

Cascading Style Sheet

The right way to specify rendering

- HTML specifications of documents should contain only information about how a document is structured into logical units such as paragraphs, pictures, lists, etc.
- HTML specifications should not contain rendering information
- And yet:
 - *we often want to specify how we want a document rendered*

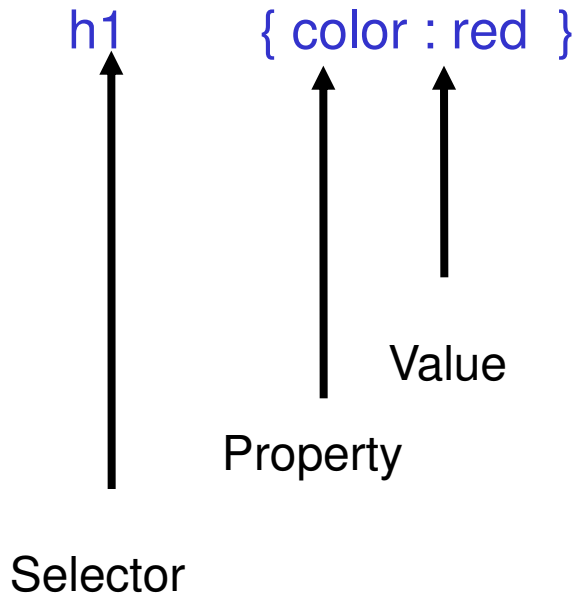
Structure versus Rendering

Question:

- How can we specify our own style of rendering without mixing rendering rules inside a document's logical specification?
- We should use Style sheets
- A style sheet contains a set of rendering specifications
- We can apply a style sheet to a HTML document specification in order to change the appearance of the document when it is rendered by a browser

A simple style rule

This style sheet specifies that all <H1> items in any document to which the style sheet is applied should be printed in red



- We can store style sheets in files with the extension “**.css**”
- We can apply a style sheet which is stored in a .css file to many HTML documents

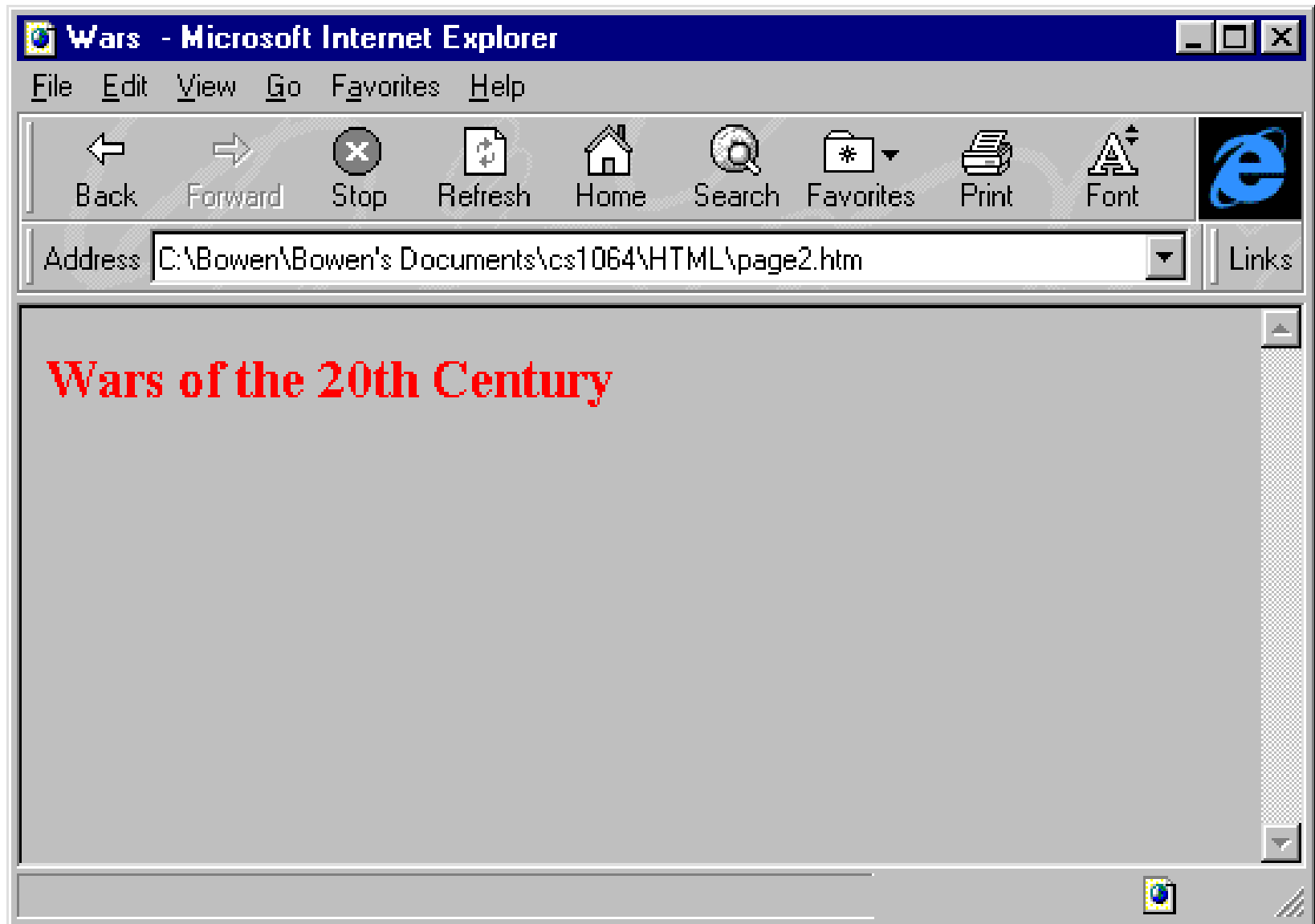
Storing style sheets:

- For example, the style sheet

```
h1 { color : red}
```

could be stored in a file called **style1.css**
- We apply a style sheet in a .css file to a document in a .html file by calling the .css file in the <HEAD> part of the .html file:

```
<HTML>
<HEAD>
<TITLE> ... </TITLE>
<LINK REL=STYLESHEET TYPE="text/css" HREF="style1.css" >
</HEAD>
<BODY>
<H1> Wars of the 20th Century </H1>
</BODY>
</HTML>
```



A more sophisticated style sheet

Specifying styles for `<p>` items as well as `<h1>` items

style2.css

```
h1    {color : red}
```

```
p     {color : blue}
```

<HTML>

<HEAD>

<TITLE> ... </TITLE>

<LINK REL=STYLESHEET TYPE="text/css" HREF="style2.css" >

</HEAD>

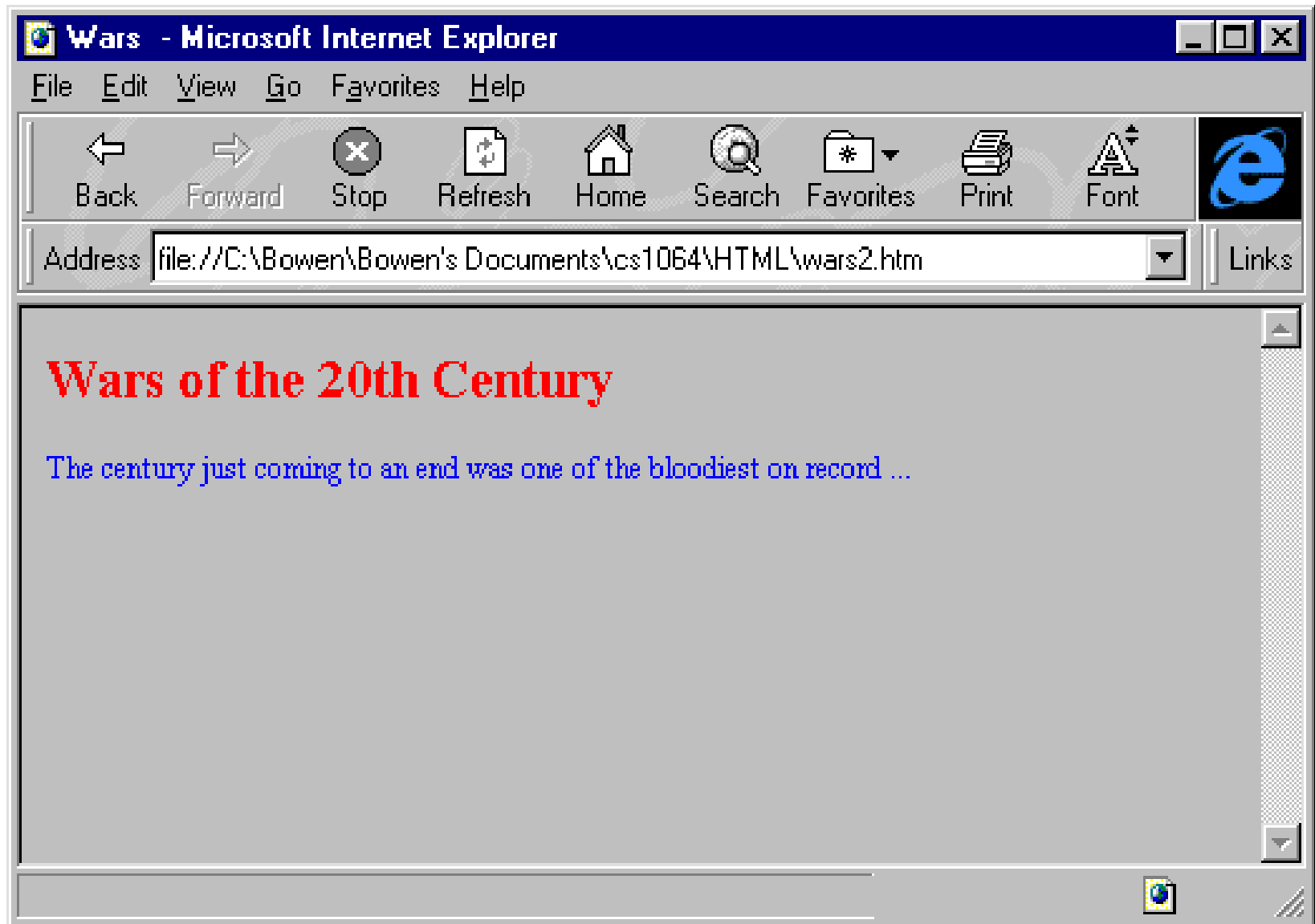
<BODY>

<H1> Wars of the 20th Century</H1>

<P>The century just coming to an end was one of the bloodiest
on record ... </P>

</BODY>

</HTML>



Several H1 style attributes

h1 {color : red; font-size : 50}

p {color : blue}

h1 { color : red;
font-size : 50 ; font-family : arial }

p { color : blue }

- We can, of course, specify styles for most HTML tags
- In case no style specified **default** style is used
 - Browser default style
 - User default style
 - Author defined style

What is the precedence ?

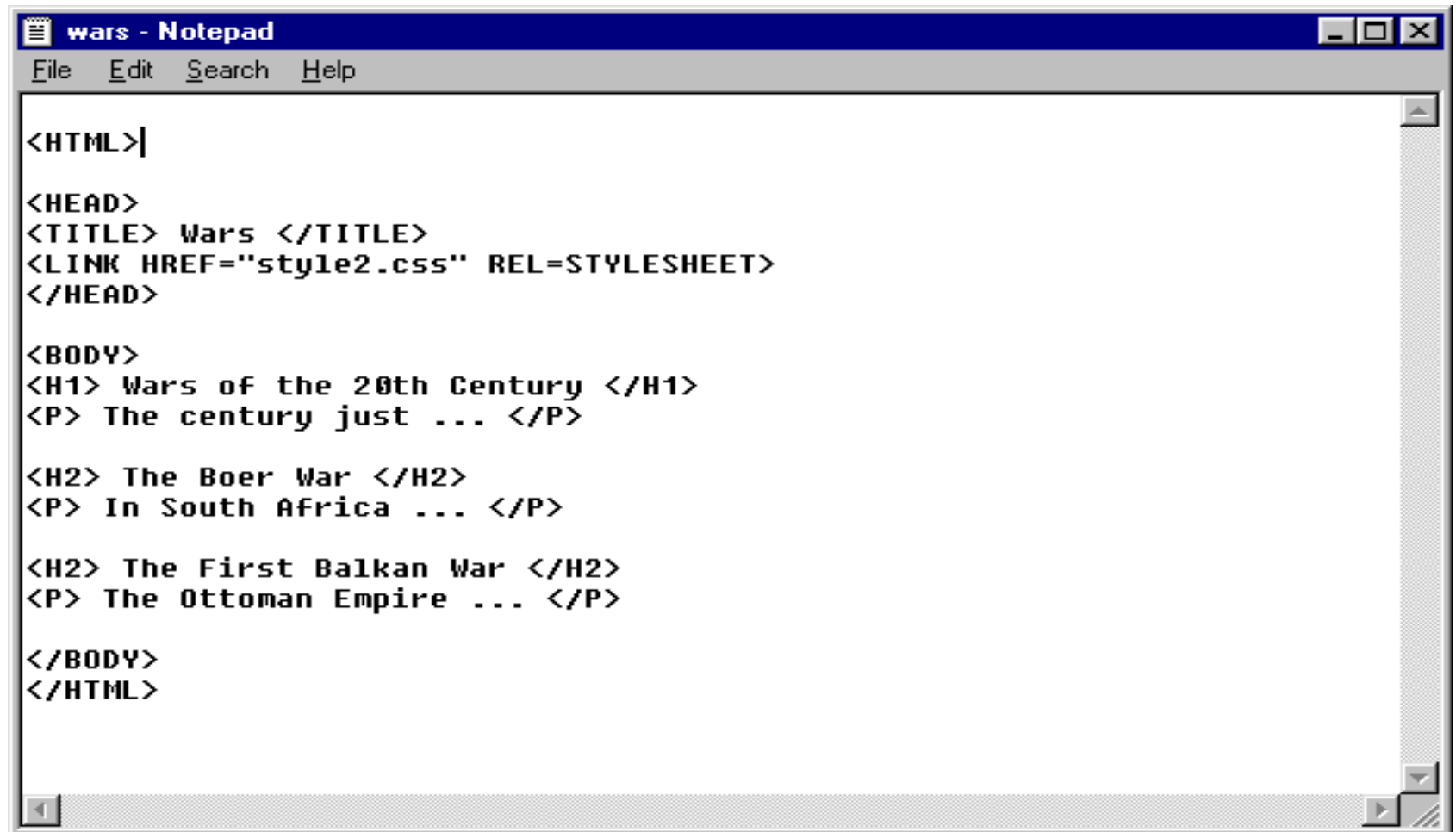
By default, rules in author style sheets have more weight than rules in user style sheets.

Precedence is reversed, however, for **"!important"** rules.

```
body {  
    color: black !important;  
    background: white !important;  
}
```

```
* {  
    color: inherit !important;  
    background: transparent !important;  
}
```

wars2.htm



```
<HTML>|

<HEAD>
<TITLE> Wars </TITLE>
<LINK HREF="style2.css" REL=STYLESHEET>
</HEAD>

<BODY>
<H1> Wars of the 20th Century </H1>
<P> The century just ... </P>

<H2> The Boer War </H2>
<P> In South Africa ... </P>

<H2> The First Balkan War </H2>
<P> The Ottoman Empire ... </P>

</BODY>
</HTML>
```

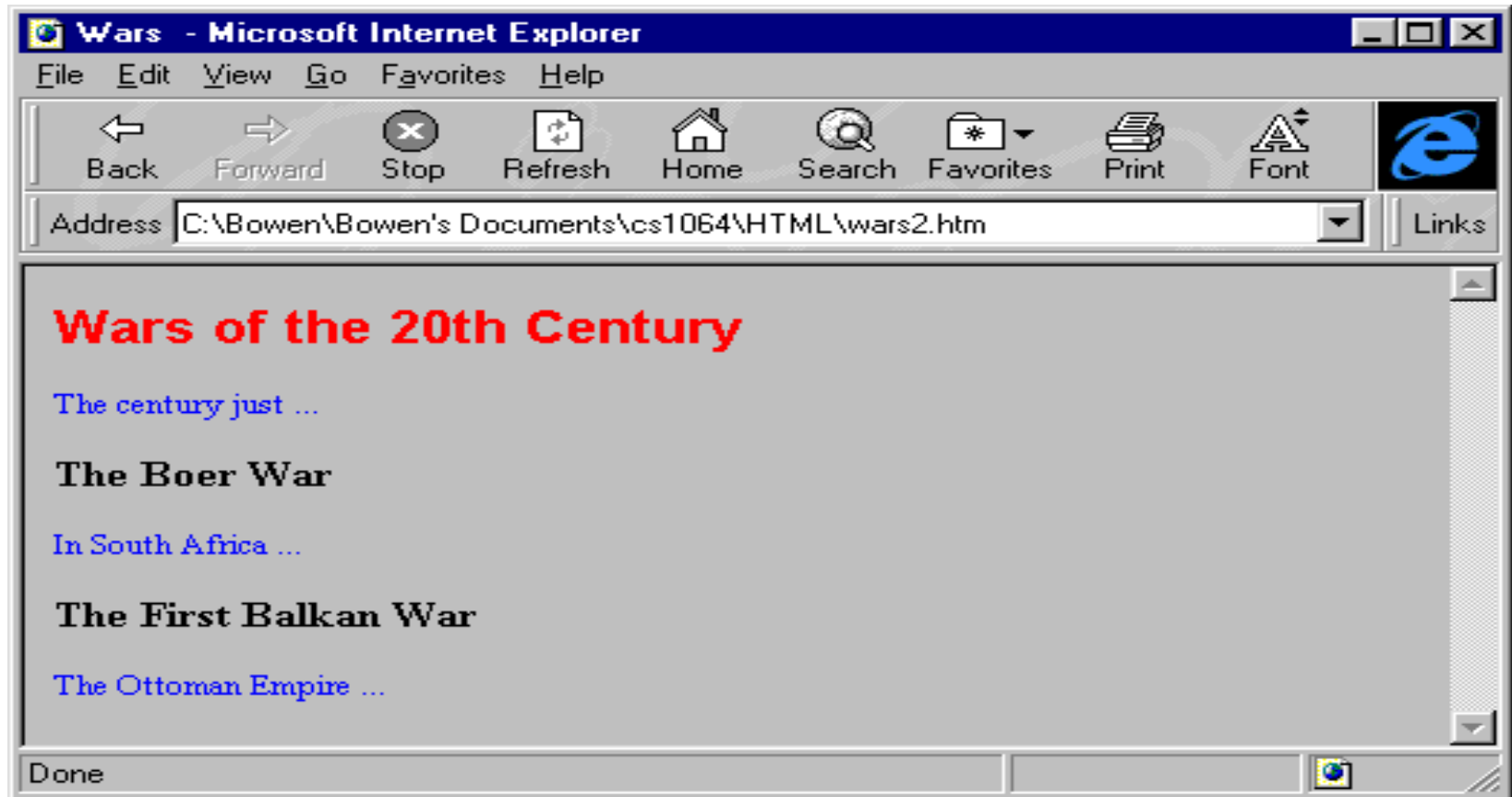
style2.css

- This first style-sheet does not specify anything for the H2 headings:

```
h1      {color : red ;  
         font-size : 20 ; font-family : arial}  
p       {color : blue}
```

- So, as we see on the next slide, H2 headings have the default style

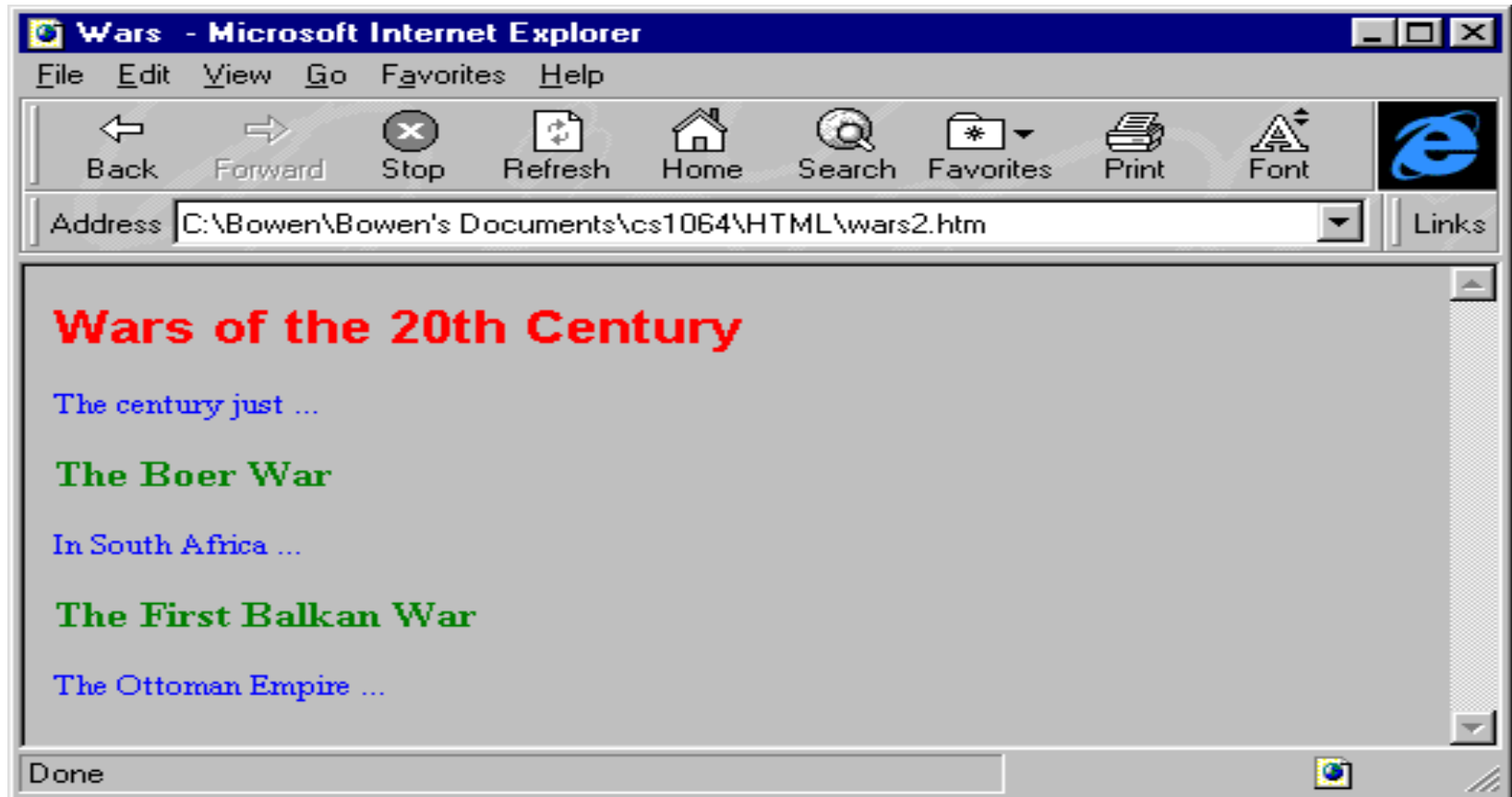
Applying style2.css to wars2.htm



style3.css

```
h1 { color : red ;  
      font-size : 20 ; font-family : arial }  
h2 { color : green }  
p   { color : blue }
```


Applying style3.css to wars2.htm



Style sheets can also affect the BODY

style4.css

```
body    { background-color : white}
h1      { color : red;
          font-size : 50; font-family : arial}
h2      { color : green}
p       { color : blue}
```

Notice carefully:

- The HTML language is evolving
- This means, unfortunately, that it is not as elegant as it could be
- A defect is that there are inconsistencies between various generations of the language
- Remember attribute **bgcolor** -- you should **not** use it because
 - In style sheets, the same effect is achieved with **background-color**

FONT-STYLE

```
body    { background-color : white}
h1      { color : red;
          font-size : 50; font-family : arial}
h2      { color : green}
p       { color : blue}
```

- A style sheet comprises a sequence of one or more items of this type:

selector { declaration }

- The **selector** specifies a type of logical item whose rendering we want to control
- The **declaration** specifies the rendering

Places to specify style sheets

- In a separate file and called by a LINK (this makes a style sheet available for lots of documents)
- Within the HEAD of a document, but this restricts its usage to the parent document
- Embedded within the BODY content of a document -- there are only rare situations where this is needed or desirable

in-HEAD style sheets

- An in-HEAD style sheet must be delimited by a pair of tags: <STYLE> and </STYLE>
- Example:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE> Wars of the 20th Century </TITLE>
```

```
<STYLE>
```

```
    BODY {BACKGROUND-COLOR : white}
```

```
    H1    {COLOR : red ; FONT-SIZE : 50 ; FONT-FAMILY : arial}
```

```
    H2    {COLOR : green ; FONT-STYLE : italic}
```

```
    P     {COLOR : blue}
```

```
</STYLE>
```

```
</HEAD>
```

```
<BODY>
```

```
....
```

Prioritization (Cascading)

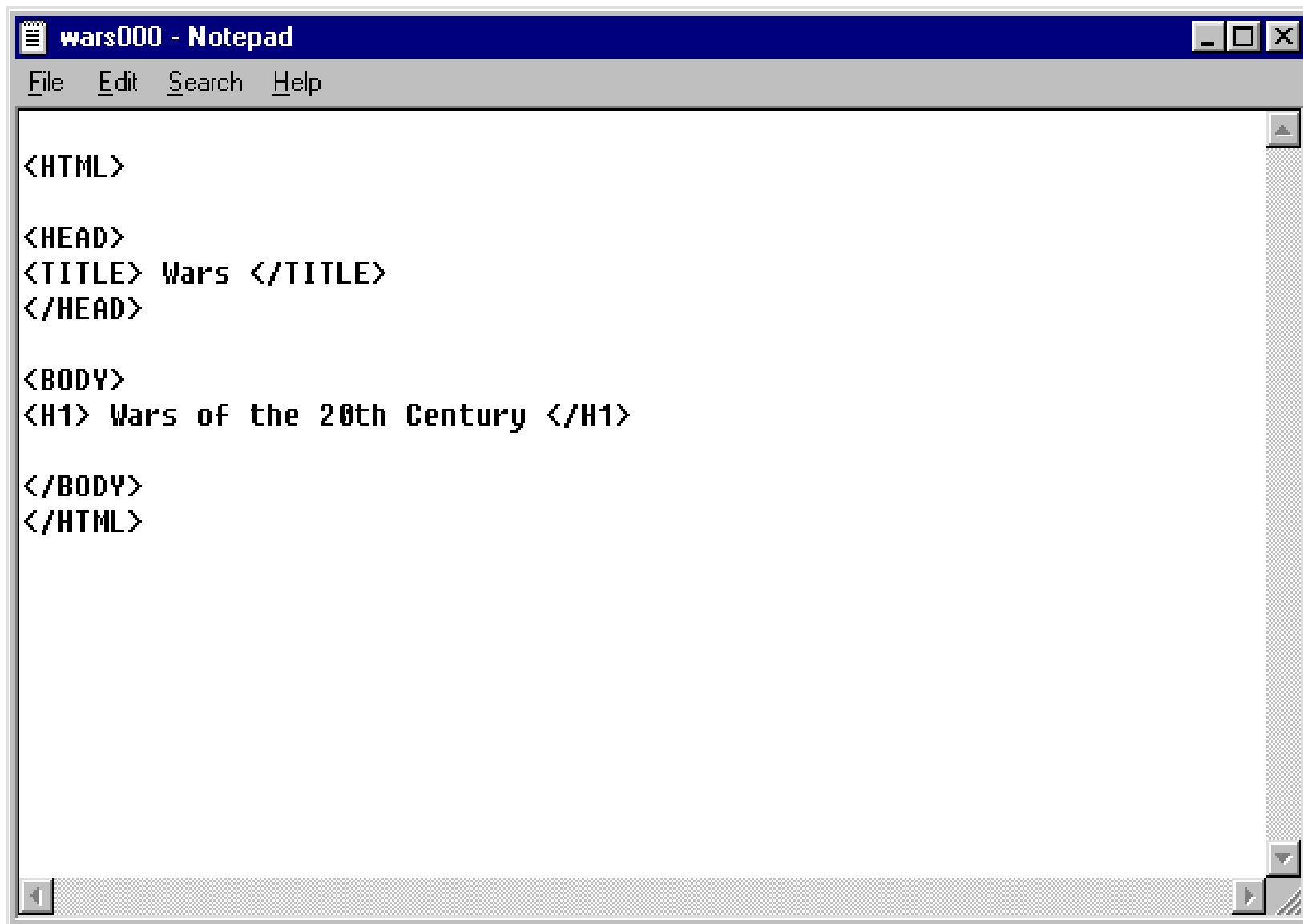
- More specific style specifications override more general ones
- If you use linked, HEAD and in-BODY style sheets, in-BODY is given first preference, then HEAD then linked sheets
- A style for a child element overrides one for its parent/ancestor

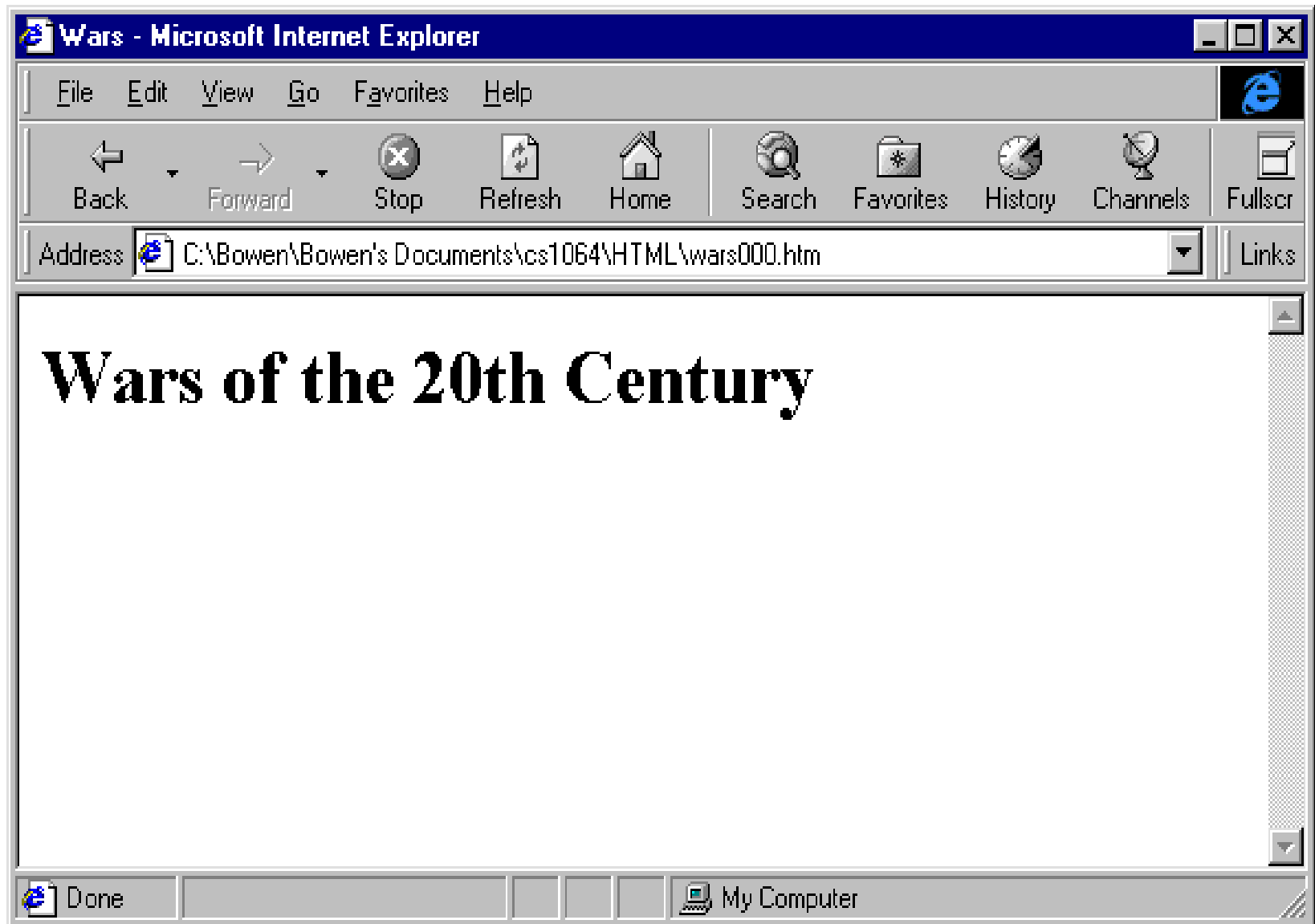
Logical items and their “boxes”

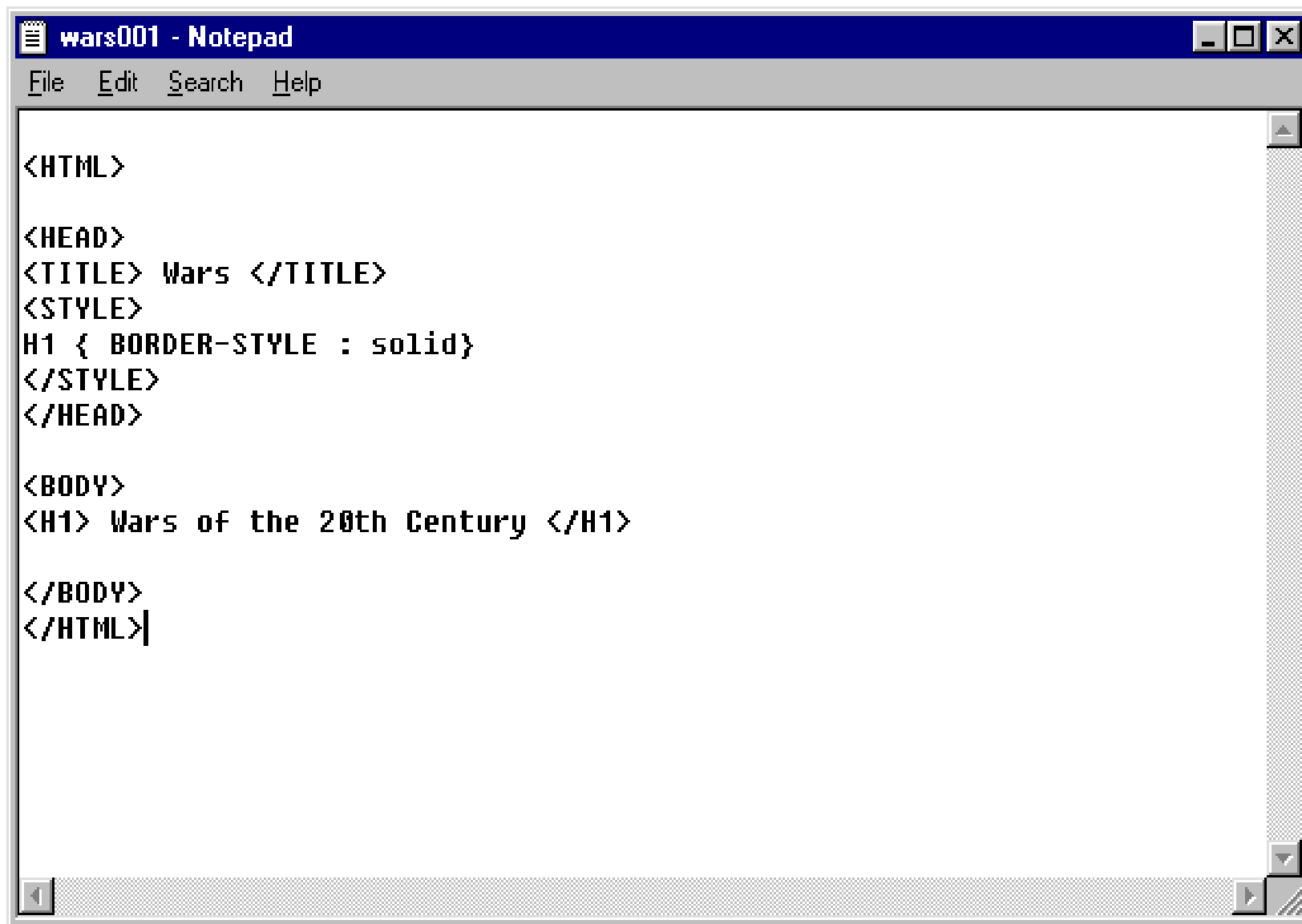
- Most logical items can be regarded as being placed, by the browser, in a “box”
- A box has various attributes, such as
 - empty margin outside the box, border of the box, padding between border of the box and its content, background of the box
- These attributes can be controlled
- The content of a box has attributes, such as color, font, spacing, alignment, decoration, indentation, transformation, and others

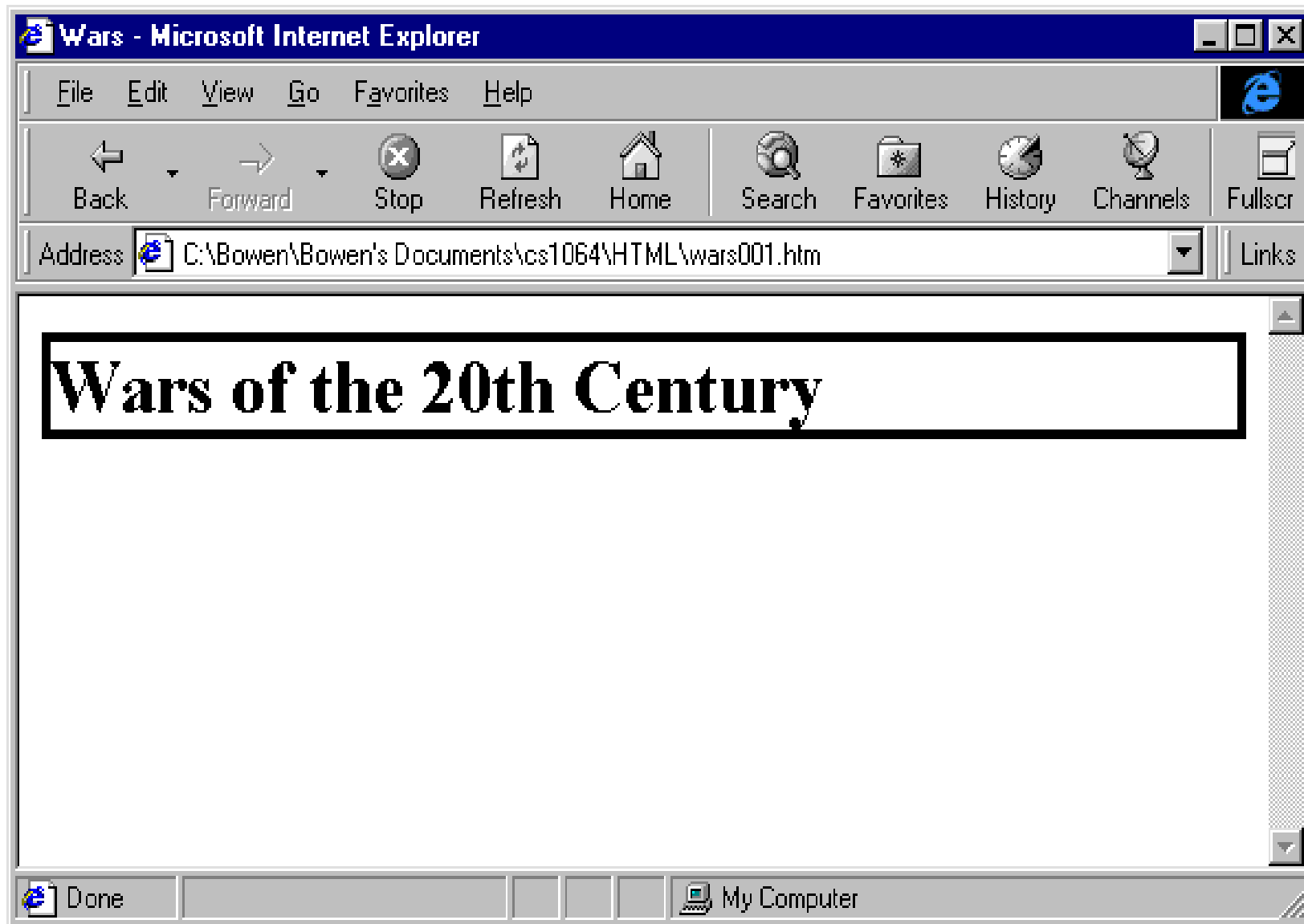
Controlling Box attributes

- A box has a BORDER
- A border has a STYLE which can have the following values:
 - none, dotted, dashed, solid, double, groove, ridge, inset and outset
 - none is the default value
- Not all browsers may support all values of the STYLE, displaying many of them as if they were solid



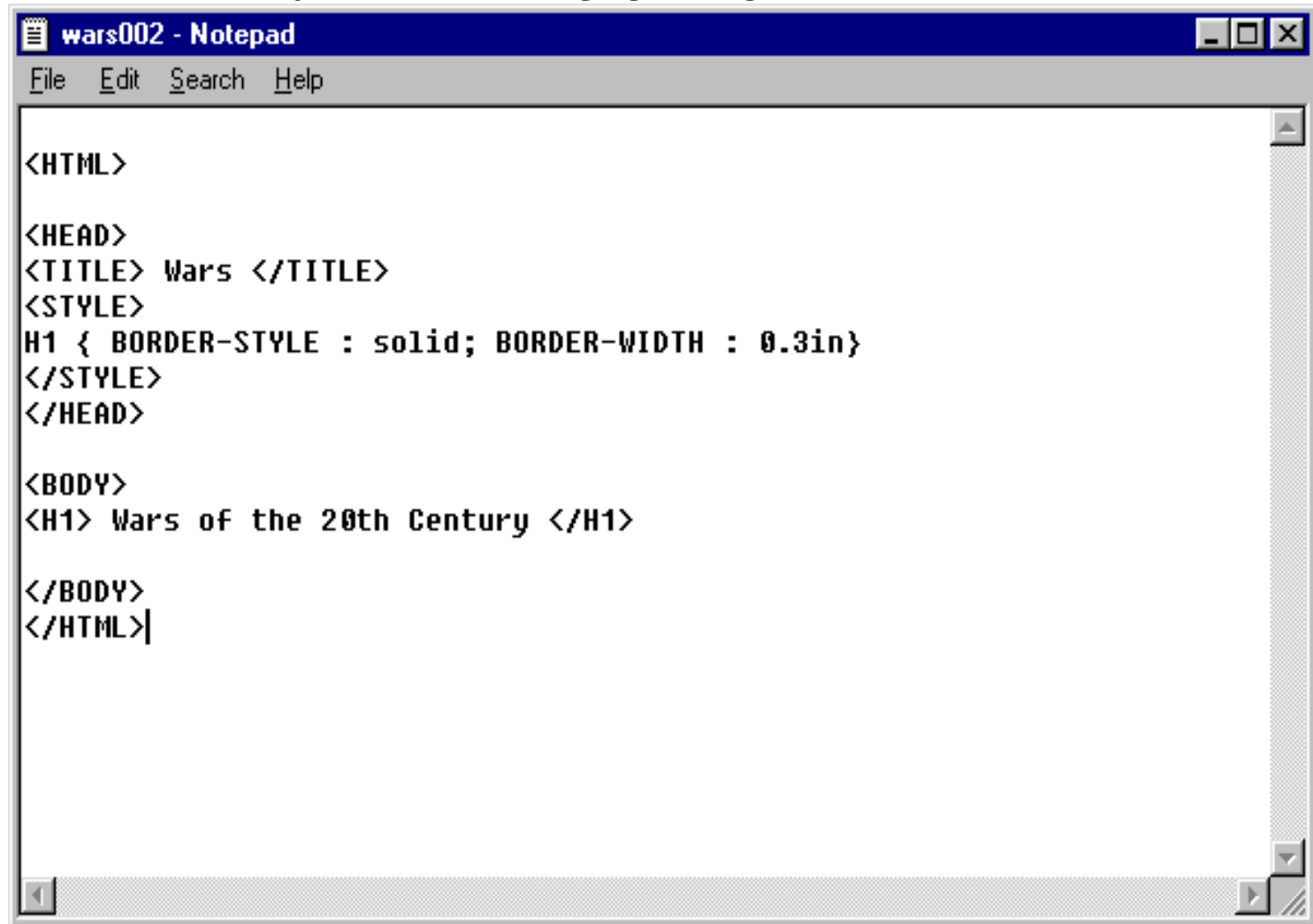






Box BORDER

- A border has a WIDTH which can be specified in various ways, including giving a value in inches

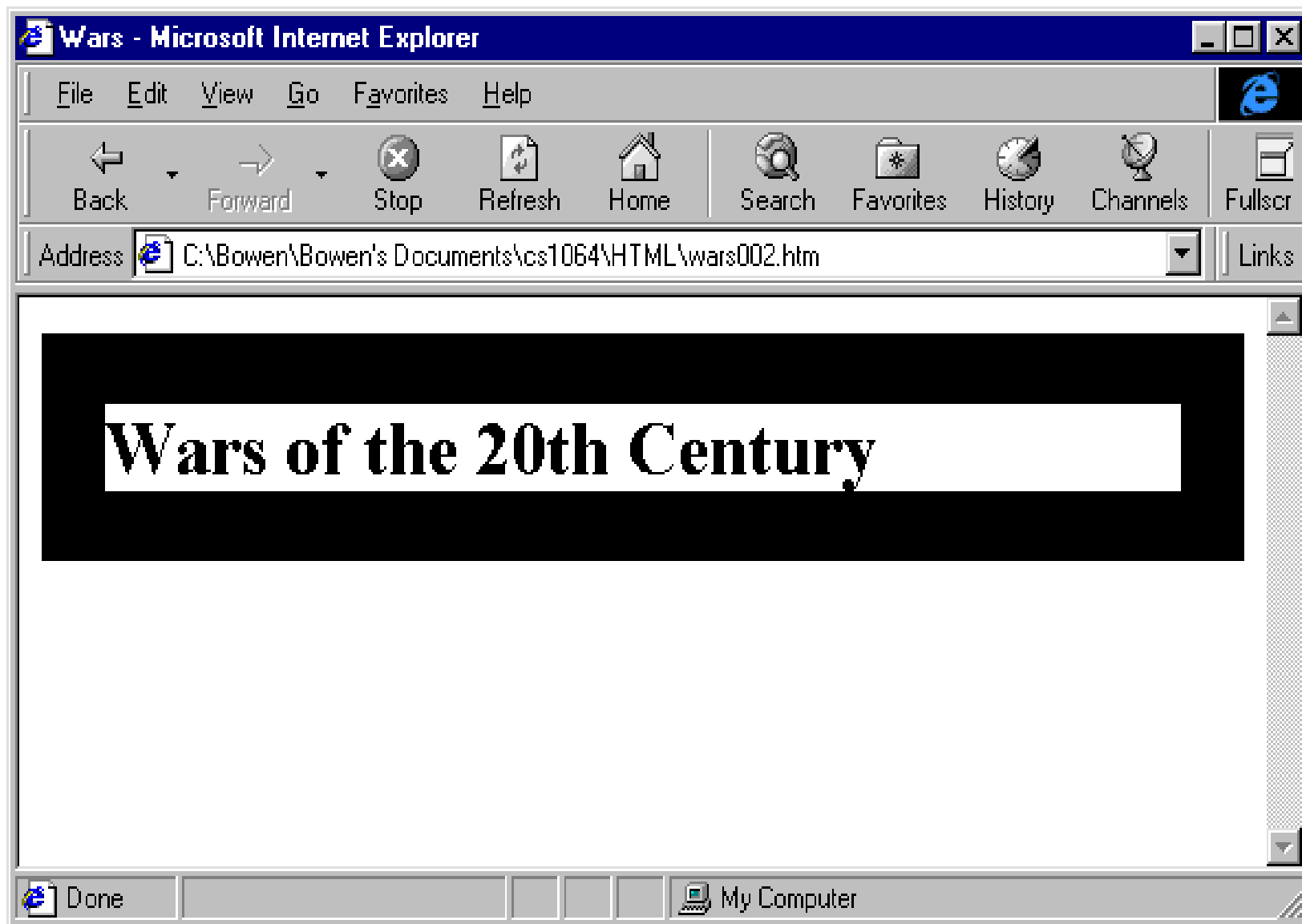


```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-STYLE : solid; BORDER-WIDTH : 0.3in}
</STYLE>
</HEAD>

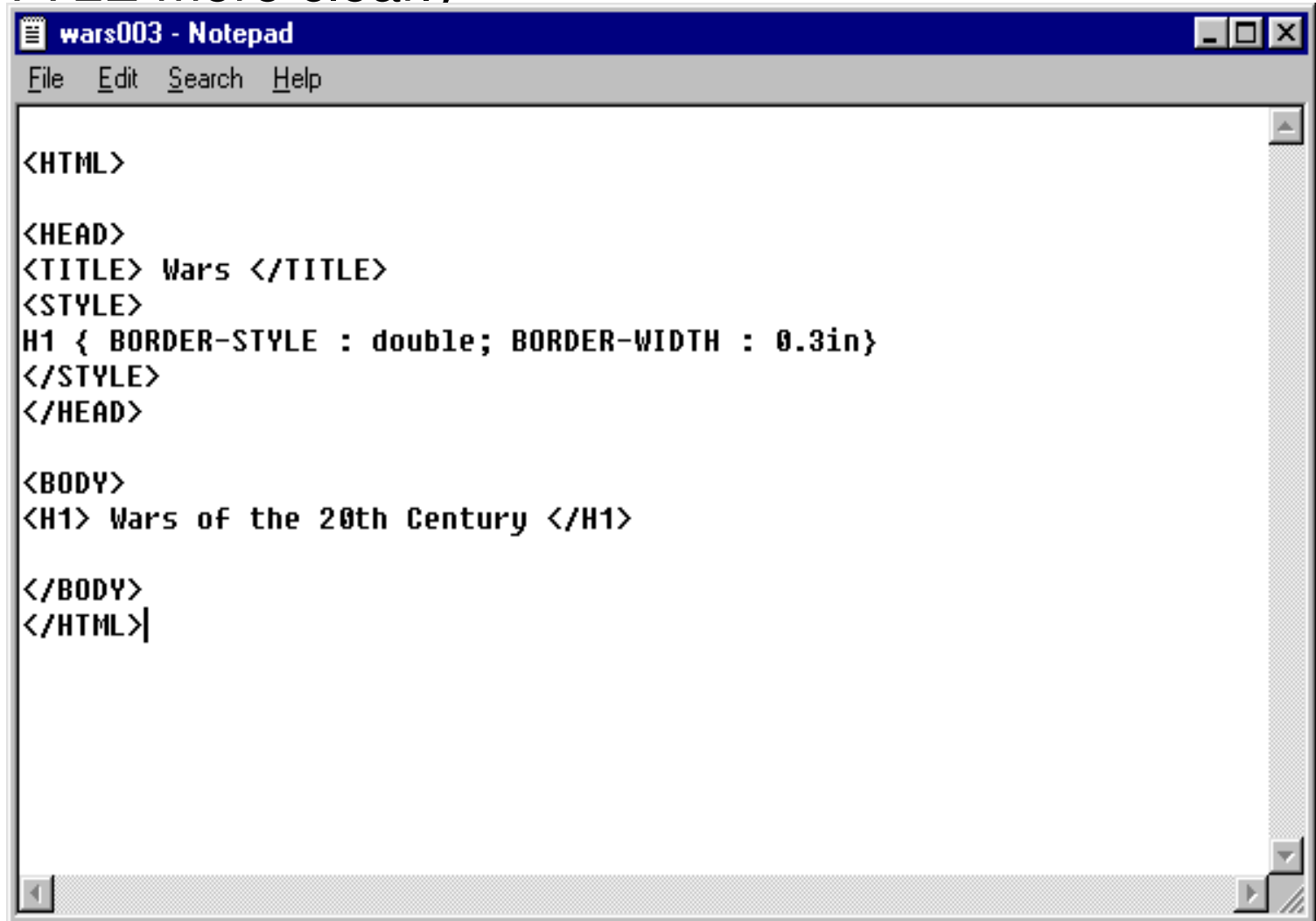
<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```



Box BORDER

- Now that we have a thick BORDER, we can see the effect of STYLE more clearly

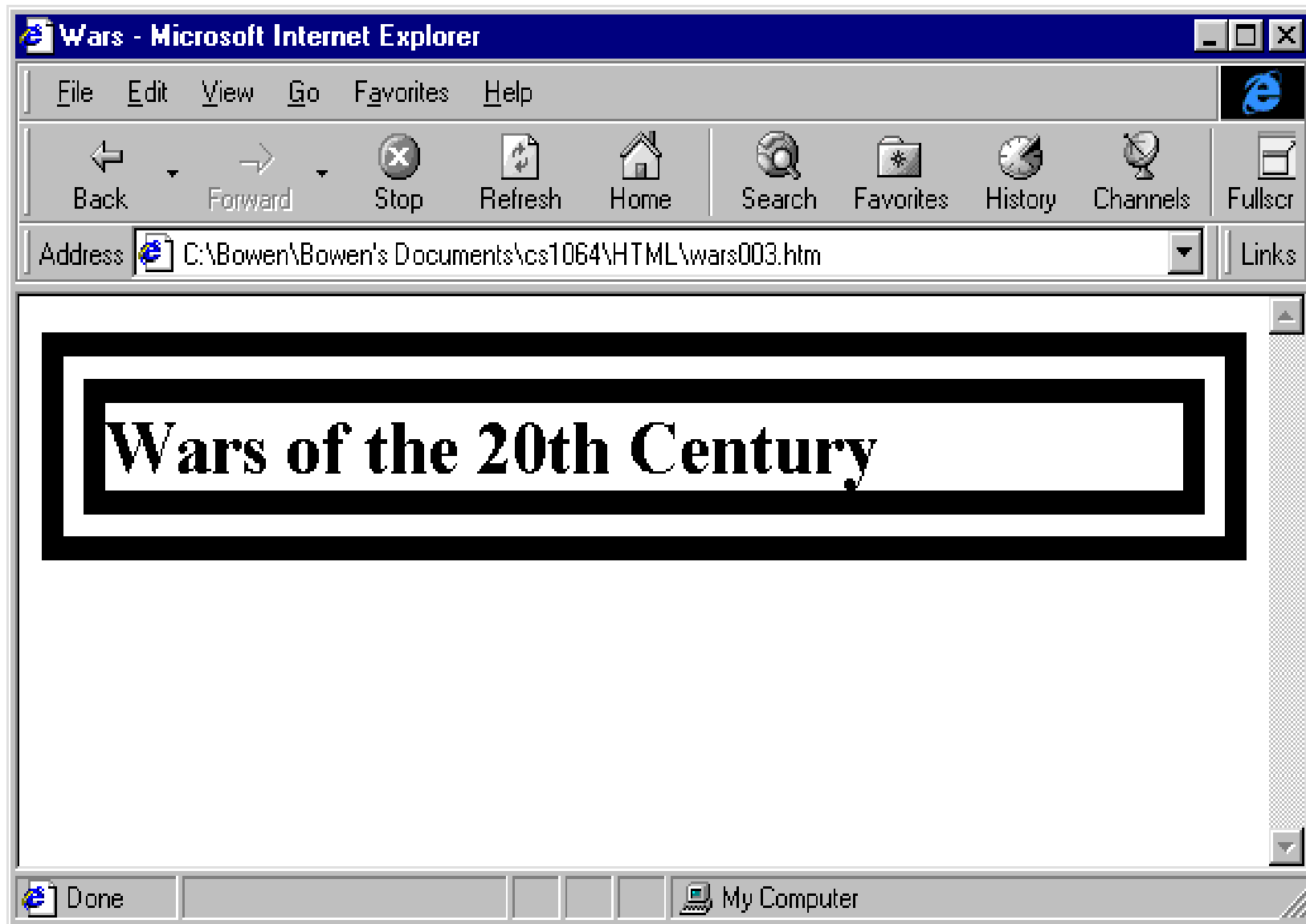


```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-STYLE : double; BORDER-WIDTH : 0.3in}
</STYLE>
</HEAD>

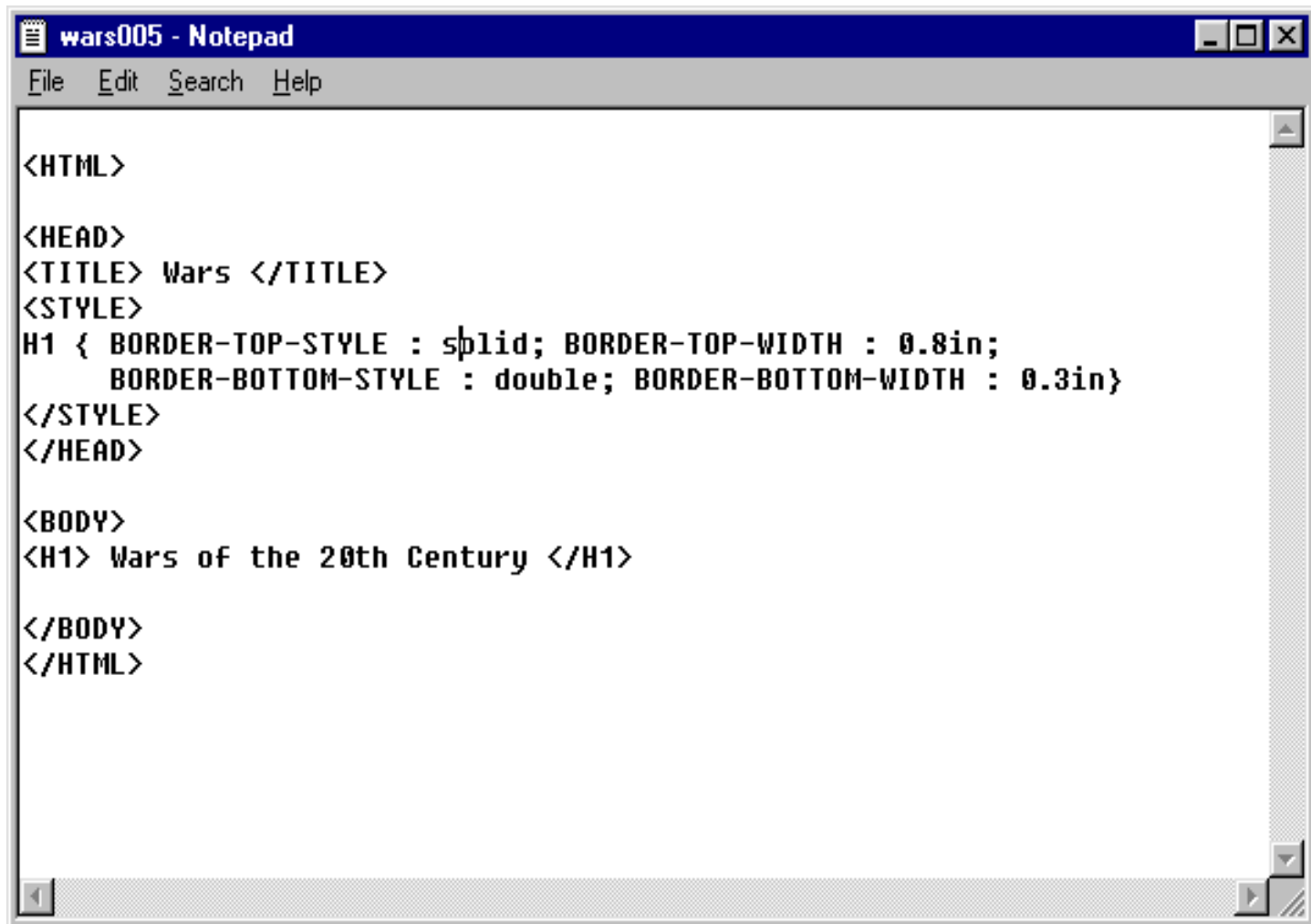
<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```

Box BORDER

- A BORDER has a TOP, BOTTOM, LEFT and RIGHT, whose WIDTH and STYLE can be set separately

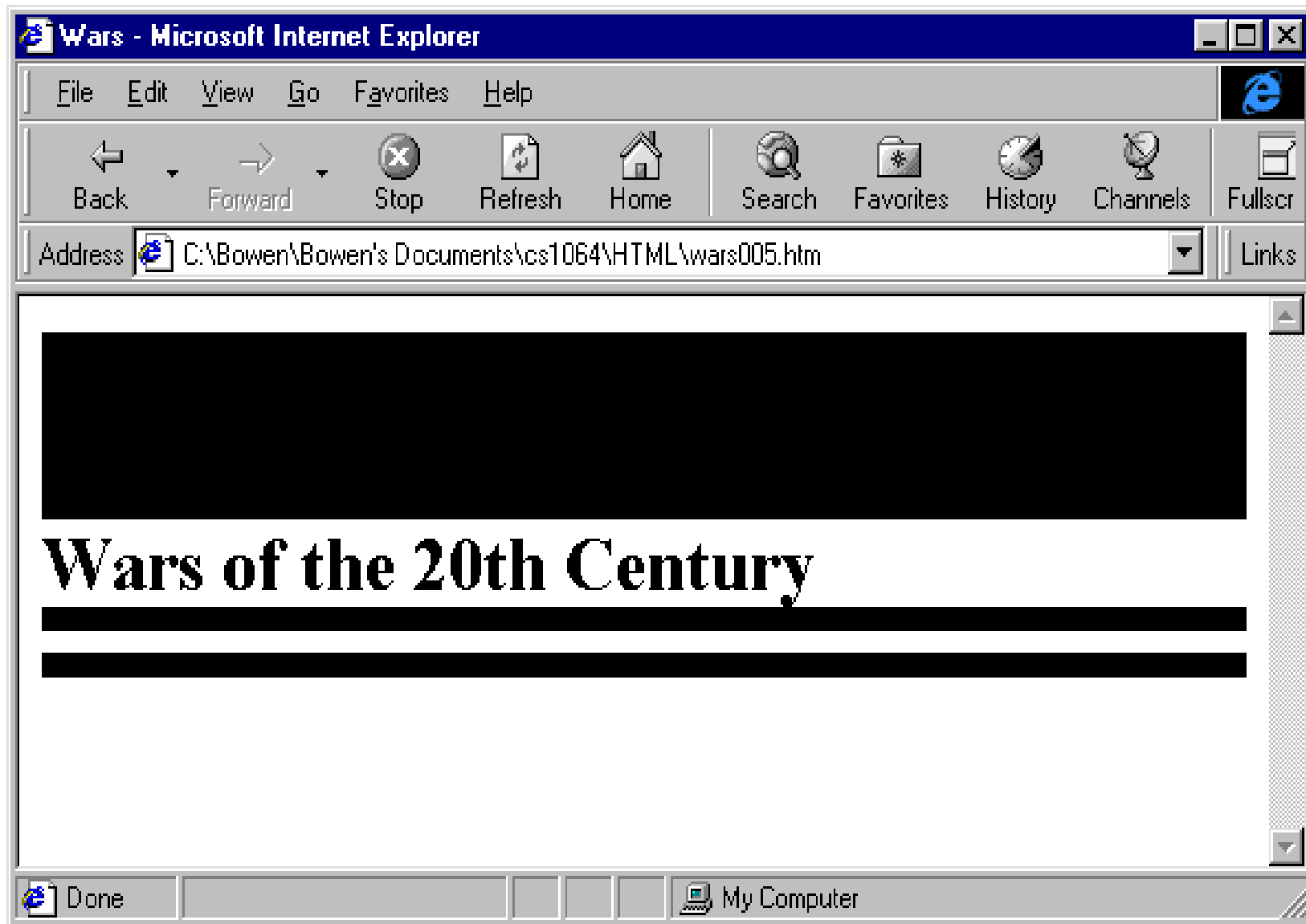


```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-TOP-STYLE : solid; BORDER-TOP-WIDTH : 0.8in;
      BORDER-BOTTOM-STYLE : double; BORDER-BOTTOM-WIDTH : 0.3in}
</STYLE>
</HEAD>

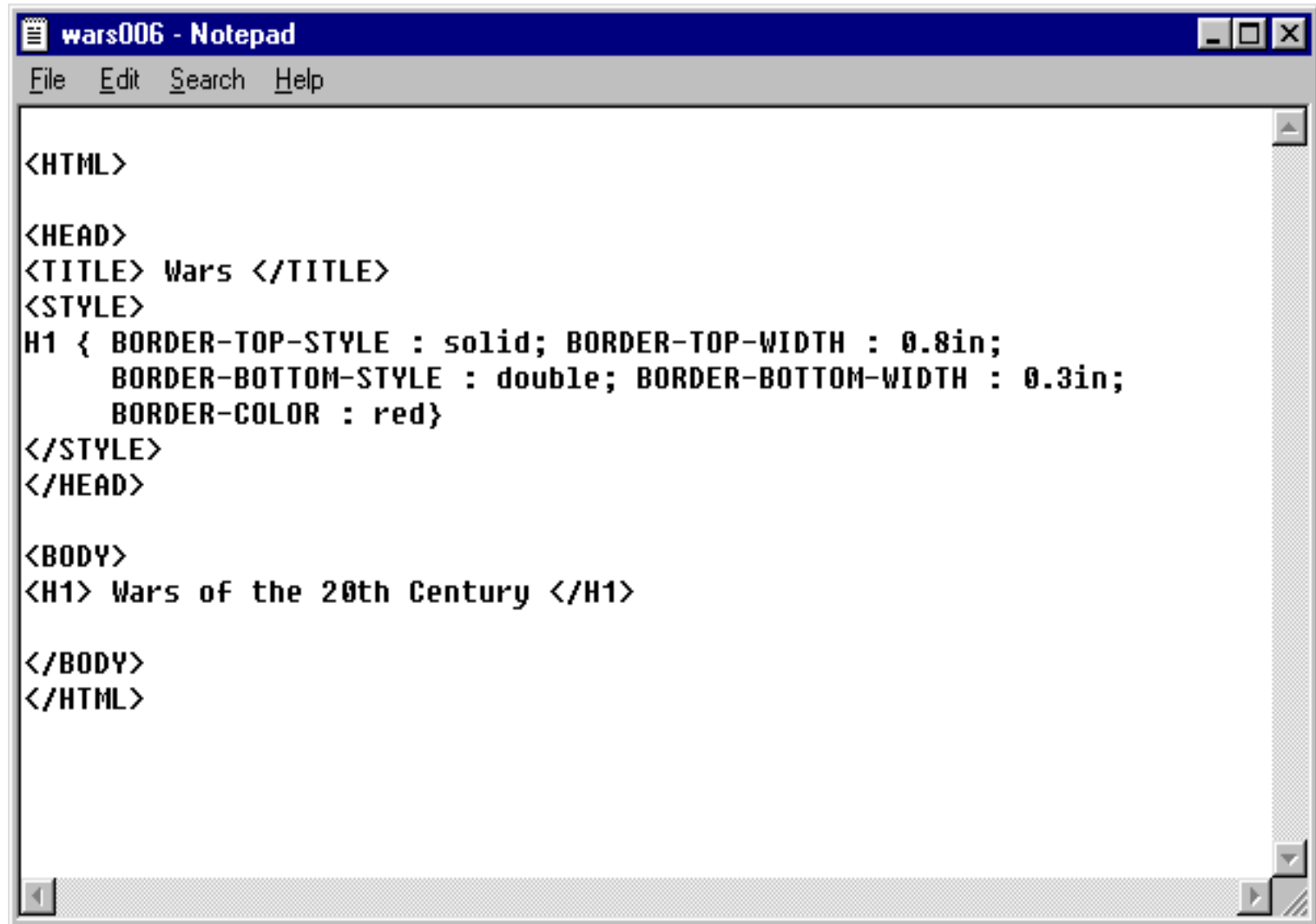
<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```



Box BORDER

- A BORDER has a COLOR but we cannot give different sides of the border different colors

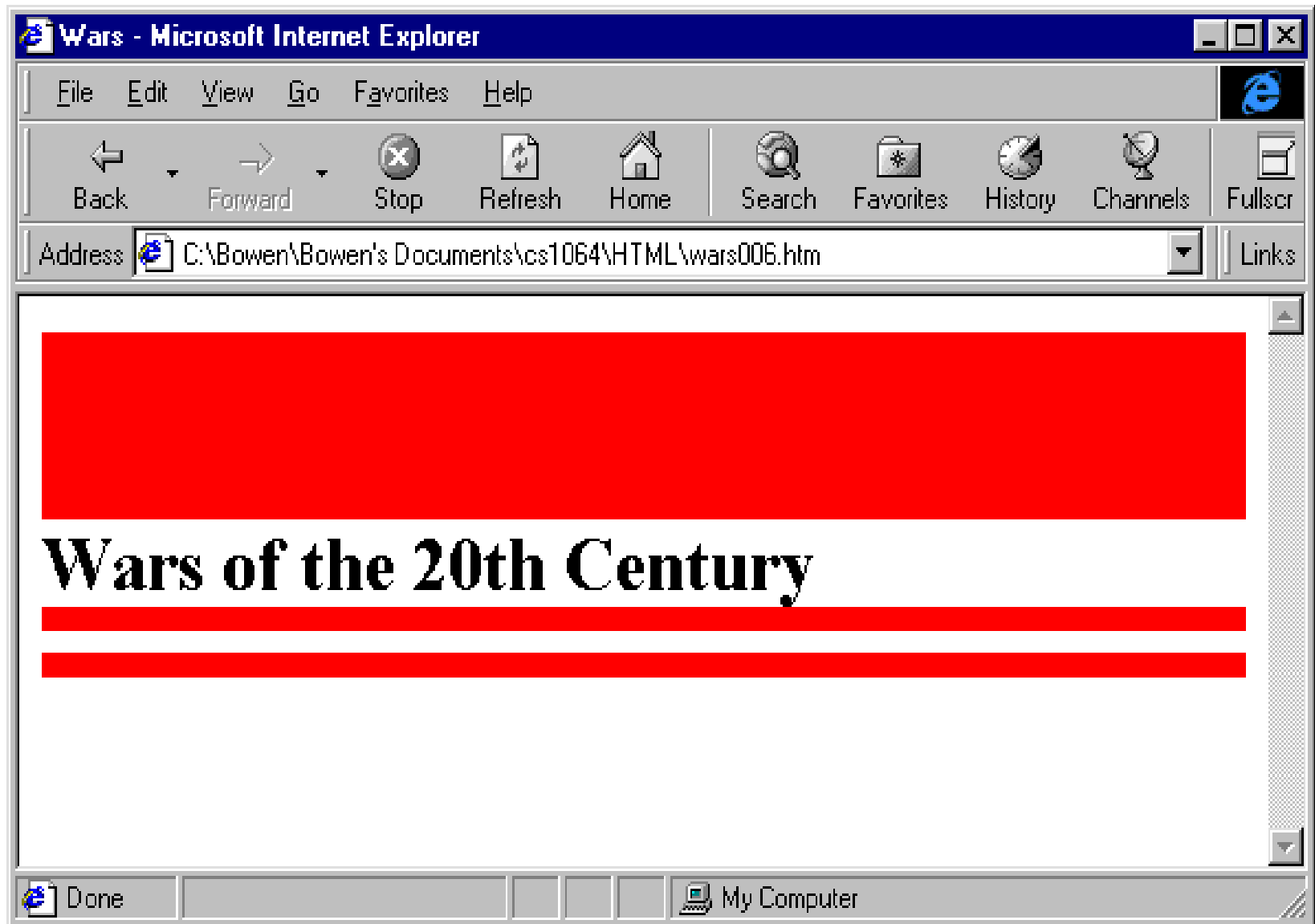
A screenshot of a Notepad window titled "wars006 - Notepad". The window has a menu bar with "File", "Edit", "Search", and "Help". The text area contains the following HTML code:

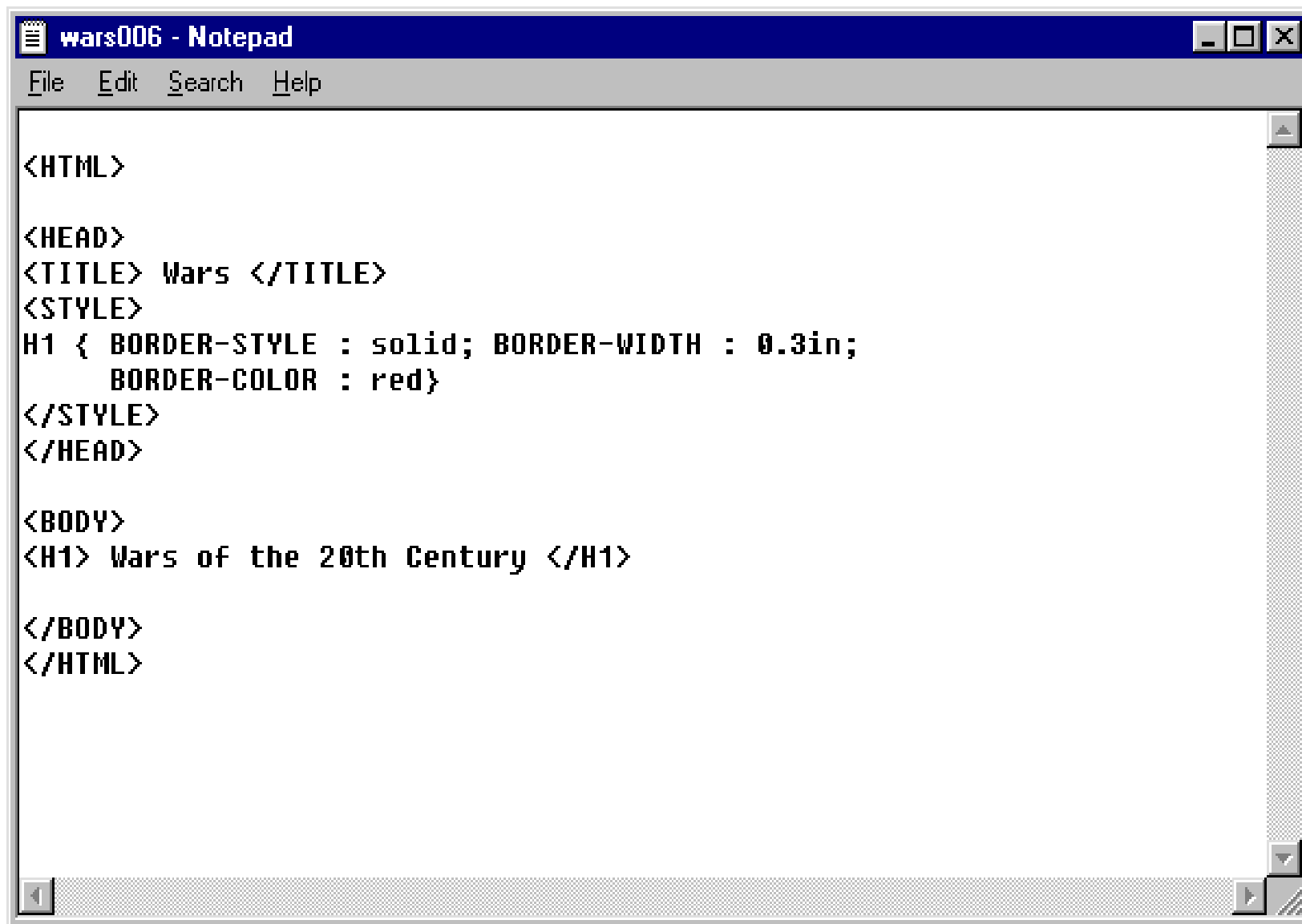
```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-TOP-STYLE : solid; BORDER-TOP-WIDTH : 0.8in;
      BORDER-BOTTOM-STYLE : double; BORDER-BOTTOM-WIDTH : 0.3in;
      BORDER-COLOR : red}
</STYLE>
</HEAD>

<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```



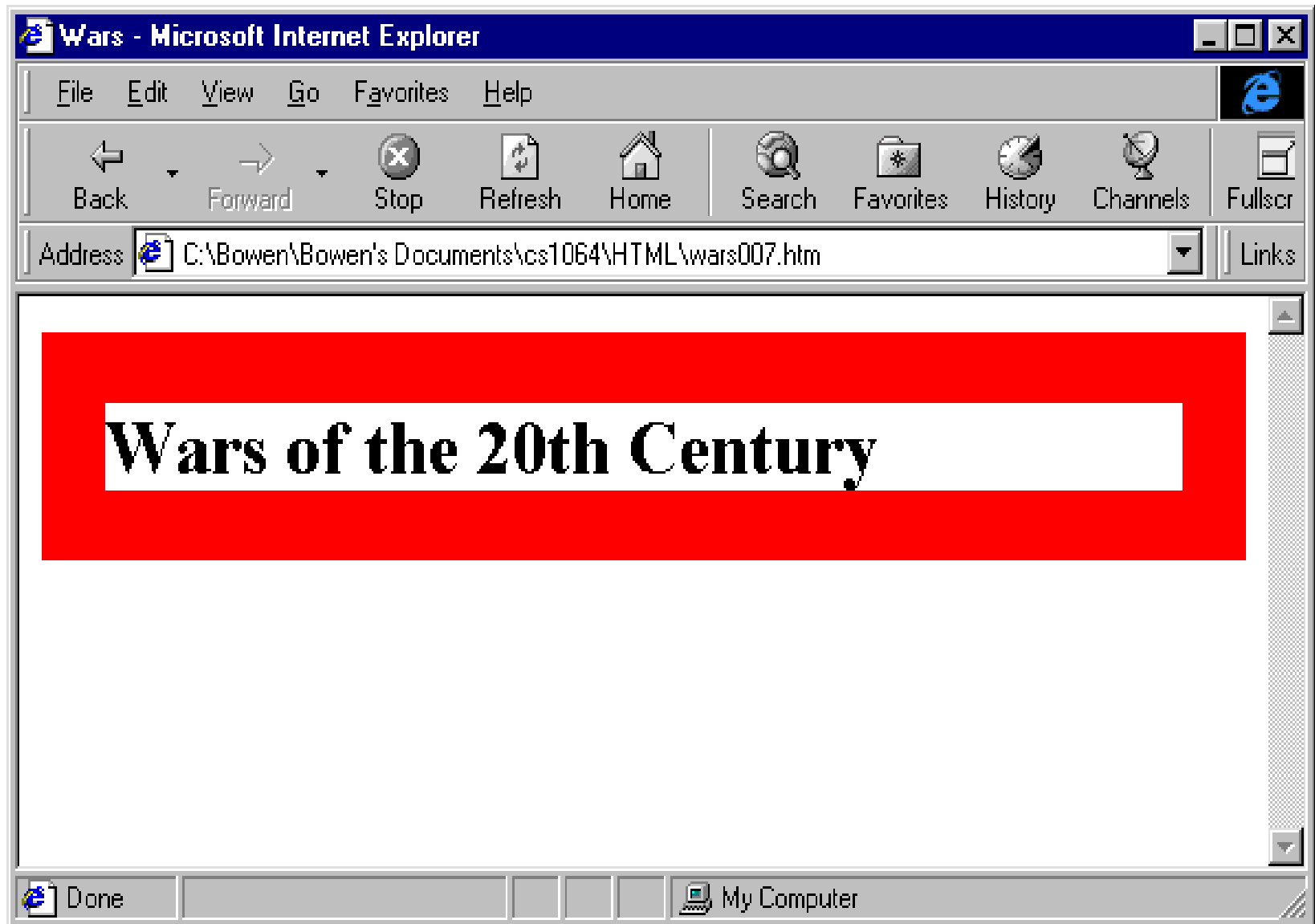


```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-STYLE : solid; BORDER-WIDTH : 0.3in;
      BORDER-COLOR : red}
</STYLE>
</HEAD>

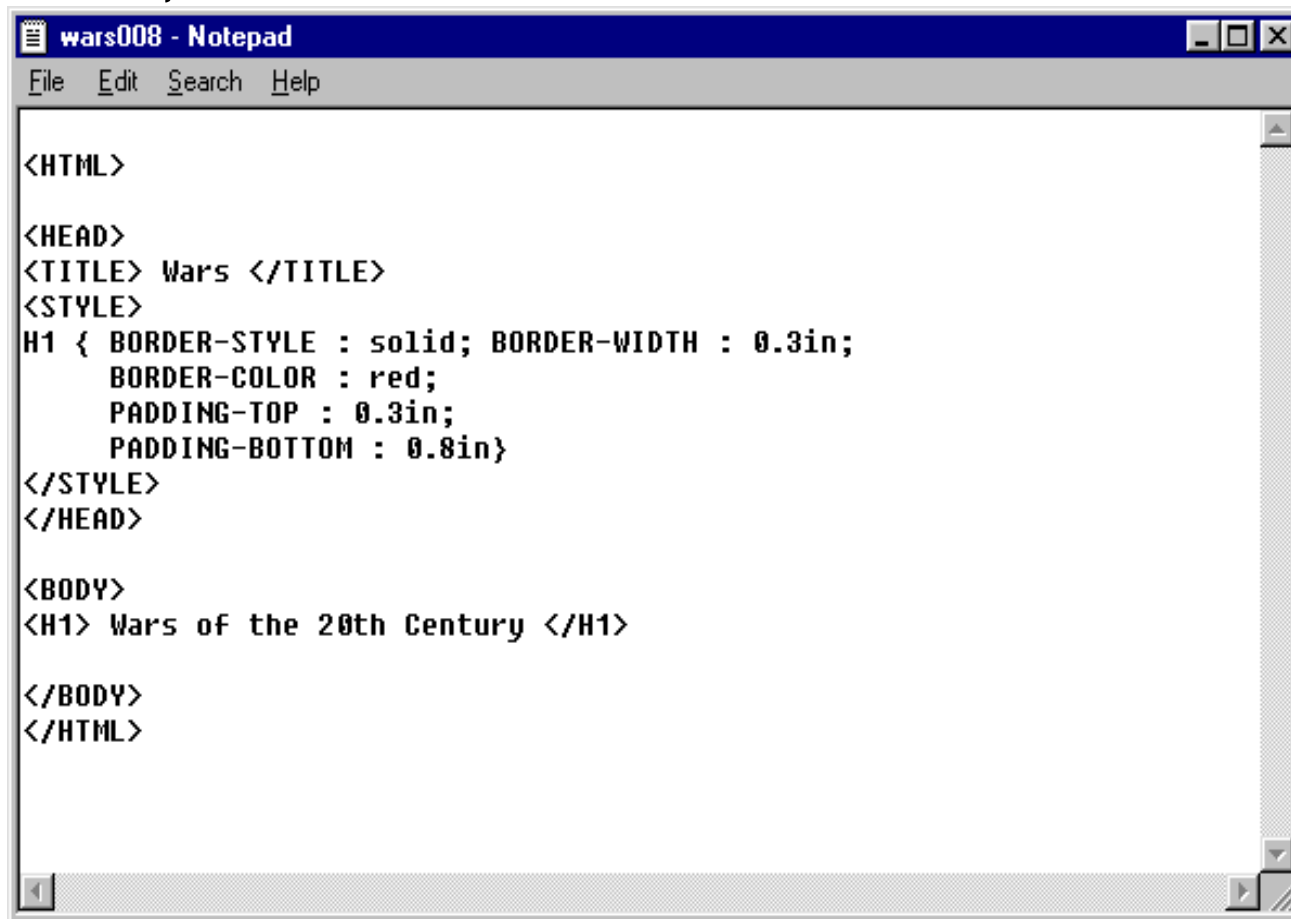
<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```



Box PADDING

- We can control the PADDING between the BORDER of a box and its content
- We can control separately the PADDING on the TOP, BOTTOM, LEFT and RIGHT of the box

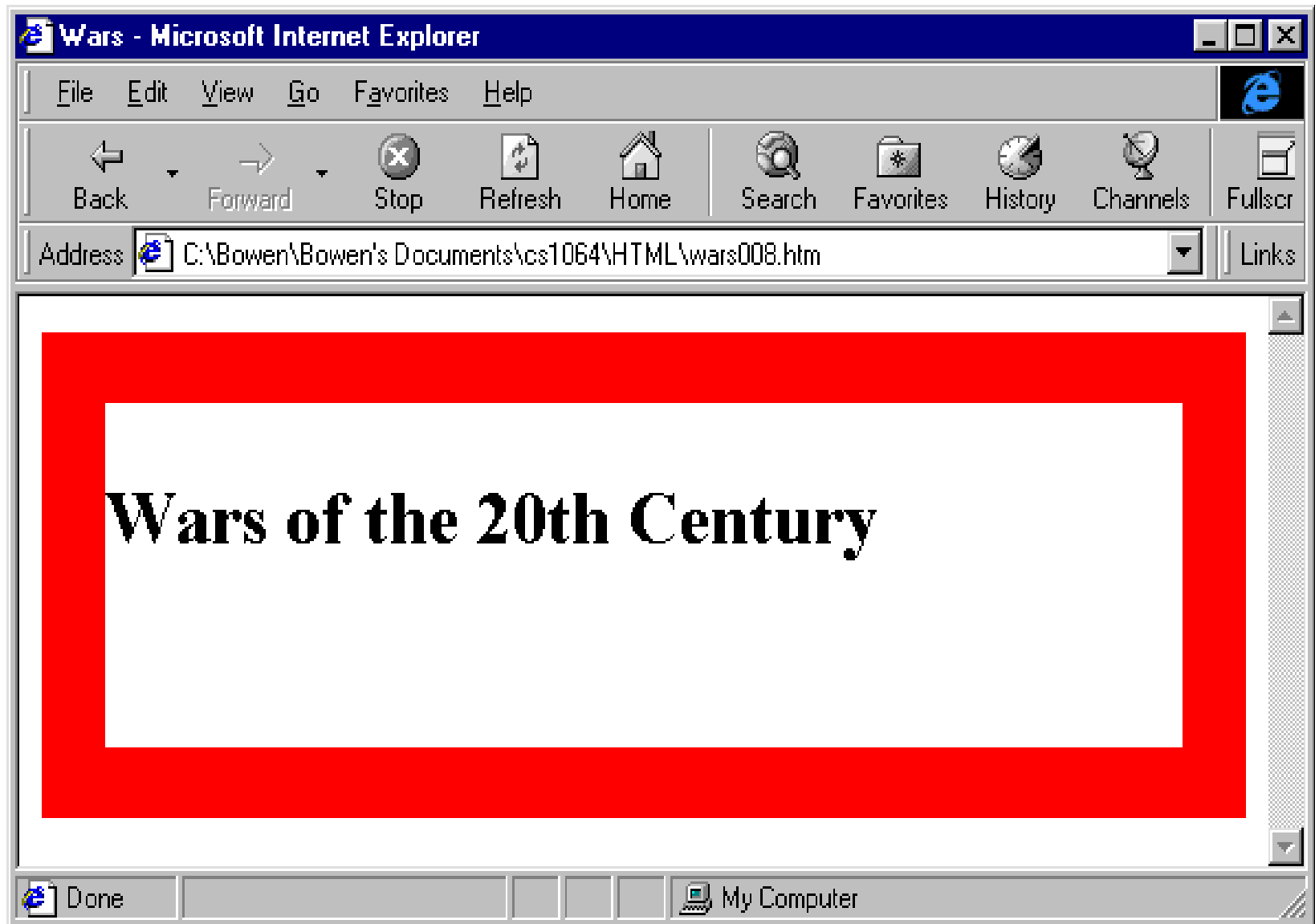


```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-STYLE : solid; BORDER-WIDTH : 0.3in;
      BORDER-COLOR : red;
      PADDING-TOP : 0.3in;
      PADDING-BOTTOM : 0.8in}
</STYLE>
</HEAD>

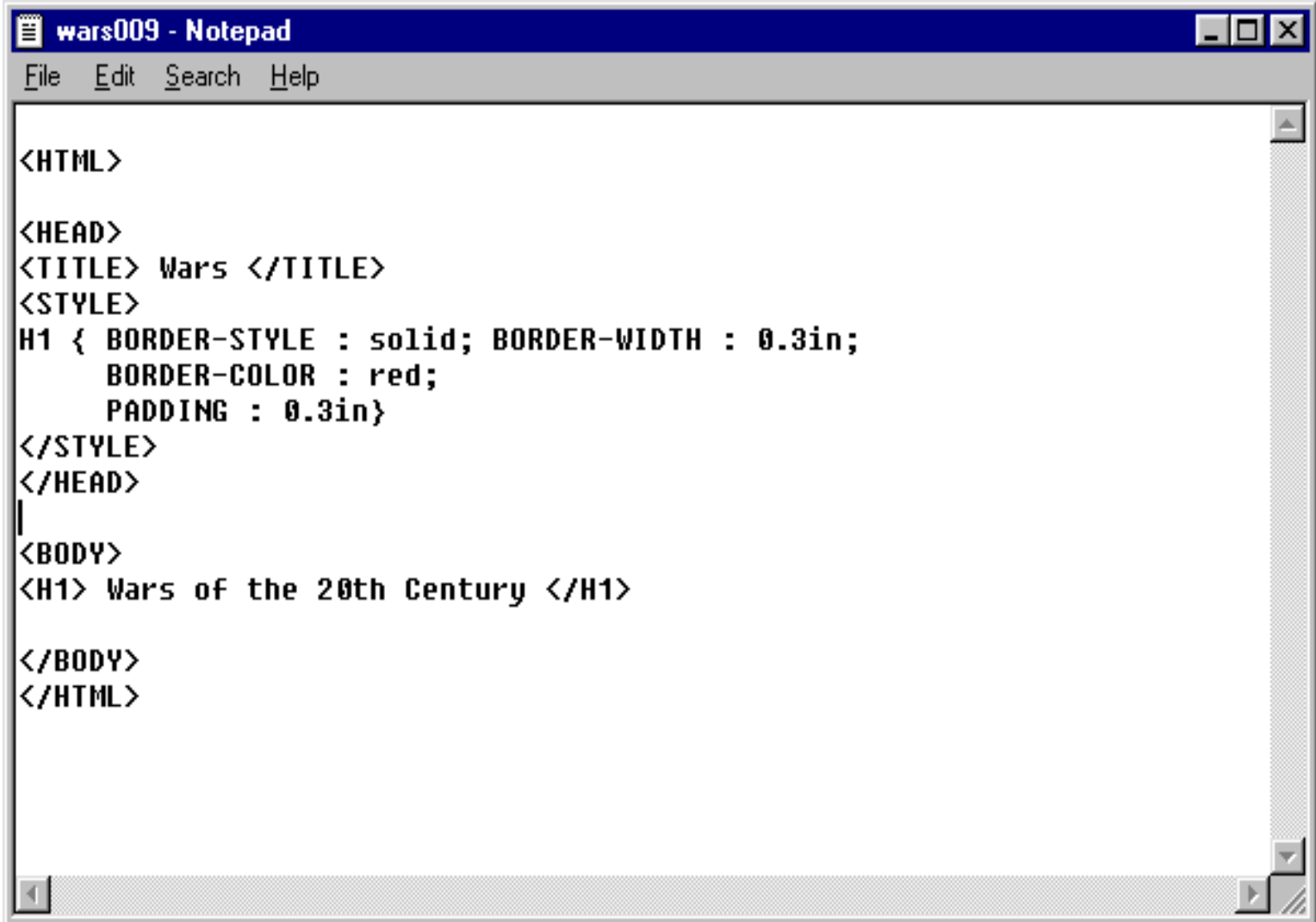
<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```

Box PADDING

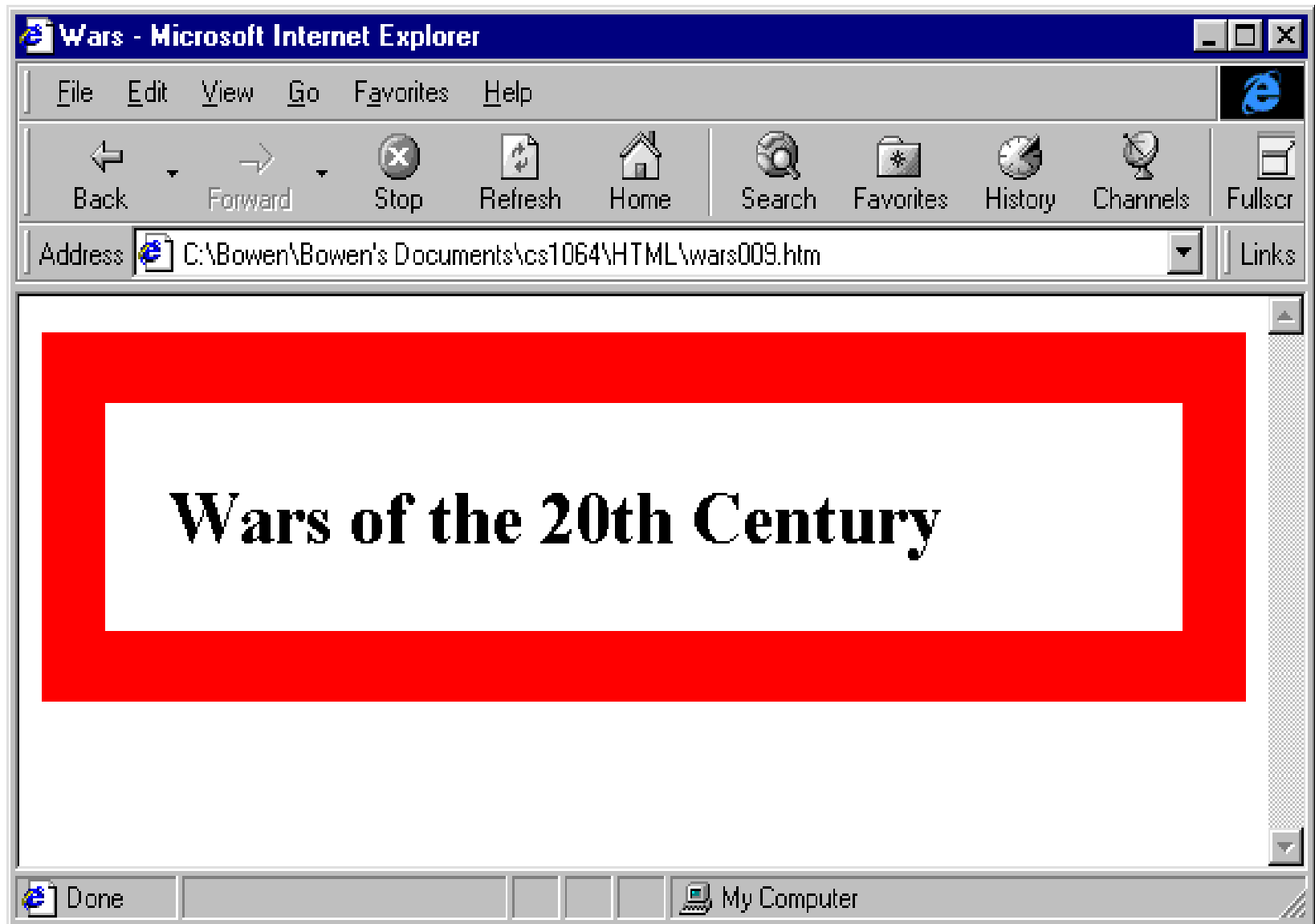
- Of course, we can just specify a PADDING that we want on all four sides

A screenshot of a Windows Notepad window titled "wars009 - Notepad". The window has a menu bar with "File", "Edit", "Search", and "Help". The text area contains the following HTML code:

```
<HTML>

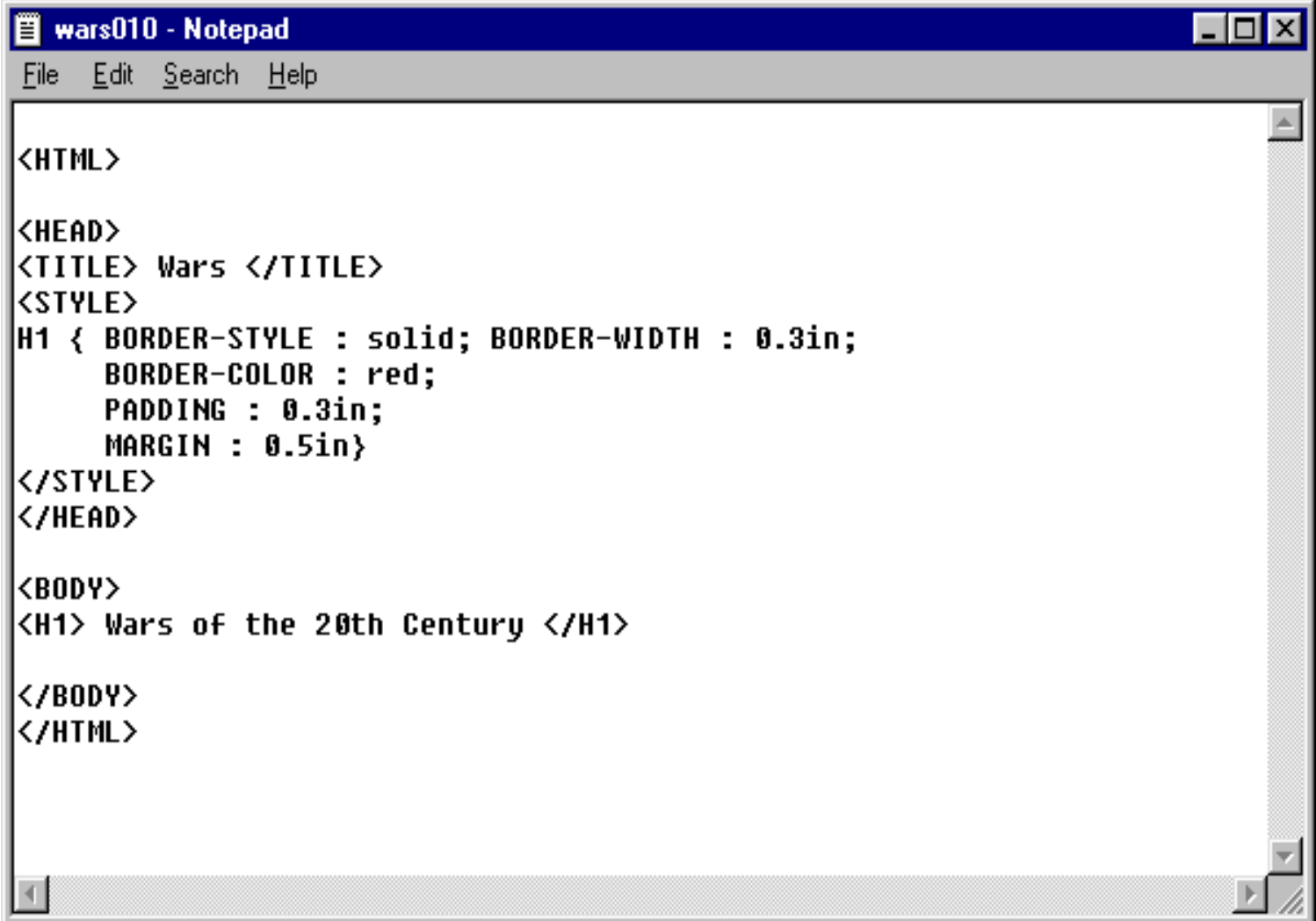
<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-STYLE : solid; BORDER-WIDTH : 0.3in;
      BORDER-COLOR : red;
      PADDING : 0.3in}
</STYLE>
</HEAD>
|
<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```



Box MARGIN

- We can control the MARGIN of clear space which exists outside a box

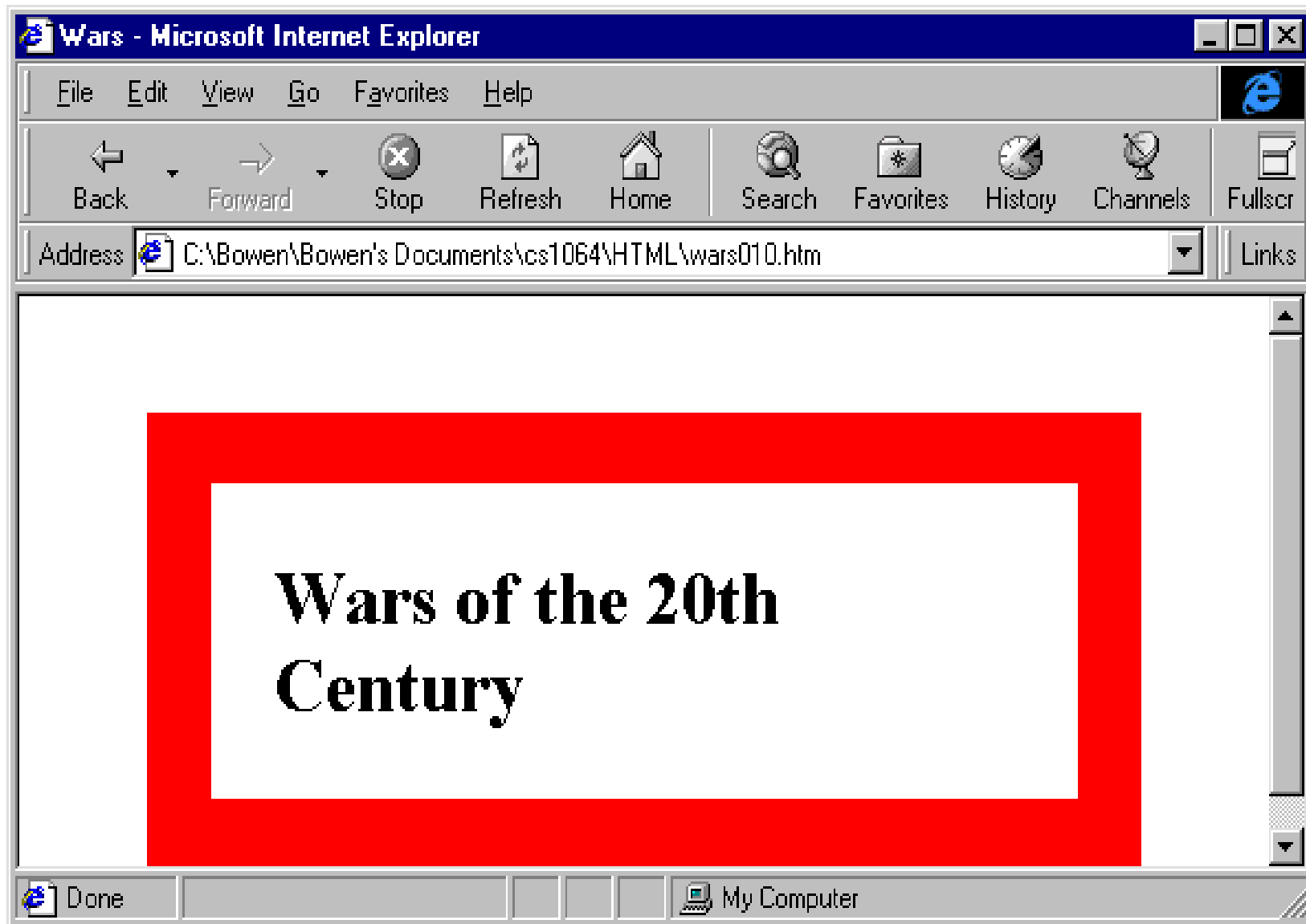


```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-STYLE : solid; BORDER-WIDTH : 0.3in;
      BORDER-COLOR : red;
      PADDING : 0.3in;
      MARGIN : 0.5in}
</STYLE>
</HEAD>

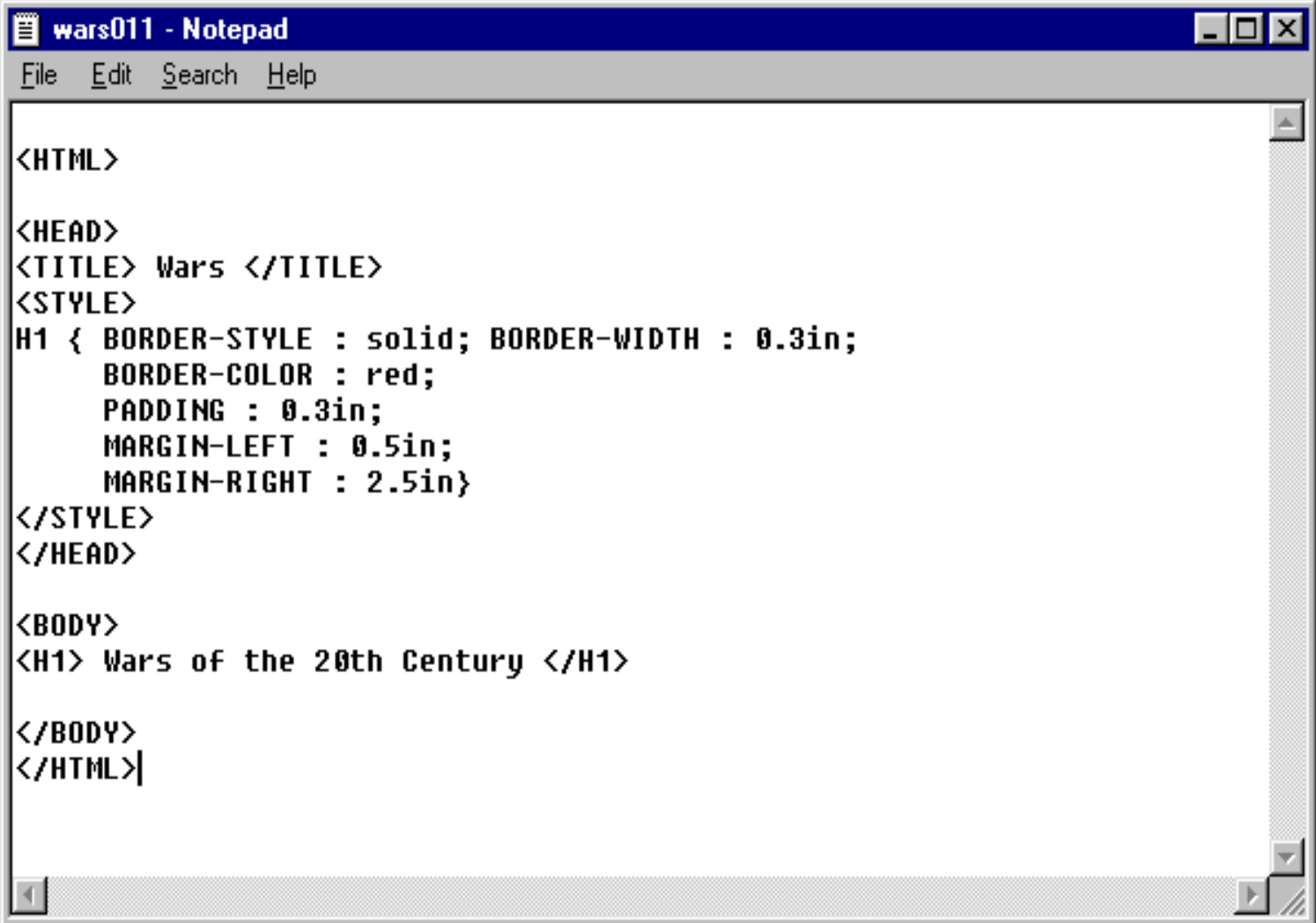
<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```



Box MARGIN

- We can control separately the MARGIN outside the TOP, BOTTOM, LEFT and RIGHT of a box

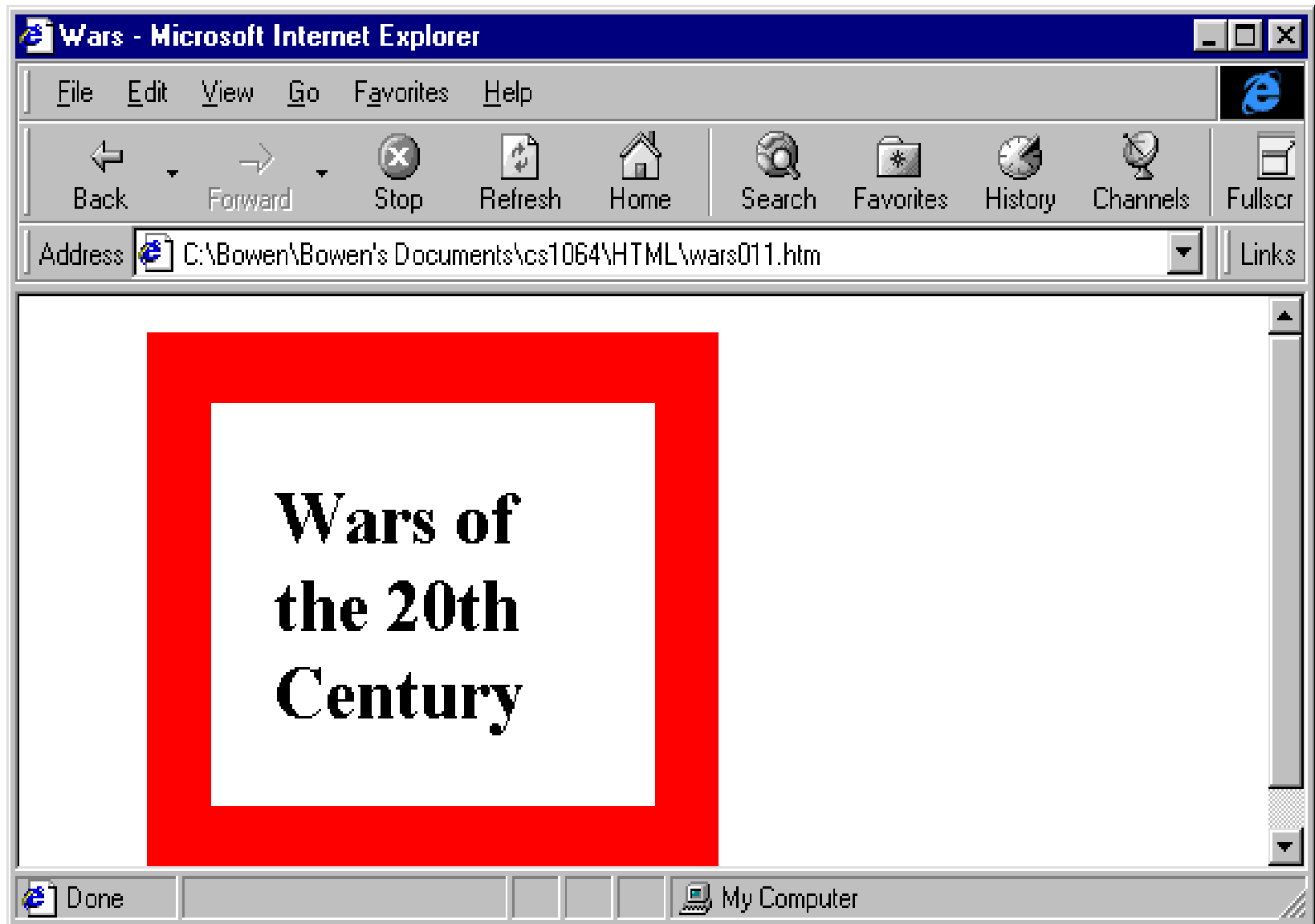


```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-STYLE : solid; BORDER-WIDTH : 0.3in;
      BORDER-COLOR : red;
      PADDING : 0.3in;
      MARGIN-LEFT : 0.5in;
      MARGIN-RIGHT : 2.5in}
</STYLE>
</HEAD>

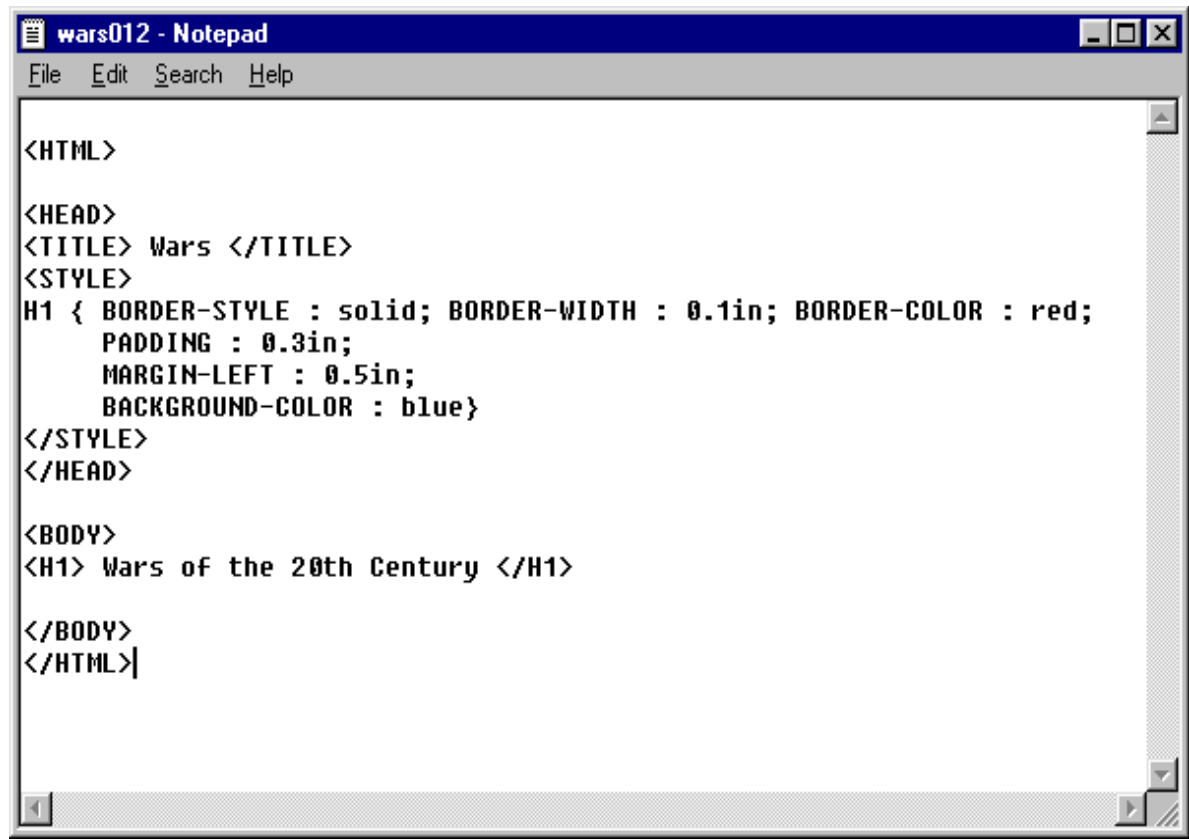
<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```



Box BACKGROUND

- A box has a foreground (the content of the box) and a BACKGROUND
- We can control the appearance of the BACKGROUND, by specifying its COLOR or by controlling the use of an IMAGE as the basis of the background

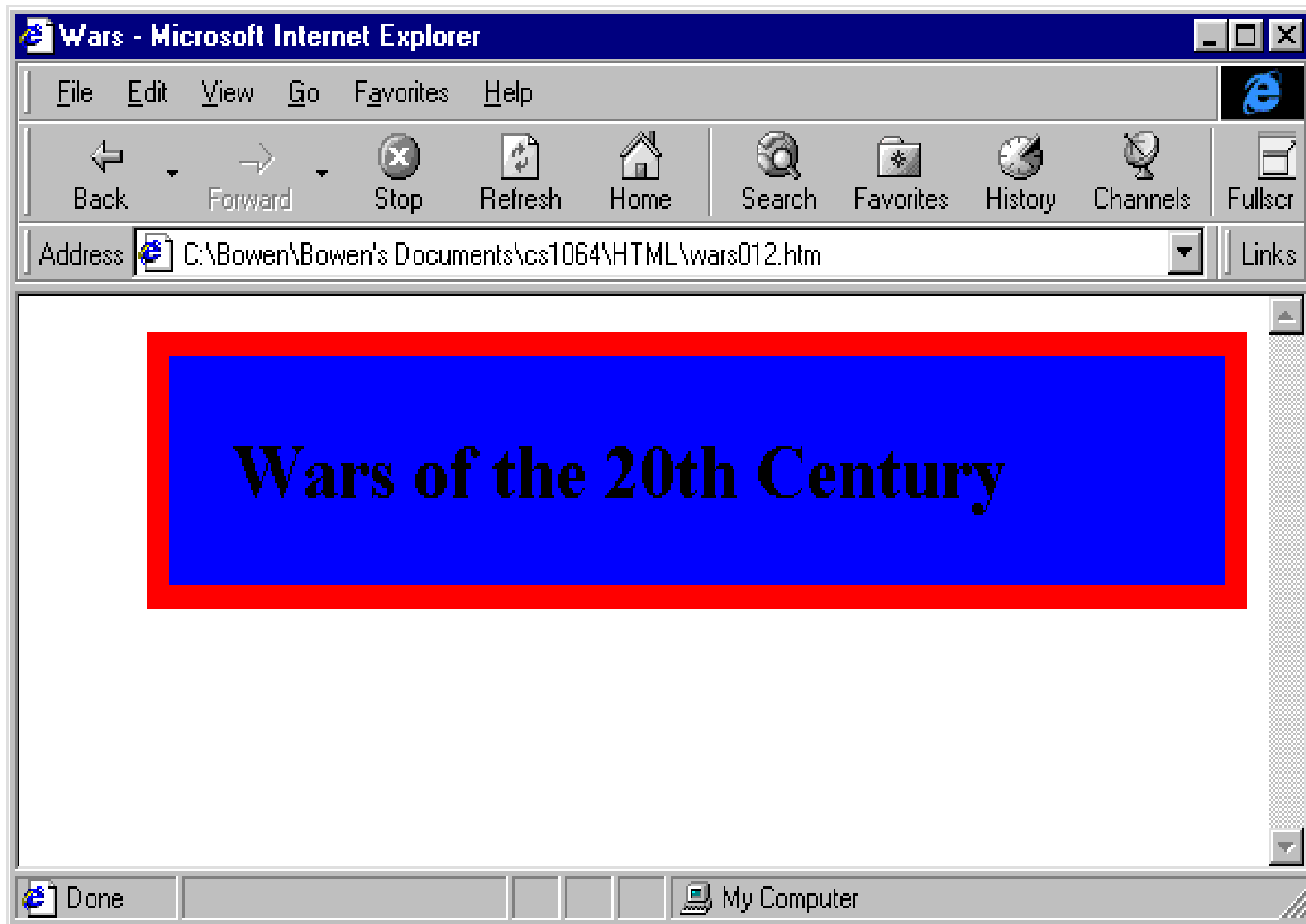


```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-STYLE : solid; BORDER-WIDTH : 0.1in; BORDER-COLOR : red;
      PADDING : 0.3in;
      MARGIN-LEFT : 0.5in;
      BACKGROUND-COLOR : blue}
</STYLE>
</HEAD>

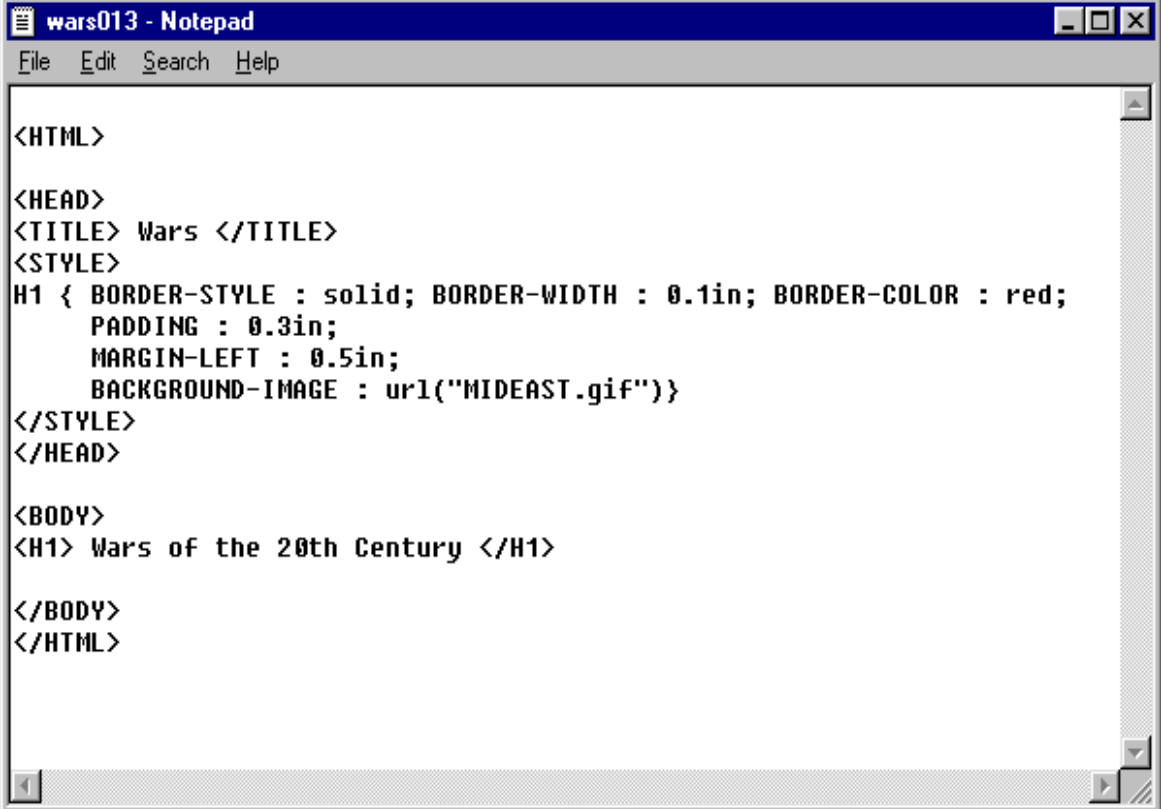
<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```

BACKGROUND IMAGE

- We specify the image which is to be used as the basis of the BACKGROUND by giving its **url**, which, of course, can be a relative or an absolute url
- The specified image is used as a tile to cover the BACKGROUND

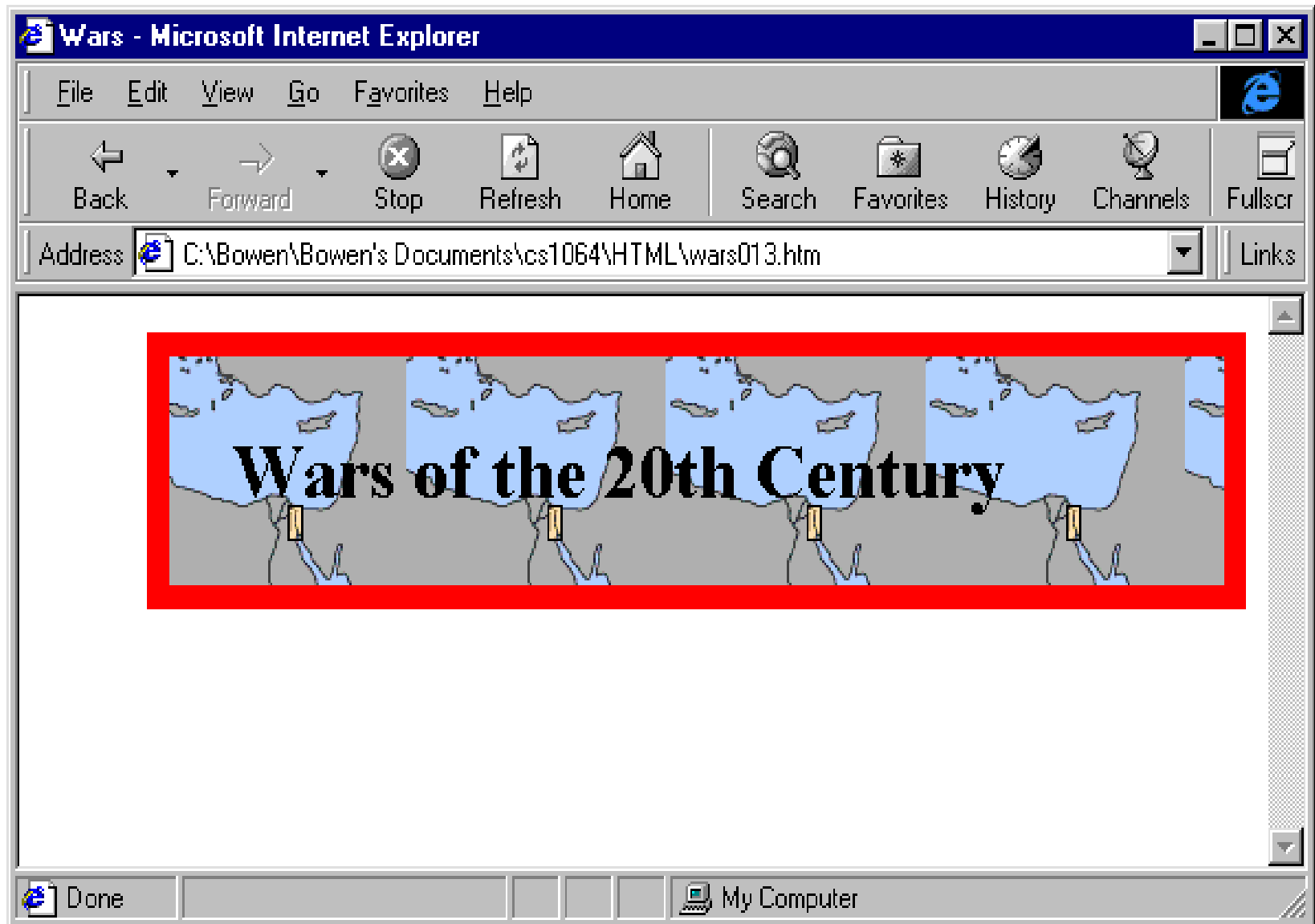


```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-STYLE : solid; BORDER-WIDTH : 0.1in; BORDER-COLOR : red;
      PADDING : 0.3in;
      MARGIN-LEFT : 0.5in;
      BACKGROUND-IMAGE : url("MIDEAST.gif")}
</STYLE>
</HEAD>

<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```

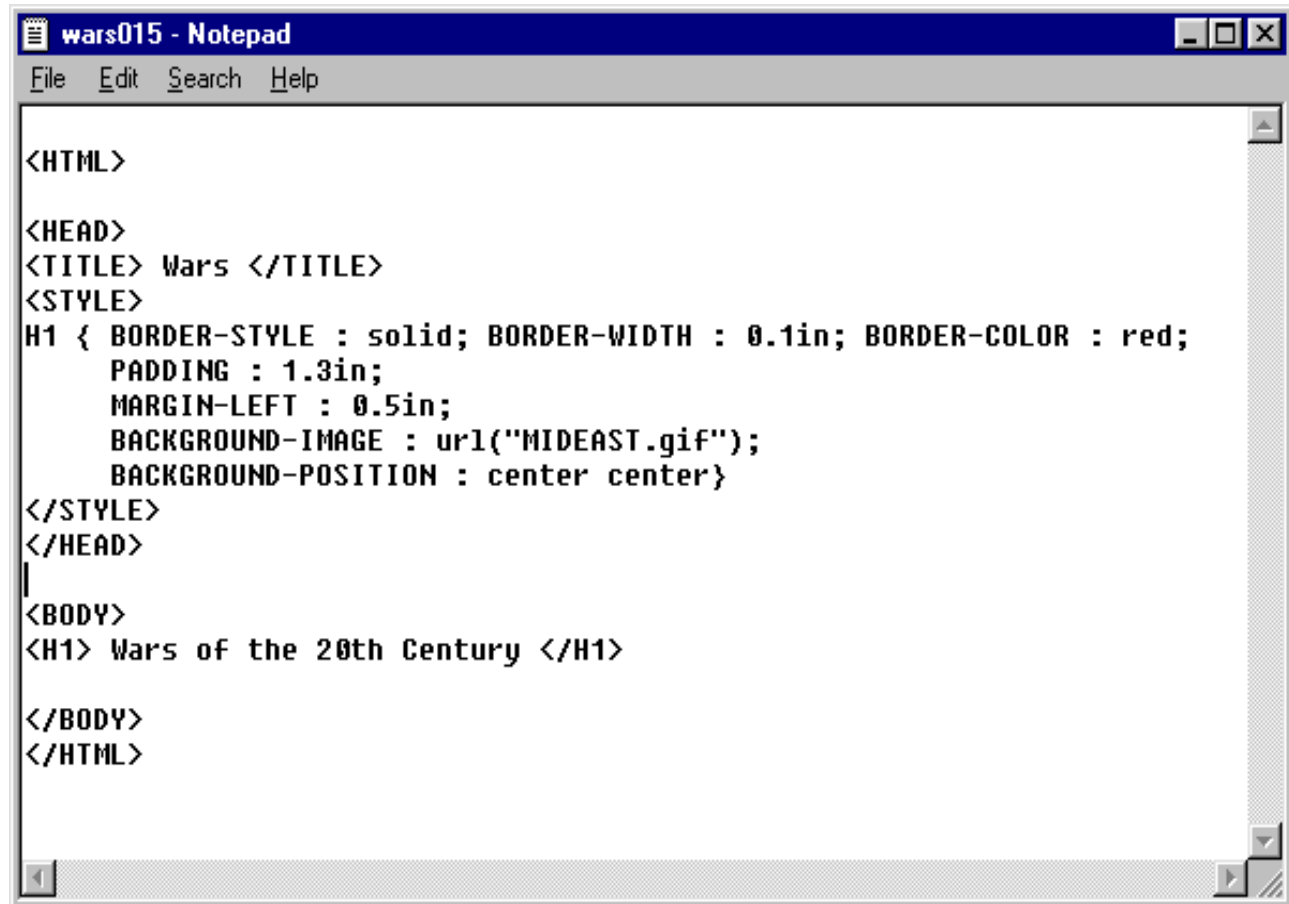


BACKGROUND IMAGE

- We can specify the POSITION, relative to the top left corner of the box, where we want tiling to start
- We can specify the POSITION
 - symbolically
 - using percentages
 - using absolute coordinates

BACKGROUND IMAGE

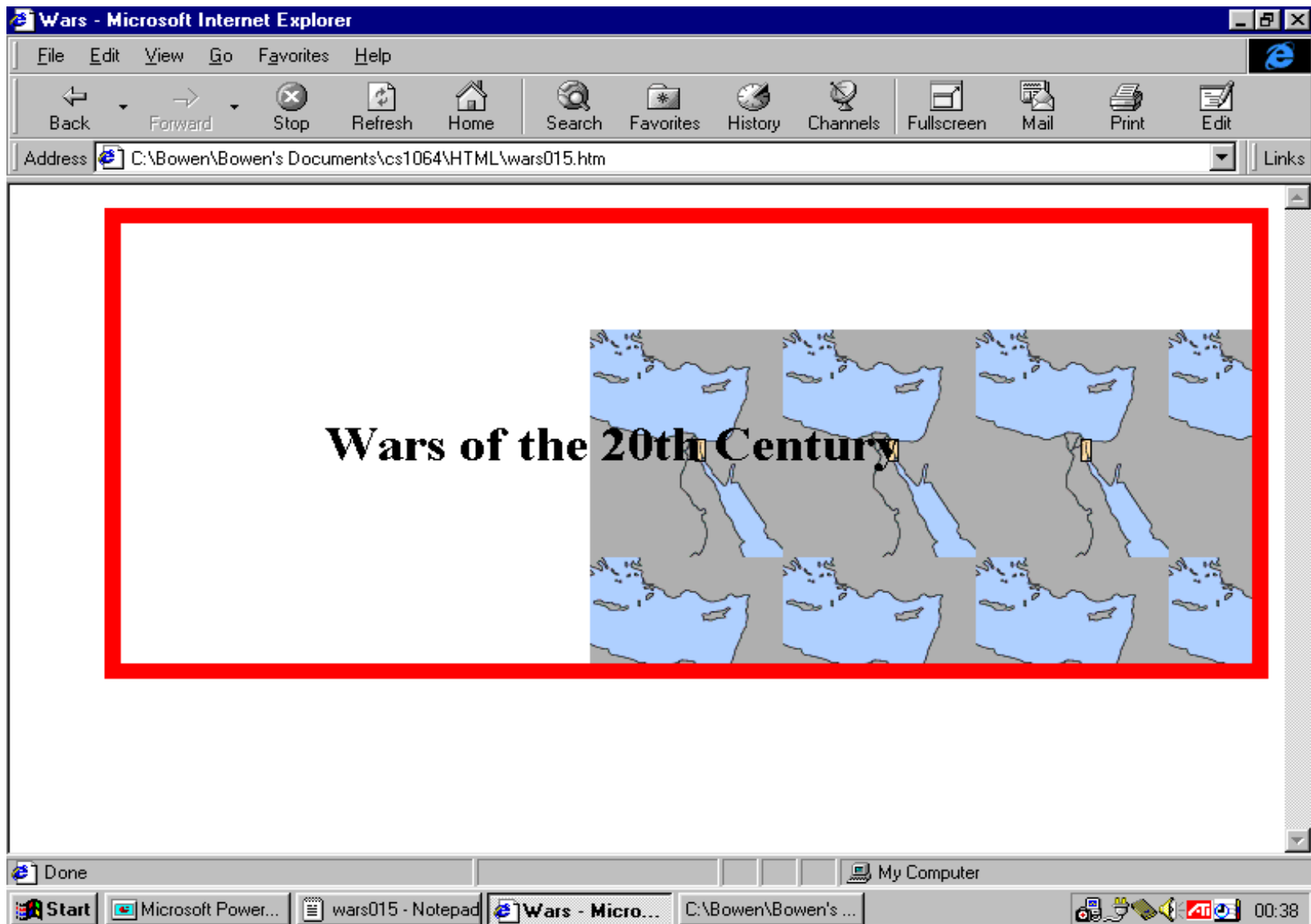
- Specifying a POSITION symbolically:
 - use **left**, **center** or **right** for horizontal positioning
 - use **top**, **center** or **bottom** for vertical positioning

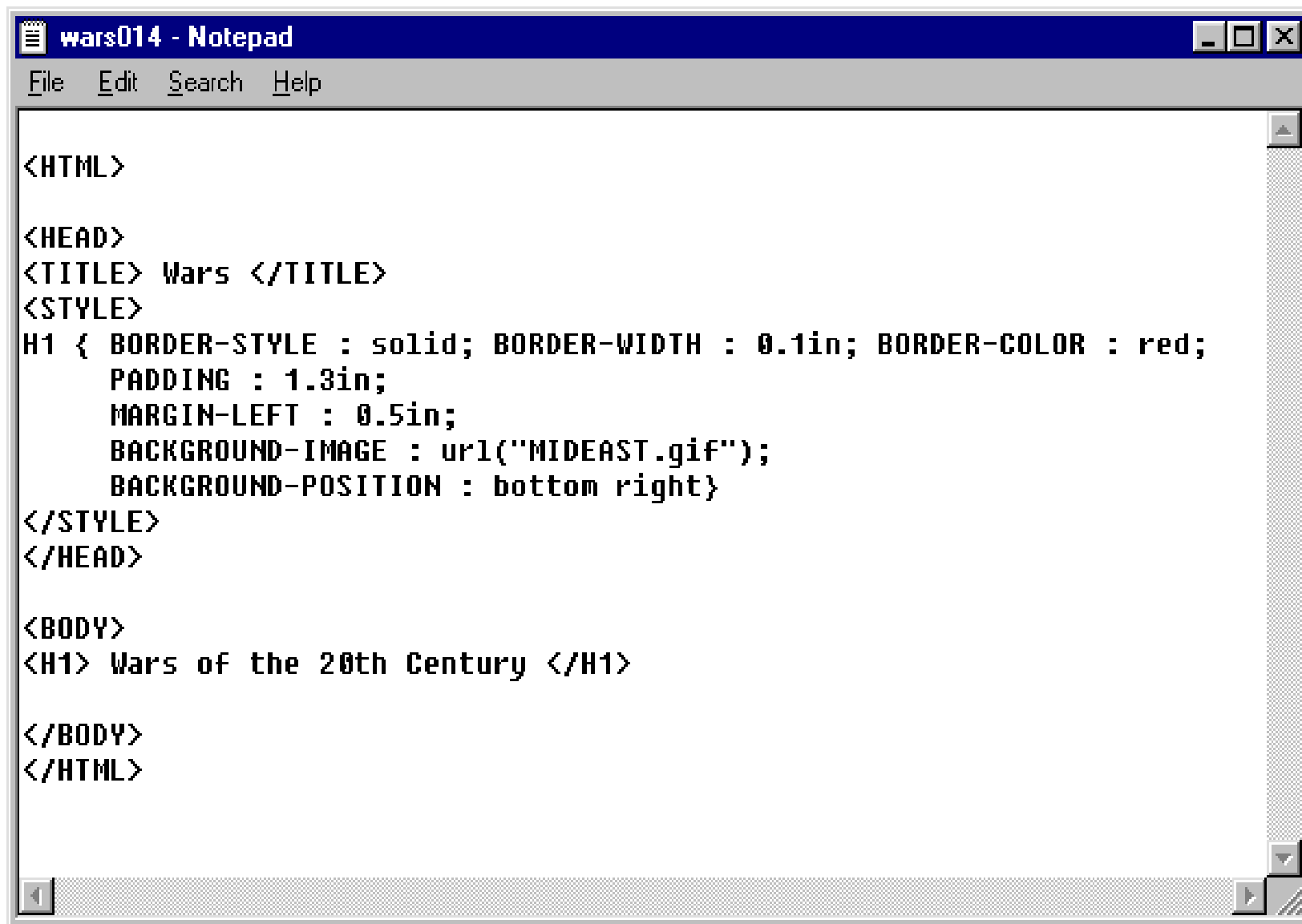


```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-STYLE : solid; BORDER-WIDTH : 0.1in; BORDER-COLOR : red;
      PADDING : 1.3in;
      MARGIN-LEFT : 0.5in;
      BACKGROUND-IMAGE : url("MIDEAST.gif");
      BACKGROUND-POSITION : center center}
</STYLE>
</HEAD>
|
<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```



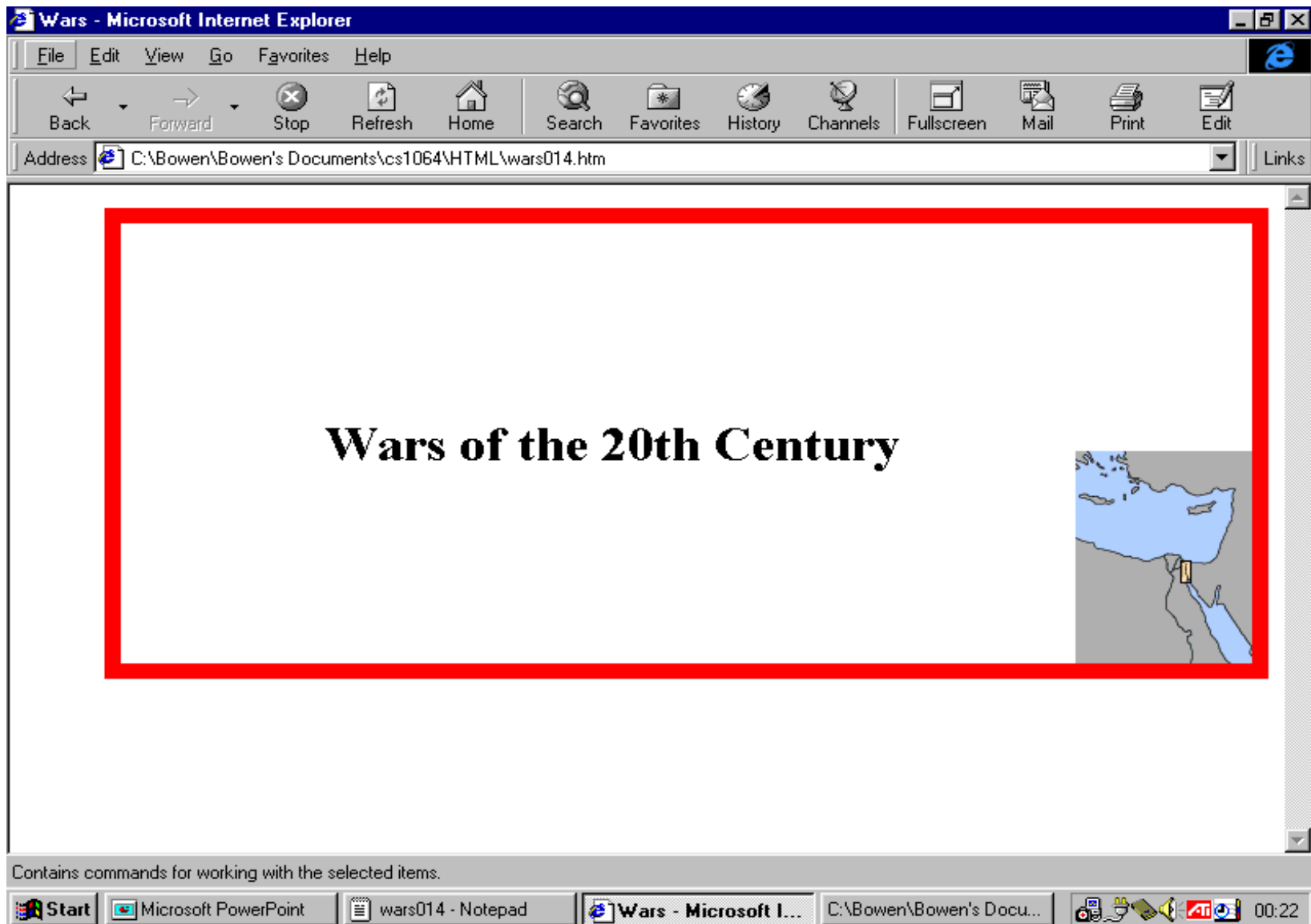


```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-STYLE : solid; BORDER-WIDTH : 0.1in; BORDER-COLOR : red;
      PADDING : 1.3in;
      MARGIN-LEFT : 0.5in;
      BACKGROUND-IMAGE : url("MIDEAST.gif");
      BACKGROUND-POSITION : bottom right}
</STYLE>
</HEAD>

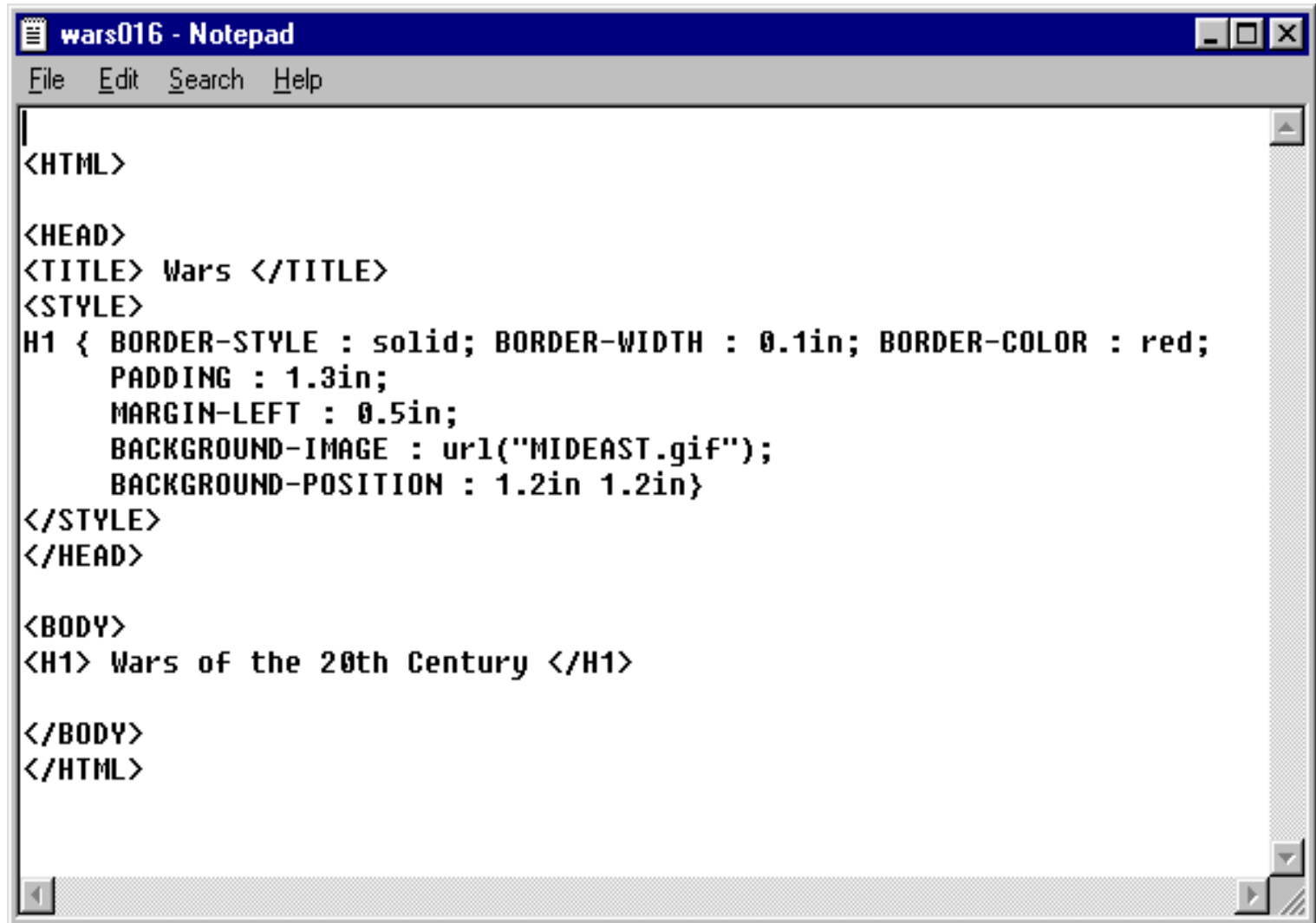
<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```



BACKGROUND IMAGE

- We specify a POSITION absolutely by using coordinates, specified in our favourite units, from the top-left corner of the box

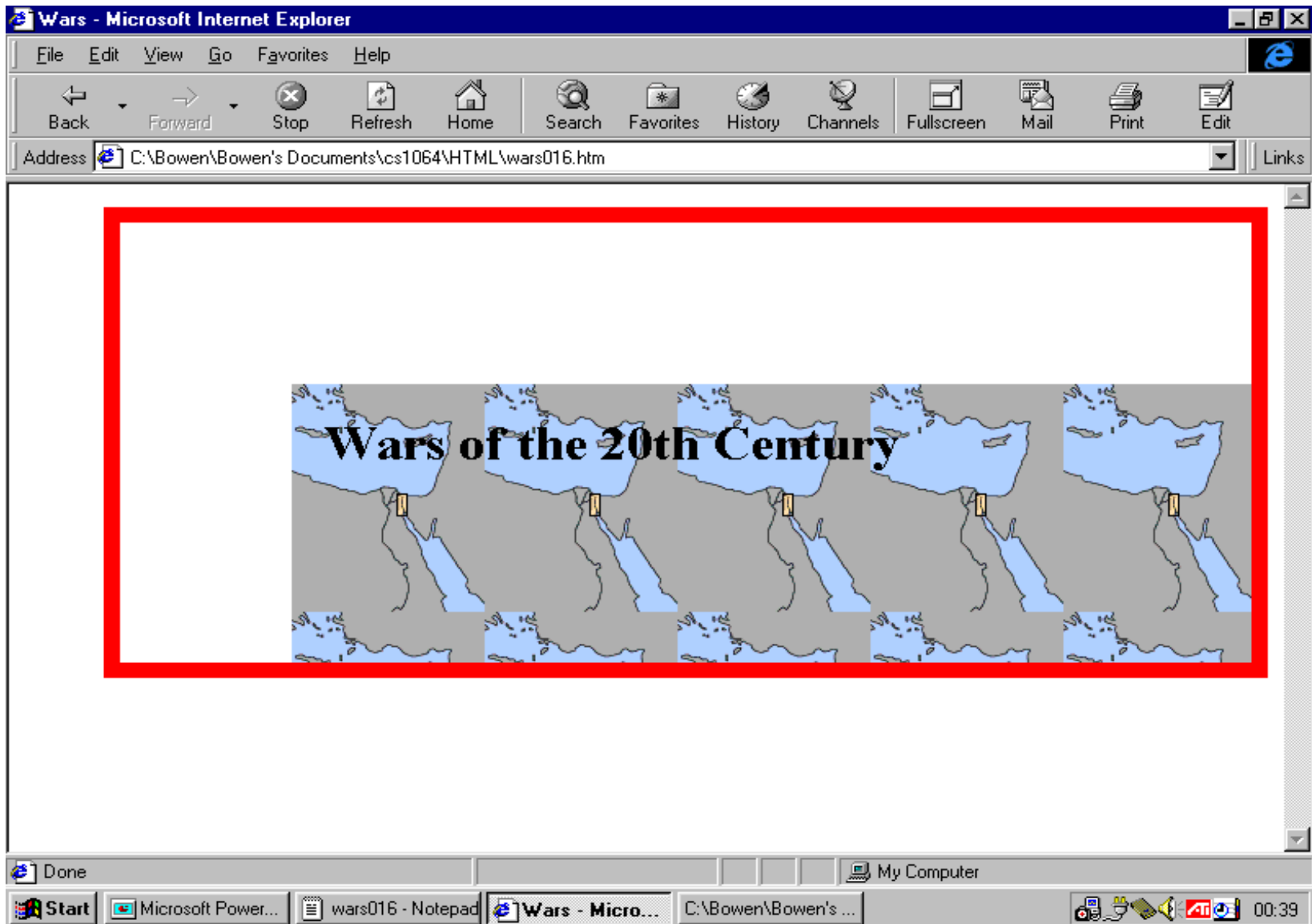


```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-STYLE : solid; BORDER-WIDTH : 0.1in; BORDER-COLOR : red;
      PADDING : 1.3in;
      MARGIN-LEFT : 0.5in;
      BACKGROUND-IMAGE : url("MIDEAST.gif");
      BACKGROUND-POSITION : 1.2in 1.2in}
</STYLE>
</HEAD>

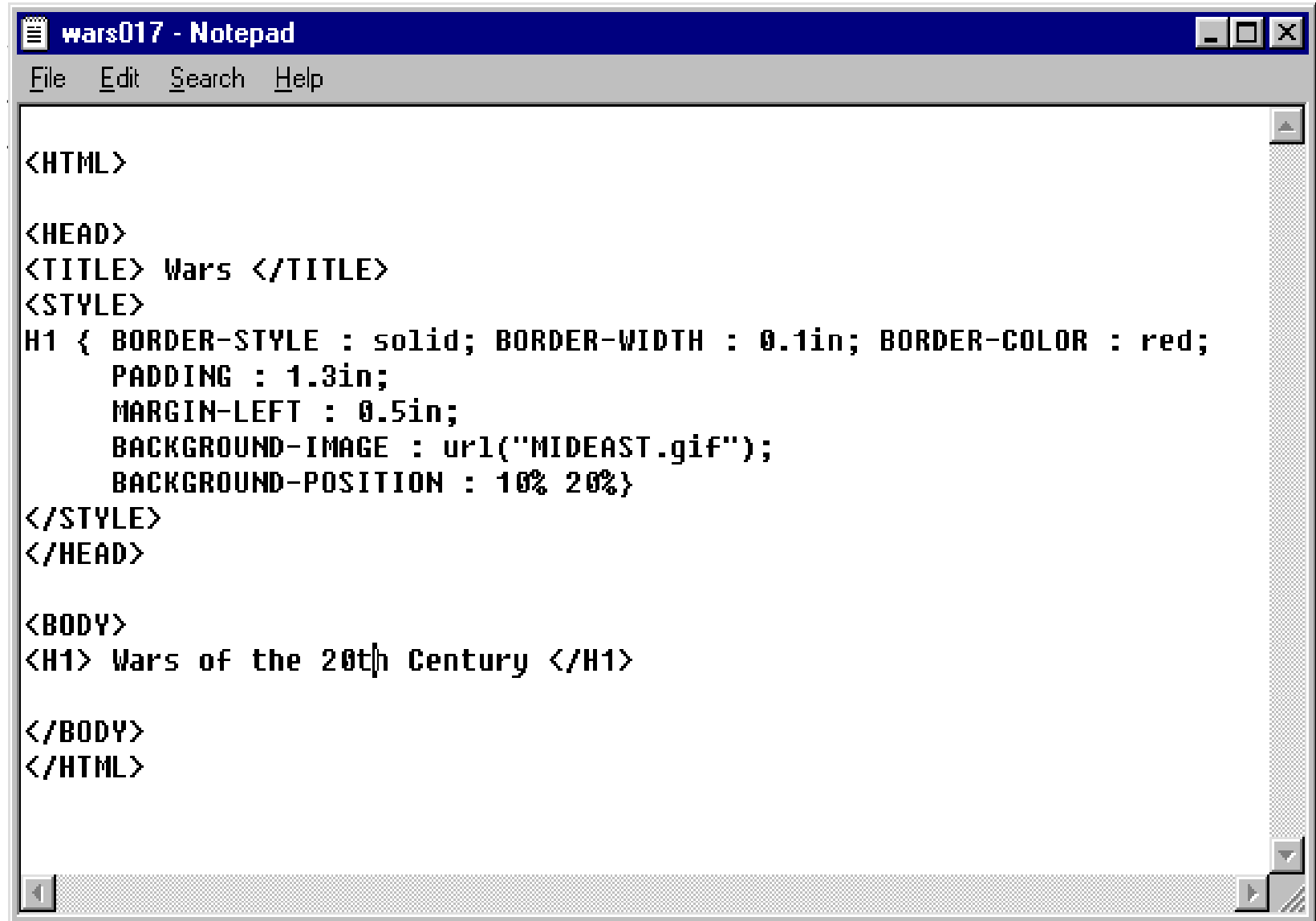
<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```



BACKGROUND IMAGE

-



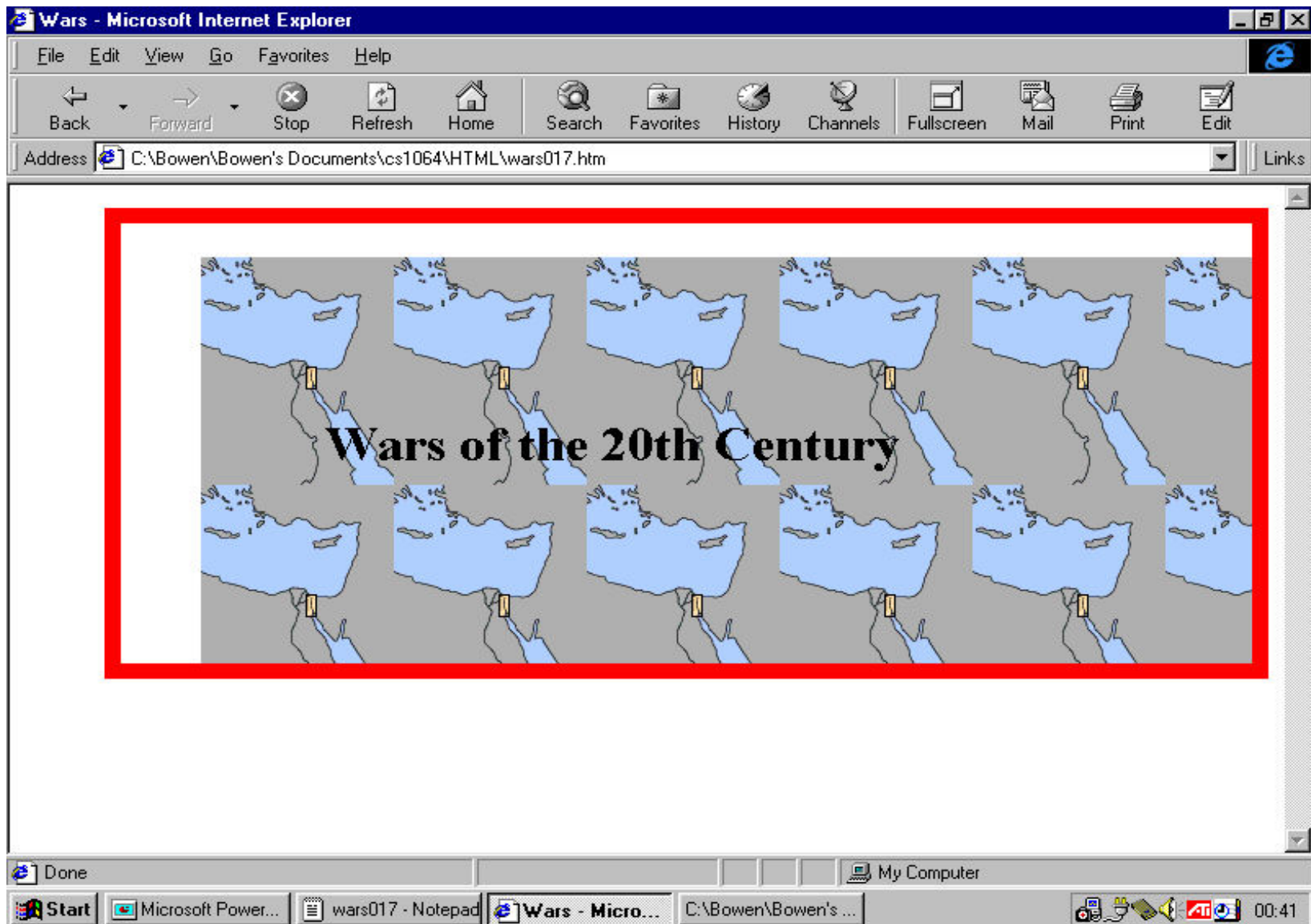
```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-STYLE : solid; BORDER-WIDTH : 0.1in; BORDER-COLOR : red;
      PADDING : 1.3in;
      MARGIN-LEFT : 0.5in;
      BACKGROUND-IMAGE : url("MIDEAST.gif");
      BACKGROUND-POSITION : 10% 20%;
    }
</STYLE>
</HEAD>

<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```

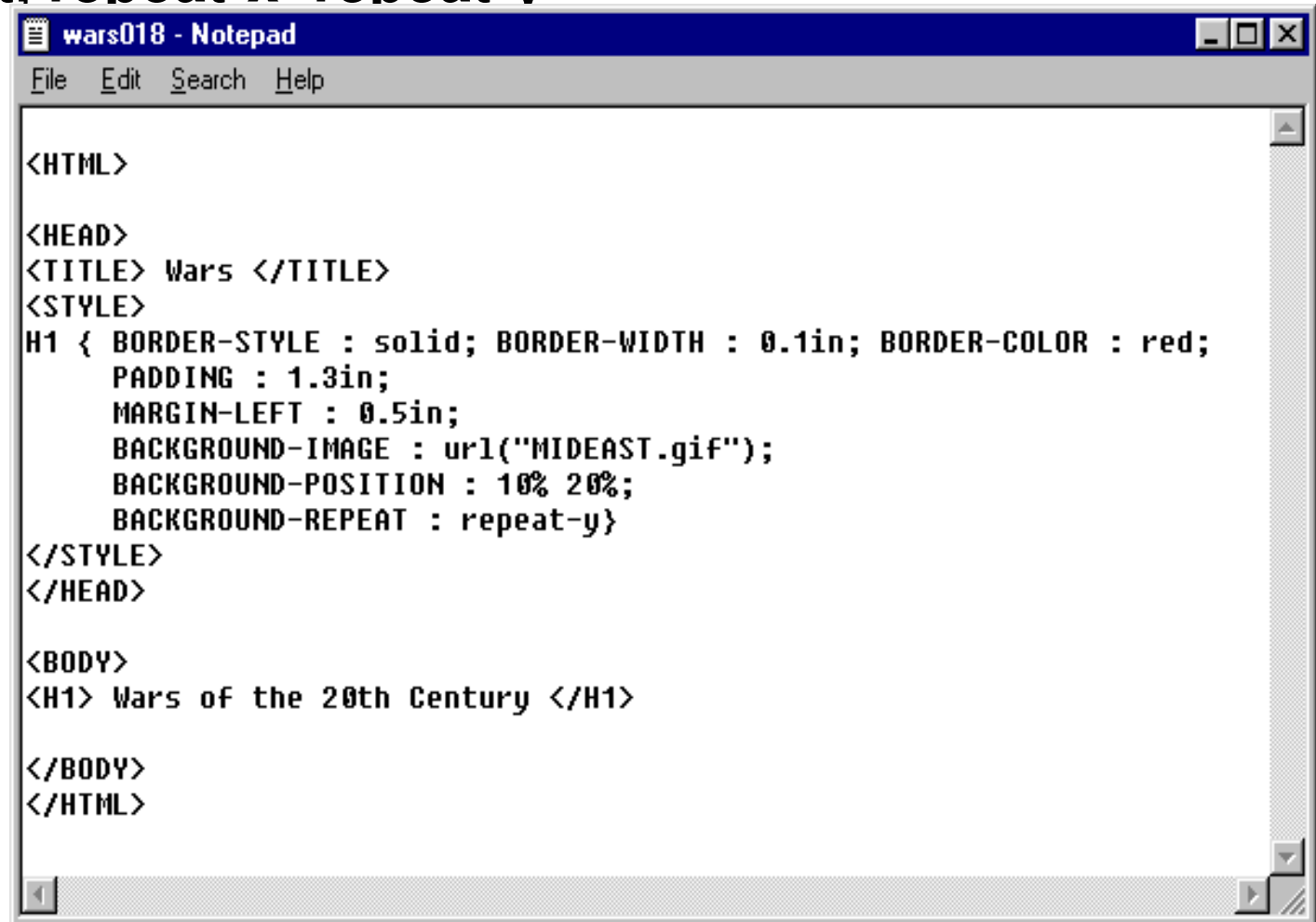
of
om



BACKGROUND IMAGE

- We can also control the form of REPEAT used in the tiling, using one of the three values:

repeat, repeat-x repeat-y



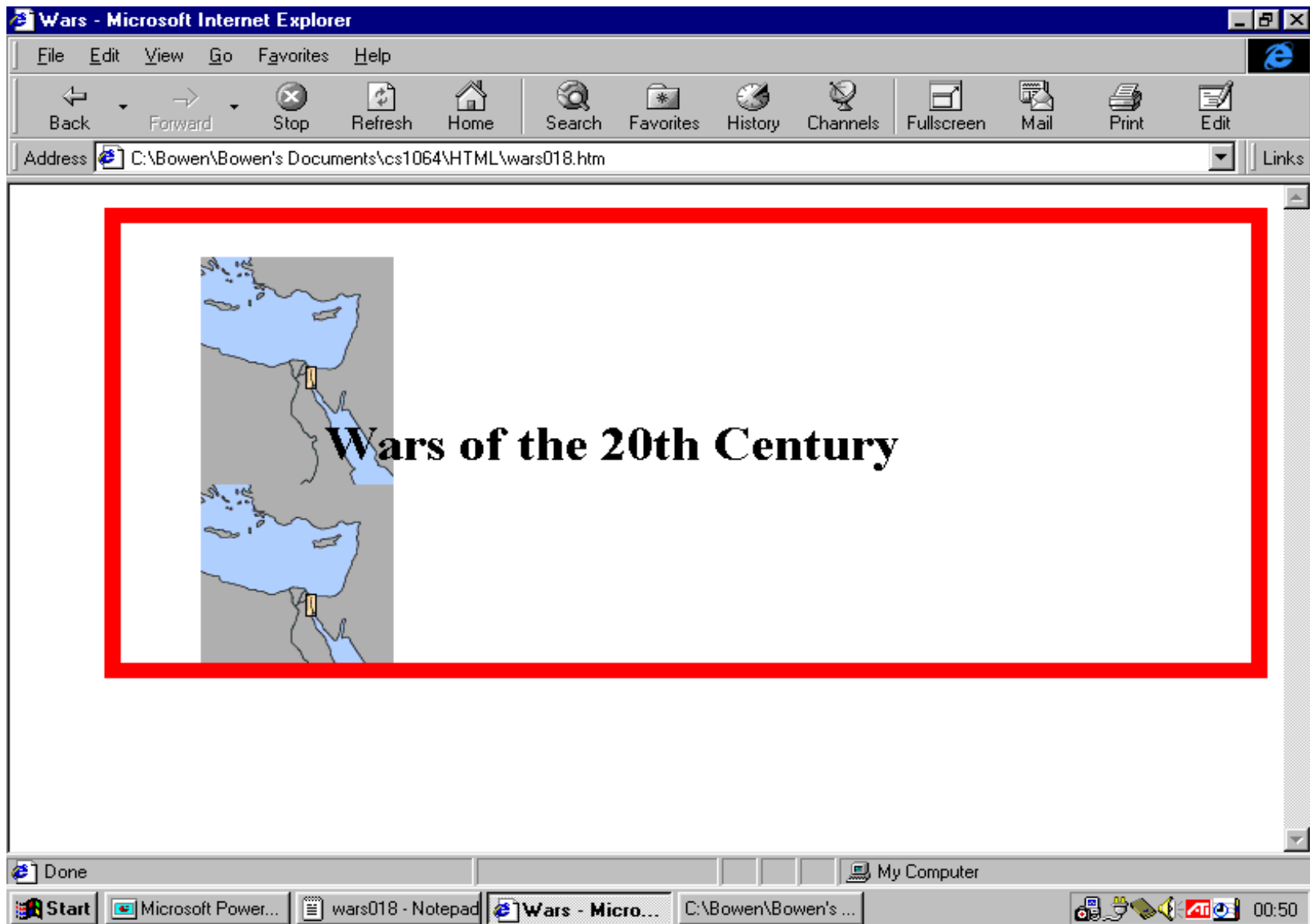
```
wars018 - Notepad
File Edit Search Help

<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-STYLE : solid; BORDER-WIDTH : 0.1in; BORDER-COLOR : red;
      PADDING : 1.3in;
      MARGIN-LEFT : 0.5in;
      BACKGROUND-IMAGE : url("MIDEAST.gif");
      BACKGROUND-POSITION : 10% 20%;
      BACKGROUND-REPEAT : repeat-y}
</STYLE>
</HEAD>

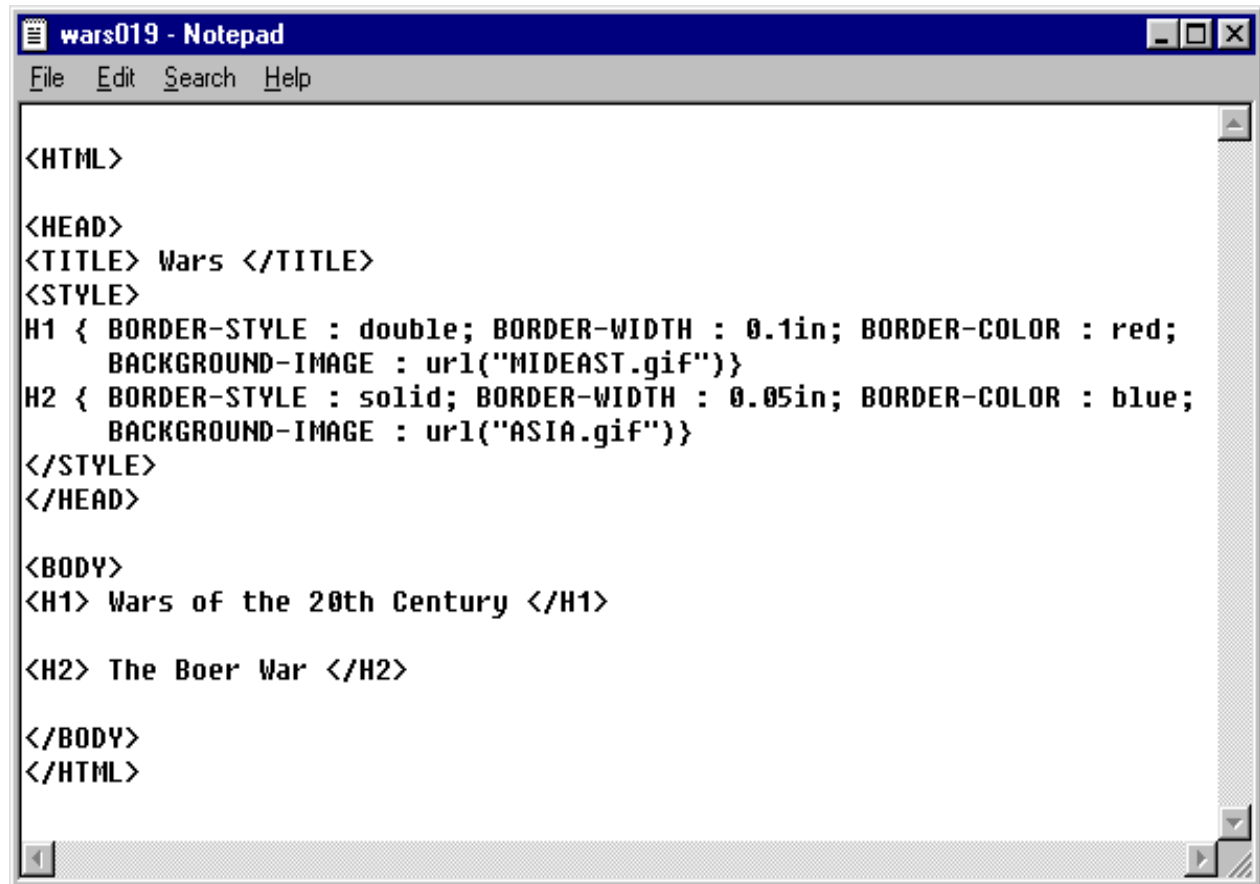
<BODY>
<H1> Wars of the 20th Century </H1>

</BODY>
</HTML>
```



Different Box elements

- All our examples to date have been shown in terms of the box for H1 elements
- Many other elements also have a box, including H2 elements



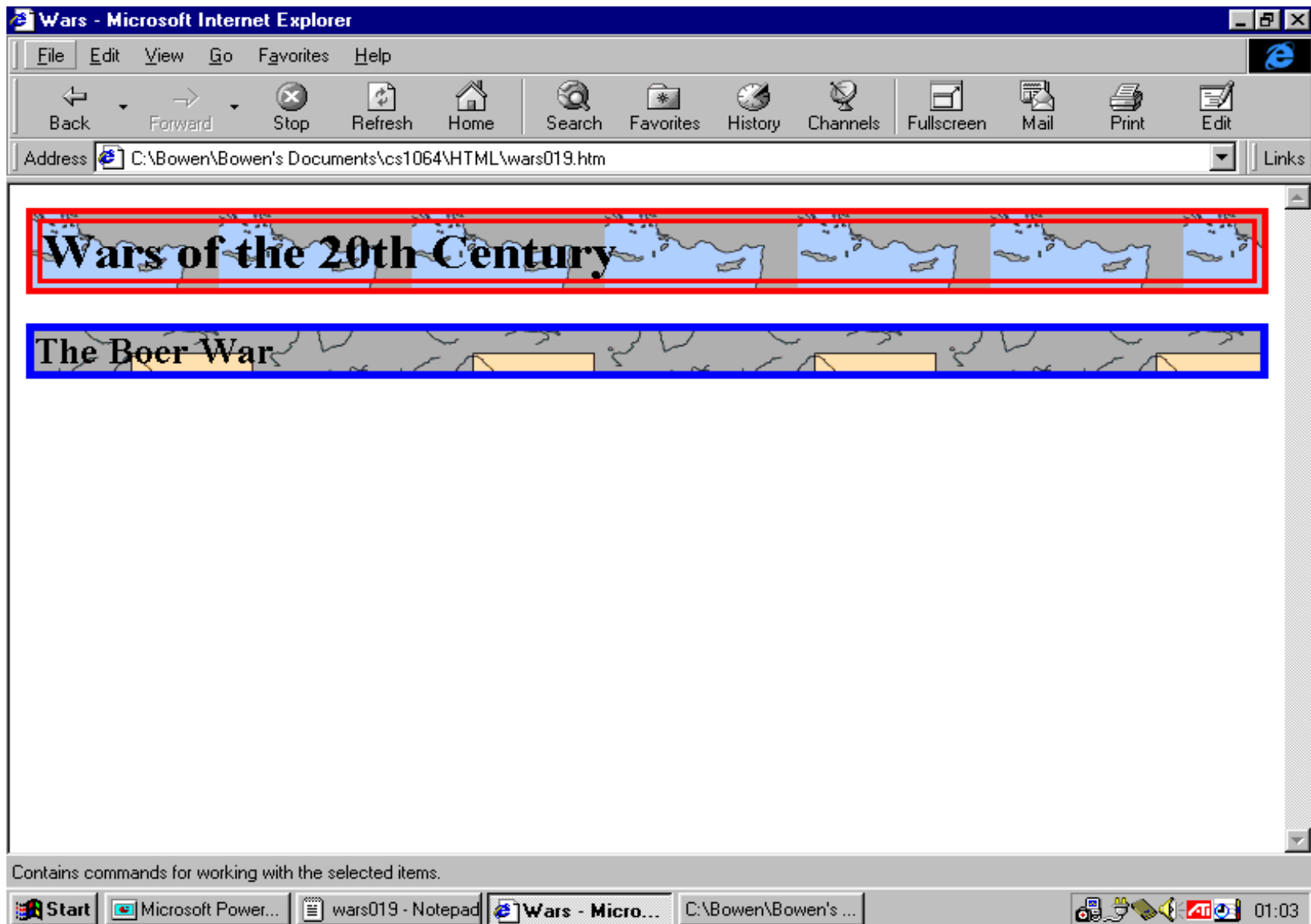
```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { BORDER-STYLE : double; BORDER-WIDTH : 0.1in; BORDER-COLOR : red;
      BACKGROUND-IMAGE : url("MIDEAST.gif")}
H2 { BORDER-STYLE : solid; BORDER-WIDTH : 0.05in; BORDER-COLOR : blue;
      BACKGROUND-IMAGE : url("ASIA.gif")}
</STYLE>
</HEAD>

<BODY>
<H1> Wars of the 20th Century </H1>

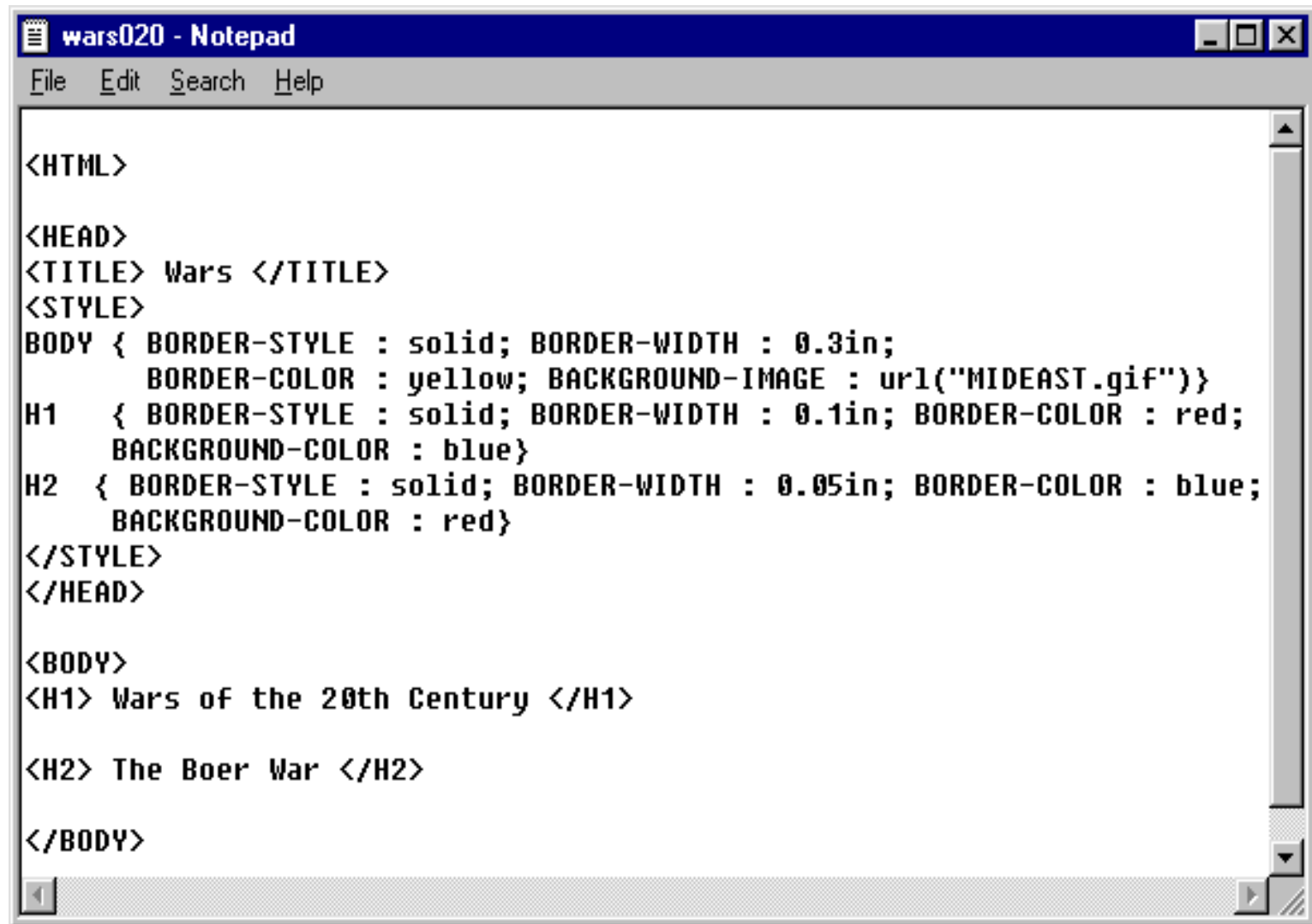
<H2> The Boer War </H2>

</BODY>
</HTML>
```



Box elements

- A BODY also has a box



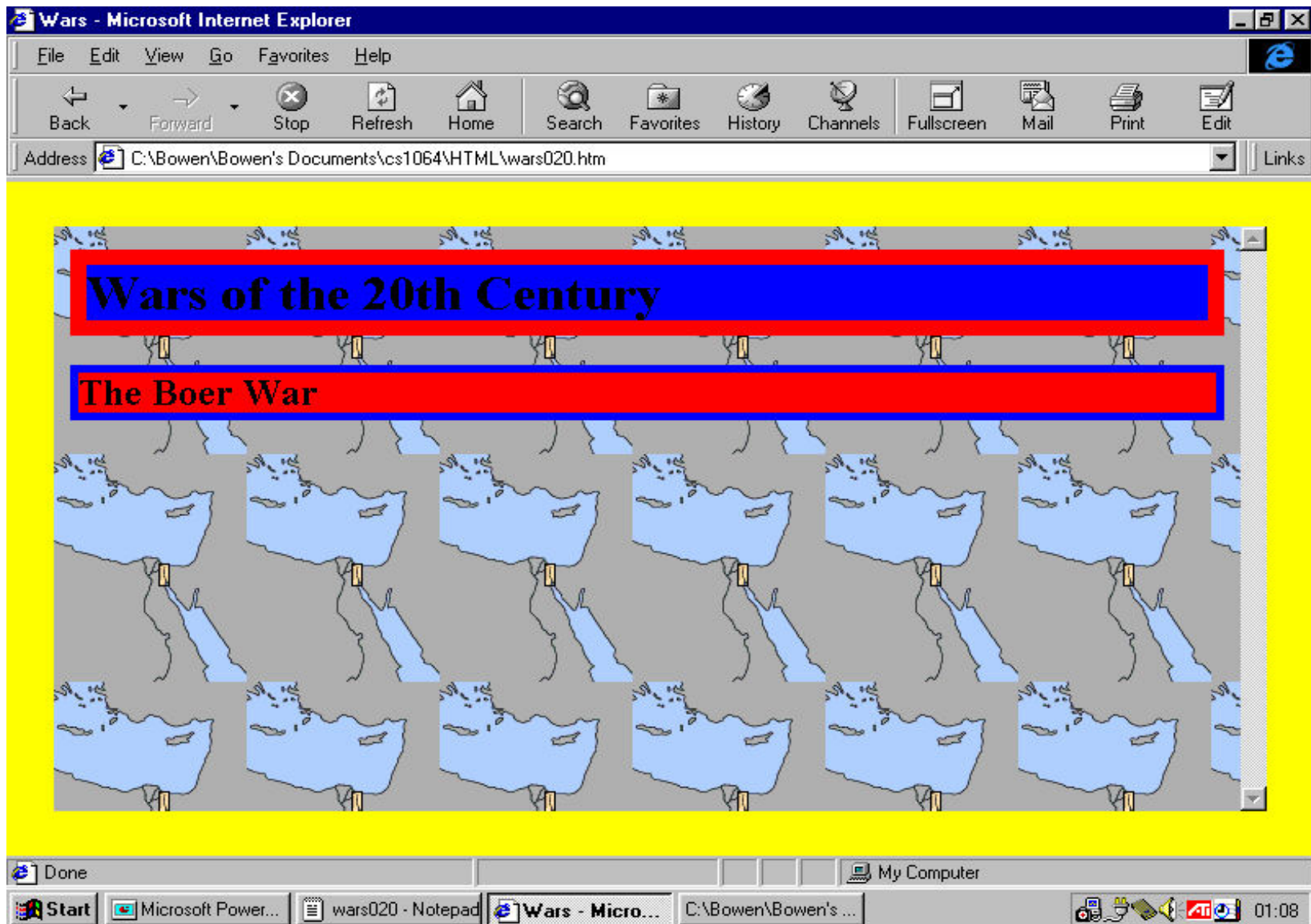
```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
BODY { BORDER-STYLE : solid; BORDER-WIDTH : 0.3in;
      BORDER-COLOR : yellow; BACKGROUND-IMAGE : url("MIDEAST.gif")}
H1   { BORDER-STYLE : solid; BORDER-WIDTH : 0.1in; BORDER-COLOR : red;
      BACKGROUND-COLOR : blue}
H2   { BORDER-STYLE : solid; BORDER-WIDTH : 0.05in; BORDER-COLOR : blue;
      BACKGROUND-COLOR : red}
</STYLE>
</HEAD>

<BODY>
<H1> Wars of the 20th Century </H1>

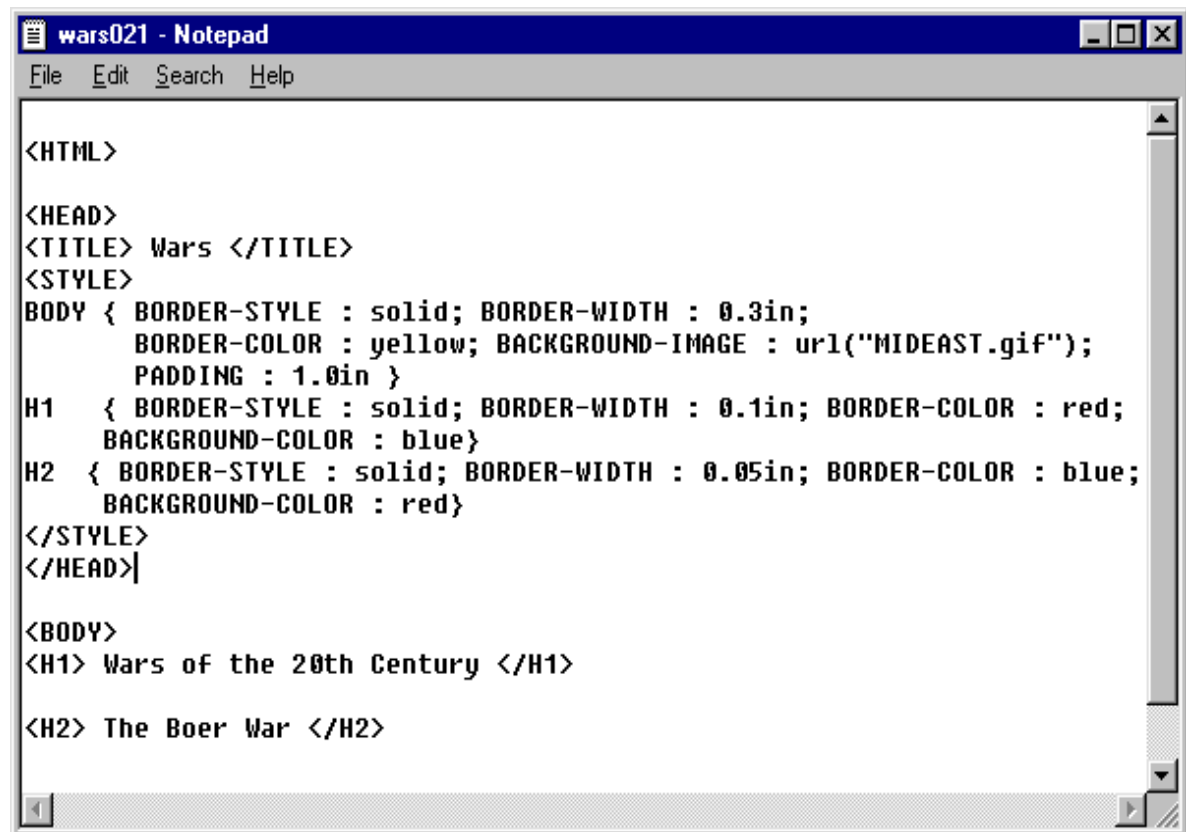
<H2> The Boer War </H2>

</BODY>
```



Box elements

- The box for a BODY element has the full range of box attributes that we have seen including, for example, the PADDING attribute, which specifies the empty space between the border of the box and its content

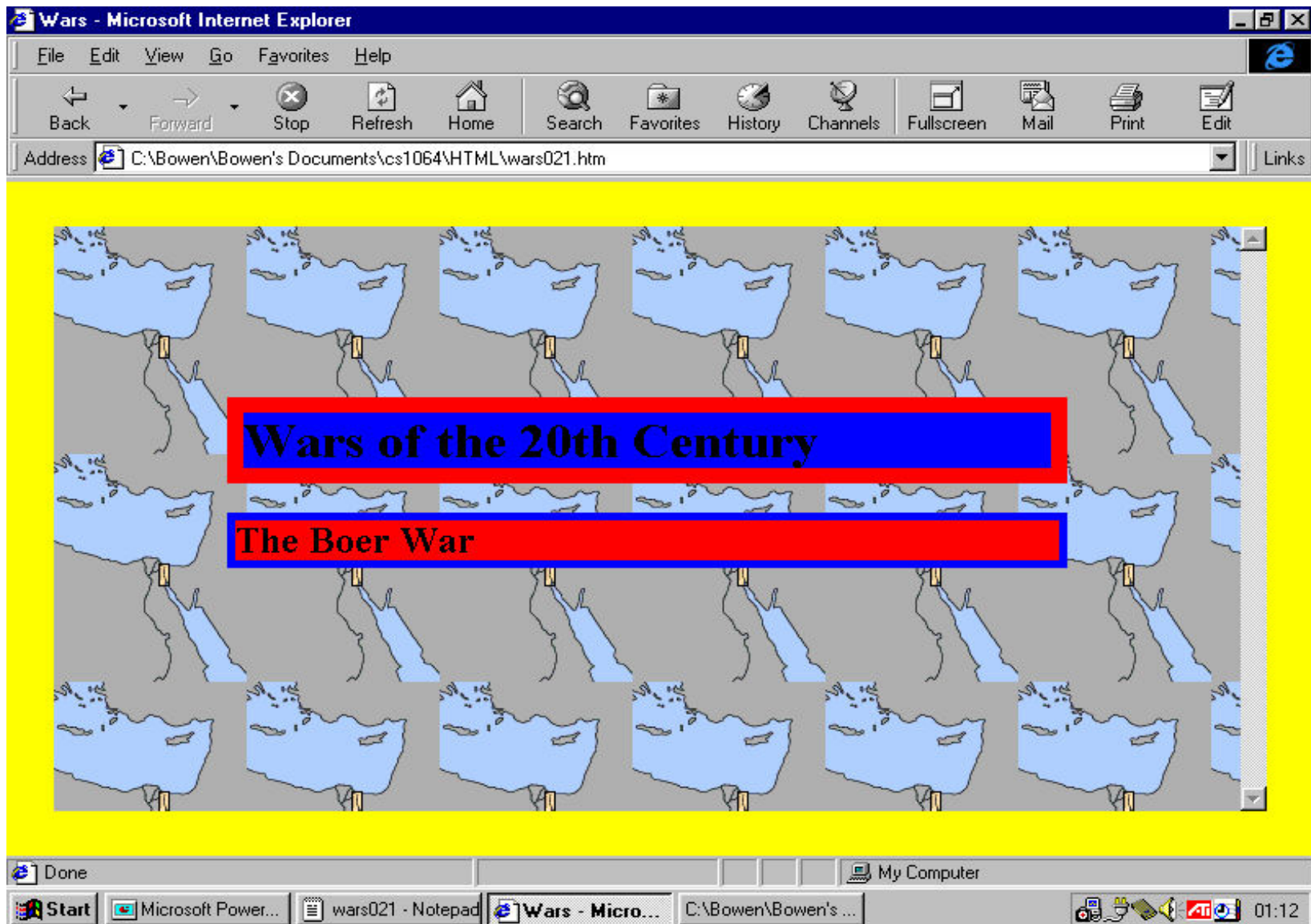


```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
BODY { BORDER-STYLE : solid; BORDER-WIDTH : 0.3in;
      BORDER-COLOR : yellow; BACKGROUND-IMAGE : url("MIDEAST.gif");
      PADDING : 1.0in }
H1   { BORDER-STYLE : solid; BORDER-WIDTH : 0.1in; BORDER-COLOR : red;
      BACKGROUND-COLOR : blue}
H2   { BORDER-STYLE : solid; BORDER-WIDTH : 0.05in; BORDER-COLOR : blue;
      BACKGROUND-COLOR : red}
</STYLE>
</HEAD>

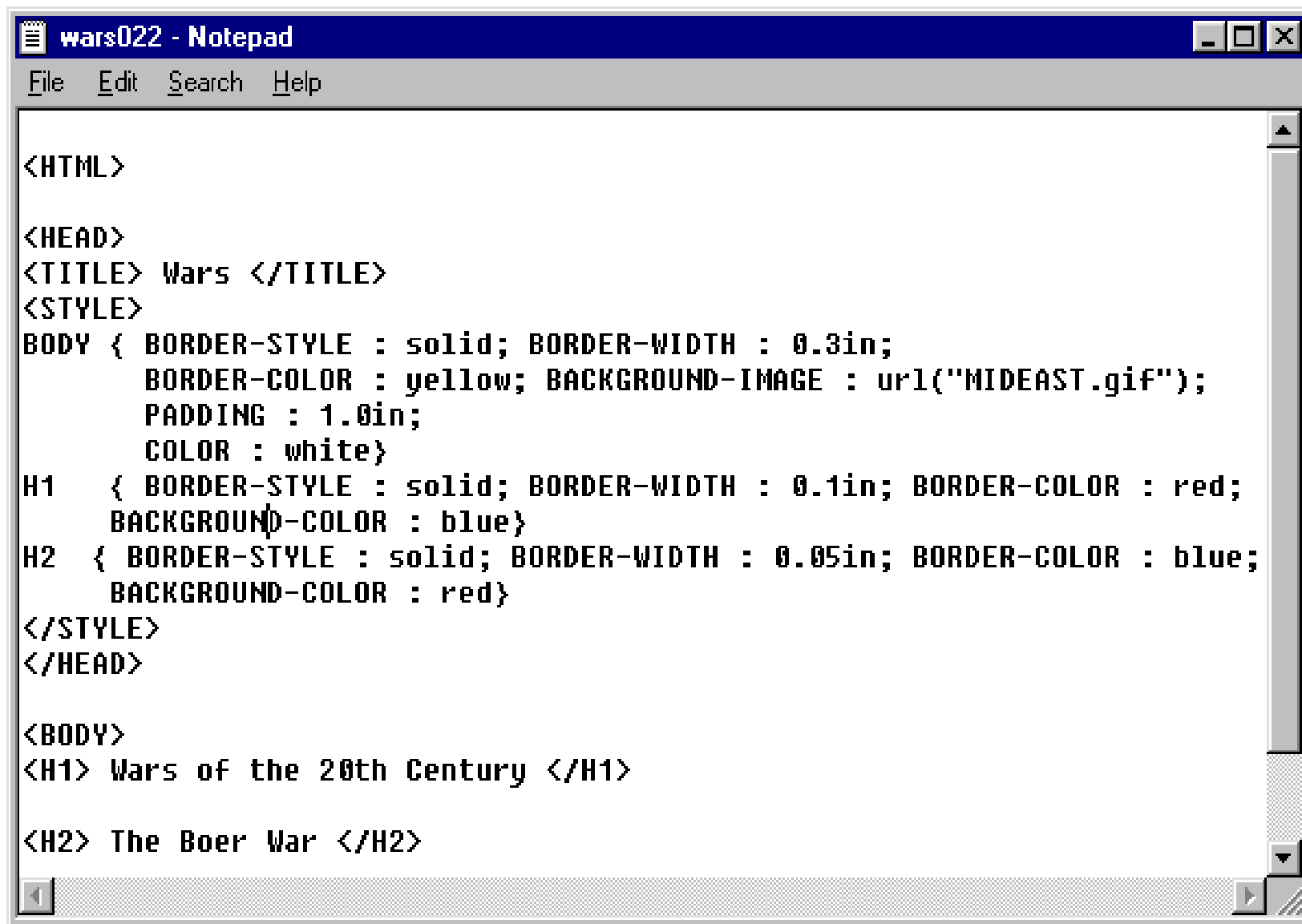
<BODY>
<H1> Wars of the 20th Century </H1>

<H2> The Boer War </H2>
```



Foreground (content) of a box

- We have discussed the BACKGROUND of a box quite extensively
- The foreground of the box is its content
- The most typical content is text
- Content has the attribute COLOR
- In the following example, notice that the boxes within the BODY inherit its value for the COLOR attribute

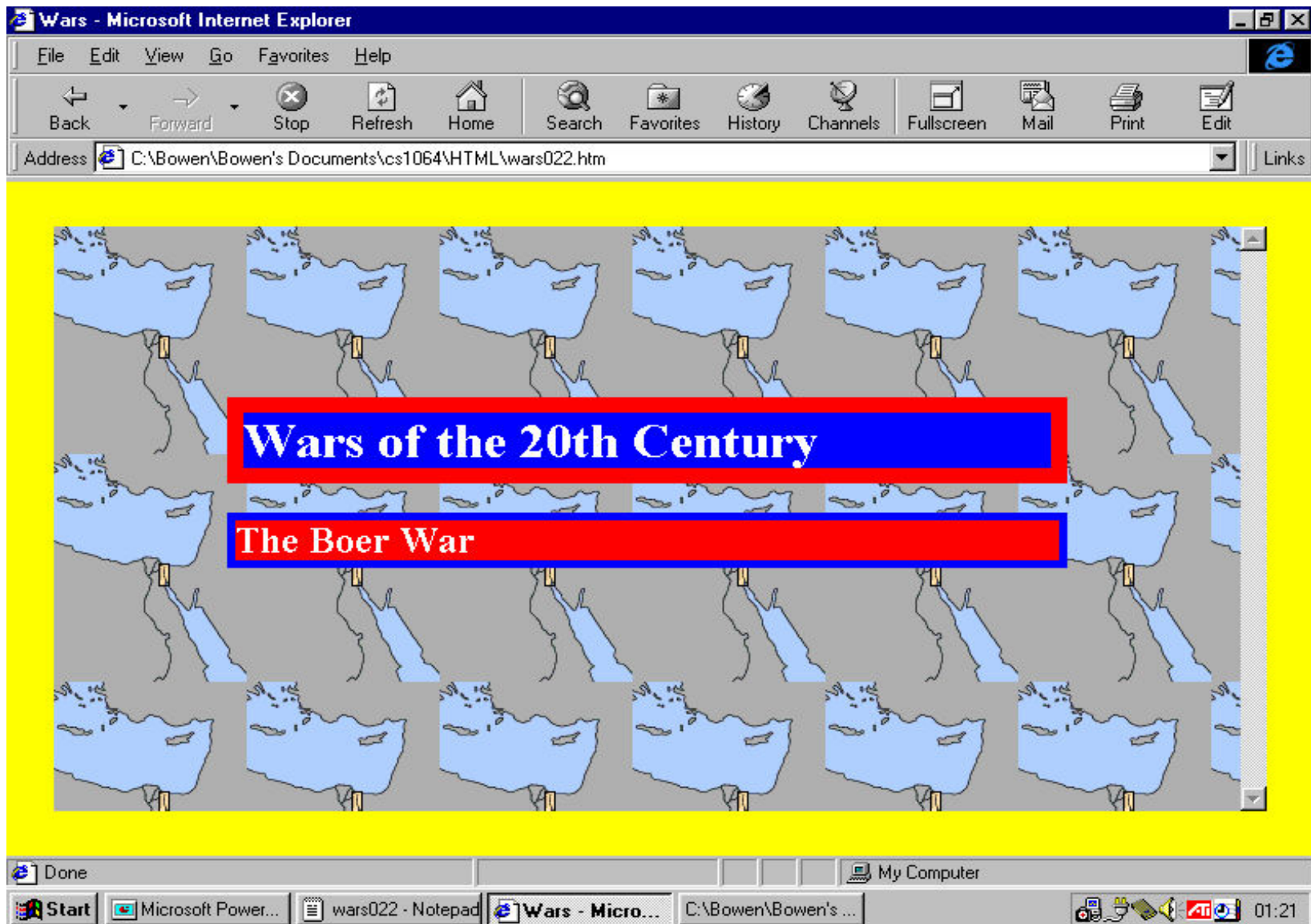


```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
BODY { BORDER-STYLE : solid; BORDER-WIDTH : 0.3in;
      BORDER-COLOR : yellow; BACKGROUND-IMAGE : url("MIDEAST.gif");
      PADDING : 1.0in;
      COLOR : white}
H1   { BORDER-STYLE : solid; BORDER-WIDTH : 0.1in; BORDER-COLOR : red;
      BACKGROUND-COLOR : blue}
H2   { BORDER-STYLE : solid; BORDER-WIDTH : 0.05in; BORDER-COLOR : blue;
      BACKGROUND-COLOR : red}
</STYLE>
</HEAD>

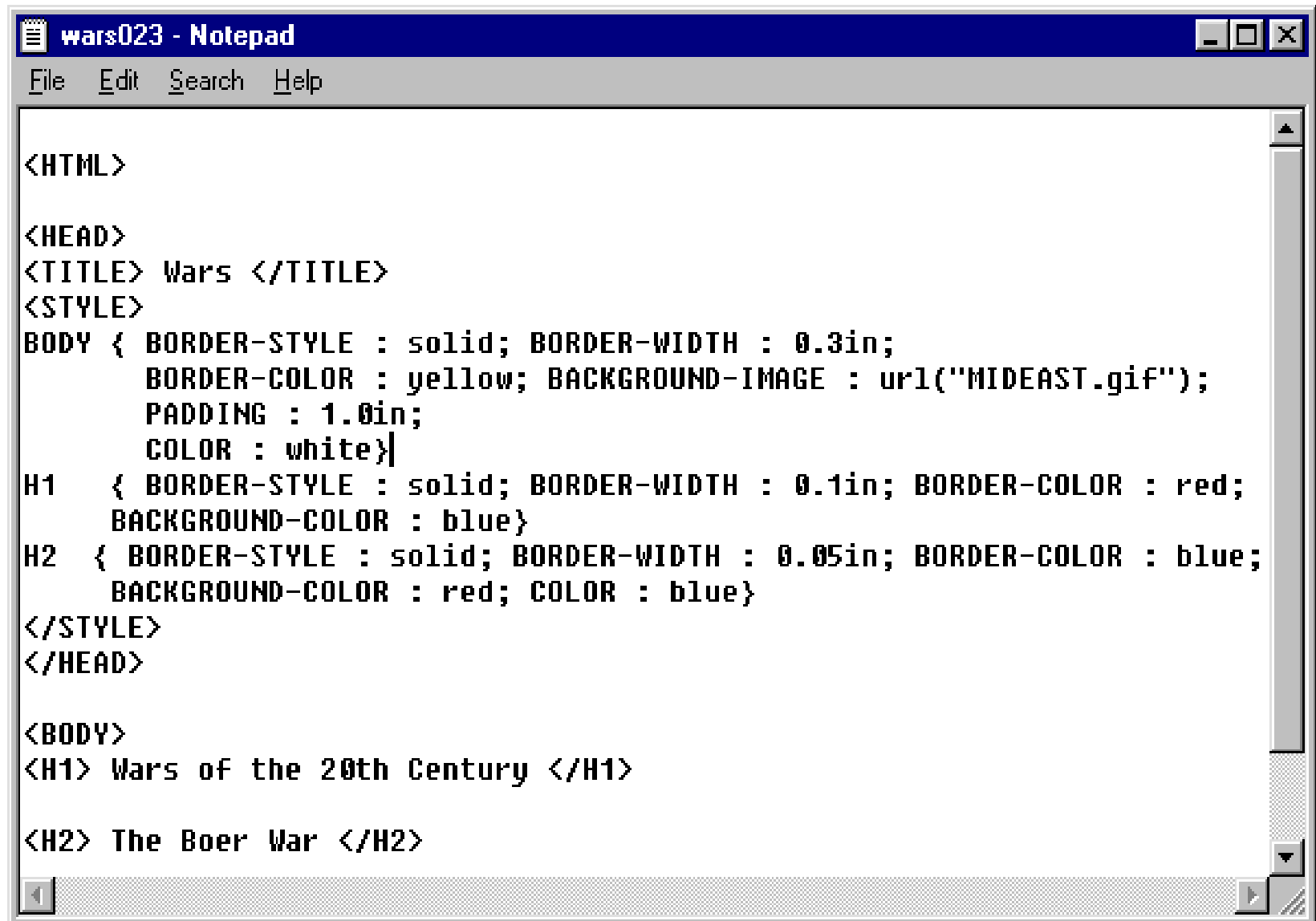
<BODY>
<H1> Wars of the 20th Century </H1>

<H2> The Boer War </H2>
```



Overriding inherited values

-



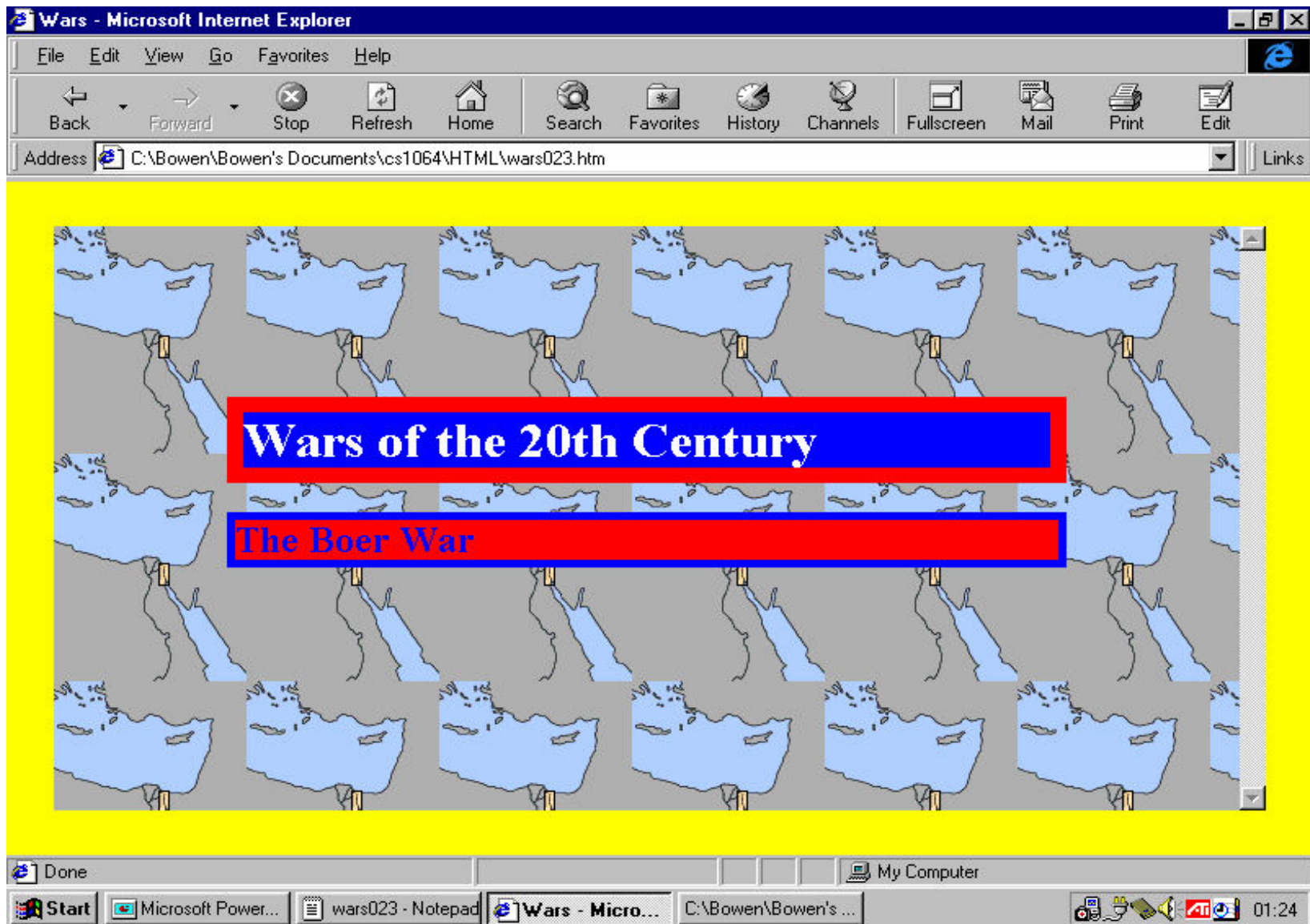
```
wars023 - Notepad
File Edit Search Help

<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
BODY { BORDER-STYLE : solid; BORDER-WIDTH : 0.3in;
      BORDER-COLOR : yellow; BACKGROUND-IMAGE : url("MIDEAST.gif");
      PADDING : 1.0in;
      COLOR : white}
H1   { BORDER-STYLE : solid; BORDER-WIDTH : 0.1in; BORDER-COLOR : red;
      BACKGROUND-COLOR : blue}
H2   { BORDER-STYLE : solid; BORDER-WIDTH : 0.05in; BORDER-COLOR : blue;
      BACKGROUND-COLOR : red; COLOR : blue}
</STYLE>
</HEAD>

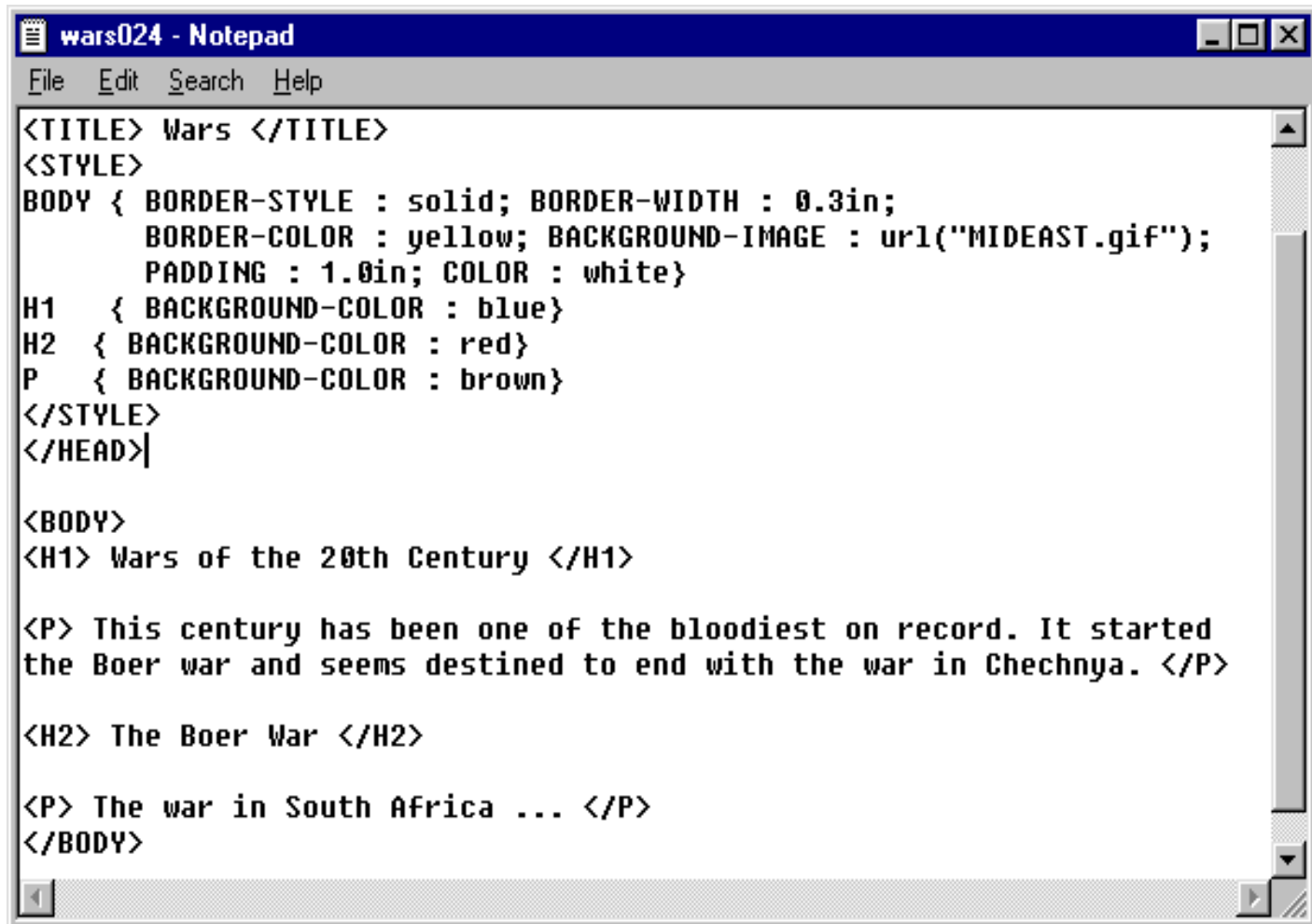
<BODY>
<H1> Wars of the 20th Century </H1>

<H2> The Boer War </H2>
```

Box elements

- Notice that even a P (paragraph) element has it box and that it inherits attributes, such as the COLOR of its content from its ancestral BODY



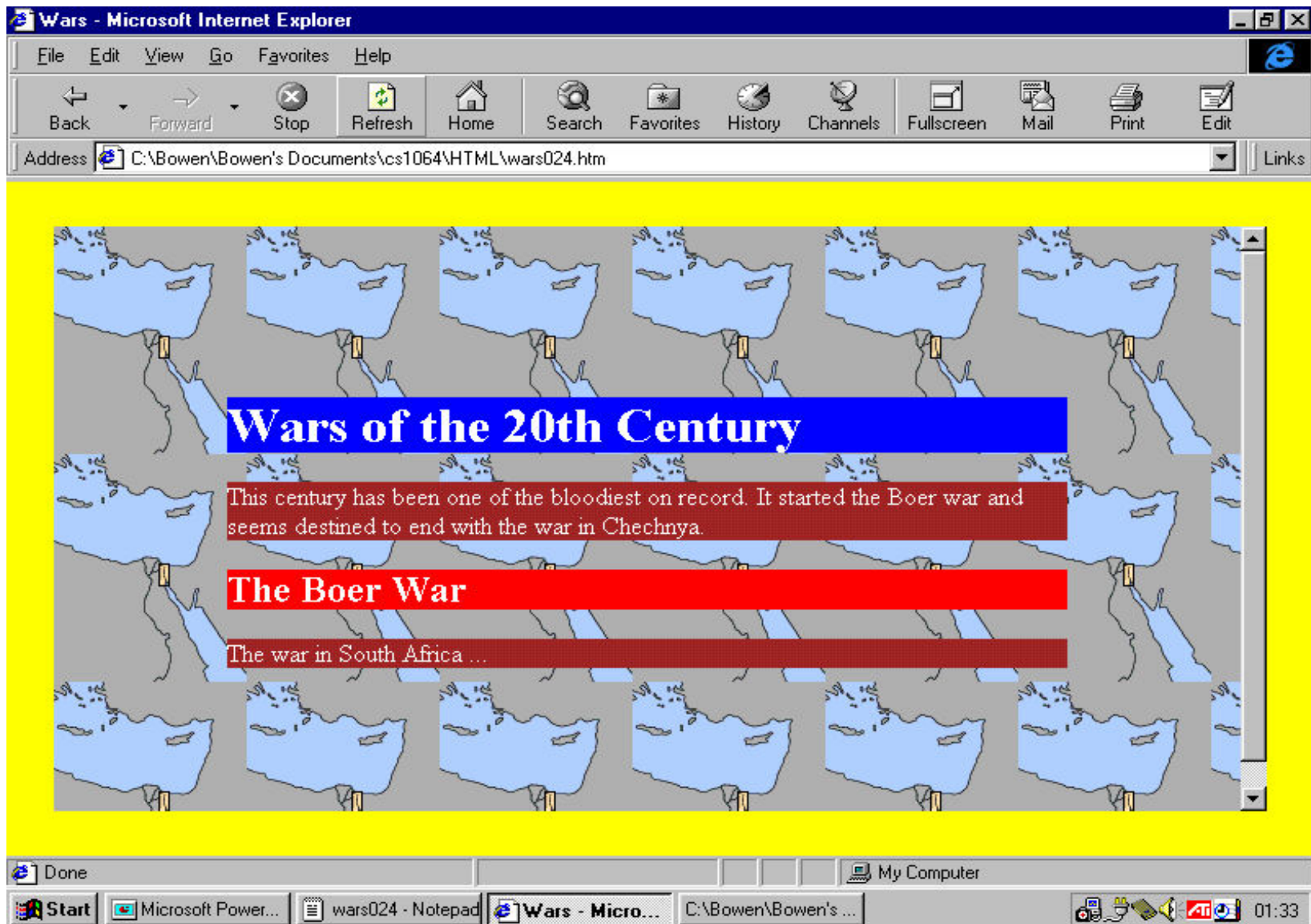
```
<TITLE> Wars </TITLE>
<STYLE>
BODY { BORDER-STYLE : solid; BORDER-WIDTH : 0.3in;
      BORDER-COLOR : yellow; BACKGROUND-IMAGE : url("MIDEAST.gif");
      PADDING : 1.0in; COLOR : white}
H1   { BACKGROUND-COLOR : blue}
H2   { BACKGROUND-COLOR : red}
P     { BACKGROUND-COLOR : brown}
</STYLE>
</HEAD>

<BODY>
<H1> Wars of the 20th Century </H1>

<P> This century has been one of the bloodiest on record. It started
the Boer war and seems destined to end with the war in Chechnya. </P>

<H2> The Boer War </H2>

<P> The war in South Africa ... </P>
</BODY>
```

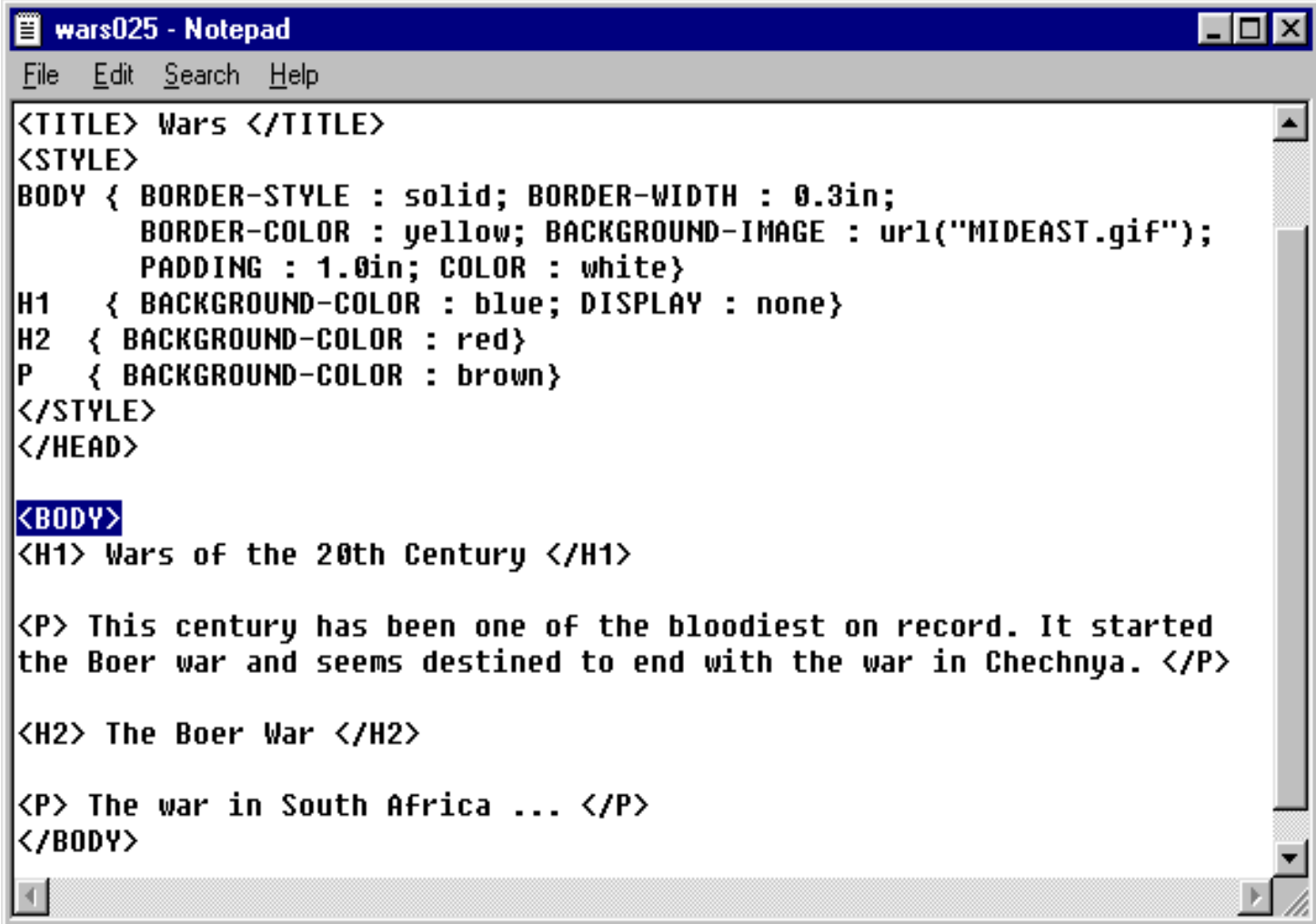


The **DISPLAY** of an element

- Every element has a **DISPLAY** attribute
- Depending on the type of element the default value of its **DISPLAY** attribute is one of
 - **inline**
 - **block**
 - **list-item** (**many more**)
- However, we can also set it to **none** which prevents the element from being rendered at all!

DISPLAY

- In the following example notice that the style sheet stops H1 elements being rendered at all, even though one is present in the document



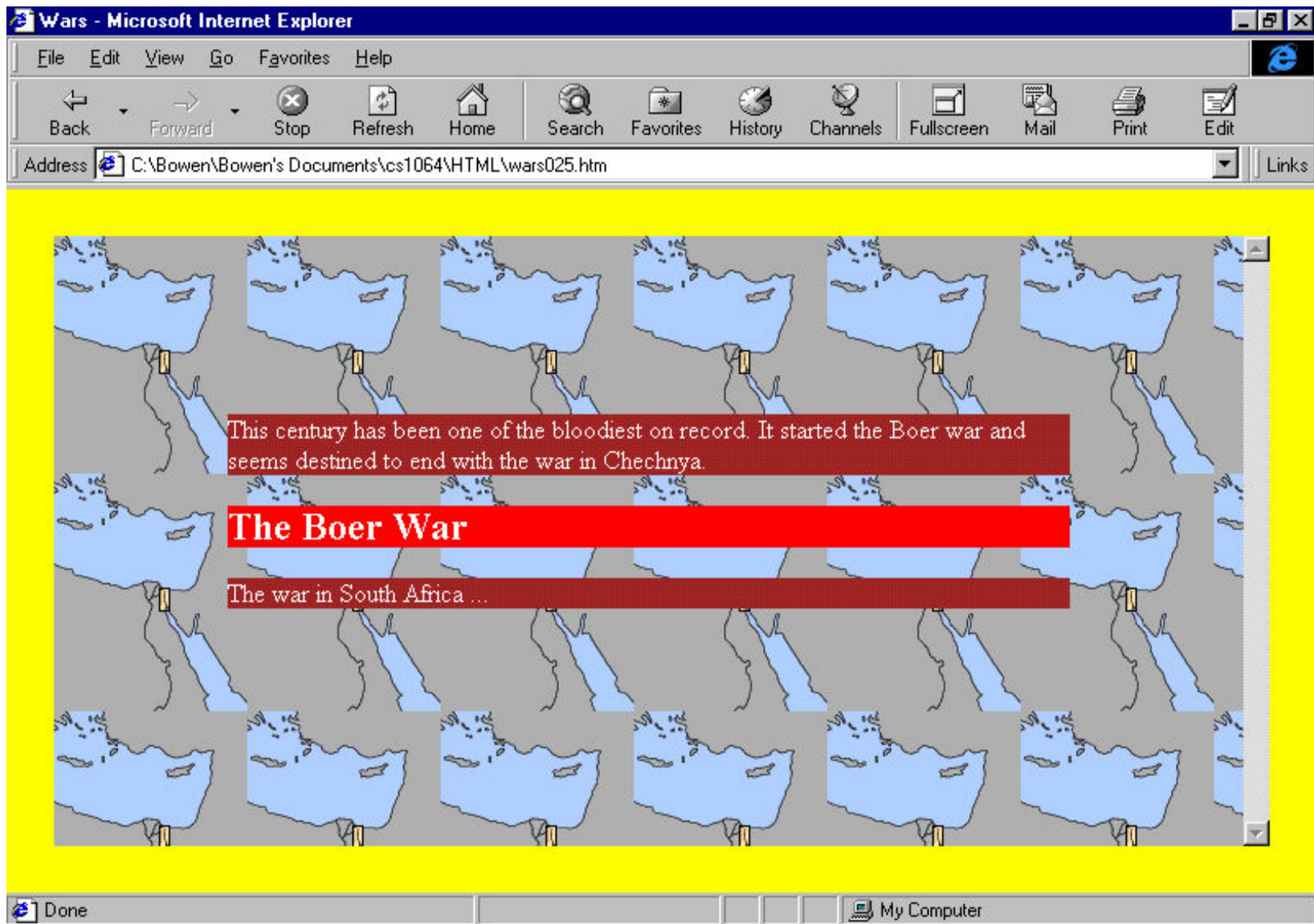
```
<TITLE> Wars </TITLE>
<STYLE>
BODY { BORDER-STYLE : solid; BORDER-WIDTH : 0.3in;
        BORDER-COLOR : yellow; BACKGROUND-IMAGE : url("MIDEAST.gif");
        PADDING : 1.0in; COLOR : white}
H1   { BACKGROUND-COLOR : blue; DISPLAY : none}
H2   { BACKGROUND-COLOR : red}
P     { BACKGROUND-COLOR : brown}
</STYLE>
</HEAD>

<BODY>
<H1> Wars of the 20th Century </H1>

<P> This century has been one of the bloodiest on record. It started
the Boer war and seems destined to end with the war in Chechnya. </P>

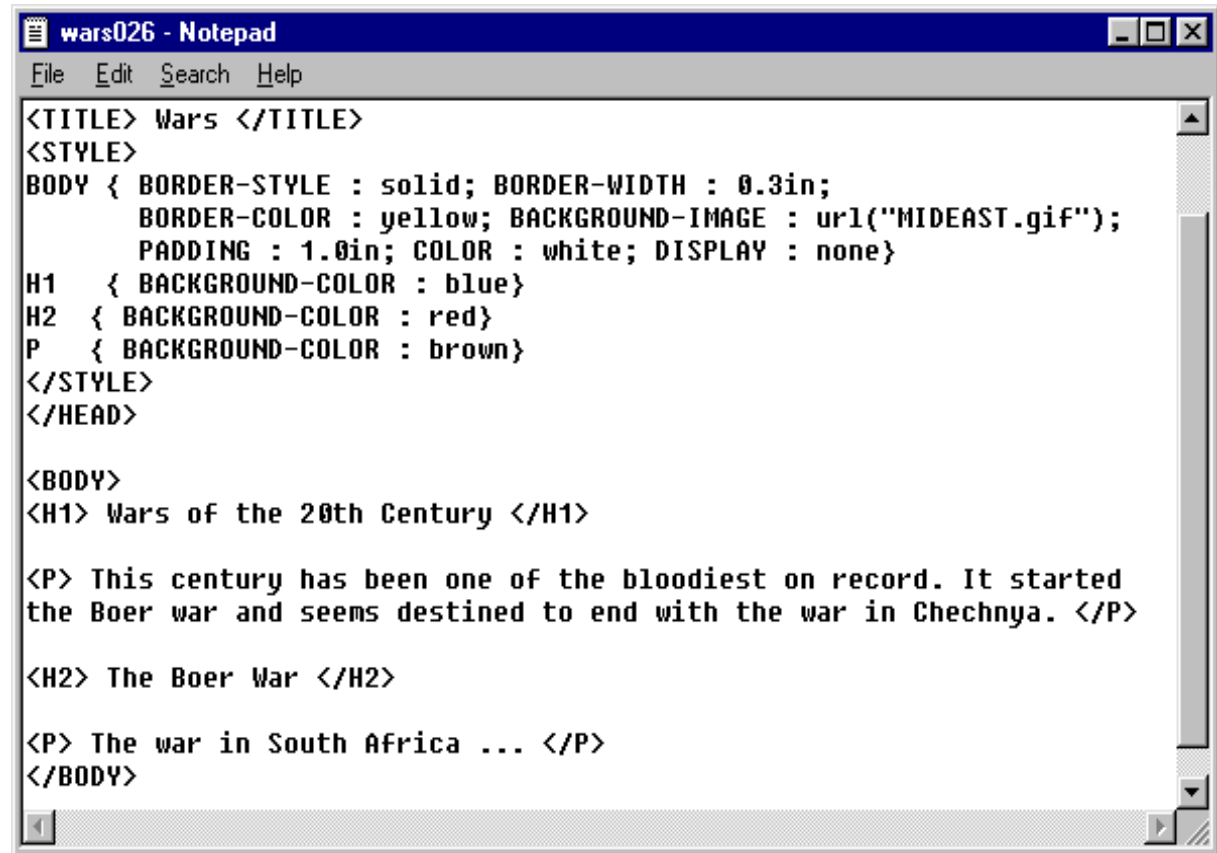
<H2> The Boer War </H2>

<P> The war in South Africa ... </P>
</BODY>
```

DISPLAY

- The DISPLAY attribute is inherited by children elements from their ancestor
- In the following notice how setting the DISPLAY attribute of the BODY to **none** stops any of the document being rendered



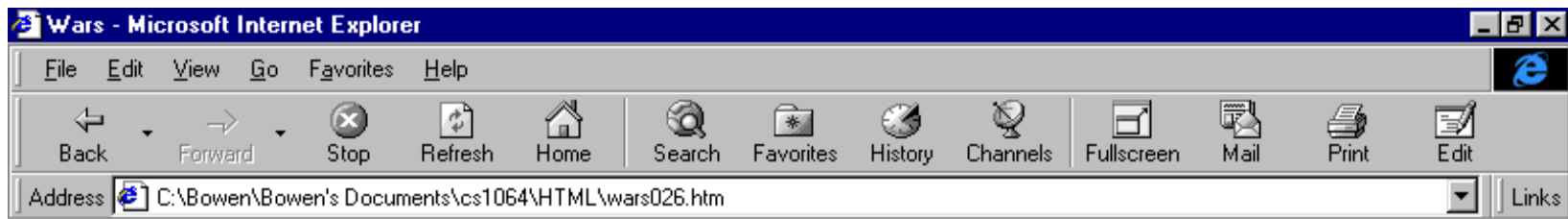
```
wars026 - Notepad
File Edit Search Help
<TITLE> Wars </TITLE>
<STYLE>
BODY { BORDER-STYLE : solid; BORDER-WIDTH : 0.3in;
      BORDER-COLOR : yellow; BACKGROUND-IMAGE : url("MIDEAST.gif");
      PADDING : 1.0in; COLOR : white; DISPLAY : none}
H1   { BACKGROUND-COLOR : blue}
H2   { BACKGROUND-COLOR : red}
P     { BACKGROUND-COLOR : brown}
</STYLE>
</HEAD>

<BODY>
<H1> Wars of the 20th Century </H1>

<P> This century has been one of the bloodiest on record. It started
the Boer war and seems destined to end with the war in Chechnya. </P>

<H2> The Boer War </H2>

<P> The war in South Africa ... </P>
</BODY>
```



In-BODY styles

- It was explained earlier that the word “Cascading” refers to the prioritization given to the three levels of style sheet:
 - linked
 - in-head
 - in-body
- with in-body styles being allowed to over-ride in-head styles which can over-ride linked styles
- It was also stated that we should rarely need to use in-body styles
- However, we will consider them now, because there ARE times when they are needed

General format of in-body style-sheets

- We can insert a stylesheet in the opening tag for any logical element for which it is possible to define a style

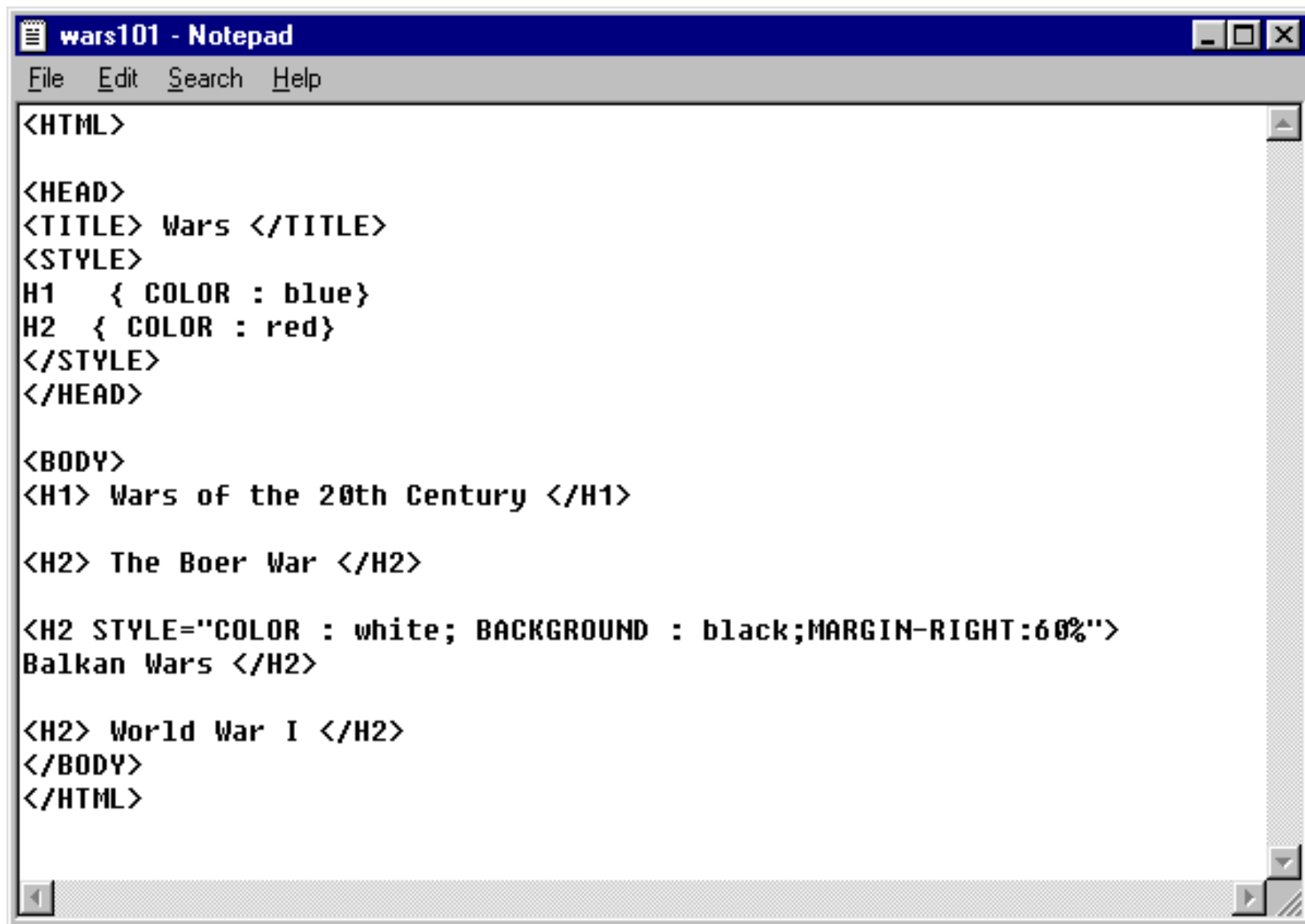
- The format is

`< some-tag style="some-stylesheet">`

- For example:

`<h1 style="color : red">`

- In the next slide, an in-body stylesheet for a specific H2 over-rides the general style specified for H2 in the in-head stylesheet



```
<HTML>

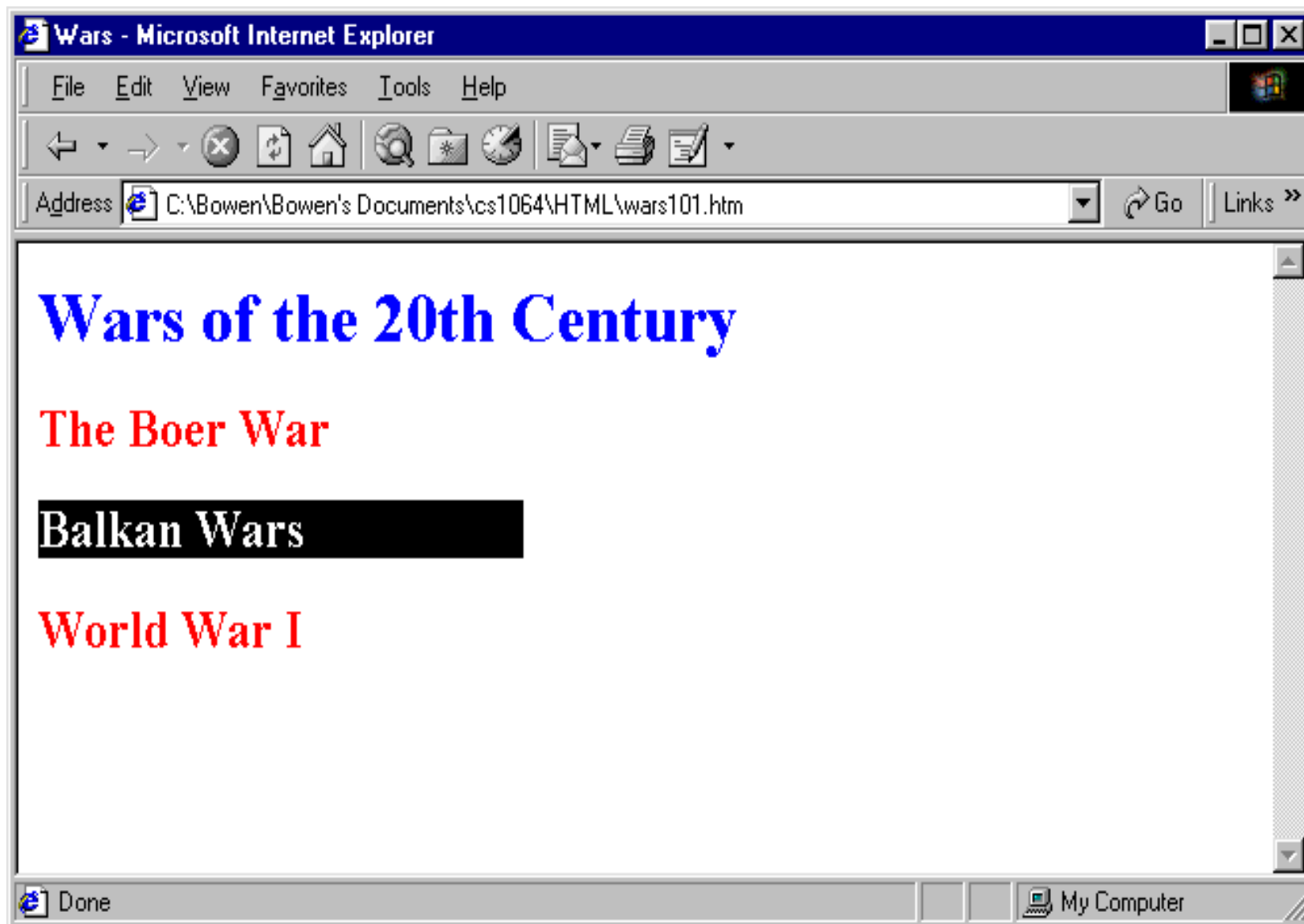
<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1  { COLOR : blue}
H2  { COLOR : red}
</STYLE>
</HEAD>

<BODY>
<H1> Wars of the 20th Century </H1>

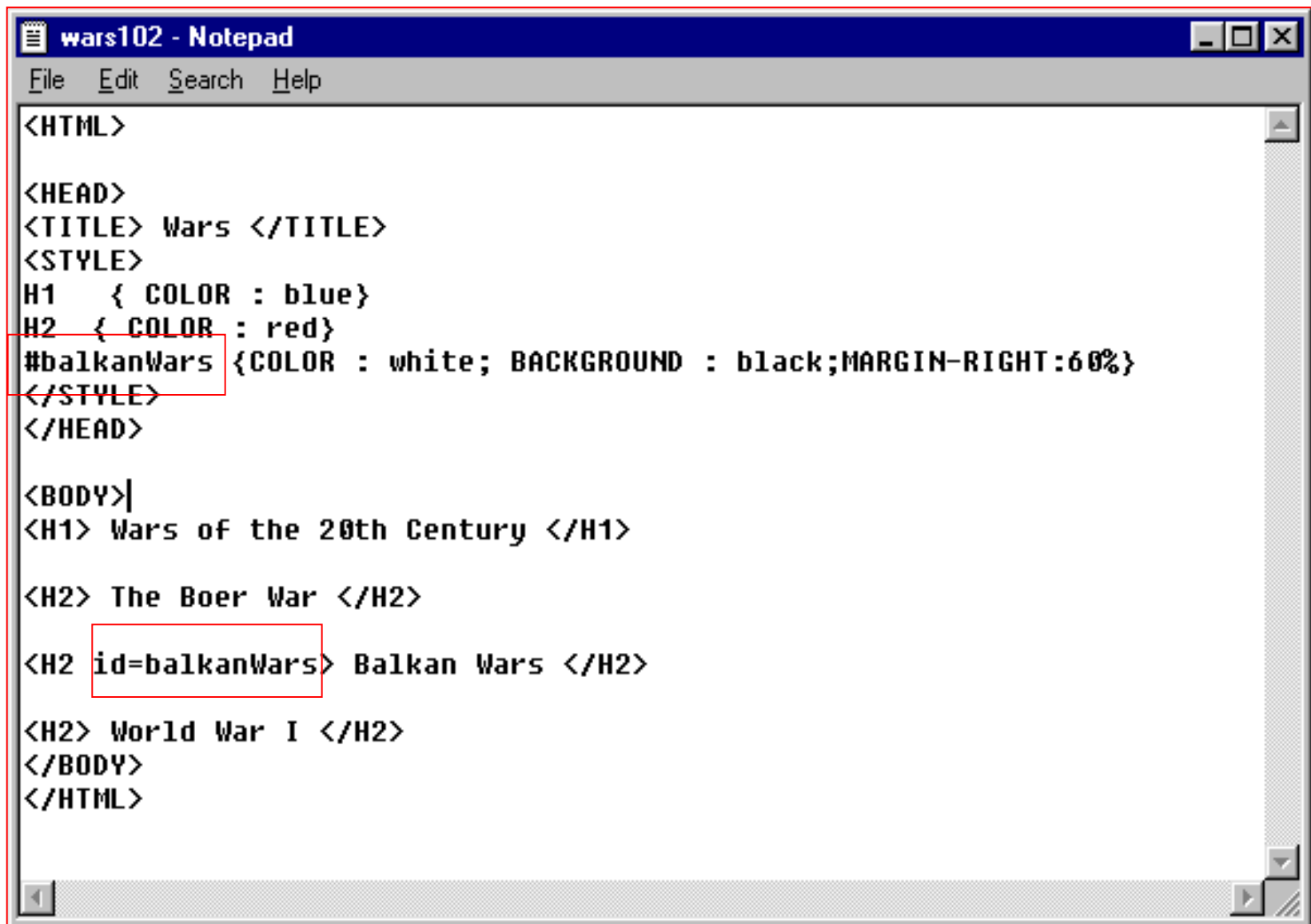
<H2> The Boer War </H2>

<H2 STYLE="COLOR : white; BACKGROUND : black;MARGIN-RIGHT:60%">
Balkan Wars </H2>

<H2> World War I </H2>
</BODY>
</HTML>
```



- There is, however, another way of achieving the same thing:
 - we can give a unique identifier to specific elements in our document
 - and specify a style for this specific element in a global stylesheet such as an in-head stylesheet or a linked style-sheet
- In the document on the next slide, we give the identifier **balkanWars** to the H2 which introduces the Balkan Wars
- In the in-head stylesheet we specify a style for the element with the identifier **balkanWars**
- We could have done the same thing in a linked style-sheet



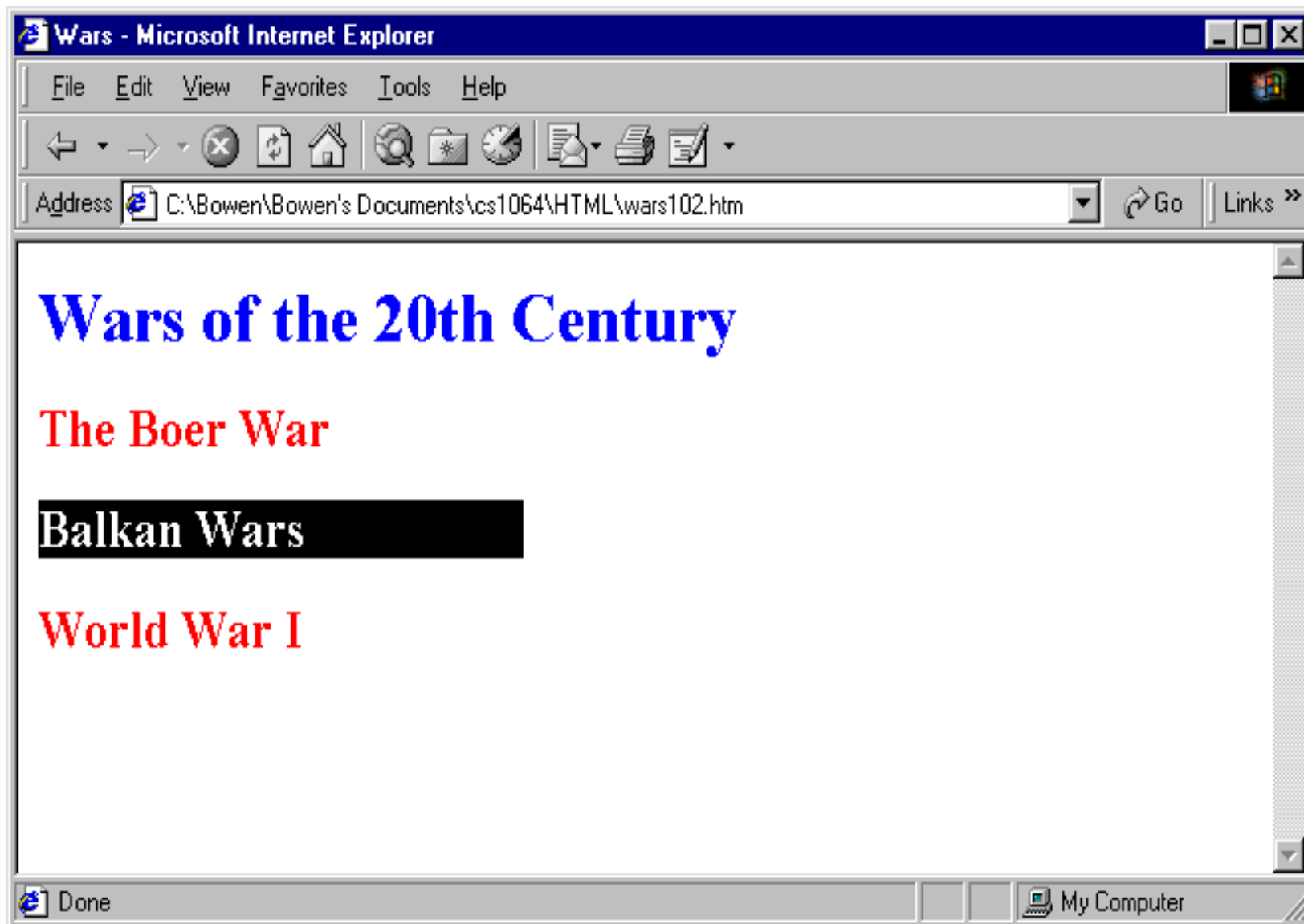
```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { COLOR : blue}
H2 { COLOR : red}
#balkanWars {COLOR : white; BACKGROUND : black;MARGIN-RIGHT:60%}
</STYLE>
</HEAD>

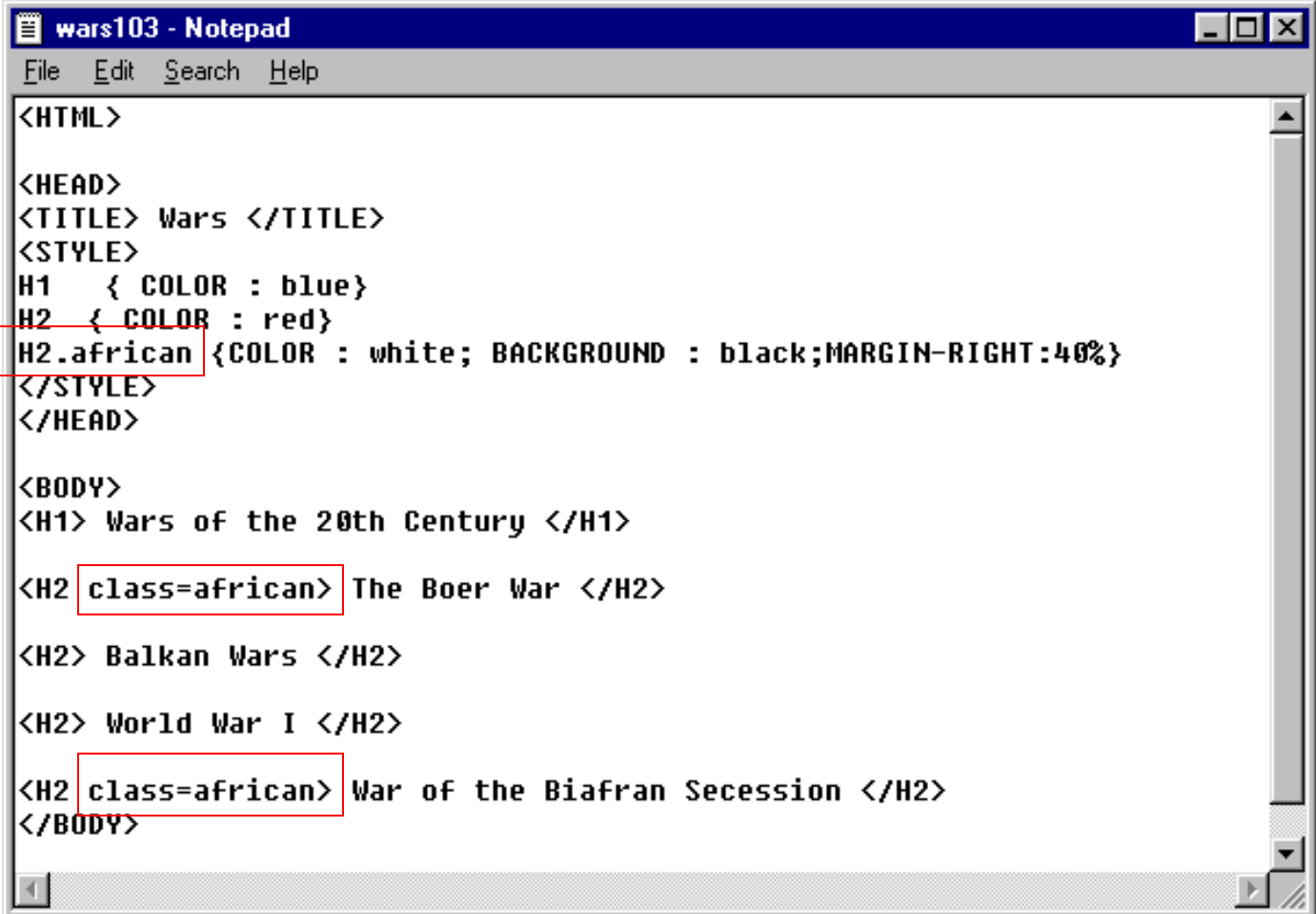
<BODY>
<H1> Wars of the 20th Century </H1>

<H2> The Boer War </H2>
<H2 id=balkanWars> Balkan Wars </H2>

<H2> World War I </H2>
</BODY>
</HTML>
```



- We can do the same thing for whole groups of elements, instead of just for individuals
- In the next slide, we give one style for all H2s but we then over-ride it with a different style for H2s which introduce African wars



```
<HTML>

<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1  { COLOR : blue}
H2  { COLOR : red}
H2.african {COLOR : white; BACKGROUND : black;MARGIN-RIGHT:40%}
</STYLE>
</HEAD>

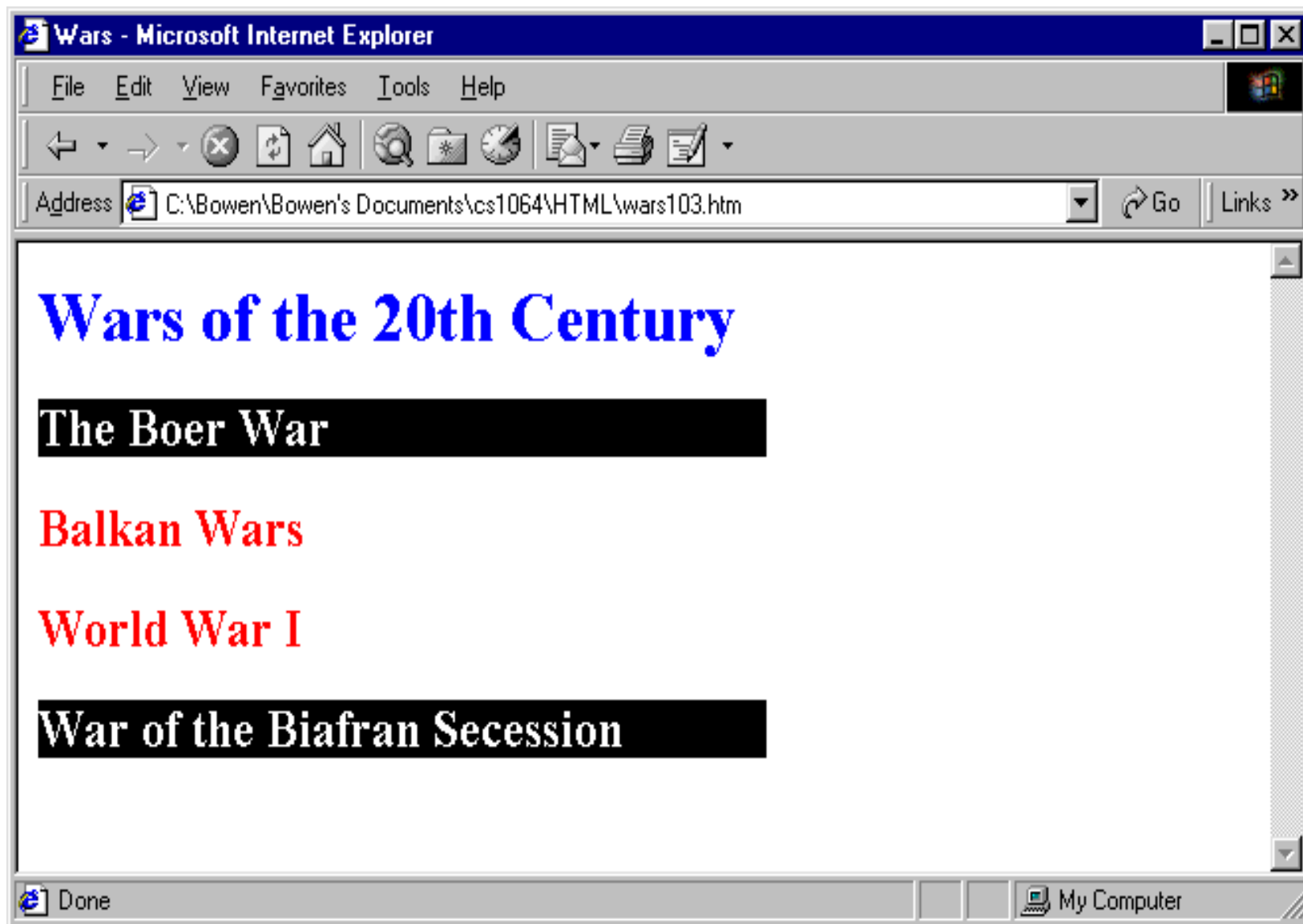
<BODY>
<H1> Wars of the 20th Century </H1>

<H2 class=african> The Boer War </H2>

<H2> Balkan Wars </H2>

<H2> World War I </H2>

<H2 class=african> War of the Biafran Secession </H2>
</BODY>
```

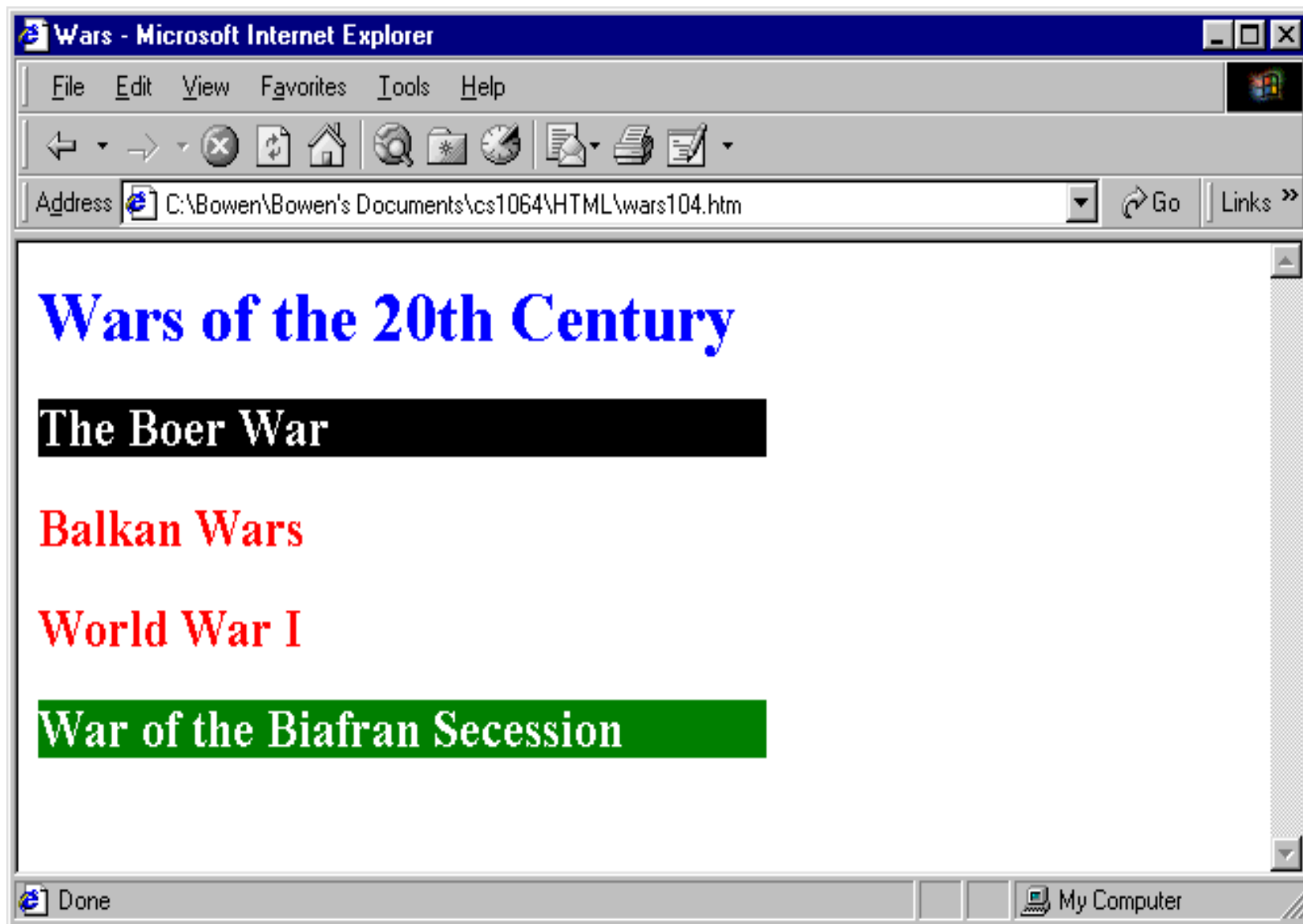



- Of course, we use both **id** and **class** in the same stylesheet
- In the next slide, we give a general style for all H2
- But we override it with a style for H2s of **class=african**
- But we then over-ride part of that with a different **background-color** for the H2 with **id=biafra**

```
wars104 - Notepad
File Edit Search Help

<HTML>
<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { COLOR : blue}
H2 { COLOR : red}
H2.african {COLOR : white; BACKGROUND : black;MARGIN-RIGHT:40%}
#biafra {BACKGROUND-COLOR : green}
</STYLE>
</HEAD>

<BODY>
<H1> Wars of the 20th Century </H1>
<H2 class=african> The Boer War </H2>
<H2> Balkan Wars </H2>
<H2> World War I </H2>
<H2 class=african id=biafra> War of the Biafran Secession </H2>
</BODY>
</HTML>
```



Selector Format

*	Matches any element.
E	Matches any E element (i.e., an element of type E).
E F	Matches any F element that is a descendant of an E element.
E > F	Matches any F element that is a child of an element E.
E:first-child	Matches element E when E is the first child of its parent.
E:link	
E:visited	Matches element E if E is the source anchor of a hyperlink of which the target is not yet visited (:link) or already visited (:visited).
E:active	
E:hover	
E:focus	Matches E during certain user actions.
E:lang(c)	Matches element of type E if it is in (human) language c (the document language specifies how language is determined).

Selector Format

E + F	Matches any F element immediately preceded by a sibling element E.
E[foo]	Matches any E element with the "foo" attribute set (whatever the value).
E[foo="warning"]	Matches any E element whose "foo" attribute value is exactly equal to "warning".
E[foo~="warning"]	Matches any E element whose "foo" attribute value is a list of space-separated values, one of which is exactly equal to "warning".
E[lang ="en"]	Matches any E element whose "lang" attribute has a hyphen-separated list of values beginning (from the left) with "en".
DIV.Warning	(In HTML, the same as DIV[class~="warning"].)
E#myid	Matches any E element with ID equal to "myid".

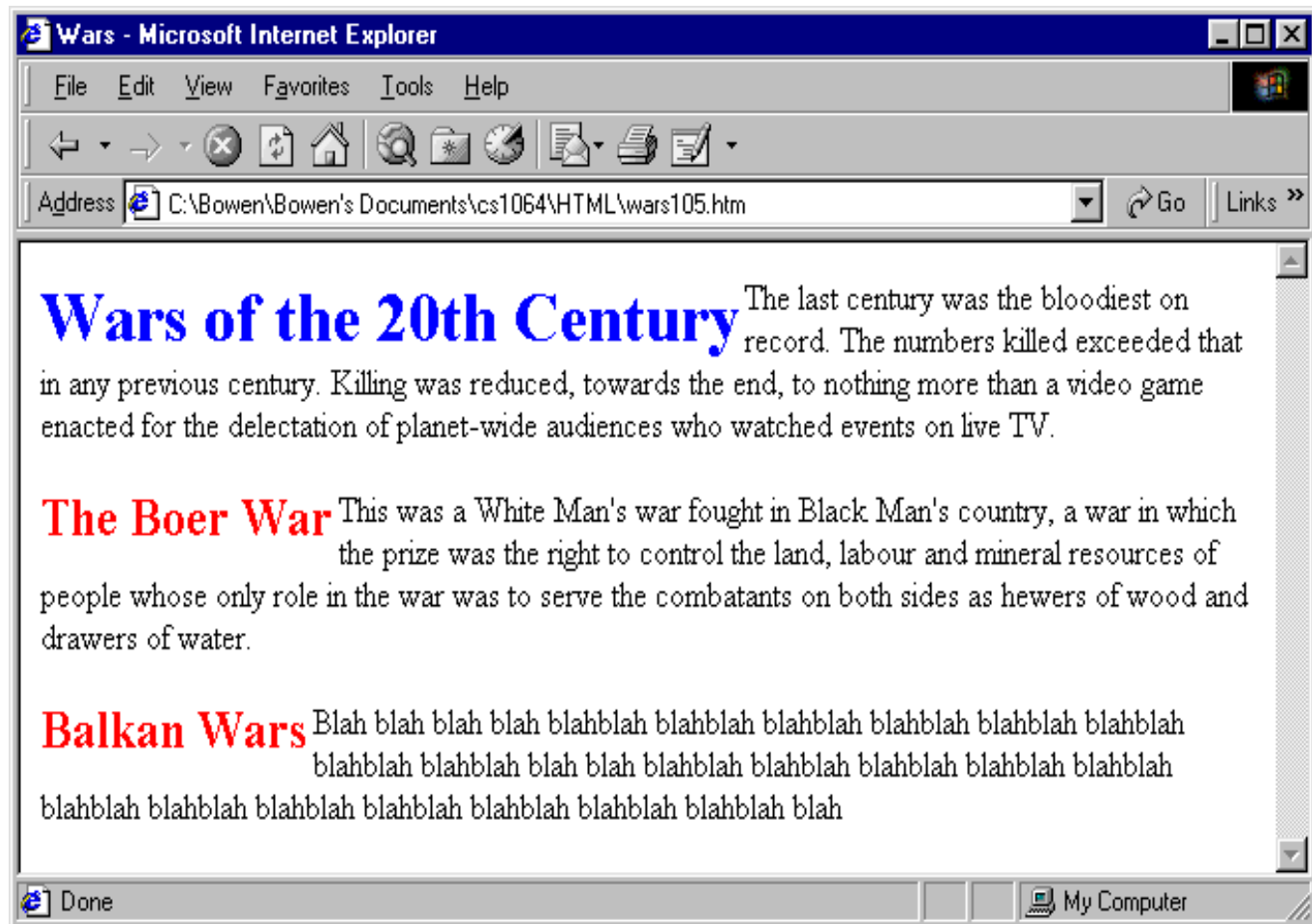
A selector's specificity is calculated as follows:

- count 1 if the selector is a 'style' attribute rather than a selector, 0 otherwise (=a) (In HTML, values of an element's "style" attribute are style sheet rules. These rules have no selectors, so a=1, b=0, c=0, and d=0.)
- count the number of ID attributes in the selector (= b)
- count the number of other attributes and pseudo-classes in the selector (= c)
- count the number of element names and pseudo-elements in the selector (= d)

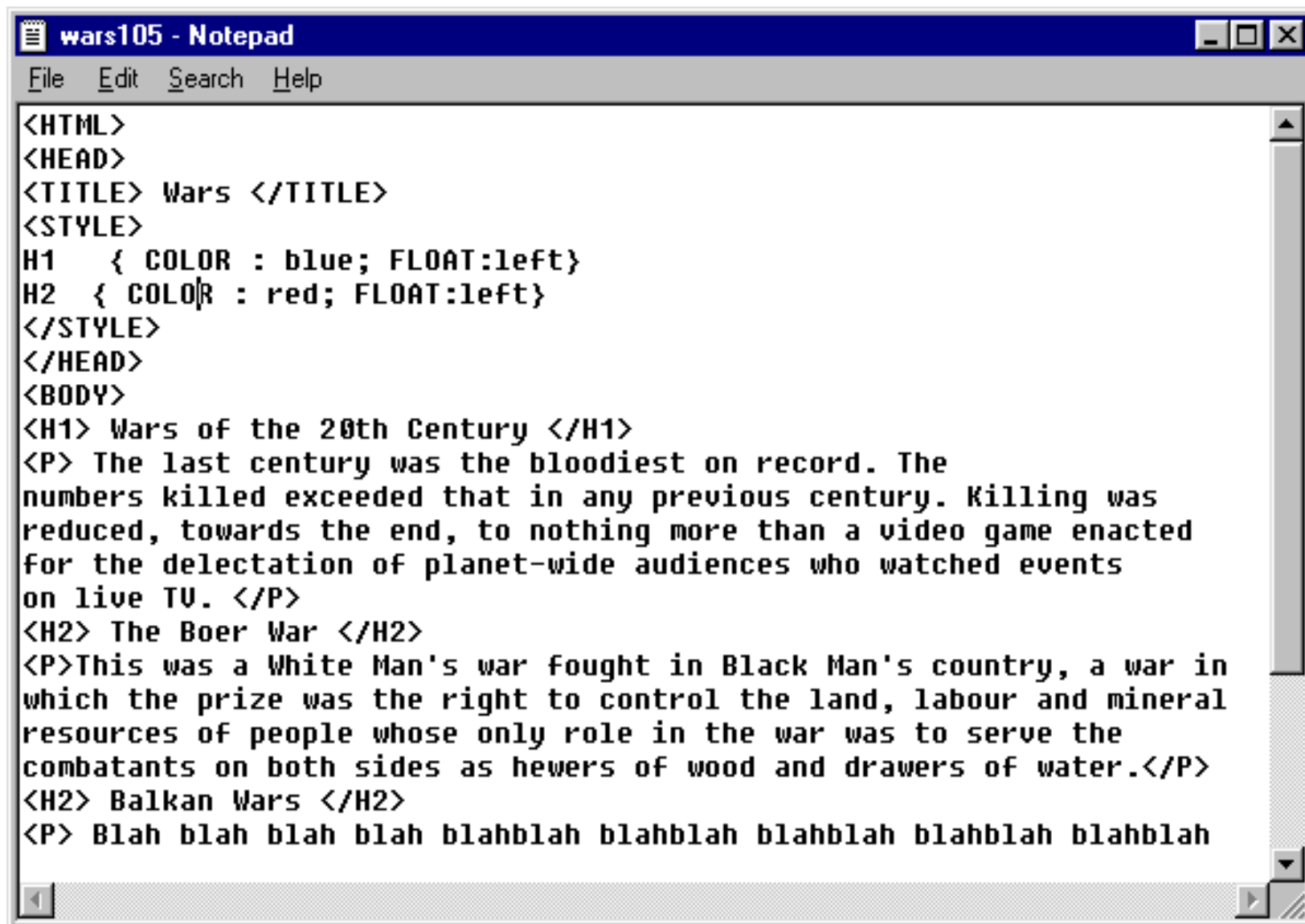
*	{}	/* a=0 b=0 c=0 d=0 -> specificity = 0,0,0,0 */
li	{}	/* a=0 b=0 c=0 d=1 -> specificity = 0,0,0,1 */
li:first-line	{}	/* a=0 b=0 c=0 d=2 -> specificity = 0,0,0,2 */
ul li	{}	/* a=0 b=0 c=0 d=2 -> specificity = 0,0,0,2 */
ul ol+li	{}	/* a=0 b=0 c=0 d=3 -> specificity = 0,0,0,3 */
h1 + *[rel=up]	{}	/* a=0 b=0 c=1 d=1 -> specificity = 0,0,1,1 */
ul ol li.red	{}	/* a=0 b=0 c=1 d=3 -> specificity = 0,0,1,3 */
li.red.level	{}	/* a=0 b=0 c=2 d=1 -> specificity = 0,0,2,1 */
#x34y	{}	/* a=0 b=1 c=0 d=0 -> specificity = 0,1,0,0 */
style=""		/* a=1 b=0 c=0 d=0 -> specificity = 1,0,0,0 */

Letting Elements Float

- We can let elements float
- Consider the way that the headers float in picture below

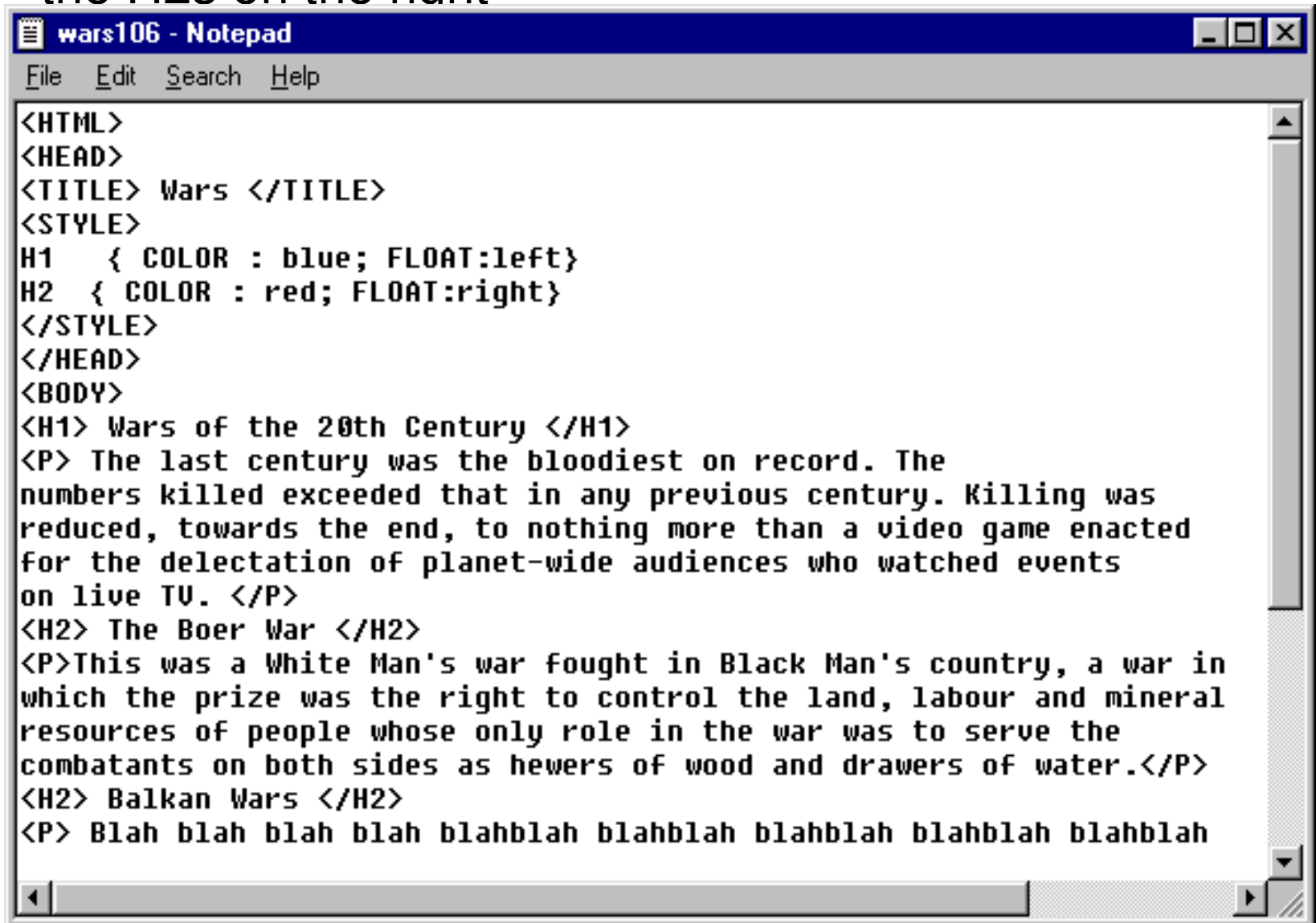


- This achieved by using the FLOAT attribute in the styles specified for the headers
- This attribute takes three values: **left**, **right**, and **none**
- **none** is the default value
- In the previous slide, the headers were allowed to float on the left, as we can see from the document specification on the next slide

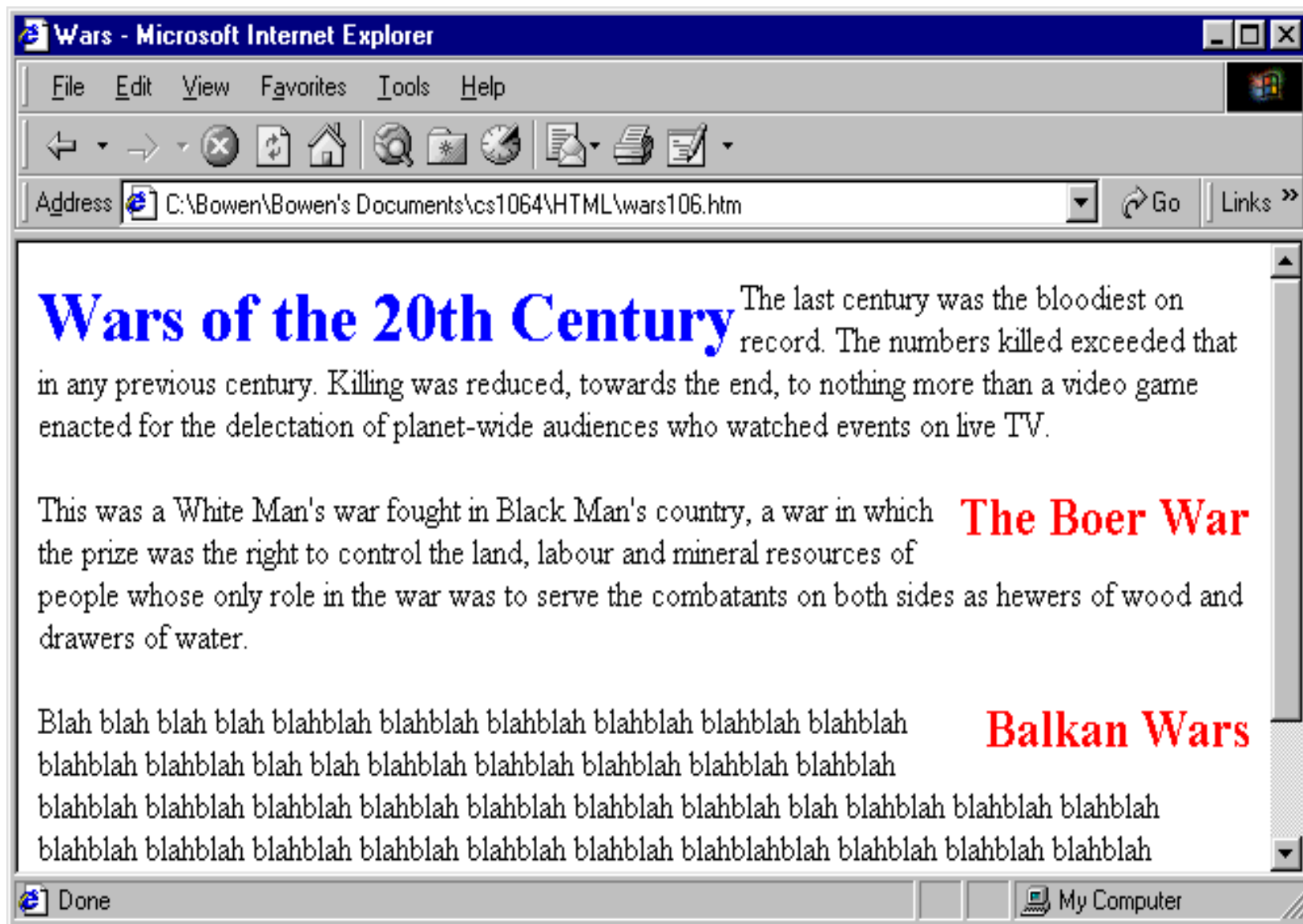


```
<HTML>
<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { COLOR : blue; FLOAT:left}
H2 { COLOR : red; FLOAT:left}
</STYLE>
</HEAD>
<BODY>
<H1> Wars of the 20th Century </H1>
<P> The last century was the bloodiest on record. The
numbers killed exceeded that in any previous century. Killing was
reduced, towards the end, to nothing more than a video game enacted
for the delectation of planet-wide audiences who watched events
on live TV. </P>
<H2> The Boer War </H2>
<P>This was a White Man's war fought in Black Man's country, a war in
which the prize was the right to control the land, labour and mineral
resources of people whose only role in the war was to serve the
combatants on both sides as hewers of wood and drawers of water.</P>
<H2> Balkan Wars </H2>
<P> Blah blah blah blah blahblah blahblah blahblah blahblah
```

- Here we specify that the H1s will float on the left and the H2s on the right

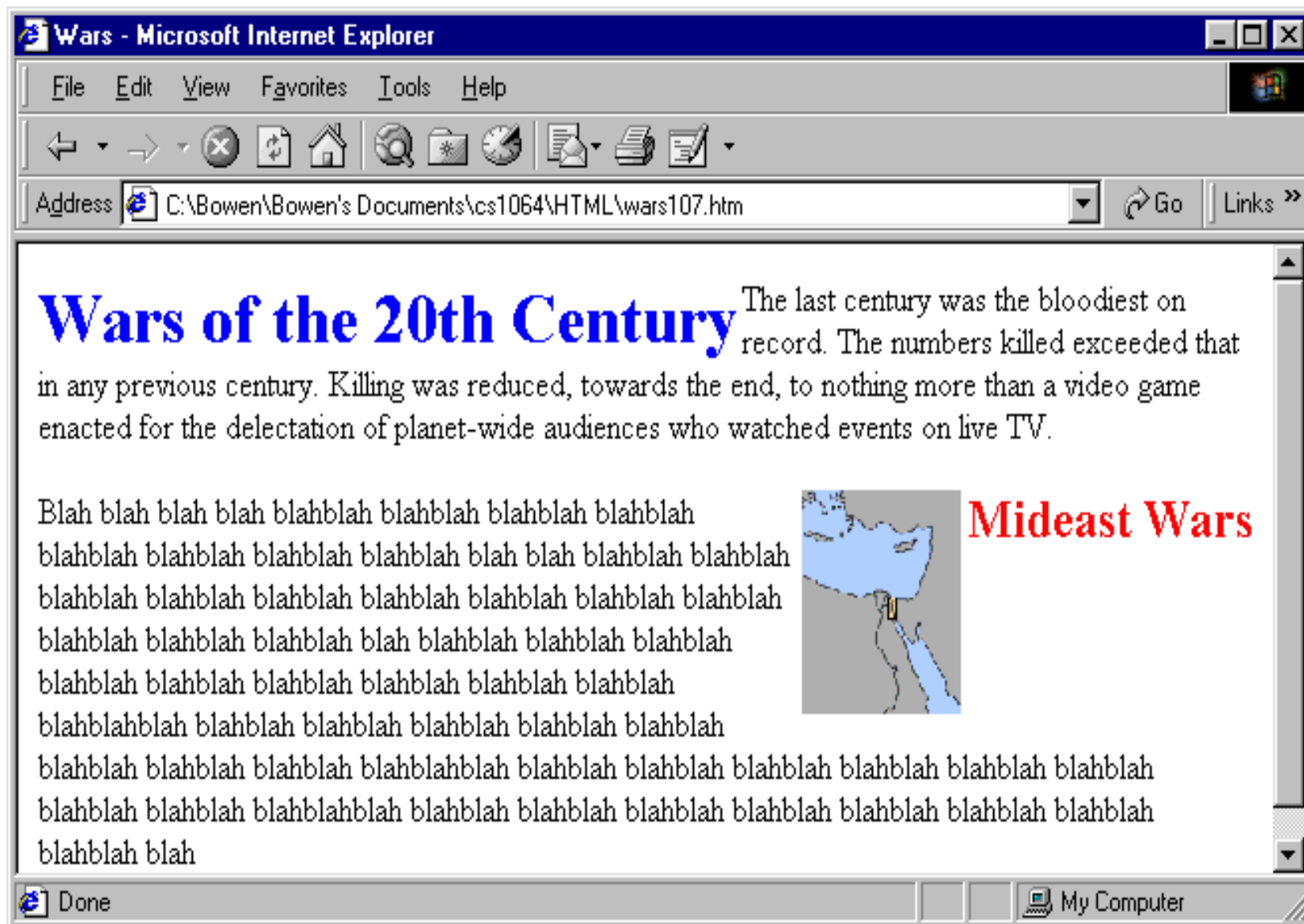


```
<HTML>
<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1  { COLOR : blue; FLOAT:left}
H2  { COLOR : red; FLOAT:right}
</STYLE>
</HEAD>
<BODY>
<H1> Wars of the 20th Century </H1>
<P> The last century was the bloodiest on record. The
numbers killed exceeded that in any previous century. Killing was
reduced, towards the end, to nothing more than a video game enacted
for the delectation of planet-wide audiences who watched events
on live TV. </P>
<H2> The Boer War </H2>
<P>This was a White Man's war fought in Black Man's country, a war in
which the prize was the right to control the land, labour and mineral
resources of people whose only role in the war was to serve the
combatants on both sides as hewers of wood and drawers of water.</P>
<H2> Balkan Wars </H2>
<P> Blah blah blah blah blahblah blahblah blahblah blahblah
```



- We can use the FLOAT attribute on any element
- For example, the next document uses FLOAT = right

```
wars107 - Notepad
File Edit Search Help
<HTML>
<HEAD>
<TITLE> Wars </TITLE>
<STYLE>
H1 { COLOR : blue; FLOAT:left}
H2 { COLOR : red; FLOAT:right}
IMG { FLOAT : right ; HEIGHT : 100; WIDTH : 80 }
</STYLE>
</HEAD>
<BODY>
<H1> Wars of the 20th Century </H1>
<P> The last century was the bloodiest on record. The
numbers killed exceeded that in any previous century. Killing was
reduced, towards the end, to nothing more than a video game enacted
for the delectation of planet-wide audiences who watched events
on live TV. </P>
<H2> Mideast Wars </H2>
<IMG SRC="MIDEAST.gif">
<P> Blah blah blah blah blahblah blahblah blahblah blahblah blahblah
blahblah blahblah blahblah blah blah blahblah blahblah blahblah
blahblah blahblah blahblah blahblah blahblah blahblah blahblah
blahblah blahblah blah blahblah blahblah blahblah blahblah blahblah bla
```



background

background-color

color-rgb

color-hex

color-name

transparent

background-image

url(URL)

none

background-repeat

repeat

repeat-x

repeat-y

no-repeat

background-attachment

scroll

fixed

background-position

top left

top center

top right

center left

center center

center right

bottom left

bottom center

bottom right

x% y%

xpos ypos

Background properties

color	<i>color</i>
direction	ltr rtl
line-height	normal <i>number</i> <i>length</i> <i>%</i>
letter-spacing	normal <i>length</i>
text-align	left right center justify
text-decoration	none underline overline line-through blink
text-indent	<i>length</i> <i>%</i>
text-shadow	none <i>color</i> <i>length</i>

Text Properties

text-transform

none

capitalize

uppercase

lowercase

white-space

normal

pre

nowrap

word-spacing

normal

length

vertical-align

baseline

sub

super

top

text-top

middle

bottom

text-bottom

inherit

length

%

Text Properties

Font

font-style

font-variant

font-weight

font-size/line-height

font-family

caption

icon

menu

message-box

small-caption

status-bar

font-family	<i>family-name</i> <i>generic-family</i>
--------------------	---

font-size	xx-small x-small small medium large x-large xx-large smaller larger <i>length</i> <i>%</i>
------------------	---

font-size-adjust	none <i>number</i>
-------------------------	------------------------------

Font Properties

font-stretch	normal	font-weight	normal
	wider		bold
	narrower		bolder
	ultra-condensed		lighter
	extra-condensed		100
	condensed		200
	semi-condensed		300
	semi-expanded		400
	expanded		500
	extra-expanded		600
font-style	ultra-expanded	font-weight	700
			800
			900
font-style	normal		
	italic		
	oblique		
font-variant			
font-variant	normal		
	small-caps		

Font Properties

border

border-width

thin

medium

thick

length

border-style

none

hidden

dotted

dashed

solid

double

groove

ridge

inset

outset

border-color

color

Border properties

border-left **border-left-color**
border-left-width
border-left-style

border-top **border-top-color**
border-top-width
border-top-style

border-left **border-left-color**
border-left-width
border-left-style

border-bottom **border-bottom-color**
border-bottom-width
border-bottom-style

outline

outline-color

color
invert

outline-style

none
dotted
dashed
solid
double
groove
ridge
inset
outset

outline-width

thin
medium
thick
length

Outline Property

margin

margin-top
margin-right
margin-bottom
margin-left

auto
length
%

Margin Property

pading

pading-top
pading -right
pading -bottom
pading -left

length
%

Padding Property

list-style

list-style-type

none
disc
circle
square
decimal
decimal-leading-zero
lower-roman
upper-roman
lower-alpha
upper-alpha
lower-greek
lower-latin
upper-latin

list-style-position

inside
outside

list-style-image

none
url

List Property

border-collapse

collapse
separate

border-spacing

length [length]

caption-side

top
bottom
left
right

empty-cells

show
hide

table-layout

auto
fixed

Table Properties

height	auto <i>length</i> %
line-height	normal <i>number</i> <i>length</i> %
max-height	none <i>length</i> %
max-width	none <i>length</i> %
min-height	<i>length</i> %
min-width	<i>length</i> %
width	auto <i>length</i> %

Dimensions Properties

clear

left

right

both

none

cursor

url

auto

crosshair

default

pointer

move

e-resize

ne-resize

nw-resize

n-resize

se-resize

sw-resize

s-resize

w-resize

text

wait

help

Positioning Properties

position

static
relative
absolute
fixed

top
bottom
right
left

clip
overflow

z-index

visibility

Positioning Properties

Absolute length units are only useful when the physical properties of the output medium are known. The absolute units are:

1. **in**: inches — 1 inch is equal to 2.54 centimeters.
2. **cm**: centimeters
3. **mm**: millimeters
4. **pt**: points — the points used by CSS 2.1 are equal to 1/72nd of an inch.
5. **pc**: picas — 1 pica is equal to 12 points.

Percentage values are always relative to another value, for example a length.

Each property that allows percentages also defines the value to which the percentage refers.

The value may be that of another property for the same element, a property for an ancestor element, or a value of the formatting context (e.g., the width of a containing block).

When a percentage value is set for a property of the root element and the percentage is defined as referring to the inherited value of some property, the resultant value is the percentage times the initial value of that property.

pseudo-element

```
p:first-line { text-transform: uppercase }
```

```
p:first-letter { font-size: 3em; font-weight: normal }
```

```
h1:before {content: counter(chapno, upper-roman) ". "}
```


The '*@import*' rule allows users to import style rules from other style sheets.

Any *@import* rules must precede all rule sets in a style sheet. The '*@import*' keyword must be followed by the URI of the style sheet to include. A string is also allowed; it will be interpreted as if it had `url(...)` around it.

Example(s):

The following lines are equivalent in meaning and illustrate both '*@import*' syntaxes (one with "`url()`" and one with a bare string):

```
@import "mystyle.css";  
@import url("mystyle.css");
```

```
<LINK REL="stylesheet" TYPE="text/css" MEDIA="print,  
handheld" HREF="foo.css">
```

```
@media print {  
    body { font-size: 10pt }
```

```
    . . .
```

```
}
```

```
@media screen {  
    body { font-size: 13px }
```

```
    . . .
```

```
}
```

```
@media screen, print {  
    body { line-height: 1.2 }
```

```
    . . .
```

```
}
```

```
@import url("fineprint.css") print;  
@import url("bluish.css") projection, tv;
```

all

Suitable for all devices.

braille

Intended for braille tactile feedback devices.

embossed

Intended for paged braille printers.

handheld

Intended for handheld devices (typically small screen, limited bandwidth).

print

Intended for paged material and for documents viewed on screen in print preview mode.

projection

Intended for projected presentations, for example projectors.

screen

Intended primarily for color computer screens.

speech

Intended for speech synthesizers. Note: CSS2 had a similar media type called 'aural' for this purpose.

tty

Intended for media using a fixed-pitch character grid (such as teletypes, terminals, or portable devices with limited display capabilities). Authors should not use pixel units with the "tty" media type.

tv

Intended for television-type devices (low resolution, color, limited-scrollability screens, sound available).