



## Tutorial 3

# Designing a Web Page

Working with Fonts, Colors, and Graphics



# Objectives

- Learn how HTML handles color
- Create foreground and background colors
- Work with font styles and designs
- Align text with surrounding content with styles
- Control spacing between letters, words, and lines with styles



# Objectives

- Learn about different image formats
- Understand transparent images and animated graphics
- Apply background images to an element
- Float images
- Set the margin size around an element



# Objectives

- Understand image maps
- Create and link image map hotspots
- Apply an image map to an inline image
- Remove a border from a linked image



# Working with Color in HTML

- Using color will make your web pages:
  - visually interesting
  - eye-catching for the reader
- HTML is a text-based language, requiring you to define your colors in textual terms.
- HTML identifies a color in one of two ways:
  - by the color value
  - by the color name



# Using Color Values

- To have more control and more choices, specify colors using **color values**.
- A **color value** is a numerical expression that precisely describes a color.

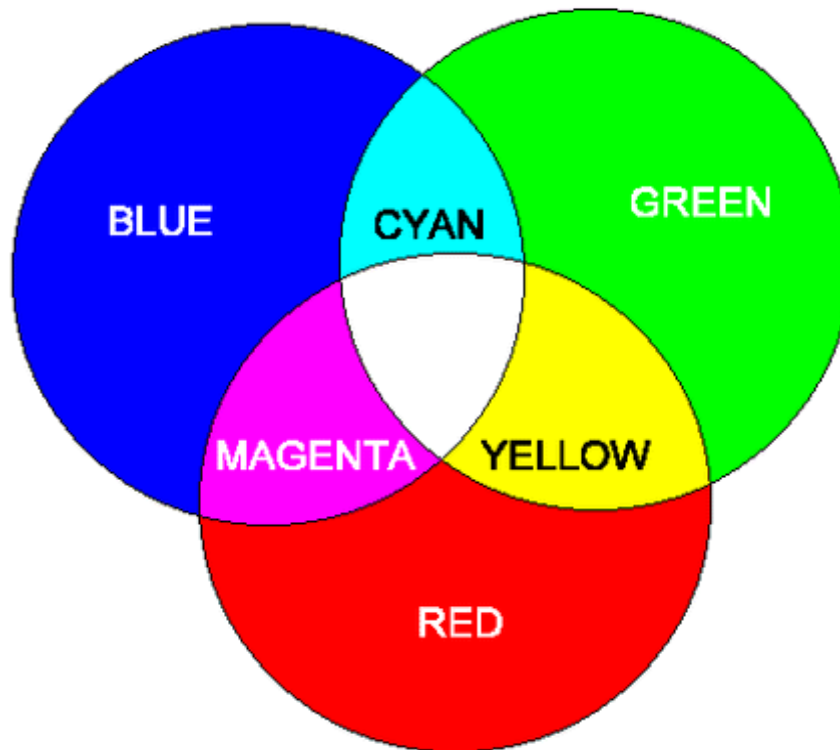


# Basic Principles of Color Theory

- Any color can be thought of as a combination of three primary colors: **red**, **green**, and **blue**.
- By varying the intensity of each primary color, you can create almost any color and any shade of color.
- This principle allows a computer monitor to combine pixels of red, green, and blue to create the array of colors you see on your screen.



# Adding the Three Primary Colors



Primary color model for light



# RGB (Red, Green, and Blue) Triplets

- Software programs, such as your Web browser, define color mathematically.
- The intensity of each of three colors (RGB) is assigned a number from 0 (absence of color) to 255 (highest intensity).
- In this way,  $255^3$ , or more than 16.7 million, distinct colors can be defined.
- Each color is represented by a triplet of numbers, called an **RGB triplet**, based on the strength of its **R**ed, **G**reen, and **B**lue components.



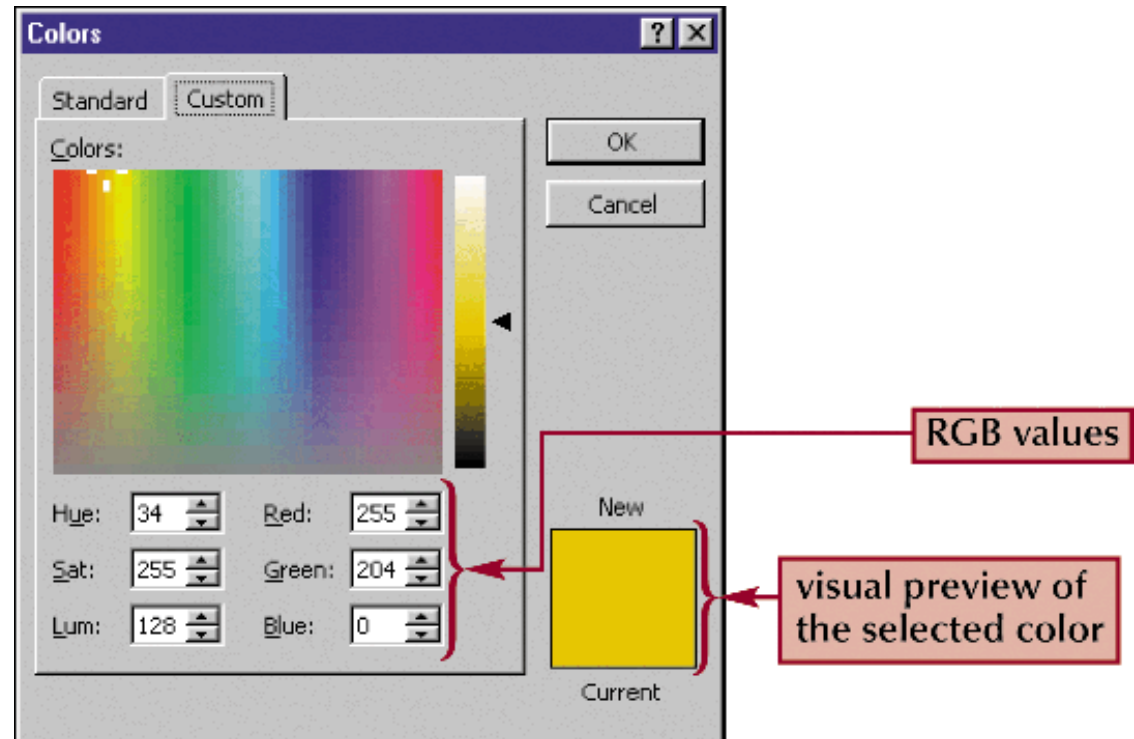
# RGB Triplets Examples

- White has a triplet of (255,255,255), indicating that red, green, and blue are equally mixed at the highest intensity.
- Yellow has the triplet (255,255,0) because it is an equal mixture of red and green with no presence of blue.



# A Typical Colors Dialog Box

- In most programs, you make your color choices with visual clues, usually without being aware of the underlying RGB triplet.
- This figure shows a typical dialog box in which you would make color selections based on the appearance of the color, rather than on the RGB values.





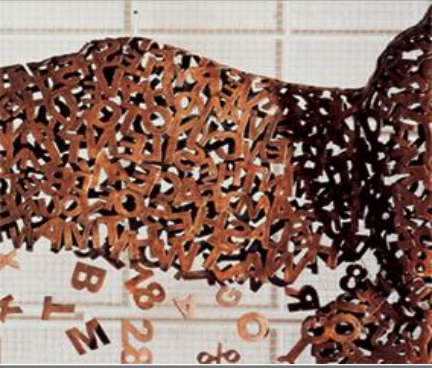
# Hexadecimal Values

- HTML requires color values be entered as hexadecimals.
- A **hexadecimal** is a number based on base-16 mathematics rather than base-10 mathematics that we use every day.
  - in base 10 counting, you use combination of 10 characters (0 through 9) to represent numerical values.
  - hexadecimals include six extra characters: A (for 10), B (for 11), C (for 12), D (for 13), E (for 14), and F (for 15).
  - for values above 15, you use a combination of the 16 characters; 16 is expressed as “10”, 17 is expressed as “11”, and so forth.
- Because of the popularity of the Web, most graphics programs will now display the hexadecimal value of the colors in their color selection dialog boxes.



# Hexadecimal Values

- To represent a number in hexadecimal terms, you convert the value to multiples of 16 plus a remainder. For example:
  - 21 is equal to  $(16 \times 1) + 5$ , so its hexadecimal representation is 15.
  - the number 255 is equal to  $(16 \times 15) + 15$ , or FF in hexadecimal format (remember that F = 15 in hexadecimal).
  - in the case of the number 255, the first F represents the number of times 16 goes into 255 (which is 15), and the second F represents the remainder of 15.
- Once you know the RGB triplet of a color, the color needs to be converted to the hexadecimal format.



# Using Color Values

- A **palette** is a selection of colors
- When a browser encounters a color that is not in its palette, it attempts to render the color; this is called **dithering**





# Using Color Values

- Use colors from the **safety palette** to avoid dithering
- The 216 colors in the safety palette are known as **Web-safe colors**



# Using Color Names

- Using the basic color names allows you to accurately display them across different browsers and operating systems
- The list of only 16 colors is limiting to Web designers.
  - in response, Netscape and Internet Explorer began to support an extended list of color names



# The 16 Basic Color Names














The 16 basic color names that are recognized by all versions of HTML and XHTML.

Color Name	RGB Triplet	Hexadecimal	Color Name	RGB Triplet	Hexadecimal
Aqua	(0,255,255)	00FFFF	Navy	(0,0,128)	000080
Black	(0,0,0)	000000	Olive	(128,128,0)	808000
Blue	(0,0,255)	0000FF	Purple	(128,0,128)	800080
Fuchsia	(255,0,255)	FF00FF	Red	(255,0,0)	FF0000
Gray	(128,128,128)	808080	Silver	(192,192,192)	C0C0C0
Green	(0,128,0)	008000	Teal	(0,128,128)	008080
Lime	(0,255,0)	00FF00	White	(255,255,255)	FFFFFF
Maroon	(128,0,0)	800000	Yellow	(255,255,0)	FFFF00



# Partial List of Extended Color Names

Partial list of extended color names

Sample	Name	RGB	Hexadecimal
	aliceblue	(240,248,255)	#F0F8FF
	antiquewhite	(250,235,215)	#FAEBD7
	<b>aqua</b>	<b>(0,255,255)</b>	<b>#00FFFF</b>
	aquamarine	(127,255,212)	#7FFFD4
	azure	(240,255,255)	#F0FFFF
	beige	(245,245,220)	#F5F5DC
	bisque	(255,228,196)	#FFE4C4
	<b>black</b>	<b>(0,0,0)</b>	<b>#000000</b>
	blanchedalmond	(255,235,205)	#FFEBCD
	<b>blue</b>	<b>(0,0,255)</b>	<b>#0000FF</b>
	blueviolet	(138,43,226)	#8A2BE2
	brown	(165,42,42)	#A52A2A
	burlywood	(222,184,135)	#DEB887



# Defining Foreground and Background Colors

- Foreground color definition:
  - **color:** *color*

*color* is either the color value or the color name
- Background color definition:
  - **background-color:** *color*
- You can apply foreground and background colors to any page element



# Defining Foreground and Background Colors

- To define the background color for an entire page:
  - Add the bgcolor attribute to the <body> tag
- To define the text color for an entire page:
  - Use the text attribute
- An example of background and text color changes including the color's hexadecimal value:
  - `<body bgcolor="yellow" text="#99CCFF">`





# Working with Fonts and Text Styles

- A **specific font** is a font such as Times New Roman, Arial, or Garamond. The font is installed on a user's computer
- A **generic font** refers to the font's general appearance



# Working with Fonts and Text Styles

## Generic fonts

	Font Samples		
serif	defg	defg	defg
sans-serif	defg	defg	defg
monospace	de fg	defg	deFg
cursive	<b>defg</b>	<i>defg</i>	defg
fantasy	<i>defg</i>	<b>defg</b>	DEFG



# Using the `<font>` Tag

- The `<font>` tag allows you to specify the color, the size, and the font to be used for text on a Web page.
- The syntax for the `<font>` tag is:

```
<font size="size" color="color" face="face">  
text </font>
```

- **size** attribute allows you to specify the font size of the text
- **color** attribute allows you to change the color of individual characters or words
- **face** attribute specifies a particular font for a section of text



# Changing the Font Color

- The **color** attribute of the `<font>` tag allows you to change the color of individual characters or words.
- Specify the color in the `<font>` tag by using either a color name or color value.
  - for example, to change the color of the word “**Arcadium**” to the hexadecimal color value 8000C0, you would enter the following HTML tag: `<font color="#8000C0"> Arcadium</font>`
- If there is no color specified in the `<body>` tag, the default colors of the Web browser is used.



# Setting the Font Size

- The style to change the font size of text within an element:
  - `font-size: length`
- **Absolute units** define a font size using one of five standard units of measurement:
  - Millimeters (mm)
  - Centimeters (cm)
  - Inches (in)
  - Points (pt)
  - Picas (pc)



# Spacing and Indentation

- **Tracking** is the amount of space between words and phrases
- **Kerning** is the amount of space between pairs of letters
- **Leading** is the space between lines of text



# Controlling the Spacing and Indentation

- **Tracking**
  - `word-spacing: value`
- **Kerning**
  - `letter-spacing: value`
- **Leading**
  - `line-height: length`



# Font Styles and Weights

- To specify font styles, use the following style:
  - `font-style: type`
- To control font weight for any page element, use the following style:
  - `font-weight: weight`
- To change the appearance of your text, use the following style:
  - `text-decoration: type`



# Decorative Features

- Underline, overline:
  - `text-decoration: underline overline`
- Capitalize:
  - `text-transform: capitalize`
- Uppercase letters, small font:
  - `font-variant: type`
- See the reference window on page HTML 127 for more information on setting font styles



# Text Alignment

Vertical Alignment	Description
baseline	Aligns the element with the bottom of lowercase letters in surrounding text. (the default)
bottom	Aligns the bottom of the element with the bottom of the lowest element in surrounding content.
middle	Aligns the middle of the element with the middle of the surrounding content.
sub	Subscripts the element.
super	Superscripts the element.
text-bottom	Aligns the bottom of the element with the bottom of the font of the surrounding content.
text-top	Aligns the top of the element with the top of the font of the surrounding content.
top	Aligns the top of the element with the top of the tallest object in the surrounding content.



# Choosing an Image Format

- **GIF (Graphics Interchange Format)** is the most commonly used image format on the Web.
- Compatible with virtually all browsers.
- GIF files are limited to displaying 256 colors.
- Often used for graphics requiring fewer colors, such as clip art images, line art, logos, and icons.
- Images that require more color depth, such as photographs, can appear grainy when saved as GIF files.



# Interlaced and Noninterlaced GIFs

- **Interlacing** refers to the way the GIF is saved by the graphics software.
- Normally, with a **noninterlaced** GIF the image is saved one line at a time, starting from the top of the graphic and moving downward.
- With interlaced GIFs, the image is saved and retrieved “**stepwise.**”
  - for example, every fifth line of the image might appear first, followed by every sixth line, and so forth through the remaining rows



# Interlaced and Noninterlaced GIFs

- Interlacing is an effective format if you have a large graphic and want to give users a preview of the final image as it loads.
- Interlacing can increase the size of a GIF file by anywhere from 3 to 20 kilobytes, depending on the image.



# Noninterlaced Graphic

A noninterlaced GIF appears as it is slowly retrieved by the Web browser.

If the graphic is large, it might take several minutes for the entire image to appear, which can frustrate the visitors to your Web page.

top appears first



Image appears one line at a time



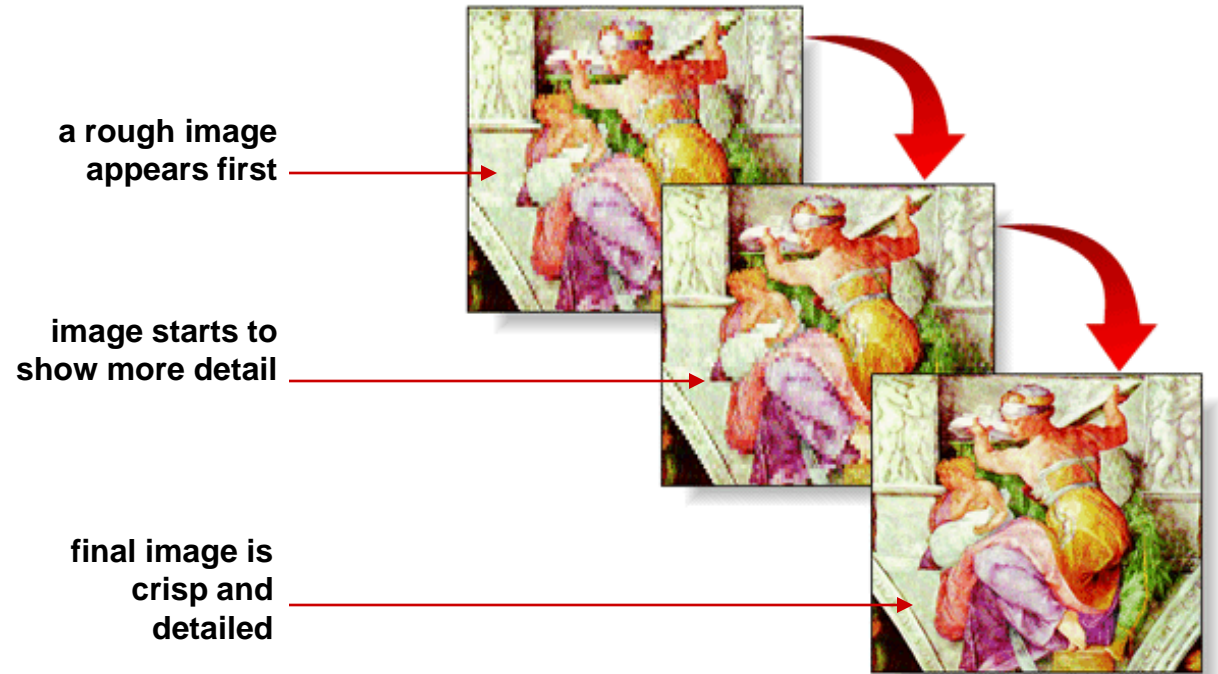
entire image is retrieved





# Interlaced Graphic

This figure shows the effect of interlacing, which is when the graphic starts out as a blurry representation of the final image, then gradually comes into focus—unlike the noninterlaced graphic, which is always a sharp image as it's being retrieved, although an incomplete one.





# Transparent GIFs

- A **transparent color** is a color from the image that is not displayed when the image is viewed in an application.
- In place of a transparent color, the browser will display whatever is on the page background.
- Creating a transparent color depends on the graphic software used.
- Many applications include the option to designate transparent color when saving the image i.e. PhotoShop.



# Animated GIFs

- One of the most popular uses of GIFs is to create animated images.
- Animated GIFs are easy to create and smaller in size.
- An **animated GIF** is composed of several images that are displayed one after the other in rapid succession.
- Animated GIFs are an effective way to compose slide shows or to simulate motion.



# Animated GIFs

- Animated GIF files are typically larger than static GIF images.
- The use of animated GIFs can greatly increase the size of a Web page.
- Be careful not to overwhelm the user with animated images.
- Animated GIFs are limited to 256 colors and can use transparent colors.
- Early browser versions may not support animated GIFs.



# Portable Network Graphics (PNG)

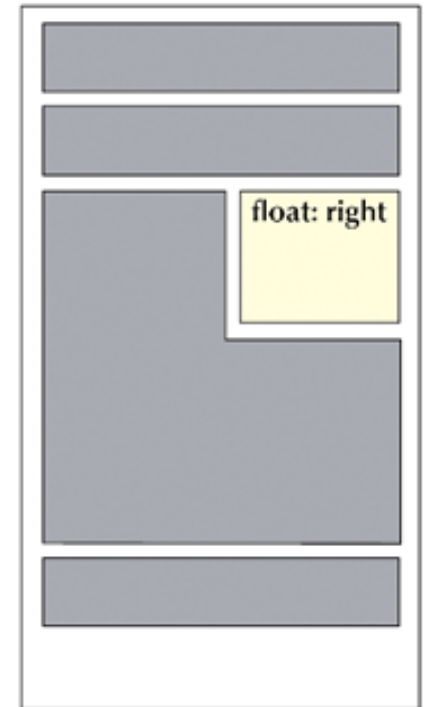
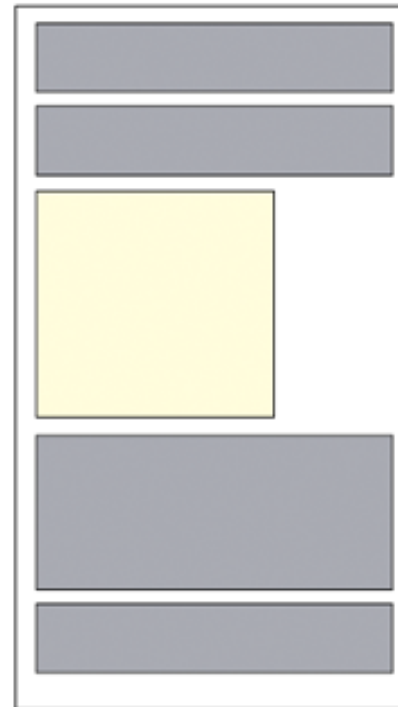
- A new file format called **PNG (Portable Network Graphics)** has been created.
- PNG files use a free and open file format and can display more colors than GIFs.
- PNGs cannot be used for animated graphics.
- PNGs do allow transparent colors, but not all browsers support this feature.



# Controlling Image Alignment

- The syntax for the float style is:

**`float: position`**





# Controlling Image Alignment

- The **align** attribute can control the alignment of an image with the `<img>` tag.
- The syntax for the align attribute is:  
`<img align="position" />`
  - ***alignment*** indicates how you want the graphic aligned in relation to the surrounding text



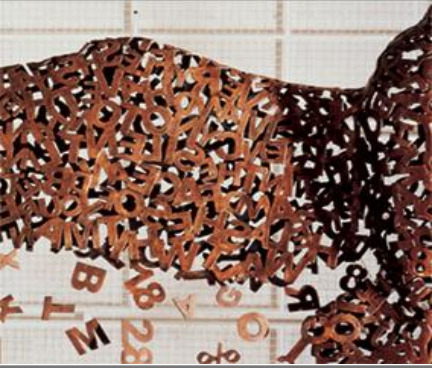
# Alignment Options

<b>align=</b>	<b>Description</b>
absbottom	Aligns the bottom of the object with the absolute bottom of the surrounding text. The absolute bottom is equal to the baseline of the text minus the height of the largest descender in the text.
absmiddle	Aligns the middle of the object with the middle of the surrounding text. The absolute middle is the midpoint between the absolute bottom and text top of the surrounding text.
baseline	Aligns the bottom of the object with the baseline of the surrounding text.
bottom	Aligns the bottom of the object with the bottom of the surrounding text. The bottom is equal to the baseline minus the standard height of a descender in the text.
left	Aligns the object to the left of the surrounding text. All preceding and subsequent text flows to the right of the object.
middle	Aligns the middle of the object with the surrounding text.
right	Aligns the object to the right of the surrounding text. All subsequent text flows to the left of the object.
texttop	Aligns the top of the object with the absolute top of the surrounding text. The absolute top is the baseline plus the height of the largest ascender in the text.
top	Aligns the top of the object with the top of the text. The top of the text is the baseline plus the standard height of an ascender in the text.



# Margins

- Control your margins with the following four styles:
  - `margin-top: length`
  - `margin-right: length`
  - `margin-bottom: length`
  - `margin-left: length`
- Margin values can also be negative- this creates an overlay effect by forcing the browser to render one element on top of another
- You can also combine the four margin styles into a single style



# Inserting a Background Image

- The **background** attribute allows an image file for the background of a Web page.
- The syntax for inserting a background image is: **background-image: url(*url*)**
  - **URL** is the location and filename of the graphic file you want to use for the background of the Web page.
    - for example, to use an image named “**bricks.gif**” as a background image, you would use the tag: **<body background=“brick.gif”>**



# Background Image

- In choosing a background image, you should remember the following:
  - use an image that will not detract from the text on the Web page, making it hard to read
  - do not use a large image file (more than 20 kilobytes)
    - large and complicated backgrounds will increase the time it takes a page to load
  - be sure to take into consideration how an image file looks when it is tiled in the background



# Background Image Options

`background-repeat: type`



background image



`background-image: repeat`



`background-image: repeat-x`



`background-image: repeat-y`



`background-image: no-repeat`

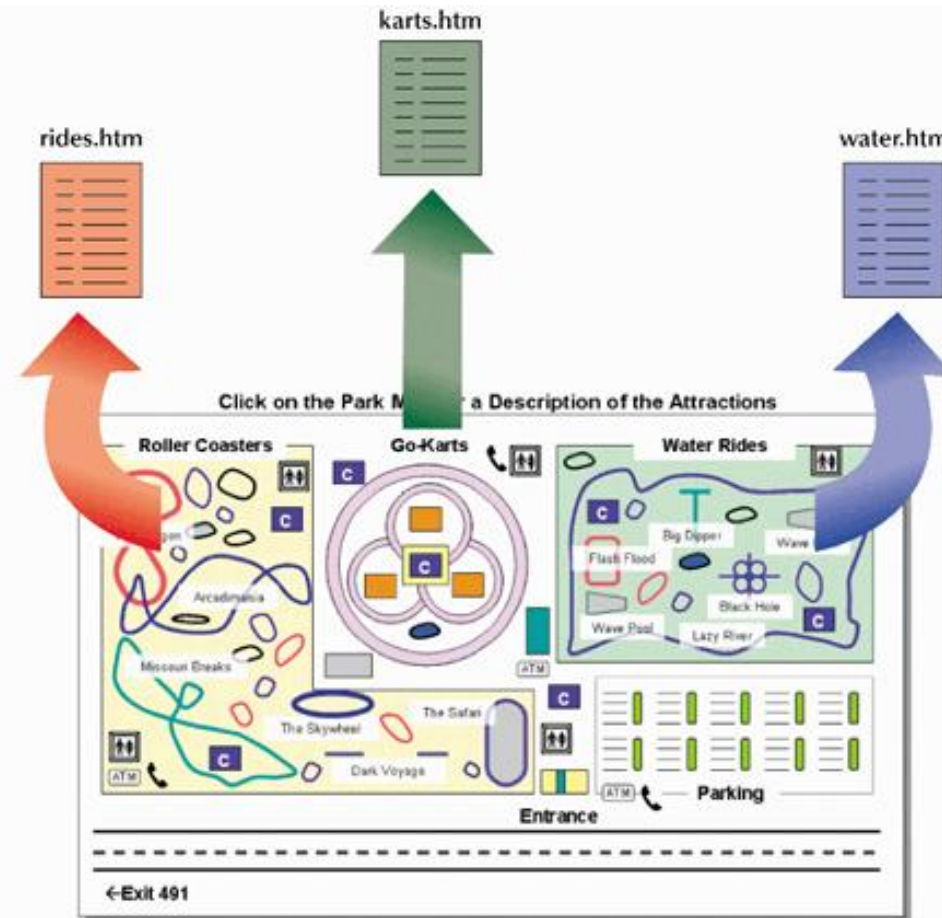


# Understanding Image Maps

- To use a single image to access multiple targets, you must set up hotspots within the image.
- A **hotspot** is a defined area of the image that acts as a hypertext link.
- When a user clicks within a hotspot, the hyperlink is activated.
- Hotspots are defined through the use of **image maps**, which list the positions of all hotspots within an image.
- There are two types of image maps: **server-side image maps** and **client-side image maps**.



# Image Maps





# Server-Side Image Maps

- In a **server-side image map**, the image map is stored on the Web server.
- Server-side image maps are supported by most graphical browsers.
- Server-side image maps can be slow to operate.
- The browser's status bar does not display the target of each hotspot.



# Server-Side Image Maps

User clicks a hotspot on the image map



The server consults the image map and accesses the link indicated on the map



The server sends the destination document back to the user



# Client-Side Image Maps

- A **client-side image map** is inserted in an image map into the HTML file.
- The browser locally processes the image map.
- Because all of the processing is done locally, you can easily test Web pages.
- More responsive than server-side maps.
- The browser's status bar displays the target of each hotspot.
- Older browsers do not support client-side images.



# Defining Image Map Hotspots

- Define a hotspot using two properties:
  - Its location in the image
  - Its shape

- Syntax of the hotspot element:

```
<area shape="shape" coords="coordinates" href="url" alt="text" />
```



# Defining Image Map Hotspots

- Use a special program that determines the image map coordinates.
- Most image map programs generate the coordinates for hotspots, as well as, the necessary HTML code.



# Creating a Rectangular Hotspot

- Two points define a **rectangular hotspot**:
  - the upper-left corner
  - the lower-right corner
- A sample code for a **rectangular hotspot** is:

```
<area shape="rect" coords="384,61,499,271" href="water.htm">
```

  - **coordinates** are entered as a series of four numbers separated by commas
  - HTML expects that the first two numbers represent the coordinates for the upper-left corner of the rectangle, and the second two numbers indicate the location of the lower-right corner
  - the **hotspot** is a hypertext link to water.htm



# Creating a Circular Hotspot

- A **circular hotspot** is defined by the location of its center and its radius.
- A sample code for a **circular hotspot** is:

```
<area shape="circle" coords="307,137,66"  
      href="karts.htm">
```

- **coordinates** are (307, 137), and it has a radius of 66 pixels.
- the **hotspot** is a hypertext link to karts.htm



# Creating a Polygonal Hotspot

- To create a polygonal hotspot, you enter the coordinates for each vertex in the shape.
- A sample code for a **polygonal hotspot** is:

```
<area shape="polygon"  
  coords="13,60,13,270,370,270,370,225,230  
  ,225,230,60" href="rides.htm">
```

- **coordinates** are for each vertex in the shape.
- the **hotspot** is a hypertext link to rides.htm