



HTML & CSS I

Modern Web Programming

(<http://my.ss.sysu.edu.cn/wiki/display/WEB/> supported by Deep Focus)

School of Data and Computer Science, Sun Yat-sen University

Part I

Outline

- **Basic HTML**
- Basic CSS
- Thinking...

Hypertext Markup Language (HTML)

- 1993: Initial official proposed description of **HTML** submitted to the **IETF** standards group
- 1995: **HTML 2** becomes an official standard language **RFC 1866**
- 1996-97: **HTML 3.2** standardizes various features
 - forms, tables, image maps, and internationalization
- 1997: **HTML 4** is proposed by **W3C** standard body
 - style sheets, scripting, frames, embedding objects, internationalization, and accessibility for disabilities
- 1999: **HTML 4.01** the last major version by **W3C**
 - A majority of pages on the Web today still use it as their starting language.
- 2001: **HTML**, HTML based on XML
- 2014: **HTML5**, RIA enabled HTML
 - initiated by **WHATWG** in 2004 , and **W3C** turned around from **HTML 2.0** to **HTML5** in 2012

Hypertext Markup Language (HTML)

- describes the **content** and **structure** of *information* on a Web page
 - not the same as the **presentation** (appearance on screen)
- surrounds text contents with opening and closing **tags**
- each tag's name is called an **element**
 - syntax: <*element*> *content* </*element*>
 - example: <p>This is an paragraph</p>
- most whitespace is insignificant in HTML (ignored or collapsed to a single space)

Hypertext Markup Language (HTML)

- **HTML coding convention:** the structure of HTML is a tree.
 - indent nested elements
 - separate siblings with blank line when it makes reading easy.
- **Responsibility of HTML languages**
 - **HTML** describes the content and structure
 - Style Sheets (**CSS**) describes the appearance of the document
 - Script (**JavaScript**) describes the behavior of the document
- **index.html**
 - <http://www.sysu.edu.cn/> = <http://www.sysu.edu.cn/index.html>

Comments: <!-- ... -->

comments to document your HTML file or “comment out” text

```
<!-- My web page, by Tim Student SS 12345, Spring  
2048 -->  
<p>SS courses are <!-- NOT --> a lot of fun!</p>
```

HTML

SS courses are a lot of fun!

output

- comments are still useful for disabling sections a page
- comments cannot be nested and cannot contain a –
- many web pages are not thoroughly commented (or at all)
 - comment is a communicative approach, to explain your designs and purposes to your colleagues, or even yourself sometime later.
 - comment is not for browsers of end users, but for developers and designers.

Structure of HTML page

```
<!DOCTYPE html>
<html>
  <head>
    information about the page
  </head>
  <body>
    page contents
  </body>
</html>
```

HTML5

- the header describes the page and the body contains the page's contents
- an HTML page is saved into a file ending with extension **.html**

Page title: <title>

describes the title of the Web page

```
<title>Chapter 2: HTML Basics</title>
```

HTML

- placed within the **head** of the page
- displayed in the Web browser's title bar and when bookmarking the page

Page meta data: <meta>

describe meta data of the Web page

```
<meta name="description" content="introduction of SYSU" />
```

HTML

```
<meta http-equiv="Content-Type" content="text/html;  
charset=gbk" />
```

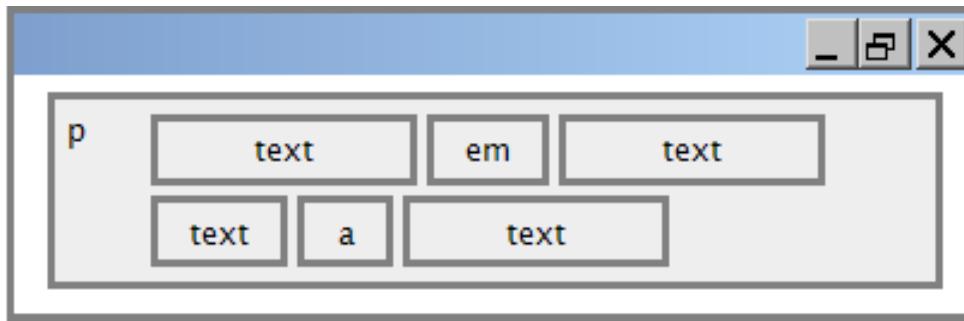
HTML

- placed within the **head** of the page
- **charset** is very significant in practice, and we often use **utf-8** for language other than English
 - character encoding and decoding, where, when, and how?

Character Encoding

- manipulating and representing of characters in computer
 - the bit length of the character
 - and the proper visual symbol
 - encoding vs. decoding
- charset
 - ASCII(basic 7b, extension 1B),
 - **iso-8859-1/latin-1** (West Europe, 1B)
 - **GB2312** (2B, Simplified Chinese)
 - **GBK**(2B, S. & T. Chinese)
 - **BIG5** (2B, Traditional Chinese)
 - GB18030 (1,2,4B, Eastern Asia)
 - Unicode (650 languages)
 - **UTF-8** (1,2,3,4B , Chinese 3B)
 - **UTF-16** (2B, 4B, Chinese 2B)
 - UTF-32 (4B, future)
 - UCS
 - UCS-2 (2B,
comparable with UTF-16)
 - UCS-4 (4B, future)

Block and inline elements



- **block** elements contains an entire large region of content
 - examples: paragraphs, lists, table cells
 - the browser places a margin of whitespace between block elements for separation
- **inline** elements effects a small amount of content
 - examples: bold text, code fragments, images
 - the browser allows many inline elements to appear on the same line
 - must be nested inside a block element

Paragraph: <p>

```
<p>You're not your job. You're not how much money you have  
in the bank. You're not the car you drive. You're not the  
contents of your wallet. You're not your khakis. You're  
the all-singing, all-dancing crap of the world.</p>
```

HTML

You're not your job. You're not how much money you have in the bank. You're not the car you drive. You're not the contents of your wallet. You're not your khakis. You're the all-singing, all-dancing crap of the world.

output

- placed within the **body** of the page
- [more paragraph examples](#)

Line break:

forces a line break in the middle of a block element (*inline*)

```
<p>Teddy said it was a hat, <br /> So I put it  
on.</p> <p>Now Daddy's sayin', <br /> Where the  
heck's the toilet plunger gone?</p>
```

HTML

Teddy said it was a hat,
So I put it on.
Now Daddy's sayin',
Where the heck's the toilet plunger gone?

output

- **br** should be immediately closed with **/>**

Headings: <h1>, <h2>, ...<h6>

headings to separate major areas of the page (*block*)

```
<h1>Sun Yat-Sen University</h1>
<h2>School of Software</h2>
<h3>Support by Google</h3>
```

HTML

Sun Yat-Sen University

School of Software

Support by Google

output

- more heading examples

Semantic HTML

- If you find the following code is shown too big in your browser, what will you do?

```
<h1>Sun Yat-Sen University</h1>
```

HTML

Sun Yat-Sen University

output

- make it from **h1** to **h3**?
- Semantic HTML – **Separation of concerns**
 - choosing tags based on the meaning of the content rather than its appearance
 - flexible and reusable

Horizontal rule: <hr />

headings to separate major areas of the page (*block*)

```
<p>First paragraph</p>
<hr />
<p>Second paragraph</p>
```

HTML

First paragraph

Second paragraph

output

- should immediately closed with */>*

More about HTML tags

- some tags can contain additional information called **attributes**
 - syntax:

```
<element attribute1="value1" attribut2="vaule2">  
content</element>
```
 - example: `Next page`
- some tags don't contain content; can be opened and closed in one tag
 - syntax: `<element attribute1="v1" attribut2="v2" />`
 - example: `<hr />`
 - example: ``

Links: <a>

links, or “anchors”, to other pages (*inline*)

```
<p>
    Search
    <a href="http://www.google.com/">Google</a> or
    our
    <a href="lectures.html">Lecture Notes</a>
</p>
```

HTML

Search Google or our Lecture Notes.

output

- uses the **href** attribute to specify the destination URL
 - can be **absolute** (to another Web site) or **relative** (to another page on this site)
- anchors are inline elements; must be placed in a block element such as **p** or **h1**

Links: <a>

- hover a link in a browser, its destination URL will be shown on the status bar
- **be descriptiveness!**

Click here to check your course schedule

output

Please check your course schedule

output

Course Schedule (please check yours before March 15!)"

output

- What's principle applied here?
- **Kind.**
 - you are kind to your Web page readers by making the page descriptive, which in turn let them understand easier.

Images:

inserts a graphical image into the page (*inline*)

```

```

HTML



output

- the **src** attribute specifies the image URL
- HTML also has an optional **alt** attribute describing the image

More about images

```
<a href="http://theonering.net/">  
      
</a>
```

HTML



output

- if placed inside an anchor, the image will become a link
- the **title** attribute specifies an optional tooltip
- images/gandalf.jpg vs. /images/gandalf.jpg

Phrase elements: ``, ``

`em`: emphasized text (usually rendered in *italic*)

`strong`: strongly emphasized text (usually rendered in **bold**)

```
<p>  
    HTML is <em>really</em>,  
    <strong>REALLY</strong> fun!  
</p>
```

HTML

HTML is *really*, **REALLY** fun!

output

- as usual, the tags must be properly nested for a valid page
- `em` vs. `i`, `strong` vs. `b`
 - SoC again~!

Nesting tags

Bad:

```
<p>
    HTML is <em>really,</em>
    <strong>REALLY</strong> lots of </strong>
    fun!
</p>
```

HTML

- tags must be correctly nested
 - a closing tag must match the most recently opened tag
- the browser may render it correctly anyway, but it is invalid HTML

Unordered list: ,

ul represents a bulleted list of items (*block*)

li represents a single item within the list (*block*)

```
<ul>
  <li>No shoes</li>
  <li>No shirt</li>
  <li>No problem!</li>
</ul>
```

HTML

- No shoes
- No shirt
- No problem!

output

More about unordered lists

a list can contain other lists:

```
<ul>
  <li>Simpsons:
    <ul>
      <li>Homer</li>
      <li>Marge</li>
    </ul>
  </li>
  <li>Family Guy:
    <ul>
      <li>Peter</li>
      <li>Lois</li>
    </ul>
  </li>
</ul>
```

HTML

- Simpsons:
 - Homer
 - Marge
- Family Guy:
 - Peter
 - Lois

output

Ordered list:

ol represents a numbered list of items (*block*)

```
<p>RIAA business model:</p>
<ol>
  <li>Sue customers for copying music</li>
  <li>???</li>
  <li>Profit!</li>
</ol>
```

HTML

RIAA business model:

1. Sue customers for copying music
2. ???
3. Profit!

output

Outline

- Basic HTML
- **Basic CSS**
- Thinking...

The bad way to produce styles

```
<p>
  <font face="Arial">Welcome to Greasy Joe's.</font>
  You will <b>never</b>, <i>ever</i>, <u>EVER</u>
  beat <font size="+4" color="red">OUR</font>
prices!
</p>
```

HTML

Welcome to Greasy Joe's. You will **never**, *ever*,
EVER beat **OUR** prices!

output

- tags such as **b**, *i*, u, and **font** are discouraged in HTML
 - Why it is bad?

Cascading Style Sheets (CSS): <link>

```
<head>
    <style type="text/css" media="screen">
        ...
    </style>
</head>
```

Embedded in HTML

```
<head>
    ...
    <link href="filename" type="text/css" rel="stylesheet"
          media="screen" />
    ...
</head>
```

standalone CSS file

- **CSS** describes the appearance and layout
 - as opposed to **HTML**, which describes **content** of the page
 - either on **screen** and on **print**
- can be embedded in HTML or placed into separate **.css** file (preferred)

Basic CSS rule syntax

```
selector {  
    property 1: value 1;  
    ...  
    property n: value n;  
}
```

CSS

```
p {  
    font-family: sans-serif;  
    color:red;  
}
```

CSS

- comments : /* */
- a **CSS** file consists of one or more **rules**
- each rule starts with a **selector** that specifies an HTML element(s) and then applies style **properties** to them
 - a selector of * selects all elements

CSS properties for colors

```
p {  
    color: red;  
    background-color: yellow;  
}
```

CSS

This paragraph uses style above

output

property	description
<u>color</u>	color of the element's text
<u>background-color</u>	color that will appear behind the element

Specifying colors

```
p { color: red; }  
h2 { color: rgb(128, 0, 196); }  
h4 { color: #FF8800; }
```

CSS

This paragraph uses the first style above.

This h2 uses the second style above.

This h4 uses the third style above.

output

color names: aqua, black, blue, fuchsia, gray, green, lime, maroon, navy, olive, purple, red, silver, teal, (white), yellow

- **RGB codes**: red, green, and blue values from 0 (none) to 255 (full)
- **hex codes**: RGB values in base-16 from 00 (0, none) to FF (255, full)

CSS properties for fonts

property	description
<u>font-family</u>	which font will be used
<u>font-size</u>	how large the letters will be drawn
<u>font-style</u>	used to enable/disable italic style
<u>font-weight</u>	used to enable/disable bold style
<u>Complete list of font properties</u>	

font-family

```
p { font-family: Georgia; }  
h2 { font-family: "Courier New"; }
```

CSS

This paragraph uses the Georgia font.

This h2 uses the Courier New font. output

- Font of Chinese Characters
 - most of browsers only support **SimSon** (“宋体”)
 - IE can support other fonts Windows OS supported
 - 黑体: SimHei 新宋体: NSimSun 仿宋: FangSong SimFang?
楷体: KaiTi SimKai? 仿宋_GB2312: FangSong_GB2312
楷体_GB2312: KaiTi_GB2312 微软雅黑: Microsoft YaHei
 - ...

More about font-family

```
p {  
    font-family: Garamond, "Times New Roman", serif;  
}
```

CSS

If no Garamond then uses TNR, and then uses serif.

output

- enclose multi-word font names in quotes
- can specify multiple fonts from highest to lowest priority
- generic font names:**
 - serif, sans-serif, monospace, cursive

font-size

```
p {  
    font-size: 20pt;  
}
```

CSS

This paragraph uses font size 20pt.

output

- **units**: pixels (**px**) vs. point(**pt**) vs. m-size(**em**)
 - 16px, 16pt, 1.16em
- **vague font size**: xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger
- **percentage font sizes**, e.g.: 90%, 120%

font-weight, font-style

```
p {  
    font-weight: bold;  
    font-style: italic  
}
```

CSS

This paragraph is bold and italic.

output

- either of the above can be set to normal to turn them off

Outline

- Basic HTML
- Basic CSS
- **Thinking...**

Thinking ...

- What's the difference?

```
void swamp(int a, int b)
{
    int temp;
    temp = a;
    a = b;
    b = temp;
}
```

C

```
<html>
    ...
<body>
    <h1>Supper Man</h1>
    <p>
        The guy teaches you Web.
    </p>
    ...

```

HTML

```
<?php
$file="1.txt";
$fp=fopen($file,"r");
$content= fread(
    $fp,filesize($file));
fclose($fp);
?>
```

PHP

```
body {
    background-color: #997788;
    font-family: SimSon;
}
h1 {
    color: blue
}
```

CSS

Declarative Programming

- Imperative vs. Declarative
 - **declarative programming** is a programming paradigm that expresses the logic of a computation without describing its control flow.
-- Lloyd, J.W., *Practical Advantages of Declarative Programming*
 - in contrast with imperative programming, which requires an explicitly provided algorithm.
- Subparadigms
 - functional programming: Scheme, Erlang, Haskell, ...
 - logic programming: Prolog
 - domain-specific languages: SQL, CSS, HTML, XSLT, SVG, XAML, regular expressions
 - constraint programming: often used as a complement to other paradigms
 - Hybrid languages: Makefiles, yacc

Summary

- HTML
 - HTML & HTML
 - HTML Tags: **title, meta, p, h1, ..., hr, a, img, br, comments, em, strong, ul, ol, li**
 - block vs. inline
 - character encoding
- CSS
 - why? how?
 - link, rule
 - properties: for color, for fonts
- Thinking:
 - SoC
 - Declarative Programming

Exercises

- write a web page for yourself which contains your self-introduction, recent photos, courses selected this semester, and favorite movies
 - your introduction should be more than one paragraph
 - your courses should be a ordered list
 - your favorite movies should be a unordered list
 - having a link to the course site of MWP
 - applying different fonts for the readability

Further Readings

- <http://en.wikipedia.org/wiki/HTML>
- [http://en.wikipedia.org/wiki/Cascading Style Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)
- Chapter 1~8, *Web Programming with HTML, HTML, and CSS* <http://my.ss.sysu.edu.cn:8080/display/W2PSC/References+and+Books>
- List of all HTML tags: <http://www.w3schools.com/tags/default.asp>
- List of HTML character entities:
http://www.w3schools.com/tags/ref_entities.asp
- HTML 1.1 Spec. <http://www.w3.org/TR/HTML11/>
- HTML 1.1 Elements Reference:
<http://www.w3.org/2007/07/HTML-basic-ref.html>
- W3 List of all CSS properties: <http://www.w3.org/TR/CSS21/propidx.html>
- W3 CSS 2.1 Specifications: <http://www.w3.org/TR/CSS21/>
- Fonts of each operating systems: <http://www.apaddedcell.com/web-fonts>

Part II

Outline

- More HTML Elements
- More Basic CSS
- CSS in practice

Web page metadata: <meta>

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<meta name="description" content="Authors' web site for Building Java
Programs." />
<meta name="keywords" content="java, textbook" />
```

HTML

- name
 - **author**
 - **description**
 - **keywords**
 - **generator**
 - **revised**
- http-equiv
 - **content-type**
 - **expires**
 - **refresh**

Table: <table>, <tr>, <td>, <th>, <caption>

```
<table>
  <caption>Smart Guys</caption>
  <tr><th>name</th><th>gender</th></tr>
  <tr><td>Bill</td><td>male</td></tr>
  <tr><td>Susan</td><td>female</td></tr>
</table>
```

HTML

Smart Guys	
name	gender
Bill	male
Susan	female

output

- Never use Table for layout~!

Definition list: **<dl>**, **<dt>**, **<dd>**

dl represents a list of definitions of terms(*block*)

dt represents each term, and **dd** its definition

```
<dl>
  <dt>newbie</dt><dd>one who does not have mad skills</dd>
  <dt>own</dt><dd>to soundly defeat
    (e.g. I owned that newbie!)</dd>
  <dt>frag</dt> <dd>a kill in a shooting game</dd>
</dl>
```

HTML

newbie

one who does not have mad skills

own

to soundly defeat (e.g. I owned that newbie!)

frag

a kill in a shooting game

output

Quotations: <blockquote>

a lengthy quotation (*block*)

```
<p>As Lincoln said in his famous Gettysburg Address:</p>
<blockquote>
  <p>Fourscore and seven years ago, our fathers brought forth on
  this continent a new nation, conceived in liberty, and dedicated
  to the proposition that all men are created equal.</p>
</blockquote>
```

HTML

As Lincoln said in his famous Gettysburg Address:

*Fourscore and seven years ago, our fathers brought forth
on this continent a new nation, conceived in liberty, and
dedicated to the proposition that all men are created equal.*

output

Inline quotations: <q>

a short quotation (*inline*)

```
<p>Quoth the Raven, <q>Nevermore.</q></p>
```

HTML

Quoth the Raven, “Nevermore”.

output

- Why not just write the following?
 - <p>Quoth the Raven, "Nevermore."</p>
- We don't use “ mark for two reasons:
 - HTML shouldn't contain literal quotation mark characters; they should be written as "
 - using <q> allows us to apply CSS styles to quotations

HTML character entities

a way of representing any Unicode character within a Web page

character (s)	entity
< >	< >
é è ñ	é è ñ
™ ©	™ ©
π δ Δ	π δ Δ
И	И
" &	" &

- Complete list of HTML entities
- How would you display the text & on a web page?

HTML-encoding text

```
&lt;p&gt; &lt;a href="http://google.com/search?  
q=marty&ie=utf-8&aq=t"&gt; Search Google for  
Marty &lt;/a&gt; &lt;/p&gt;
```

HTML

```
<p> <a href="http://google.com/search?  
q=marty&ie=utf-8&aq=t"> Search Google for Marty </a>  
</p>
```

output

- To display the link text in a Web page, its special characters must be encoded as shown above

Computer code: `<code>`

code: a short section of computer code (usually rendered in a fixed-width font)

```
<p> The <code>ul</code> and <code>ol</code> tags make lists. </p>
```

HTML

The ul and ol tags make lists.

output

Preformatted text: <pre>

a large section of pre-formatted text (*block*)

```
<pre>
```

Steve Jobs speaks loudly
 reality distortion
 Apple fans bow down

```
</pre>
```

HTML

Steve Jobs speaks loudly
 reality distortion
 Apple fans bow down

output

- displayed with exactly the whitespace / line breaks given in the text
- shown in a fixed-width font by default
- how would it look if we had instead enclosed it in code tags?

Outline

- More HTML Elements
- **More Basic CSS**
- CSS in practice

Grouping styles

```
p, h1, h2 {  
    color: green;  
}  
h2 {  
    background-color: yellow;  
}
```

CSS

This paragraph uses the above style.

This h2 uses the above style.

output

- a style can select multiple elements separated by commas
- the individual elements can also have their own style (like **h2** above)

CSS properties for text

property	description
<u>text-align</u>	alignment of text within its element
<u>text-decoration</u>	decorations such as underlining
<u>line-height</u> , <u>word-spacing</u> , <u>letter-spacing</u>	gaps between the various portions of the text
<u>text-indent</u>	indents the first letter of each paragraph
<u>Complete list of text properties</u>	

text-align

```
blockquote { text-align: justify; }  
h2 { text-align: center; }
```

CSS

The Emperor's Quote

[TO LUKE SKYWALKER] The alliance... will die. As will your friends. Good, I can feel your anger. I am unarmed. Take your weapon. Strike me down with all of your hatred and your journey towards the dark side will be complete.

output

- **text-align** can be **left**, **right**, **center**, or **justify** (which widens all full lines of the element so that they occupy its entire width)

text-decoration

```
p {  
    text-decoration: underline;  
}
```

CSS

This paragraph uses the style above.

output

- can also be `overline`, `line-through`, `blink`, or `none`
- effects can be combined:
 - `text-decoration: overline underline`

The list-style-type property

```
ol { list-style-type: upper-roman }
```

CSS

- Possible values:
 - none : No marker
 - I. disc (default), circle, square
 - II. decimal : 1, 2, 3, etc.
 - III. decimal-leading-zero : 01, 02, 03, etc.
 - IV. lower-roman : i, ii, iii, iv, v, etc.
 - V. upper-roman : I, II, III, IV, V, etc.
 - VI. lower-alpha : a, b, c, d, e, etc.
 - VII. upper-alpha : A, B, C, D, E, etc.
 - VIII. lower-greek : alpha, beta, gamma, etc.
 - IX. others: hebrew, armenian, georgian, cjk-ideographic, hiragana, katakana, hiragana-iroha, katakana-iroha

Outline

- More HTML Elements
- More Basic CSS
- **CSS in practice**

Body Styles

```
body { font-size: 16px; }
```

CSS

- to apply a style to the entire body of your page, write a selector for the **body** element
- saves you from manually applying a style to each element

Styles that conflict

```
p, h1, h2 { color: blue; font-style: italic; }  
h2 { color: red; background-color: yellow; }
```

CSS

This paragraph uses the first style above.

This heading uses both styles above.

- when two styles set conflicting values for the same property, the latter style takes precedence
- (later we will learn about more specific styles that can override more general styles)

W3C CSS Validator

```
<p>
  <a href="http://jigsaw.w3.org/css-validator/check/referer">      HTML
    
  </a>
</p>
```



output

- jigsaw.w3.org/css-validator/
- checks your CSS to make sure it meets the official CSS specifications
- more picky than the web browser, which may render malformed CSS correctly

CSS properties for backgrounds

property	description
<u>background-color</u>	color to fill background
<u>background-image</u>	image to place in background
<u>background-position</u>	placement of bg image within element
<u>background-repeat</u>	whether/how bg image should be repeated
<u>background-attachment</u>	whether bg image scrolls with page

background-image

```
body {  
    background-image: url("images/draft.jpg");  
}
```

CSS

This is the first paragraph

This is the second paragraph...

It occupies 2 lines



draft.jpg

- background image/color fills the element's content area

background-repeat

```
body {  
    background-image: url("images/draft.jpg");  
    background-repeat: repeat-x;  
}
```

CSS

This is the first paragraph

This is the second paragraph...

It occupies 2 lines

DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT DRAFT

- can be **repeat** (default), **repeat-x**, **repeat-y**, or **no-repeat**

background-position

```
body {  
    background-image: url("images/draft.jpg");  
    background-repeat: no-repeat;  
    background-position: 370px 20px;  
}
```

CSS

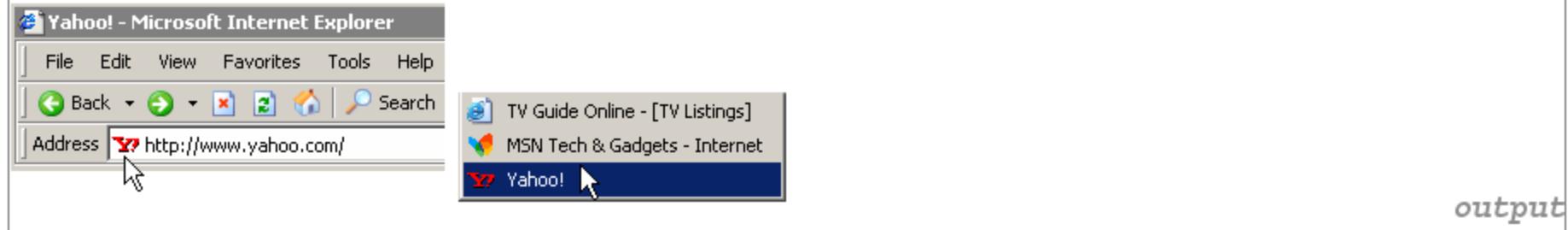
This is the first paragraph
This is the second paragraph...
It occupies 2 lines

DRAFT

- value consists of two tokens, each of which can be **top**, **left**, **right**, **bottom**, **center**, a percentage, or a length value in **px**, **pt**, etc.
- value can be negative to shift left/up by a given amount

Favorites icon (“favicon”)

```
<link href="filename" type="MIME type" rel="shortcut icon" /> HTML  
<link href="yahoo.gif" type="image/gif" rel="shortcut icon" />HTML
```



- the **link** tag, placed in the HTML page's **head** section, can specify an icon for a Web page.
- IE6: this doesn't work; must place a file in **.ico** format named **favicon.ico** in the root directory of the Web server ([instructions](#))

Summary

- More HTML Elements
 - meta
 - dl, dt, dd
 - blockquote, q
 - HTML character entities, HTML-encoding text
 - code, pre
- More Basic CSS
 - grouping style
 - comments
 - text properties: text-align, text-decoration
 - list-style-type

Summary

- CSS in practice
 - body styles
 - cascading vs. inherit
 - conflict resolve
 - W3C CSS validator
 - background properties: `background-image`, `background-repeat`, `background-position`
 - favorites icon

Exercises

- list definitions all html tags and css properties we learned this course in a web page with explanations of their purposes and usages
 - `dl`, `dt`, `dd` for definitions
 - `blockquote`, `q` for quoted sentences from [w3 school](#)
 - `code`, `pre` for examples
 - applying style with a standalone css file

Further Readings

- <http://en.wikipedia.org/wiki/HTML>
- http://en.wikipedia.org/wiki/Cascading_Style_Sheets
- Chapter 1~8, *Web Programming with HTML, HTML, and CSS* <http://my.ss.sysu.edu.cn:8080/display/W2PSC/References+and+Books>
- List of all HTML tags: <http://www.w3schools.com/tags/default.asp>
- List of HTML character entites:
http://www.w3schools.com/tags/ref_entities.asp
- HTML 1.1 Spec. <http://www.w3.org/TR/HTML11/>
- HTML 1.1 Elements Reference:
<http://www.w3.org/2007/07/HTML-basic-ref.html>
- W3 List of all CSS properties: <http://www.w3.org/TR/CSS21/propidx.html>
- W3 CSS 2.1 Specifications: <http://www.w3.org/TR/CSS21/>
- Fonts of each operating systems: <http://www.apaddedcell.com/web-fonts>

Grouping

<http://grouper.meteor.com/>

Thank you!

