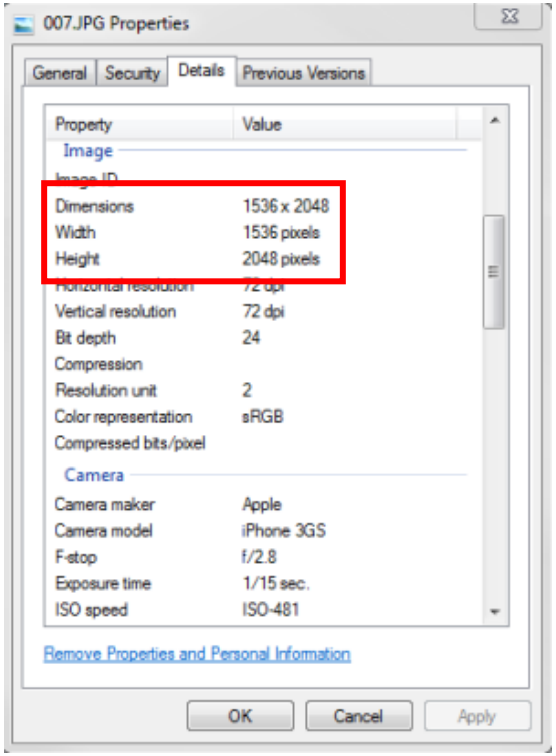


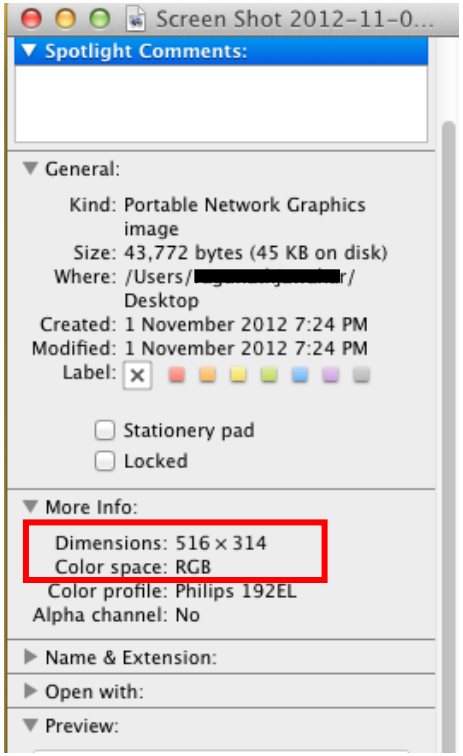
Image source: Wikipedia

HTML Elements: Images

Characteristics of an image:

- 2-dimensional
- Has width and height, expressed in pixels (px)
- You can find image details by “right-clicking” on image files and checking under “Properties” (on Windows, Linux), and “Get Info” (on Mac)

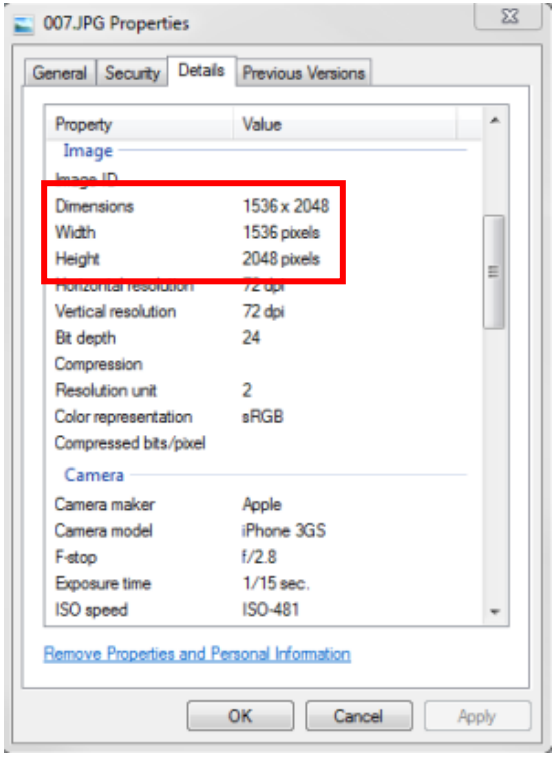


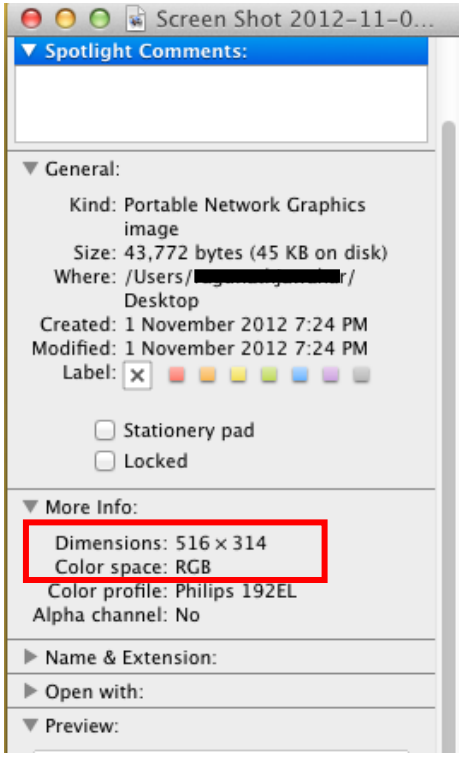


HTML Elements: Images

Note:

- By default, Mac OSX and Windows do not display the file extension of image files or any other files (i.e. whether an image file is .png or .jpg)
- Checking properties will tell you what type of image it is.
- Use the appropriate file extension.





HTML Elements: Images

What if the image dimensions are too large for your web page?

For example, typical web page widths are 1200px.
What if the image is 2400px X 1200px?
Image dimensions are larger than width of the web page dimensions!

What can you do in such cases?

You may re-size images or scale the images.
Use tools such as Photoshop, LightRoom, GIMP, etc.

Make sure that the aspect-ratio is maintained

HTML Elements: Images

Aspect ratio

Proportional relationship between its width and height.

This ratio of the width to the height (or, width/height) is required to make sure that the image is not distorted.

As web developers, it is your responsibility to make sure that logos are not distorted – so, if you re-size any image, make sure to keep the aspect ratio the same.

HTML Elements: Images

Aspect ratio

Proportional relationship between its width and height.

If image width = **640px**, image height = **480px**, and
if you want to re-size it to a width of **54px**,
what should be the height of the image if the aspect ratio is to be maintained the same (or, for the image to not be distorted)?

$\text{width1} = 640$ $\quad$ $\text{height1} = 480$

$\text{width2} = 54$ $\quad$ $\text{height2} = 54 \div \left(\frac{640}{480}\right) = 54 \div 1.3333$

$\text{height2} = 40.5 = 41$ (rounded up)

If the result has decimal points, you can either round up or down; check which one is better and make a decision.

HTML Elements: Images

Why do we need all these concepts?

Well, because you will use images in your websites, and you are required to specify the width and height when you do so.

You are expected to maintain the aspect ratio of images when you include them in your websites. You may resize images, however, ensure that you specify the appropriate width and height to maintain its aspect ratio.

Also, if you use distorted images in your assignments / labs / project, points will be deducted.

HTML Elements: Images

Structure of HTML ***img*** tag (empty element):

***Address**, i.e. file location of the image source*

***Alternate text:** Text that indicates alternate text for the image; if the image cannot be displayed*

```
<img src="path/to/file" alt="alt-text"
    style="width:98px;height:46px">
```

***Style – width and height:** Include width and height, so that the images are not awkwardly sized!*

HTML Elements: Images

Structure of HTML ***img*** tag (empty element):

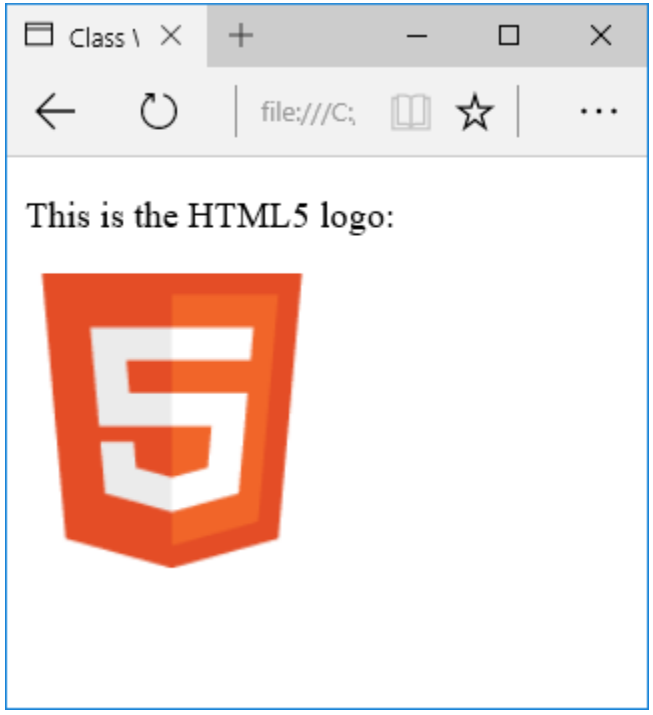
```
<img src="path/to/file" alt="alt-text"
  style="width:98px;height:46px" />
```

You may also use self-closing tag syntax. This is entirely up to you. This is also supported by HTML5.

HTML Elements: Images

Example:

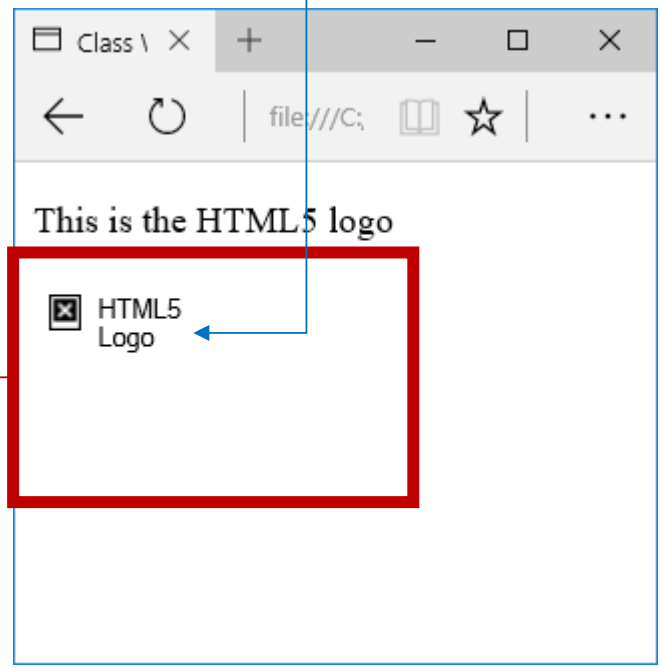
```
<p>This is the HTML5 logo:</p>
<img src="images/html5.gif" alt="HTML5 Logo" style="width:128px;height:128px">
```



HTML Elements: Images

Example:

```
<p>This is the HTML5 logo:</p>
<img src="images/html5.gif" alt="HTML5 Logo" style="width:128px;height:128px">
```



Alternate text being displayed, when image not available

HTML Elements: Images

Why must we include alt attribute?

Main reason:
if a user agent (e.g. a screen reader) is reading the web page, it will read the alternate text when it reaches the image element.

Second reason:
if the image cannot be displayed for some reason, the browser will display the alternate text instead.

HTML Elements: Images

Why must we include width and height?

It is good practice to tell the browser the dimensions of an image. (And, it can reserve the required space when the page is loaded)

As a developer, you must know what types of images you upload and their dimensions.

You must take into account that images with larger dimensions (and consequently, larger size), and scale them appropriately (while maintaining the aspect ratio).

HTML Elements: Images

Any alternative ways to specify image width and height?

Yes!
In addition to using the ***style*** attribute, you may also use the ***width*** and ***height*** attributes in the ***img*** tag.

They specify width and height in pixels.

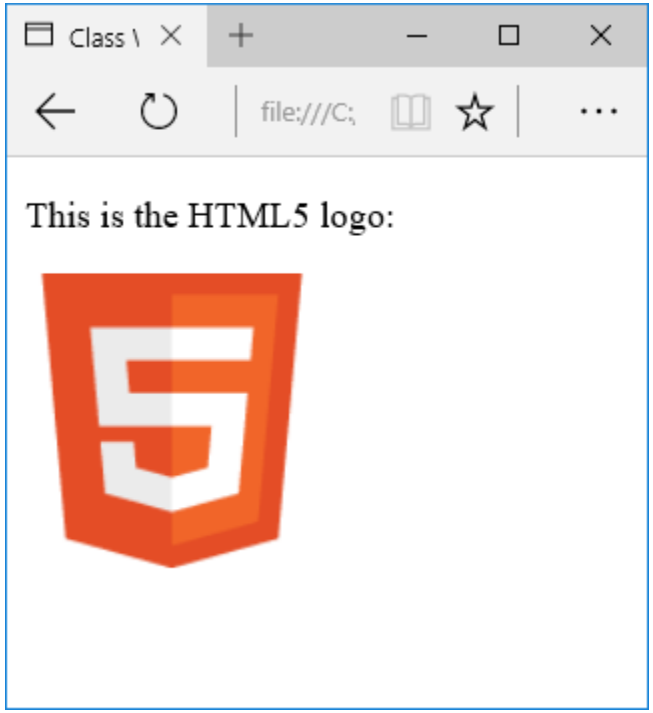
Also allowed in HTML5.

```
<img src="source" alt="alt-text"
    width="98" height="46" />
```

HTML Elements: Images

Example (using width and height attributes, instead of the style properties style:width and style:height):

```
<p>This is the HTML5 logo:</p>
<img src="images/html5.gif" alt="HTML5 Logo" width="128" height="128">
```



HTML Elements: Images

Which is better:
style:width, style:height OR width, height attributes?

Both are perfectly valid in HTML5 standard;
However, using the style attribute is better.

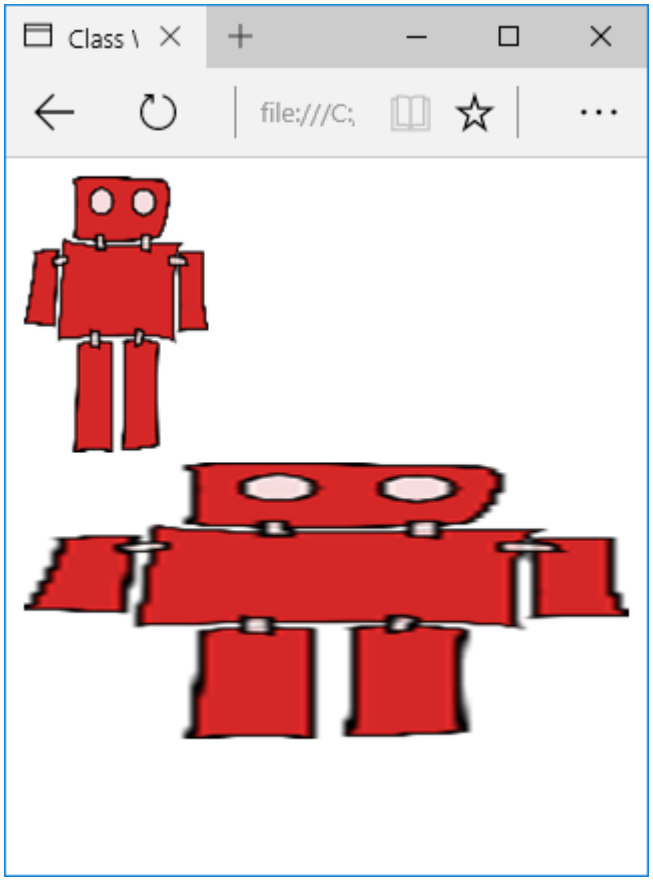
You can also use **style properties** (width and height) to set style properties for all images on a page or individual images, by appropriately coding the style sheets (CSS). We will discuss more about this later.

If you use width and height **attributes** to set dimensions of the image element, in addition to setting some styles in CSS, the CSS style property values will override those set by the attributes.

HTML Elements: Images

Example (width and height attributes, versus style:width and style:height):

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Class Work</title>
    <meta charset="UTF-8">
    <style>
      img {
        width: 100%;
      }
    </style>
  </head>
  <body>
    <img src="images/robot.png" alt="My Robot"
style="width:80px;height:120px">
    <img src="images/robot.png" alt="My Robot"
width="80" height="120">
  </body>
</html>
```

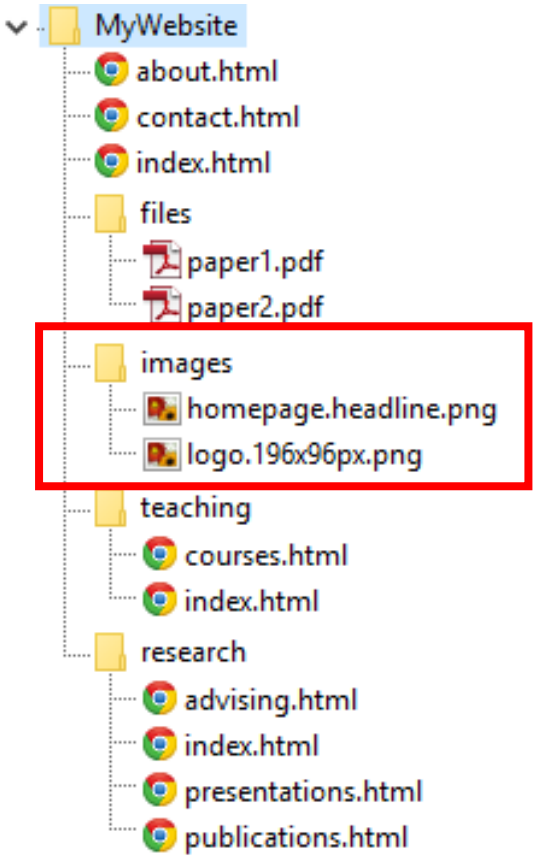


HTML Elements: Images

Storing images on websites

I suggest that you store all images in a separate folder

Why is this better?
It is easier to organize similar files in separate folder.
Makes it easier to manage all resources in your website.



HTML Elements: Images

Retrieving images on websites

You can use full image URLs (or absolute path), or use relative URLs.

Which is better?
Relative links are better – easier to change domain names, and minimal re-organization with ease.
If you use full paths, you will have to make changes in all places.

For example, if www.mywebsite.com changes to www.mysite.com, there's not much to change in terms of links if you have used relative paths.

HTML Elements: Images

Retrieving images on websites

You can use full image URLs (or absolute path), or use relative URLs.

Which is better?
Relative links are better – easier to change domain names, and minimal re-organization with ease.
If you use full paths, you will have to make changes in all places.

Using absolute paths is generally frowned upon by developers.

You can find a good comparison and detailed assessment between using one or the other here:
http://www.coffeecup.com/help/articles/absolute-vs-relative-pathslinks/

HTML Elements: Images

Retrieving images on websites

You can use full image URLs (or absolute path), or use relative URLs.

What is one application when using absolute paths is necessary?

If the image is stored on a different server.
For example, if you wanted to use the Dalhousie logo, from its originally stored location on the Dalhousie server, you could use its absolute path (and attribute credit to the actual source).

That way, any changes to the logo will be reflected on your website as well.

Can also be a problem: *if the source website decides to change the image and the image file name, your page will not display any image!!*

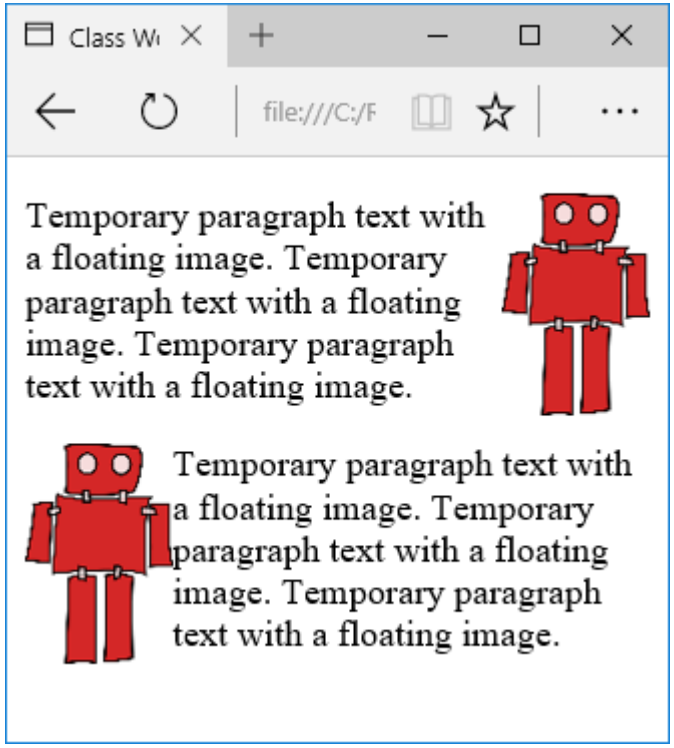
HTML Elements: Images

Image positioning relative to text

To position an image to the left or right of a paragraph, simply use the ***float*** style property, and set its value to be either right or left.

Example:

```
<p>
   <img src="images/robot.png" alt="My Robot"
style="float:right; width:64px; height:96px;">
   Temporary paragraph text with a floating image.
Temporary paragraph text with a floating image.
Temporary paragraph text with a floating image.
</p>
<p>
   <img src="images/robot.png" alt="My Robot"
style="float:left; width:64px; height:96px;">
   Temporary paragraph text with a floating image.
Temporary paragraph text with a floating image.
Temporary paragraph text with a floating image.
</p>
```



HTML Elements: Images

Using images as links

To accomplish this, simply include the <img> element as a child element of the anchor <a> element, or, you would nest <img> inside the <a>.

Example:

```
<a
href="https://en.wikipedia.org/wiki/List_of_f
ictional_robots_and_androids"
target="_blank">

   <img src="images/robot.png" alt="My Robot"
style="width:64px;height:96px;">

</a>
```