Introduction to CSS

Introduction to CSS

CSS stands for Cascading Style Sheets. It is a language designed to specify the style, layout, and appearance of various elements on webpages.CSS covers fonts, colors, margins, lines, height, width, background images, advanced positions and many other things.

It allows one to adapt the presentation to different types of devices, such as large screens, small screens, or printers. The separation of HTML from CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments.

Introduction to CSS cont…

CSS was first developed in 1997, as a way for Web developers to define the look and feel of their Web pages.
It was intended to allow developers to separate content from design so that HTML could perform more of the function that it was originally based on - describes the structure of a website , without worry about the design and layout.
CSS didn't gain in popularity until around 2000, when Web browsers began using more than the basic font and color aspects of CSS. CSS is supported by all browsers today.

CSS PROPERTIES

Text Properties

Property	Description	Values
color	Sets the color of a text	RGB, hex, keyword
line-height	Sets the distance between lines	normal, number, length, %
letter-spacing	Increase or decrease the space between characters	normal, length
text-align	Aligns the text in an element	left, right, center, justify
text-decoration	Adds decoration to text	none, underline, overline, line-through
text-indent	Indents the first line of text in an element	length, %
text-transform	Controls the letters in an element	none, capitalize, uppercase, lowercase

CSS PROPERTIES

List Properties

Property	Description	Values
list-style	Sets all the properties for a list in one declaration	list-style-type, list-style-position, list-style-image, inherit
list-style-image	Specifies an image as the list-item marker	URL, none, inherit
list-style-position	Specifies where to place the list-item marker	inside, outside, inherit
list-style-type	Specifies the type of list-item marker	none, disc, circle, square, decimal, decimal-leading-zero, armenian, georgian, lower-alpha, upper-alpha, lower-greek, lower-latin, upper-latin, lower-roman, upper-roman, inherit

CSS PROPERTIES

Border Properties

Property	Description	Values
border	Sets all the border properties in one declaration	border-width, border-style, border-color
border-bottom	Sets all the bottom border properties in one declaration	border-bottom-width, border-bottom-style, border-bottom-color
border-bottom-color	Sets the color of the bottom border	border-color
border-bottom-style	Sets the style of the bottom border	border-style
border-bottom-width	Sets the width of the bottom border	border-width
border-color	Sets the color of the four borders	color_name, hex_number, rgb_number, transparent, inherit
border-left	Sets all the left border properties in one declaration	border-left-width, border-left-style, border-left-color
border-left-color	Sets the color of the left border	border-color
border-left-style	Sets the style of the left border	border-style

CSS PROPERTIES

Border Properties

border-left-width	Sets the width of the left border	border-width
border-right	Sets all the right border properties in one declaration	border-right-width, border-right-style, border-right-color
border-right-color	Sets the color of the right border	border-color
border-right-style	Sets the style of the right border	border-style
border-right-width	Sets the width of the right border	border-width
border-style	Sets the style of the four borders	none, hidden, dotted, dashed, solid, double, groove, ridge, inset, outset, inherit
border-top	Sets all the top border properties in one declaration	border-top-width, border-top-style, border-top-color
border-top-color	Sets the color of the top border	border-color
border-top-style	Sets the style of the top border	border-style
border-top-width	Sets the width of the top border	border-width
border-width	Sets the width of the four borders	thin, medium, thick, length, inherit

CSS PROPERTIES

Font Properties

Property	Description	Values
font	Sets all the font properties in one declaration	font-style, font-variant, font-weight, font-size/line-height, font-family, caption, icon, menu, message-box, small-caption, status-bar, inherit
font-family	Specifies the font family for text	family-name, generic-family, inherit
font-size	Specifies the font size of text	xx-small, x-small, small, medium, large, x-large, xx-large, smaller, larger, length, %, inherit
font-style	Specifies the font style for text	normal, italic, oblique, inherit
font-variant	Specifies whether or not a text should be displayed in a small-caps font	normal, small-caps, inherit
font-weight	Specifies the weight of a font	normal, bold, bolder, lighter, 100, 200, 300, 400, 500, 600, 700, 800, 900, inherit Careful, many of these are not supported!

CSS PROPERTIES

align-content	Specifies the alignment between the lines inside a flexible container when the items do not use all available space
align-items	Specifies the alignment for items inside a flexible container
align-self	Specifies the alignment for selected items inside a flexible container
all	Resets all properties (except unicode-bidi and direction)
animation	A shorthand property for all the animation properties
animation-delay	Specifies a delay for the start of an animation
animation-direction	Specifies whether an animation should be played forwards, backwards or in alternate cycles
animation-duration	Specifies how long an animation should take to complete one cycle
animation-fill-mode	Specifies a style for the element when the animation is not playing (before it starts, after it ends, or both)
animation-iteration-count	Specifies the number of times an animation should be played
animation-name	Specifies a name for the @keyframes animation
animation-play-state	Specifies whether the animation is running or paused
animation-timing-function	Specifies the speed curve of an animation

CSS PROPERTIES

backface-visibility	Defines whether or not the back face of an element should be visible when facing the user
background	A shorthand property for setting all the background properties in one declaration
background-attachment	Sets whether a background image scrolls with the rest of the page, or is fixed
background-blend-mode	Specifies the blending mode of each background layer (color/image)
background-clip	Defines how far the background (color or image) should extend within an element
background-color	Specifies the background color of an element
background-image	Specifies one or more background images for an element
background-origin	Specifies the origin position of a background image
background-position	Specifies the position of a background image
background-repeat	Sets if/how a background image will be repeated
background-size	Specifies the size of the background images

CSS PROPERTIES

border	A shorthand property for border-width, border-style and border-color
border-bottom	A shorthand property for setting all the bottom border properties in one declaration
border-bottom-color	Sets the color of the bottom border
border-bottom-left-radius	Defines the radius of the border of the bottom-left corner
border-bottom-right-radius	Defines the radius of the border of the bottom-right corner
border-bottom-style	Sets the style of the bottom border
border-bottom-width	Sets the width of the bottom border
border-collapse	Sets whether table borders should collapse into a single border or be separated
border-color	Sets the color of the four borders
border-image	A shorthand property for setting all the border-image-* properties
border-image-outset	Specifies the amount by which the border image area extends beyond the border box
border-image-repeat	Specifies whether the border image should be repeated, rounded or stretched
border-image-slice	Specifies how to slice the border image
border-image-source	Specifies the path to the image to be used as a border

CSS PROPERTIES

border-image-width	Specifies the width of the border image
border-left	A shorthand property for setting all the left border properties in one declaration
border-left-color	Sets the color of the left border
border-left-style	Sets the style of the left border
border-left-width	Sets the width of the left border
border-radius	A shorthand property for setting all the four border-*-radius properties
border-right	A shorthand property for setting all the right border properties in one declaration
border-right-color	Sets the color of the right border
border-right-style	Sets the style of the right border
border-right-width	Sets the width of the right border
border-spacing	Sets the distance between the borders of adjacent cells
border-style	Sets the style of the four borders
border-top	A shorthand property for setting all the top border properties in one declaration

CSS PROPERTIES

border-top-color	Sets the color of the top border
border-top-left-radius	Defines the radius of the border of the top-left corner
border-top-right-radius	Defines the radius of the border of the top-right corner
border-top-style	Sets the style of the top border
border-top-width	Sets the width of the top border
border-width	Sets the width of the four borders
bottom	Sets the elements position, from the bottom of its parent element
box-decoration-break	Sets the behavior of the background and border of an element at page-break, or, for in-line elements, at line-break.
box-shadow	Attaches one or more shadows to an element
box-sizing	Defines how the width and height of an element are calculated: should they include padding and borders, or not
break-after	Specifies the page-, column-, or region-break behavior after the generated box
break-before	Specifies the page-, column-, or region-break behavior before the generated box
break-inside	Specifies the page-, column-, or region-break behavior inside the generated box

CSS PROPERTIES

caption-side	Specifies the placement of a table caption
caret-color	Specifies the color of the cursor (caret) in inputs, textareas, or any element that is editable
@charset	Specifies the character encoding used in the style sheet
clear	Specifies on which sides of an element floating elements are not allowed to float
clip	Clips an absolutely positioned element
color	Sets the color of text
column-count	Specifies the number of columns an element should be divided into
column-fill	Specifies how to fill columns, balanced or not
column-gap	Specifies the gap between the columns
column-rule	A shorthand property for setting all the column-rule-* properties
column-rule-color	Specifies the color of the rule between columns
column-rule-style	Specifies the style of the rule between columns


Recycle Bin
a
c
b
chris
DIGITAL DREAMS
ICT ACADEMY
HOT LINE: 07064286765

CSS PROPERTIES

column-rule-width	Specifies the width of the rule between columns
column-span	Specifies how many columns an element should span across
column-width	Specifies the column width
columns	A shorthand property for setting column-width and column-count
content	Used with the :before and :after pseudo-elements, to insert generated content
counter-increment	Increases or decreases the value of one or more CSS counters
counter-reset	Creates or resets one or more CSS counters
cursor	Specifies the mouse cursor to be displayed when pointing over an element

CSS PROPERTIES

direction	Specifies the text direction/writing direction
display	Specifies how a certain HTML element should be displayed
empty-cells	Specifies whether or not to display borders and background on empty cells in a table
filter	Defines effects (e.g. blurring or color shifting) on an element before the element is displayed
flex	A shorthand property for the flex-grow, flex-shrink, and the flex-basis properties
flex-basis	Specifies the initial length of a flexible item
flex-direction	Specifies the direction of the flexible items
flex-flow	A shorthand property for the flex-direction and the flex-wrap properties
flex-grow	Specifies how much the item will grow relative to the rest
flex-shrink	Specifies how the item will shrink relative to the rest
flex-wrap	Specifies whether the flexible items should wrap or not
float	Specifies whether or not a box should float
font	Sets all the font properties in one declaration

CSS PROPERTIES

@font-face	A rule that allows websites to download and use fonts other than the "web-safe" fonts
font-family	Specifies the font family for text
font-feature-settings	Allows control over advanced typographic features in OpenType fonts
@font-feature-values	Allows authors to use a common name in font-variant-alternate for feature activated differently in OpenType
font-kerning	Controls the usage of the kerning information (how letters are spaced)
font-language-override	Controls the usage of language-specific glyphs in a typeface
font-size	Specifies the font size of text
font-size-adjust	Preserves the readability of text when font fallback occurs
font-stretch	Selects a normal, condensed, or expanded face from a font family
font-style	Specifies the font style for text
font-synthesis	Controls which missing typefaces (bold or italic) may be synthesized by the browser

CSS PROPERTIES

font-variant	Specifies whether or not a text should be displayed in a small-caps font
font-variant-alternates	Controls the usage of alternate glyphs associated to alternative names defined in @font-feature-values
font-variant-caps	Controls the usage of alternate glyphs for capital letters
font-variant-east-asian	Controls the usage of alternate glyphs for East Asian scripts (e.g Japanese and Chinese)
font-variant-ligatures	Controls which ligatures and contextual forms are used in textual content of the elements it applies to
font-variant-numeric	Controls the usage of alternate glyphs for numbers, fractions, and ordinal markers
font-variant-position	Controls the usage of alternate glyphs of smaller size positioned as superscript or subscript regarding the baseline of the font
font-weight	Specifies the weight of a font

CSS PROPERTIES

grid	A shorthand property
grid-area	Either specifies a name for the grid item, or this property is a shorthand property for the grid-row-start, grid-column-start, grid-row-end, and grid-column-end properties
grid-auto-columns	Specifies a default column size
grid-auto-flow	Specifies how auto-placed items are inserted in the grid
grid-auto-rows	Specifies a default row size
grid-column	A shorthand property for the grid-column-start and the grid-column-end properties
grid-column-end	Specifies where to end the grid item
grid-column-gap	Specifies the size of the gap between columns
grid-column-start	Specifies where to start the grid item
grid-gap	A shorthand property for the grid-row-gap and grid-column-gap properties

CSS PROPERTIES

grid-row	A shorthand property for the grid-row-start and the grid-row-end properties
grid-row-end	Specifies where to end the grid item
grid-row-gap	Specifies the size of the gap between rows
grid-row-start	Specifies where to start the grid item
grid-template	A shorthand property for the grid-template-rows, grid-template-columns and grid-areas properties
grid-template-areas	Specifies how to display columns and rows, using named grid items
grid-template-columns	Specifies the size of the columns, and how many columns in a grid layout
grid-template-rows	Specifies the size of the rows in a grid layout

CSS PROPERTIES

hanging-punctuation	Specifies whether a punctuation character may be placed outside the line box
height	Sets the height of an element
hyphens	Sets how to split words to improve the layout of paragraphs
image-orientation	Specifies a rotation in the right or clockwise direction that a user agent applies to an image (This property is likely going to be deprecated and its functionality moved to HTML)
image-rendering	Gives a hint to the browser about what aspects of an image are most important to preserve when the image is scaled
image-resolution	Specifies the intrinsic resolution of all raster images used in/on the element
@import	Allows you to import a style sheet into another style sheet

CSS PROPERTIES

justify-content	Specifies the alignment between the items inside a flexible container when the items do not use all available space
@keyframes	Specifies the animation code
left	Specifies the left position of a positioned element
letter-spacing	Increases or decreases the space between characters in a text
line-break	Specifies how/if to break lines
line-height	Sets the line height
list-style	Sets all the properties for a list in one declaration
list-style-image	Specifies an image as the list-item marker
list-style-position	Specifies the position of the list-item markers (bullet points)
list-style-type	Specifies the type of list-item marker

CSS PROPERTIES

margin	Sets all the margin properties in one declaration
margin-bottom	Sets the bottom margin of an element
margin-left	Sets the left margin of an element
margin-right	Sets the right margin of an element
margin-top	Sets the top margin of an element
max-height	Sets the maximum height of an element
max-width	Sets the maximum width of an element
@media	Sets the style rules for different media types/devices/sizes
min-height	Sets the minimum height of an element
min-width	Sets the minimum width of an element

CSS PROPERTIES

object-fit	Specifies how the contents of a replaced element should be fitted to the box established by its used height and width
object-position	Specifies the alignment of the replaced element inside its box
opacity	Sets the opacity level for an element
order	Sets the order of the flexible item, relative to the rest
orphans	Sets the minimum number of lines that must be left at the bottom of a page when a page break occurs inside an element
outline	Sets all the outline properties in one declaration
outline-color	Sets the color of an outline
outline-offset	Offsets an outline, and draws it beyond the border edge
outline-style	Sets the style of an outline
outline-width	Sets the width of an outline
overflow	Specifies what happens if content overflows an element's box
overflow-wrap	Specifies whether or not the browser may break lines within words in order to prevent overflow (when a string is too long to fit its containing box)
overflow-x	Specifies whether or not to clip the left/right edges of the content, if it overflows the element's content area
overflow-y	Specifies whether or not to clip the top/bottom edges of the content, if it overflows the element's content area

CSS PROPERTIES

padding	Sets all the padding properties in one declaration
padding-bottom	Sets the bottom padding of an element
padding-left	Sets the left padding of an element
padding-right	Sets the right padding of an element
padding-top	Sets the top padding of an element
page-break-after	Sets the page-breake behavior after an element
page-break-before	Sets the page-breake behavior before an element
page-break-inside	Sets the page-breake behavior inside an element
perspective	Gives a 3D-positioned element some perspective
perspective-origin	Defines at which position the user is looking at the 3D-positioned element
pointer-events	Defines whether or not an element reacts to pointer events
position	Specifies the type of positioning method used for an element (static, relative, absolute or fixed)

CSS PROPERTIES

quotes	Sets the type of quotation marks for embedded quotations
resize	Defines if (and how) an element is resizable by the user
right	Specifies the right position of a positioned element
tab-size	Specifies the width of a tab character
table-layout	Defines the algorithm used to lay out table cells, rows, and columns
text-align	Specifies the horizontal alignment of text
text-align-last	Describes how the last line of a block or a line right before a forced line break is aligned when text-align is "justify"
text-combine-upright	Specifies the combination of multiple characters into the space of a single character
text-decoration	Specifies the decoration added to text
text-decoration-color	Specifies the color of the text-decoration
text-decoration-line	Specifies the type of line in a text-decoration
text-decoration-style	Specifies the style of the line in a text decoration
text-indent	Specifies the indentation of the first line in a text-block

CSS PROPERTIES

text-justify	Specifies the justification method used when text-align is "justify"
text-orientation	Defines the orientation of the text in a line
text-overflow	Specifies what should happen when text overflows the containing element
text-shadow	Adds shadow to text
text-transform	Controls the capitalization of text
text-underline-position	Specifies the position of the underline which is set using the text-decoration property
top	Specifies the top position of a positioned element
transform	Applies a 2D or 3D transformation to an element
transform-origin	Allows you to change the position on transformed elements
transform-style	Specifies how nested elements are rendered in 3D space
transition	A shorthand property for setting the four transition properties
transition-delay	Specifies when the transition effect will start
transition-duration	Specifies how many seconds or milliseconds a transition effect takes to complete
transition-property	Specifies the name of the CSS property the transition effect is for
transition-timing-function	Specifies the speed curve of the transition effect

CSS PROPERTIES

unicode-bidi	Used together with the direction property to set or return whether the text should be overridden to support multiple languages in the same document
user-select	Specifies whether the text of an element can be selected
vertical-align	Sets the vertical alignment of an element
visibility	Specifies whether or not an element is visible
white-space	Specifies how white-space inside an element is handled
widows	Sets the minimum number of lines that must be left at the top of a page when a page break occurs inside an element
width	Sets the width of an element
word-break	Specifies line breaking rules for non-CJK scripts
word-spacing	Increases or decreases the space between words in a text
word-wrap	Allows long, unbreakable words to be broken and wrap to the next line
z-index	Sets the stack order of a positioned element


Recycle Bin
a
c
b
chris
DIGITAL DREAMS
ICT ACADEMY
HOT LINE: 07064286765

CSS SELECTORS

In CSS, selectors are used to target the HTML elements on our web pages that we want to style. There are a wide variety of CSS selectors available, allowing for fine grained precision when selecting elements to style.

Simple selectors

Type selector Selects all elements that match the given node name.

Syntax: *eltname*

Example: input will match any <input> element. Class selector Selects all elements that have the given class attribute.

```
input {.....}
```

Syntax: *.classname*

Example: .index will match any element that has a class of "index". ID selector Selects an element based on the value of its id attribute. There should be only one element with a given ID in a document.

```
.classname {.....}
```

Syntax: *#idname*

Example: #toc will match the element that has the ID "toc".

```
#toc {.....}
```

CSS SELECTORS

Universal selector Selects all elements. Optionally, it may be restricted to a specific namespace or to all namespaces.
Syntax: * ns|* *|*
Example: * will match all the elements of the document. Attribute selector Selects elements based on the value of the given attribute.

```
*{.....}
```

Syntax: [attr] [attr=value] [attr~=value] [attr|=value] [attr^=value] [attr$=value] [attr*=value]
Example: [autoplay] will match all elements that have the autoplay attribute set (to any value)

```
/* <a> elements with a title attribute */
a[title] { color: purple; }
```

CSS SELECTORS

Combinators

Adjacent sibling combinator : The + combinator selects adjacent siblings. This means that the second element directly follows the first, and both share the same parent.
Syntax: *A* + *B*
Example: h2 + p will match all <p> elements that directly follow an <h2>.

```
/* Paragraphs that come immediately after any image */
 img + p { font-style: bold; }
```

General sibling combinator : The ~ combinator selects siblings. This means that the second element follows the first (though not necessarily immediately), and both share the same parent.
Syntax: *A* ~ *B*
Example: p ~ span will match all elements that follow a <p>.

```
/* Paragraphs that are siblings of and subsequent to any image */
 img ~ p { color: red; }
```

CSS SELECTORS

Child combinator The > combinator selects nodes that are direct children of the first element.
Syntax: *A* > *B*
Example: ul > li will match all <li> elements that are nested directly inside a <ul> element.

```
/* List items that are children of the "my-things" list */
 ul.my-things > li { margin: 2em; }
```

Descendant combinator The combinator selects nodes that are descendants of the first element.
Syntax: A B
Example: div span will match all elements that are inside a element.

```
/* List items that are descendants of the "my-things" list */
ul.my-things li { margin: 2em; }
```

CSS SELECTORS

Pseudo-classes

Pseudo-classes allow the selection of elements based on state information that is not contained in the document tree. **Example:** a:visited will match all <a> elements that have been visited by the user.

```
div:hover { background-color: #F89B4D; }
```

Pseudo-elements

Pseudo-elements represent entities that are not included in HTML.
Example: p::first-line will match the first line of all <p> elements.

```
/* The first line of every <p> element. */
 p::first-line {
color: blue;
text-transform: uppercase;
 }
```

CSS SELECTORS

Selector	Example	Example description
.class	.intro	Selects all elements with class="intro"
#id	#firstname	Selects the element with id="firstname"
*	*	Selects all elements
element	p	Selects all <p> elements
element,element	div, p	Selects all elements and all <p> elements
element element	div p	Selects all <p> elements inside elements
element>element	div > p	Selects all <p> elements where the parent is a element
element+element	div + p	Selects all <p> elements that are placed immediately after elements
element1~element2	p ~ ul	Selects every element that are preceded by a <p> element

CSS SELECTORS

[attribute]	[target]	Selects all elements with a target attribute
[attribute=value]	[target=_blank]	Selects all elements with target="_blank"
[attribute~=value]	[title~=flower]	Selects all elements with a title attribute containing the word "flower"
[attribute|=value]	[lang|=en]	Selects all elements with a lang attribute value starting with "en"
[attribute^=value]	a[href^="https"]	Selects every <a> element whose href attribute value begins with "https"
[attribute$=value]	a[href$=".pdf"]	Selects every <a> element whose href attribute value ends with ".pdf"
[attribute*=value]	a[href*="w3schools"]	Selects every <a> element whose href attribute value contains the substring "w3schools"
:active	a:active	Selects the active link
::after	p::after	Insert something after the content of each <p> element
::before	p::before	Insert something before the content of each <p> element
:checked	input:checked	Selects every checked <input> element
:disabled	input:disabled	Selects every disabled <input> element

CSS SELECTORS

:empty	p:empty	Selects every <p> element that has no children (including text nodes)
:enabled	input:enabled	Selects every enabled <input> element
:first-child	p:first-child	Selects every <p> element that is the first child of its parent
::first-letter	p::first-letter	Selects the first letter of every <p> element
::first-line	p::first-line	Selects the first line of every <p> element
:first-of-type	p:first-of-type	Selects every <p> element that is the first <p> element of its parent
:focus	input:focus	Selects the input element which has focus
:hover	a:hover	Selects links on mouse over
:in-range	input:in-range	Selects input elements with a value within a specified range
:invalid	input:invalid	Selects all input elements with an invalid value
:lang(language)	p:lang(it)	Selects every <p> element with a lang attribute equal to "it" (Italian)

CSS SELECTORS

:last-child	p:last-child	Selects every <p> element that is the last child of its parent
:last-of-type	p:last-of-type	Selects every <p> element that is the last <p> element of its parent
:link	a:link	Selects all unvisited links
:not(selector)	:not(p)	Selects every element that is not a <p> element
:nth-child(n)	p:nth-child(2)	Selects every <p> element that is the second child of its parent
:nth-last-child(n)	p:nth-last-child(2)	Selects every <p> element that is the second child of its parent, counting from the last child
:nth-last-of-type(n)	p:nth-last-of-type(2)	Selects every <p> element that is the second <p> element of its parent, counting from the last child
:nth-of-type(n)	p:nth-of-type(2)	Selects every <p> element that is the second <p> element of its parent
:only-of-type	p:only-of-type	Selects every <p> element that is the only <p> element of its parent
:only-child	p:only-child	Selects every <p> element that is the only child of its parent

CSS SELECTORS

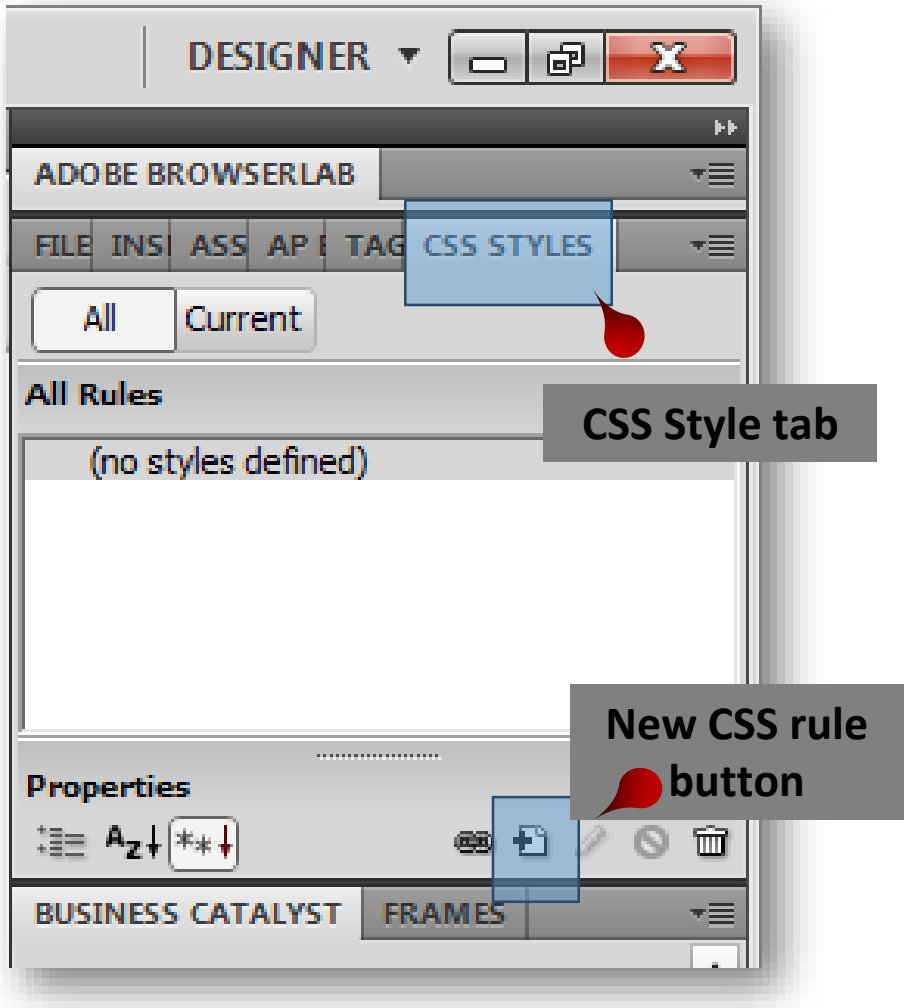
:optional	input:optional	Selects input elements with no "required" attribute
:out-of-range	input:out-of-range	Selects input elements with a value outside a specified range
:read-only	input:read-only	Selects input elements with the "readonly" attribute specified
:read-write	input:read-write	Selects input elements with the "readonly" attribute NOT specified
:required	input:required	Selects input elements with the "required" attribute specified
:root	:root	Selects the document's root element
::selection	::selection	Selects the portion of an element that is selected by a user
:target	#news:target	Selects the current active #news element (clicked on a URL containing that anchor name)
:valid	input:valid	Selects all input elements with a valid value
:visited	a:visited	Selects all visited links

Using CSS to create layout

- **Open** and **save** new html webpage in your website folder created with Dreamweaver.

- Click on **CSS style tab** on the left hand side of the Dreamweaver window

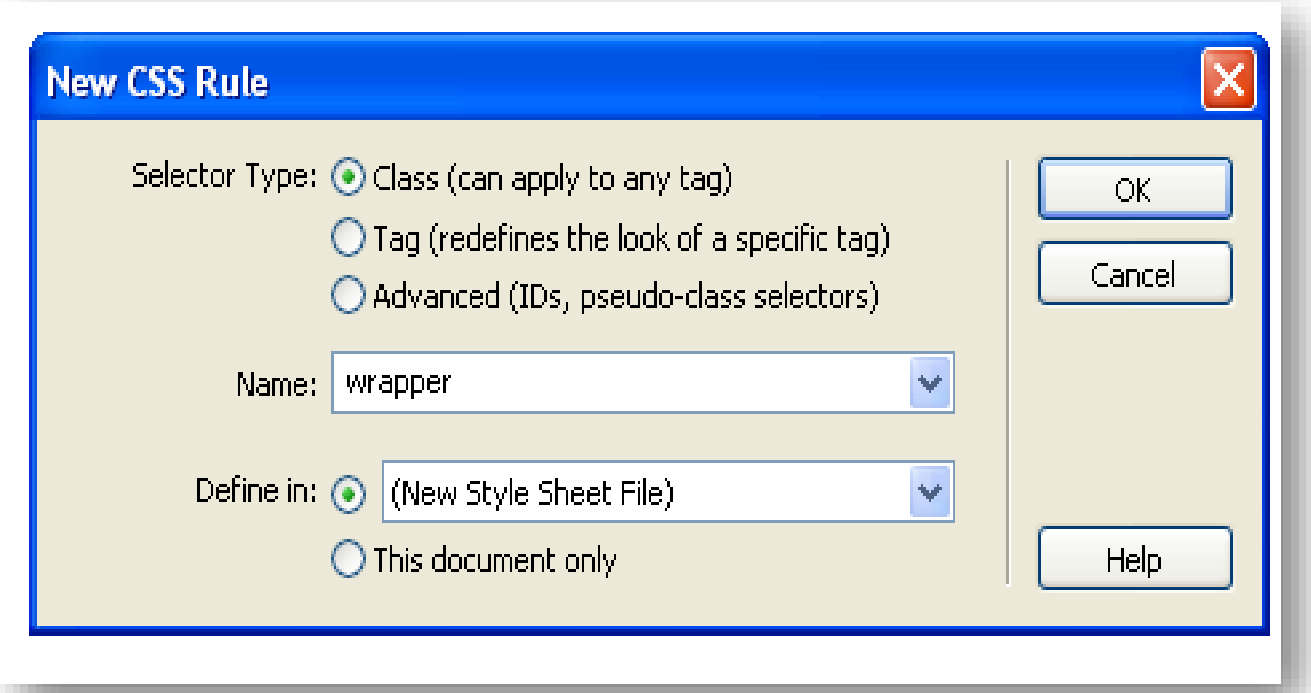
- Click on **new CSS Rule** button below.



Using CSS to create layout cont..

- Ensure that the new **CSS Rule dialogue box window** is set as shown below

- Under **name**, type **wrapper** and ok it.

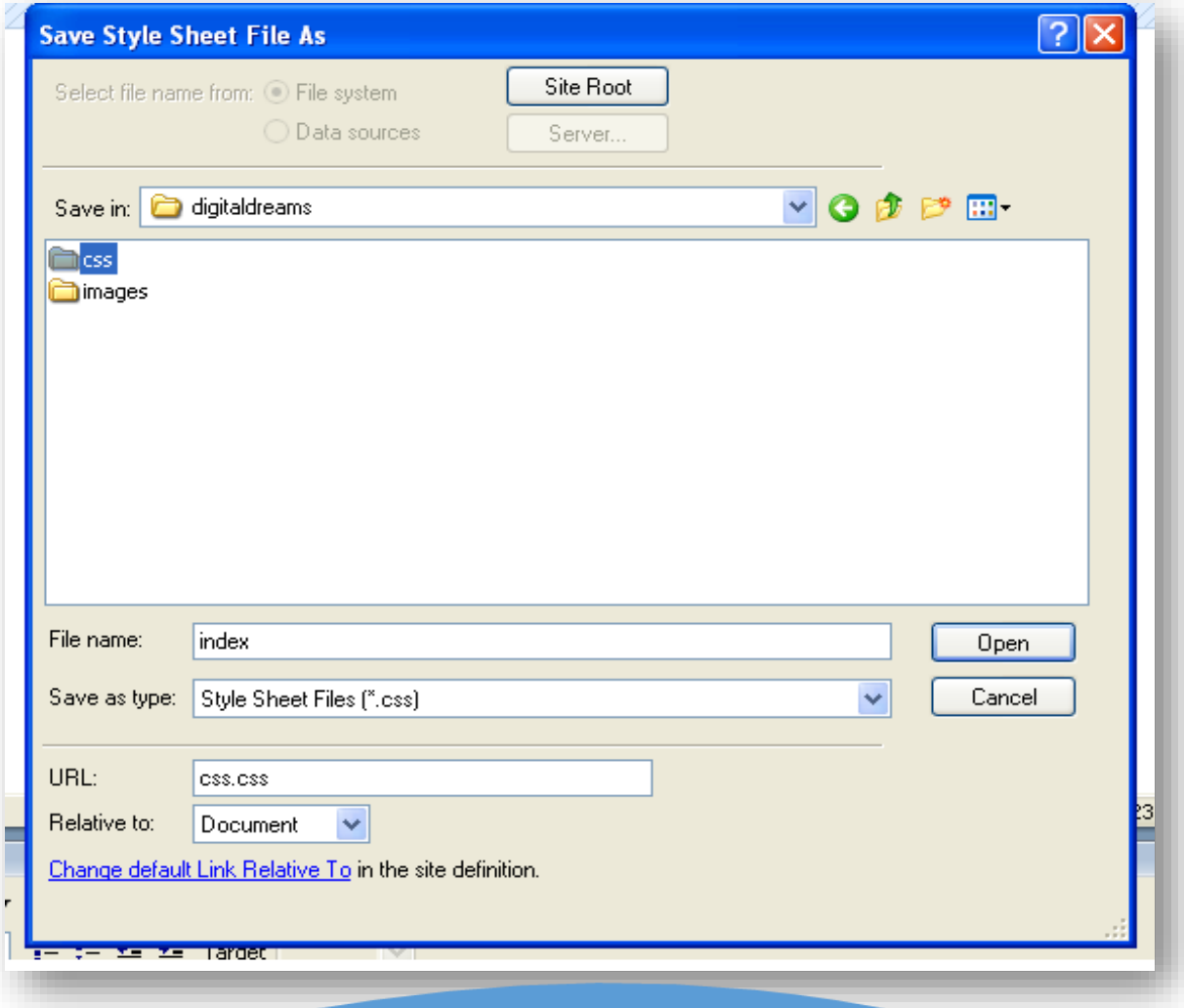


Using CSS to create layout cont..

When the following window shown below appear,

- create a **folder** and name it **CSS**.

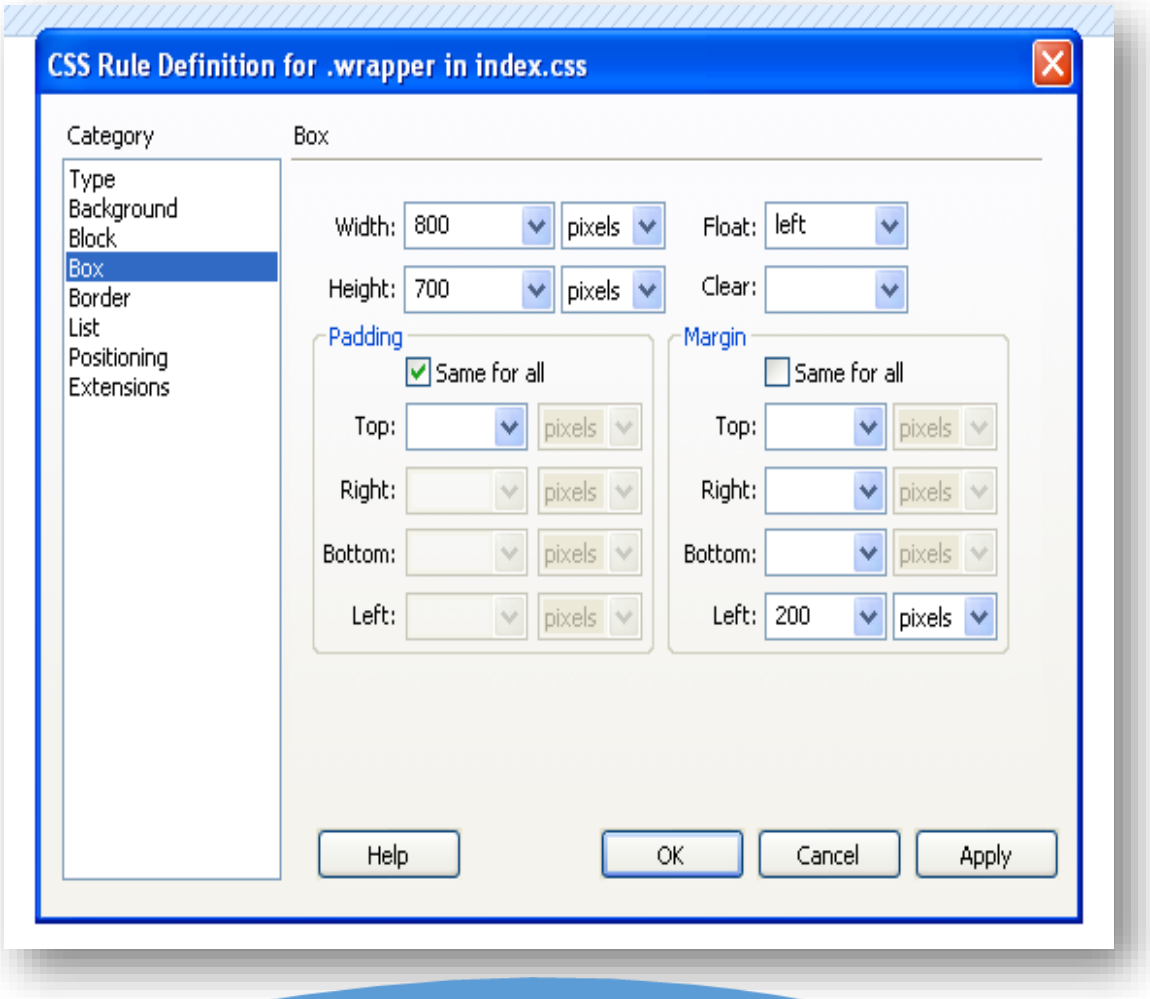
- Under **file name**, type **index.css** and **save** it in the **CSS folder**.



Setting the Wrapper property in CSS

From the CSS property window shown ,set the window as shown below:

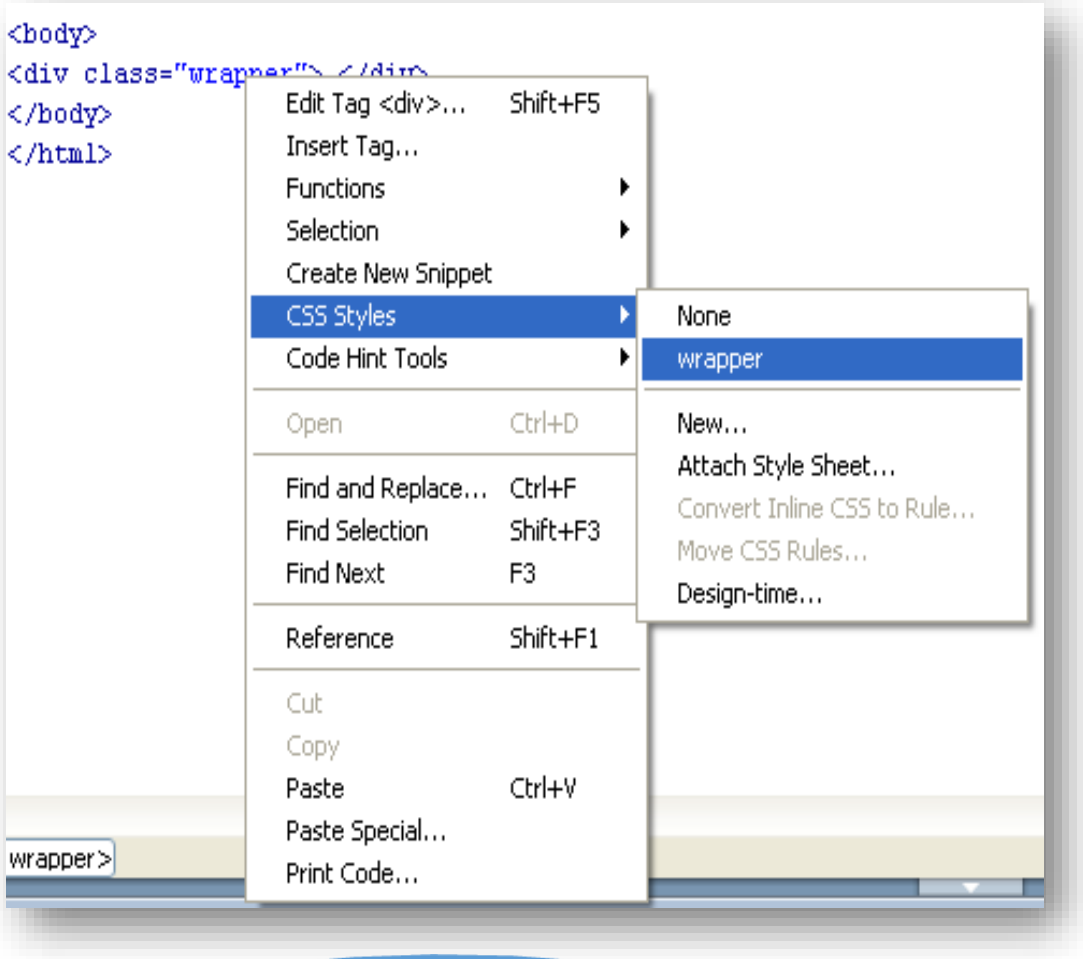
- Click on **Box**.
- **Width=800** and **height=700**
- Under **Float**, select **left**
- Under **margin** ,uncheck same for all Under **left select Left=Auto** and Under right select **right=Auto this will centralize the given box or div.(Note: this is achievable only when the box or div is not floated)**
- Click on **Background** and select any Background colour of your choice.
- Finally, click on **Apply** and **OK** button




Recycle Bin
a
c
b
chris
DIGITAL DREAMS
ICT ACADEMY
HOT LINE: 07064286765
4:04 PM

Connecting Wrapper code to CSS

- Right click on the **wrapper**,
- Click on **CSS style** and **wrapper** and as shown.
- Click on **design view** tab and **press F12** to preview the page.



Typing the CSS navbar code

- Return to Dreamweaver click on code view.
- Type the code as shown ** ** in between the wrapper code

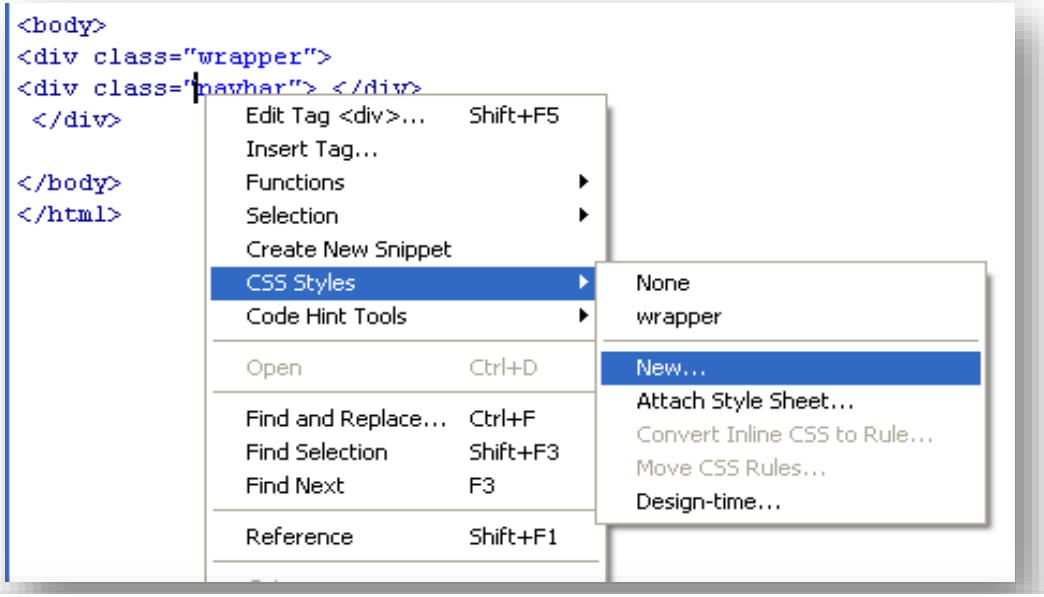
```
<html xmlns="http://www.w3.org/1999/x
<head>
<meta http-equiv="Content-Type" conte
<title>Untitled Document</title>
<link href="css/index.css" rel="style
</head>

<body>
<div class="wrapper">
<div class="navbar"> </div>
 </div>

</body>
</html>
```

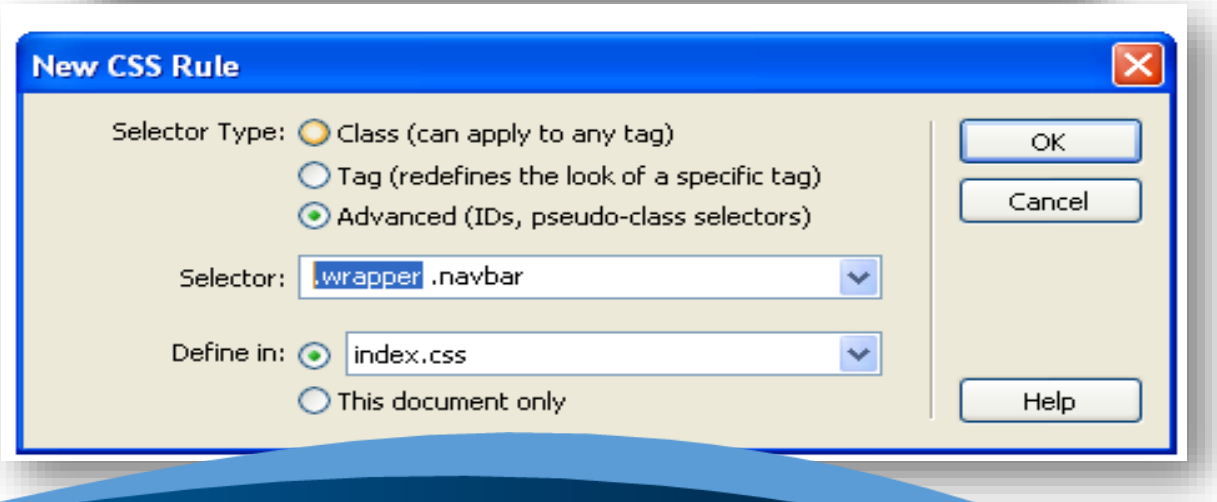
Connecting Navbar code to CSS

- Right click on the **navbar** code, click on **CSS style** and **new** from the list menu as shown.

- From the **New CSS Rule** dialogue window that appear, select **.wrapper** as shown below and **Delete** it

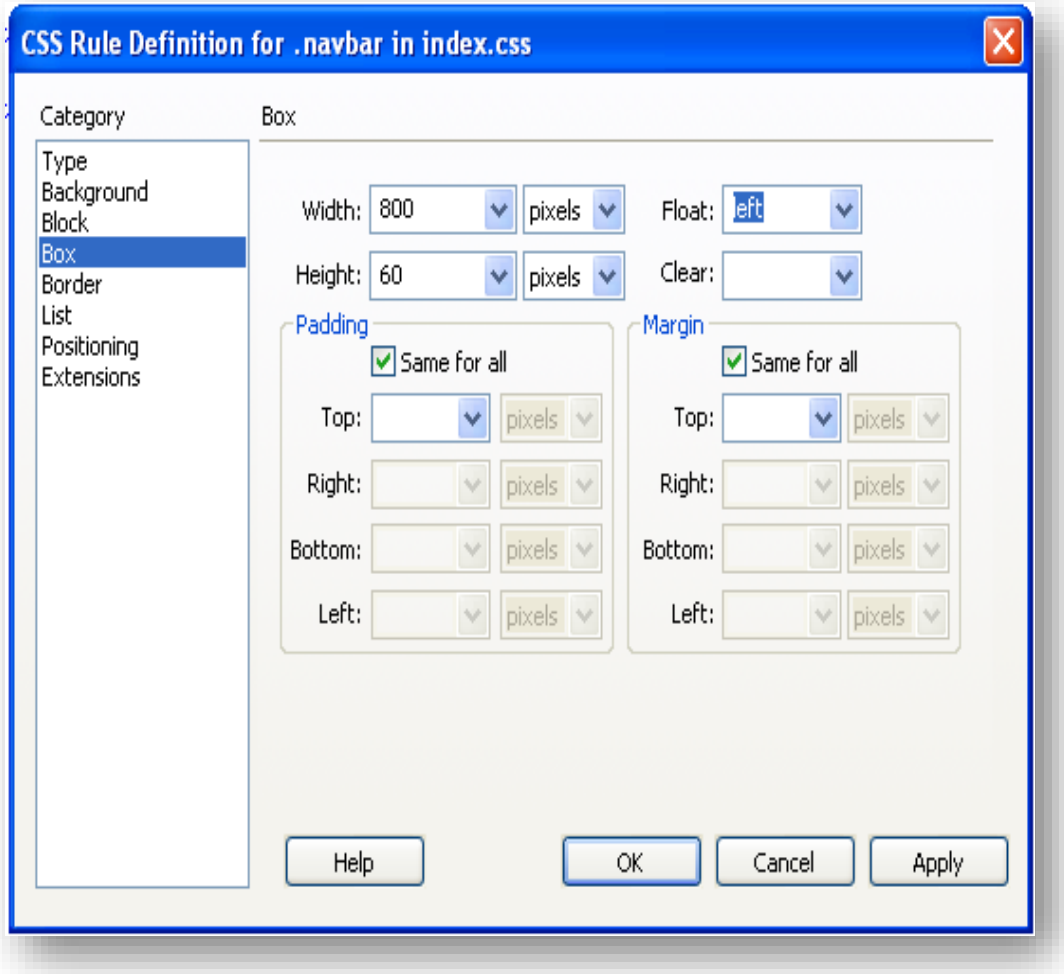
- Then click on **Ok**.





Setting the navbar property

- Set the **navbar** property as shown below.
- **Width=800 , height=60** and **Float=left**
- Click on **Background** and select any Background colour of your choice.
- Finally, click on **Apply** and **OK** button

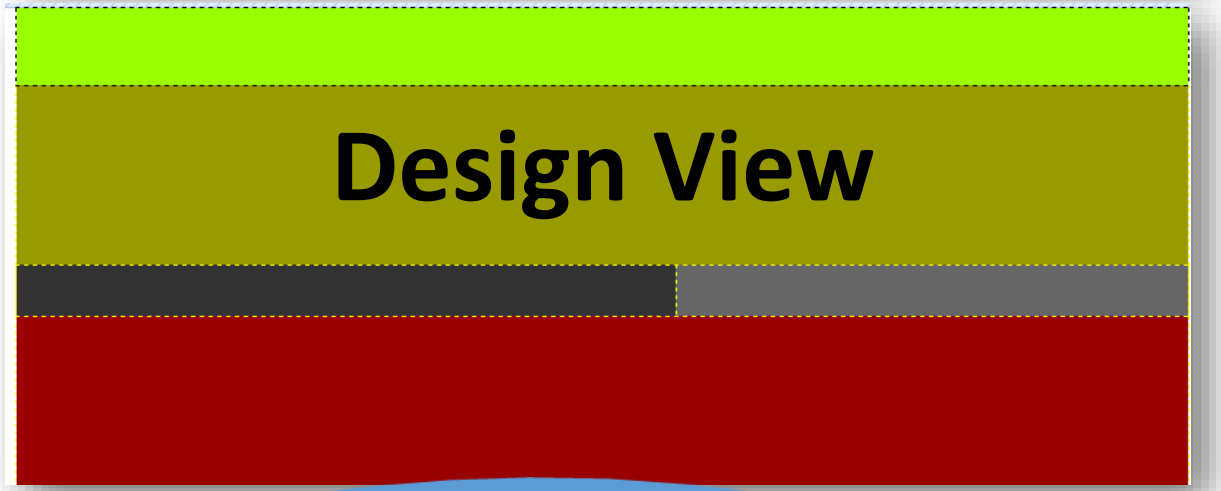


Creating other layout with CSS

- Follow the **navbar** method and create any other **CSS layout** of your choice e.g. **banner** etc as shown in the **code view** and **design view** below.

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Untitled Document</title>
<link href="css/index.css" rel="stylesheet" type="text/css" />
</head>

<body>
<div class="wrapper">
<div class="navbar"> </div>
<div class="banner"> </div>
<div class="leftbg"> </div>
<div class="right"> </div>
 </div>

</body>
</html>
```

Code View

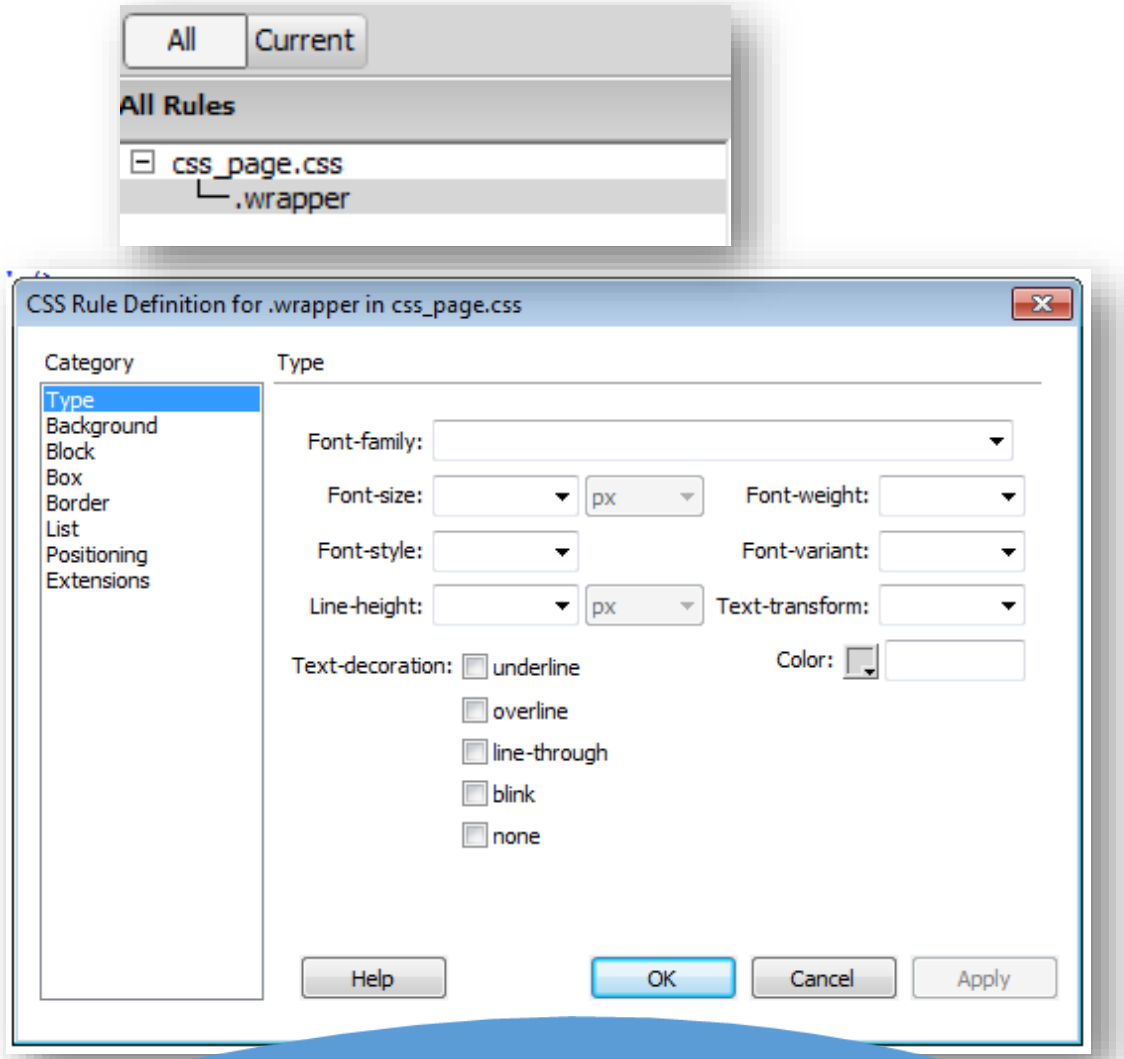


Setting other CSS properties

Text

Double click on the wrapper To display the property window

Choose a **font type**, **size**, **style**, **weight**, **line-height** and **color** from the dropdown arrows. Click **apply** then **ok.**

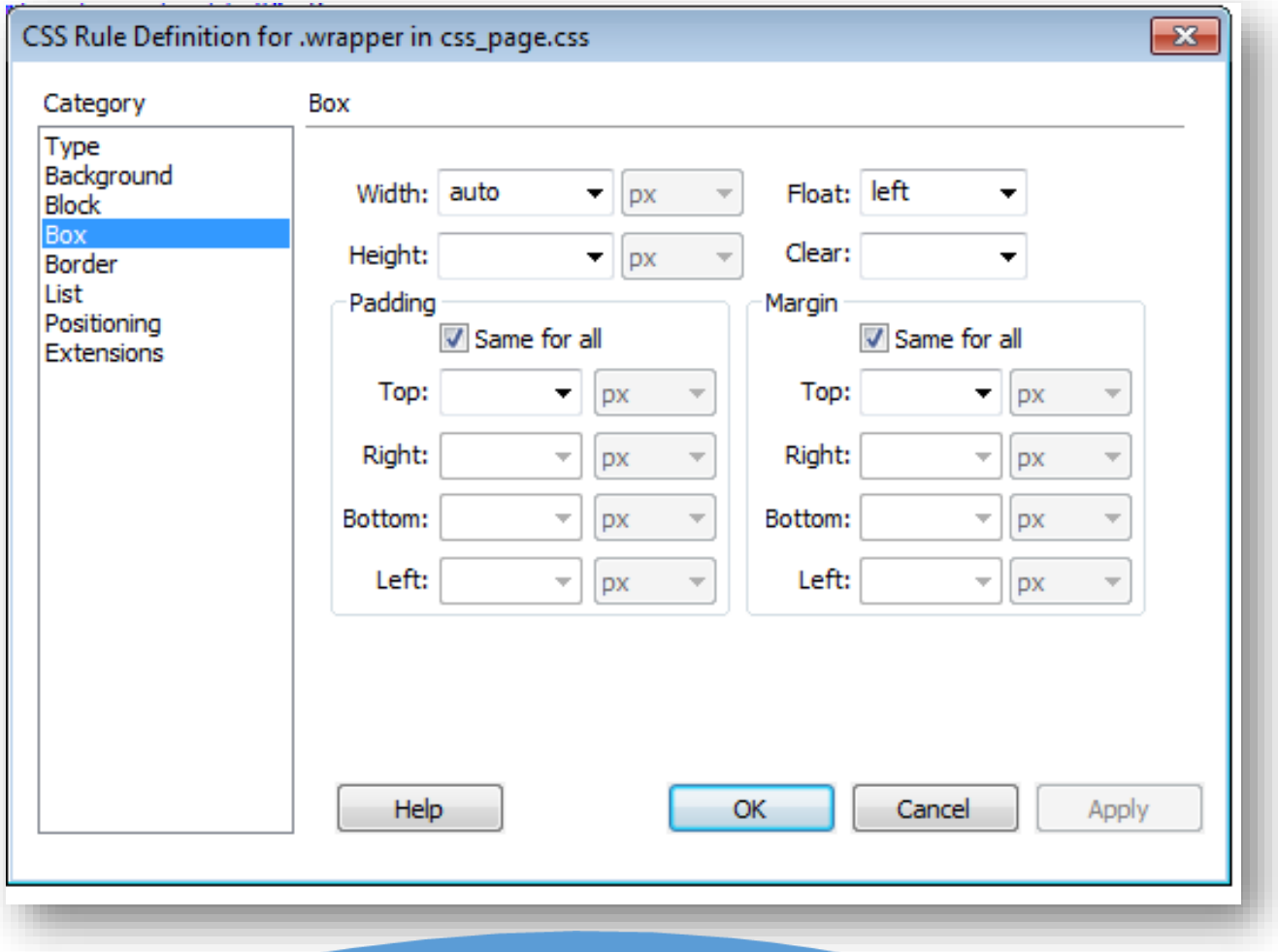


Setting other CSS properties

Margin and Padding

Click on the **Box** tab, set the **Padding** and **Margin** as desired

Padding is the space between the text and the border of a div (i.e. inner border) and Margin is the space between divs (i.e. outer border)



Advantages of CSS over Table Layout

Disadvantages of using tables

- We needed three sets of tags to present some content as opposed to the one set of div tags. As we add more to the page's design the table complexity continues to increase compared to divs.
- You will need to add at least another table cell to get another block of information compared to adding another div for another block of information, which would grow our code equally and leads to more potential errors
- They impose a more rigid structure than divs. Every table cell is dependent on the other table cells in its row to maintain the structure. Divs can work independently from each other.
- In order for a browser to render a page built with tables it needs to read the code on the page twice. First, to understand the structure and another time to present it. That extra pass at the code makes table-based layouts take longer to display.

Advantages of CSS layout over Table layout

- easier to maintain – less code and less complexity to the structure makes things easier to find and change.

- more flexible – since one div is not dependent on the other divs on the page it allows for more freedom in your design

- quicker to load – blocks of code can be presented right away instead of the browser requiring an extra pass

- That might not seem like a lot, but just those three things are enough to make a div based approach better to a table based approach. Now let's get to some of the myths on both sides that keep this debate going.

Basic causes of CSS failure

Basic causes of CSS failure

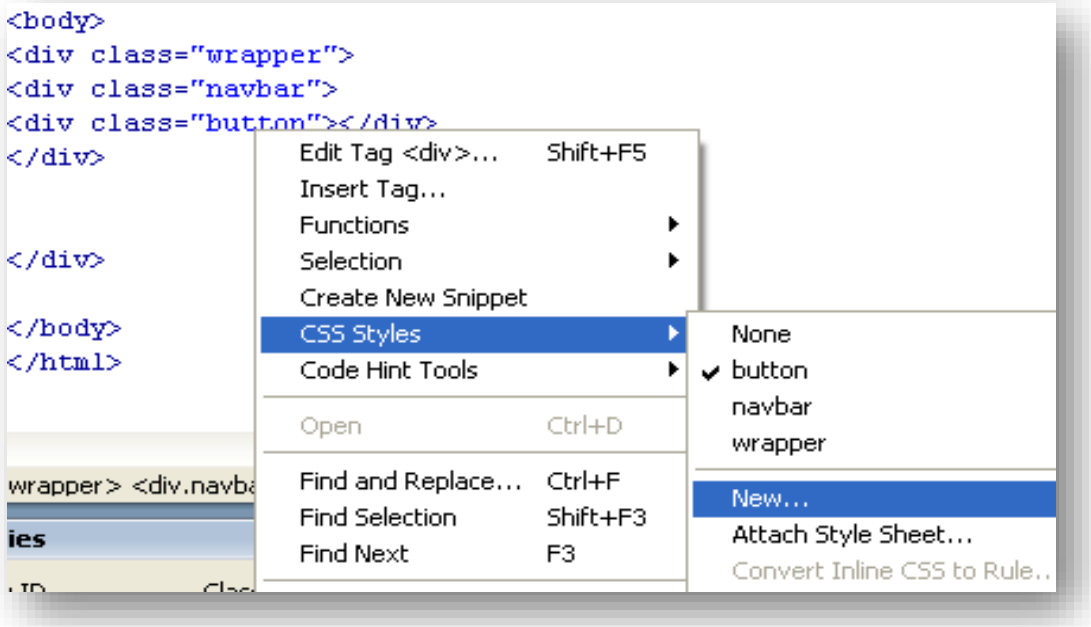
- Failure to close any of the div code
- Inconsistency in floating left etc.
- When total different individual div sizes is more than main div that contain them.
- Failure to consider padding, margin and border value when calculating the individual div size.
- Wrong position of code.
- Changing code property shared by many pages in a just one page.
- Wrong naming of Div code e.g. use of number.
- And many other silly mistake

Using CSS to create button

➢Type the **button div code** as shown

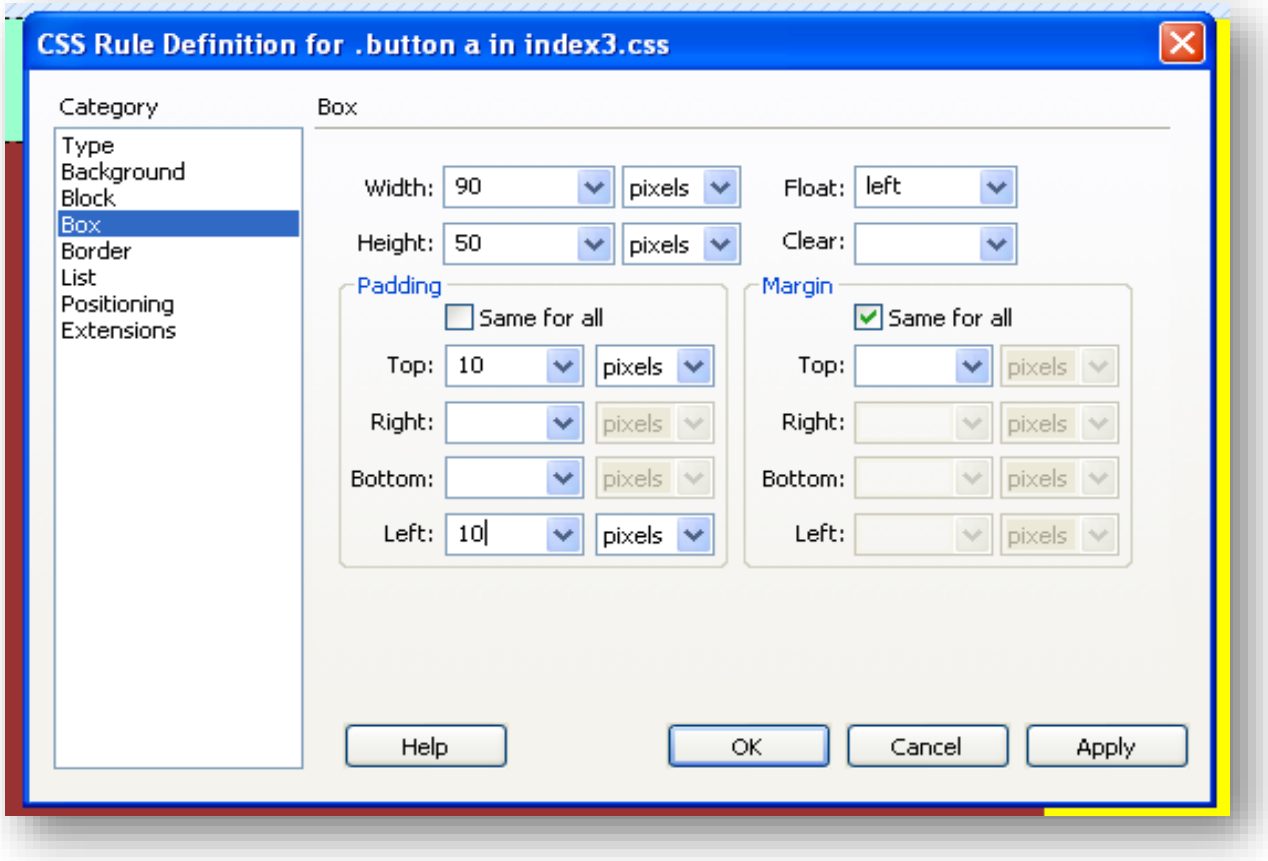
```
<body>
    <div class="wrapper">
        <div class="nav"> </div>
        <div class="button"> </div>
    </div>
</body>
```

➢Right click on the **button** code,
select **CSS style** and click on **ne**w as shown.



Using CSS to create button

- Set the property as shown Below including **bgcolor** And **apply** and **OK**

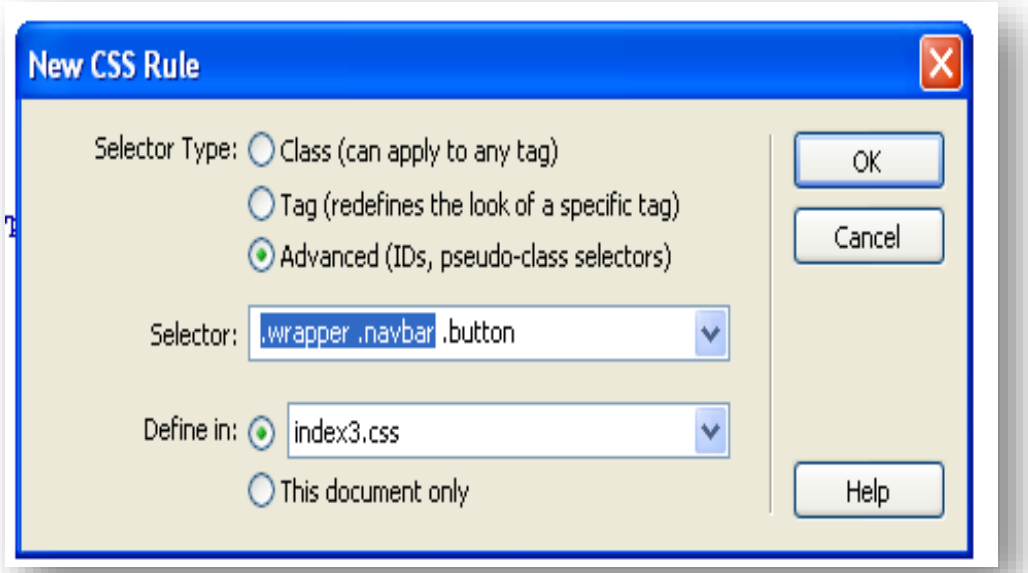


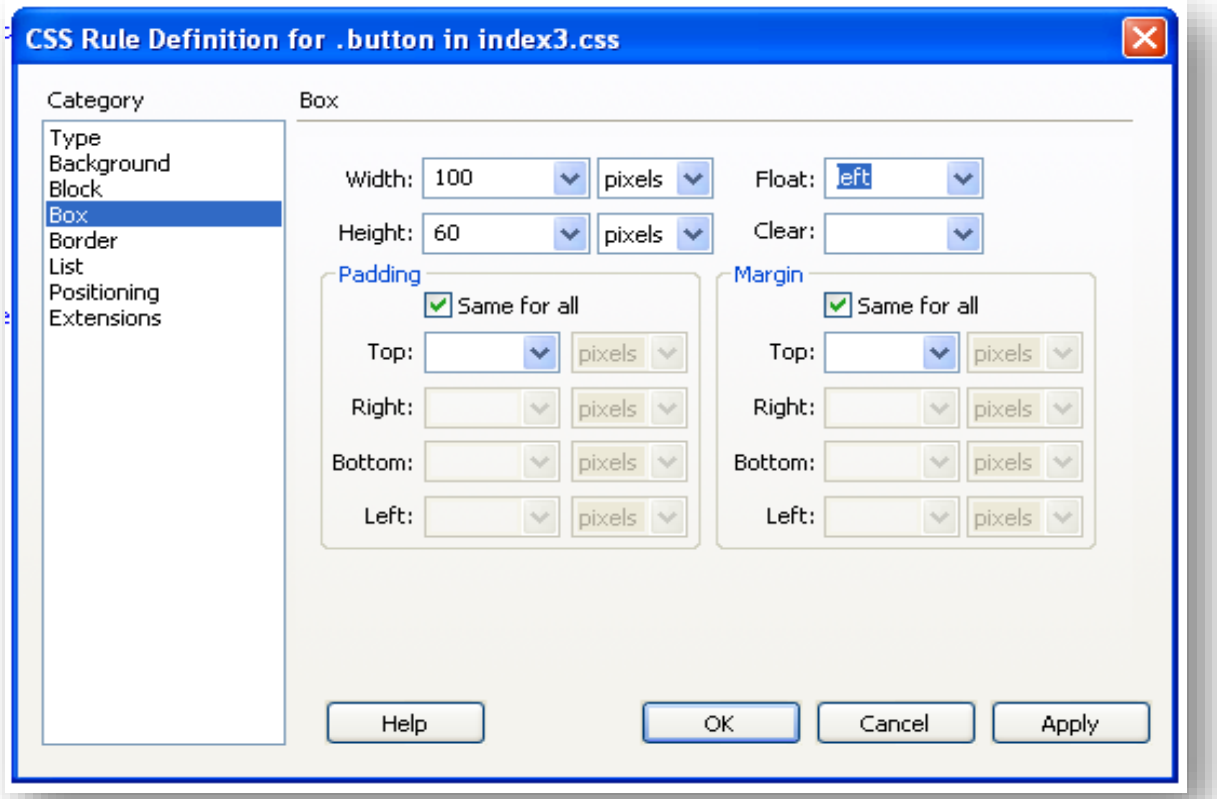

Recycle Bin
a
c
b
chris
DIGITAL DREAMS
ICT ACADEMY
HOT LINE: 07064286765
4:04 PM

Using CSS to create button

- Delete the Selected word as shown and leave **.button** and Ok it.

- Set the property as shown and **apply** and **Ok**.





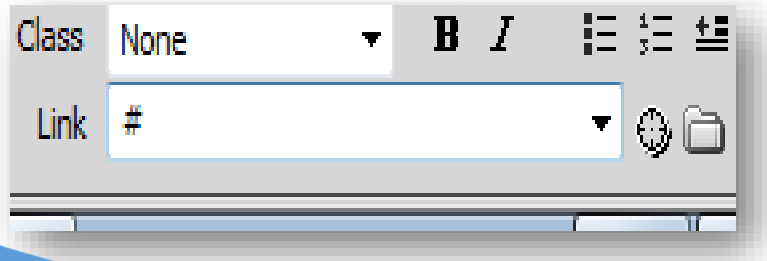
Using CSS to create button

➢Go to **code view** and type **home** as shown.

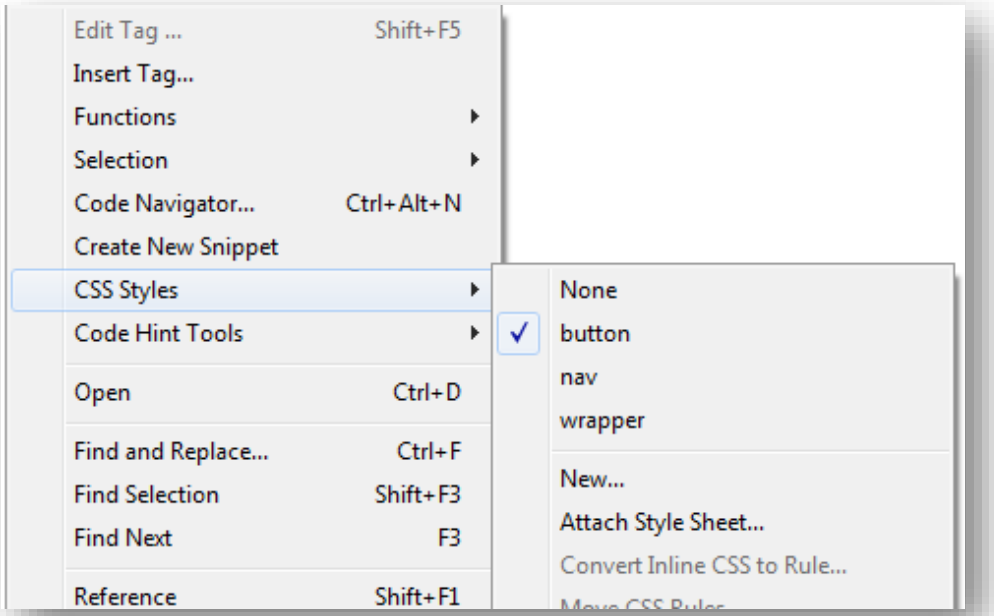
```
<body>
    <div class="wrapper">
        <div class="nav"> </div>
        <div class="button">Home page</div>
    </div>
```

➢Highlight the **hom**e and type # in link space as shown

```
<body>
<div class="wrapper">
    <div class="nav"> </div>
        <div class="button">Home page</div>
</div>
</body>
```

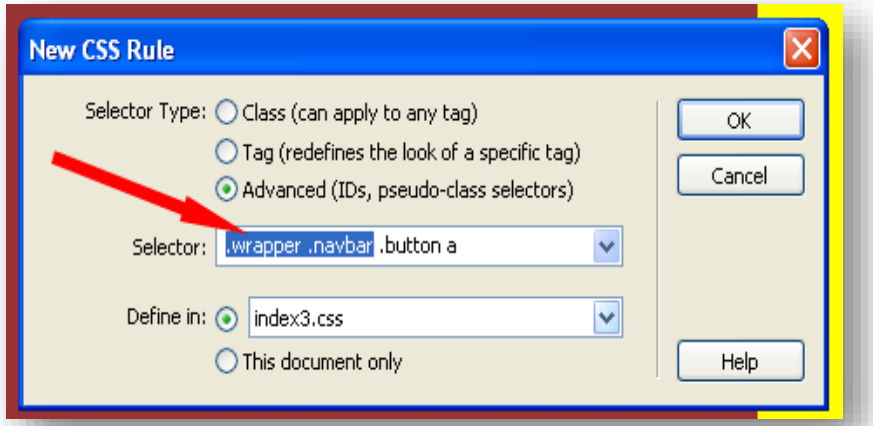


➢Right click on the **home** and select **CSS style** and **new** as shown below.

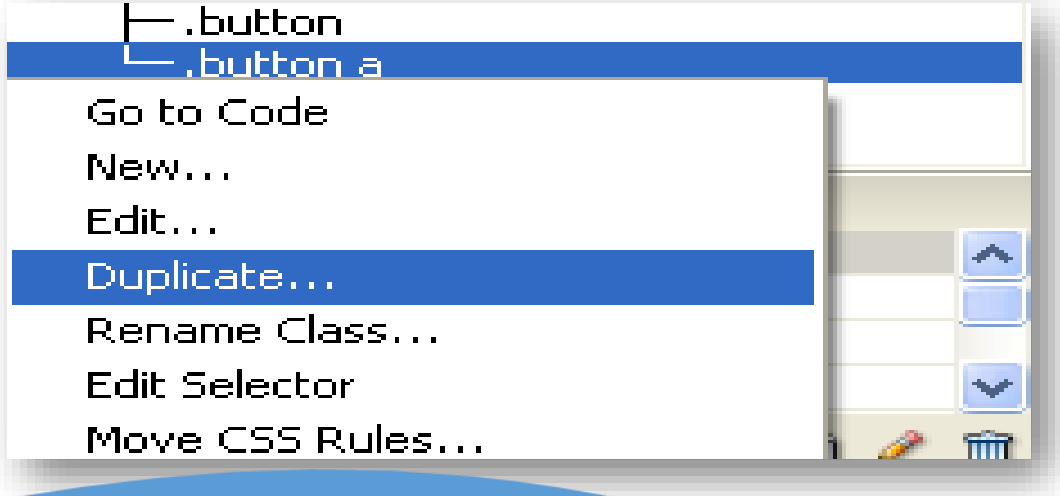


Using CSS to create button

- **Delete** selected words shown and leave **.button a**

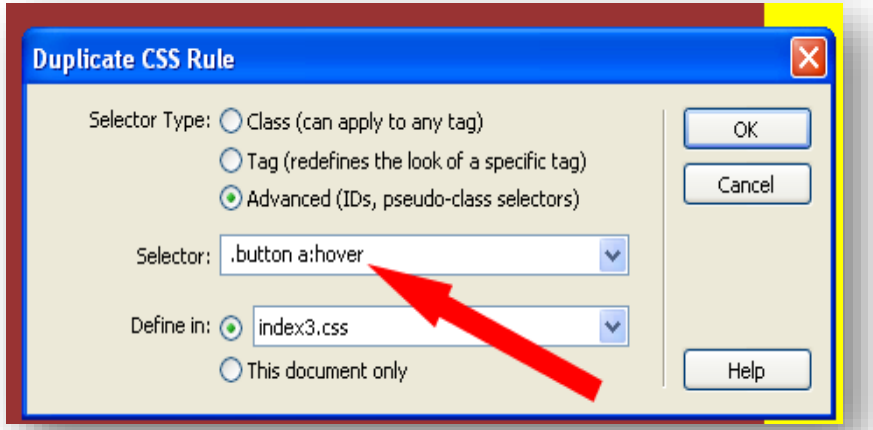


- Click on **button a** from the list, **right click** and **click on duplicate** as shown below.

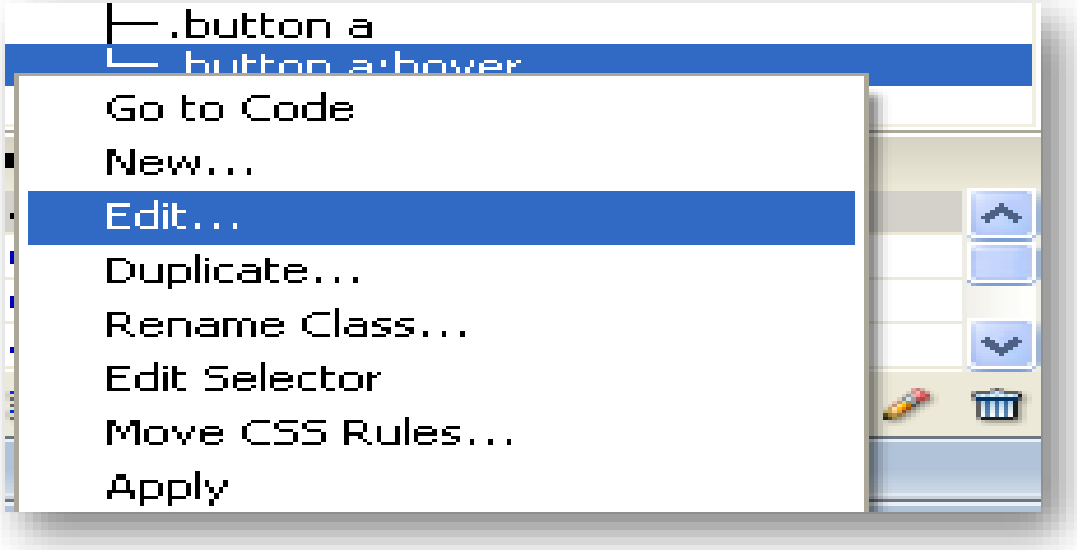


Using CSS to create button

type **:hover** in the front of **button a** as shown and Ok it.



➢Click on **button a:hover** from the list,
right click and click on **edit** as shown below.



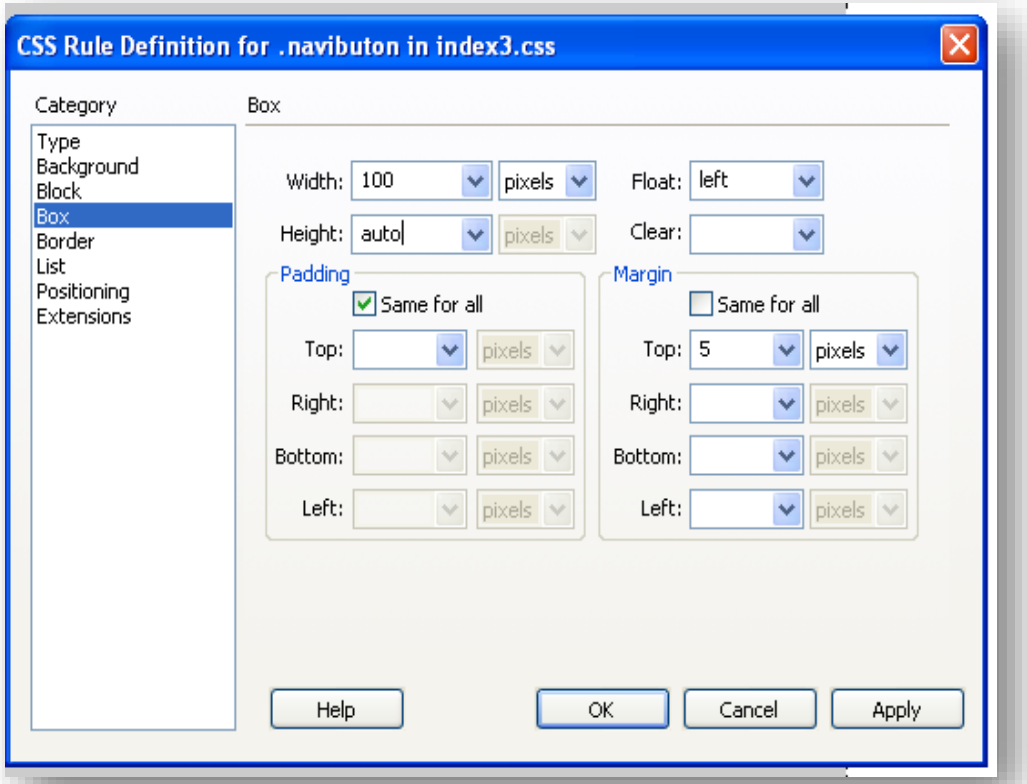

Recycle Bin
a
c
b
chris
DIGITAL DREAMS
ICT ACADEMY
HOT LINE: 07064286765
4:04 PM

Creating a dropdown menu with css

Creating a dropdown menu with css

- First design a complete one nav button with CSS

- Change the button height to **auto** as shown below and apply and Ok.

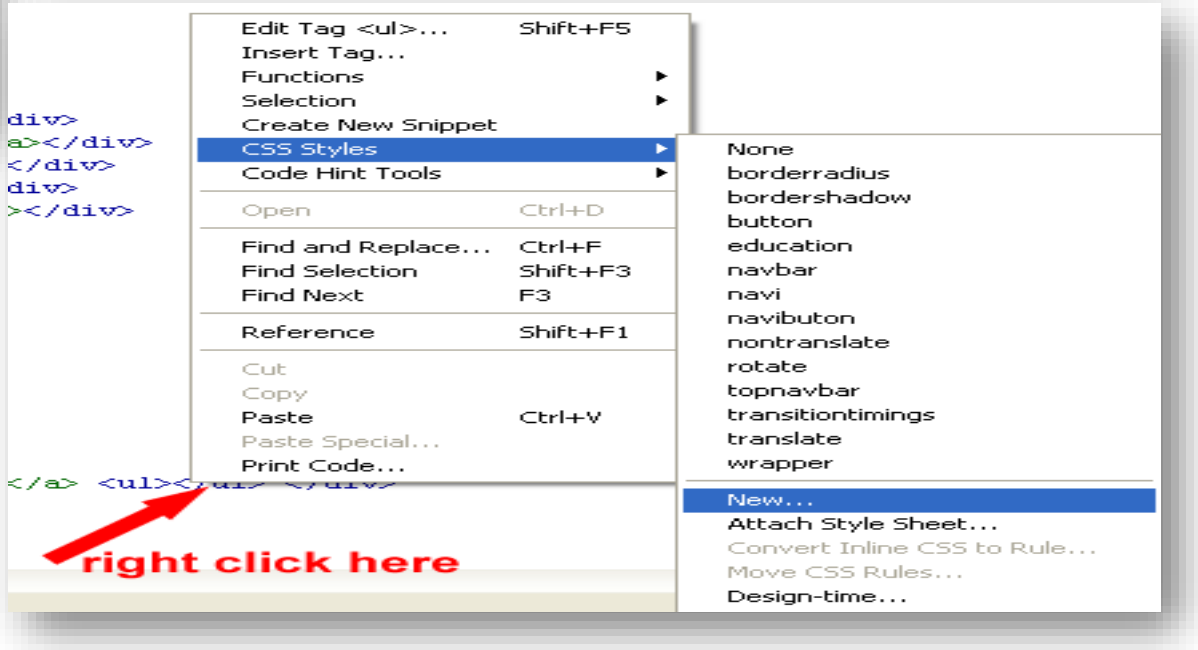


Creating a dropdown menu with css

- Now change to code view and type the following <ul> code as shown.

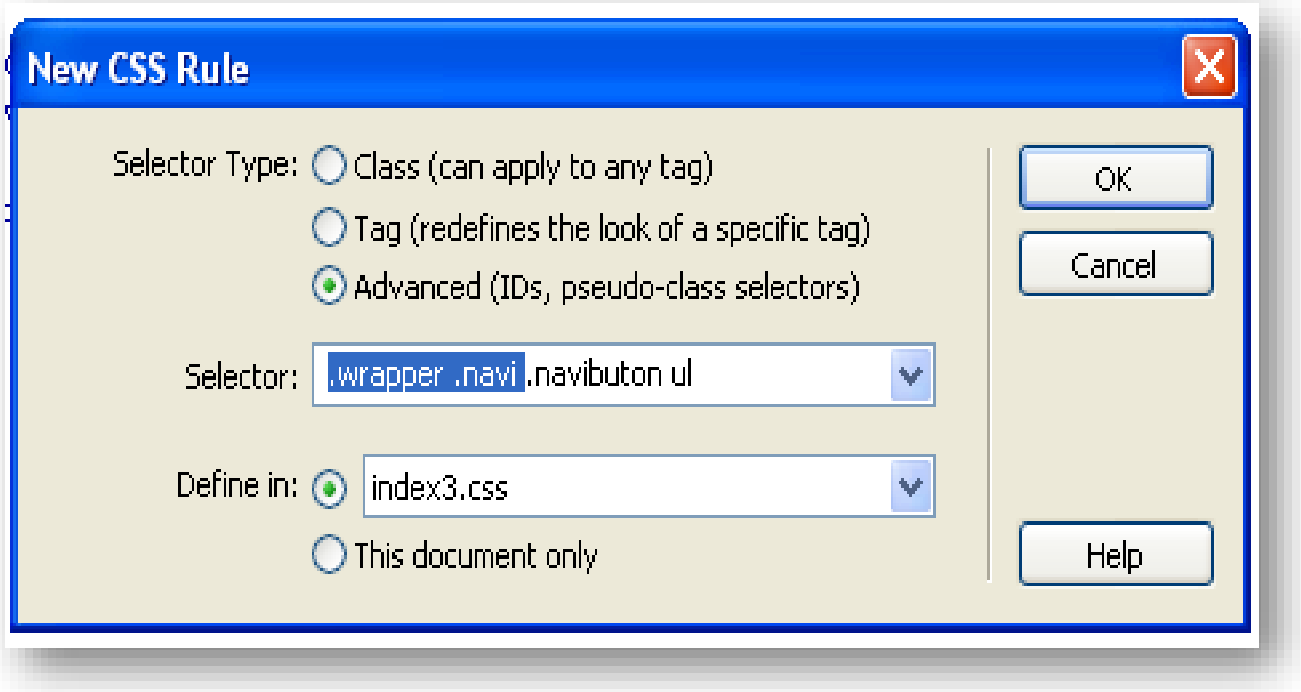
```
<div class="navi">
<div class="navibuton"><a href="#">homepage</a> <ul></ul> </div>
</div>
```

- Right click on **
** code,
- click on **css style** and **new**



Creating a dropdown menu with css

➢Delete the selected text and leave .navibutton ul behind as shown below and OK it.



Creating a dropdown menu with css

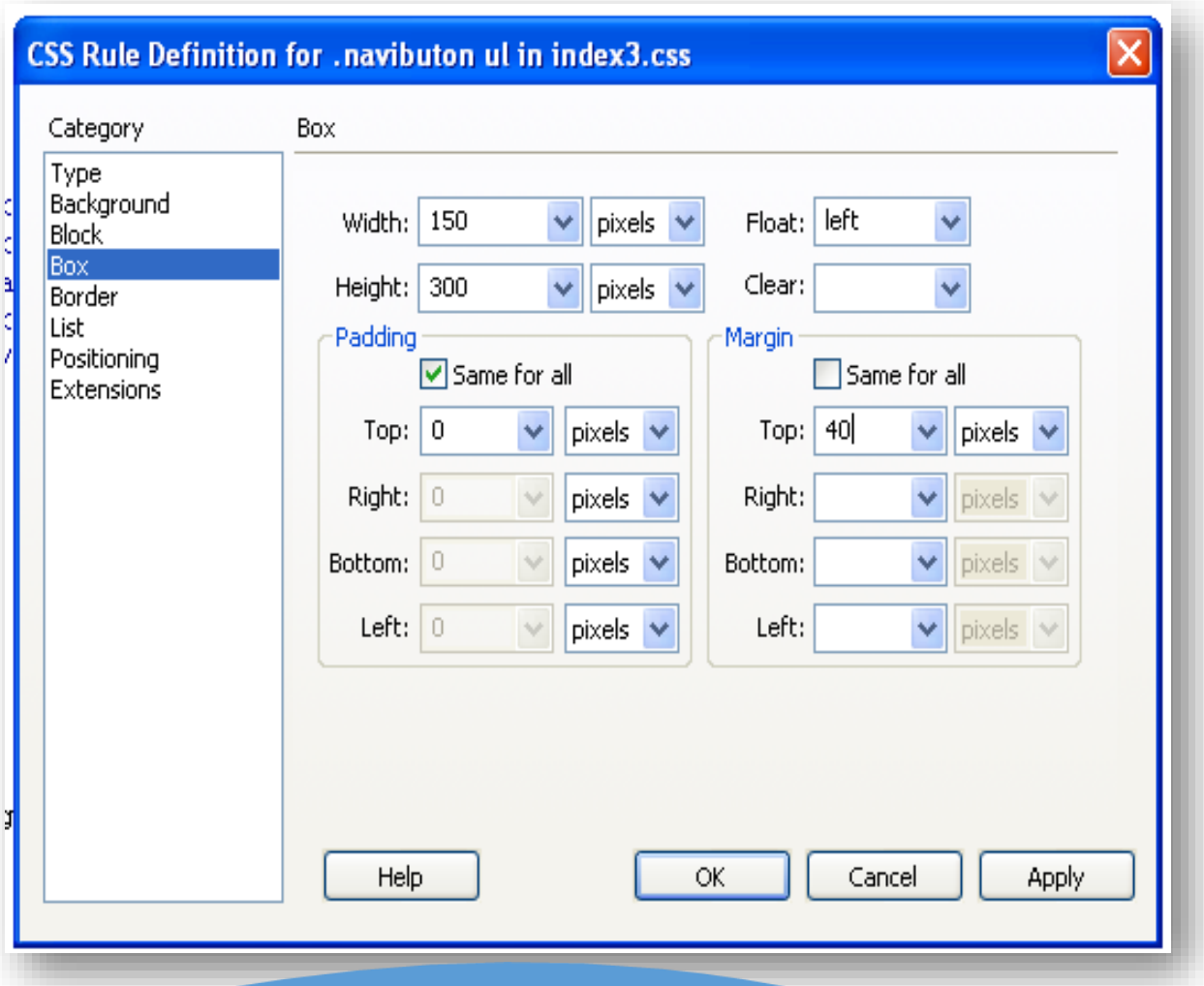
- From the property page that will display, click on **block** and change **display** to **none.**

- Click on **Position** and change position type to **absolute**.

- Click on **list** and change **type** to **None**.

- Finally click on **Box** and change property
as shown

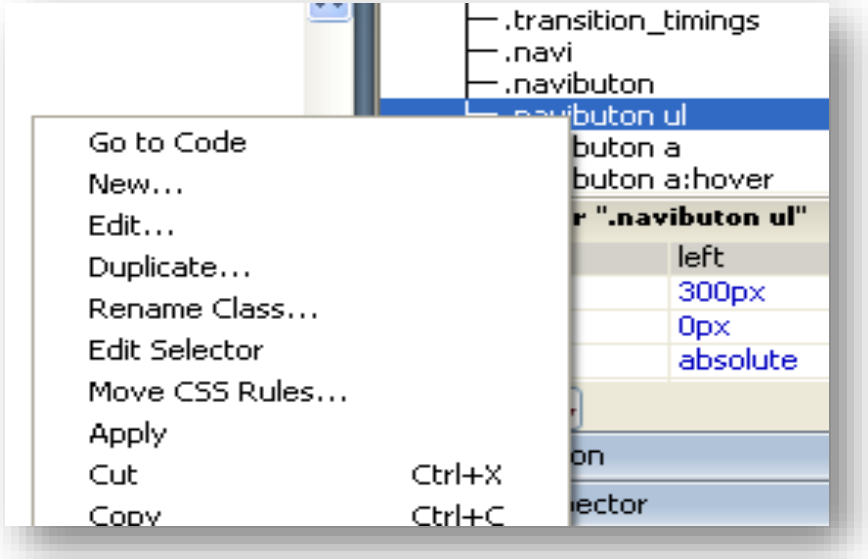
- Click **apply** and **Ok**.



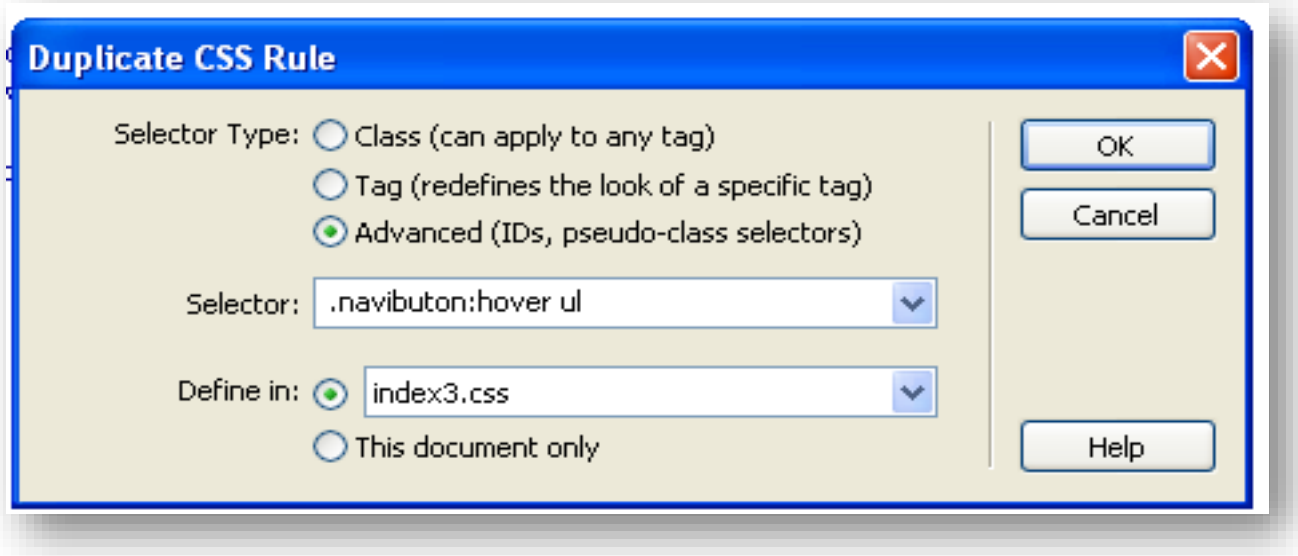
Creating a dropdown menu with css

➢Right click on **navibutton ul** from the list view as shown

and click on **Duplicate.**



Type **:hover** in between **navibutton** and **ul** as shown below and Ok it.

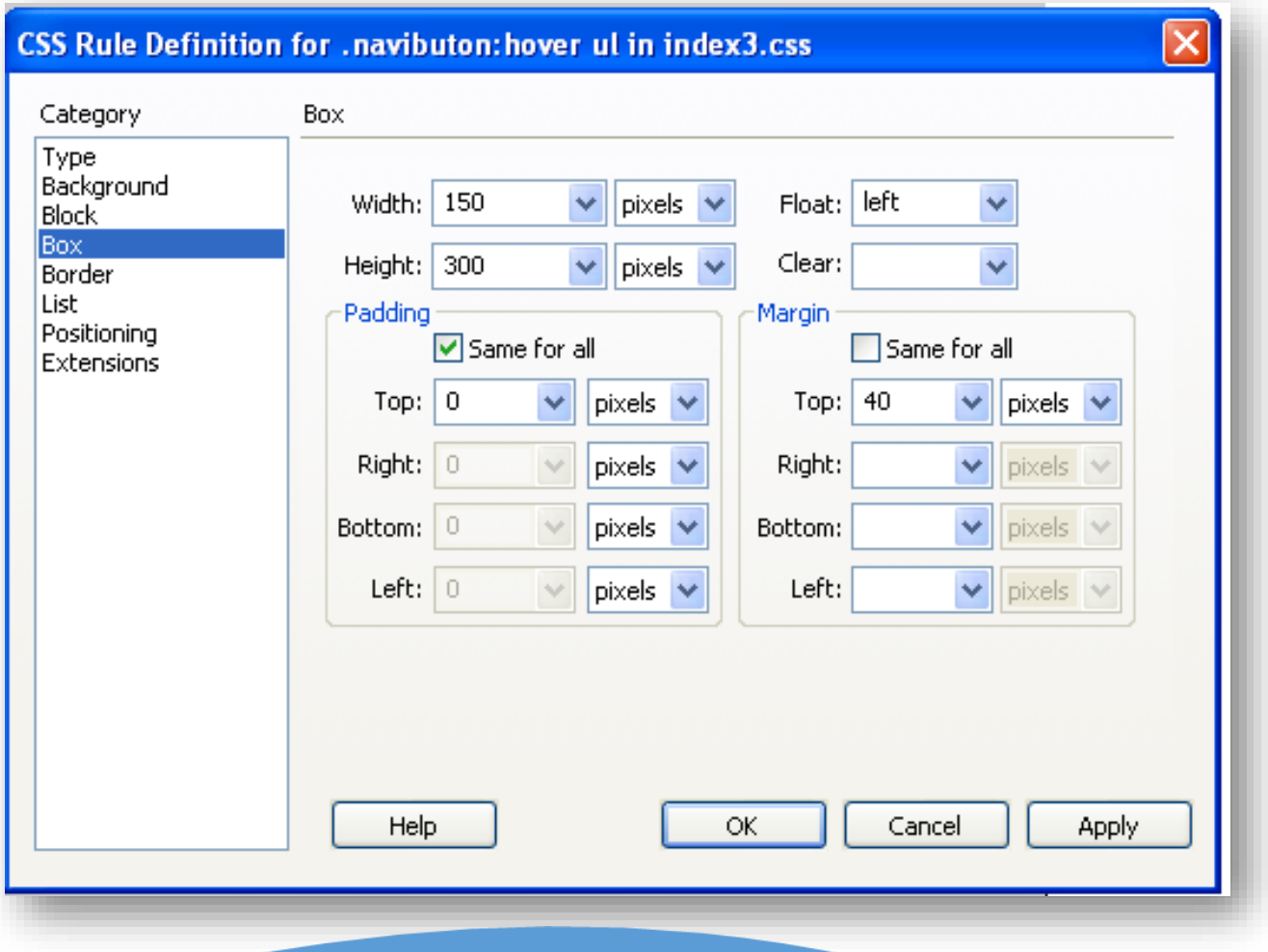


Creating a dropdown menu with css

- Go to list view and double click on **navibutton:hover ul** to open and edit it as follows.

- Go to **background color** and give it the same color as **navibutton a:hover**

- Go to **block** and change **display** from **none** to **Block.** Then **Apply** and **OK.**



Creating a dropdown menu with css

- Change to **code view** and type **
** code in between **
** code .
- Type **link name** and select it as shown below.

```
<div class="navi">
<div class="navibuton"><a href="#">homepage</a> <ul> <li>realhome</li></ul>
</div>
</div>
```

- Change to **design view** and type **#** under link spacebar.

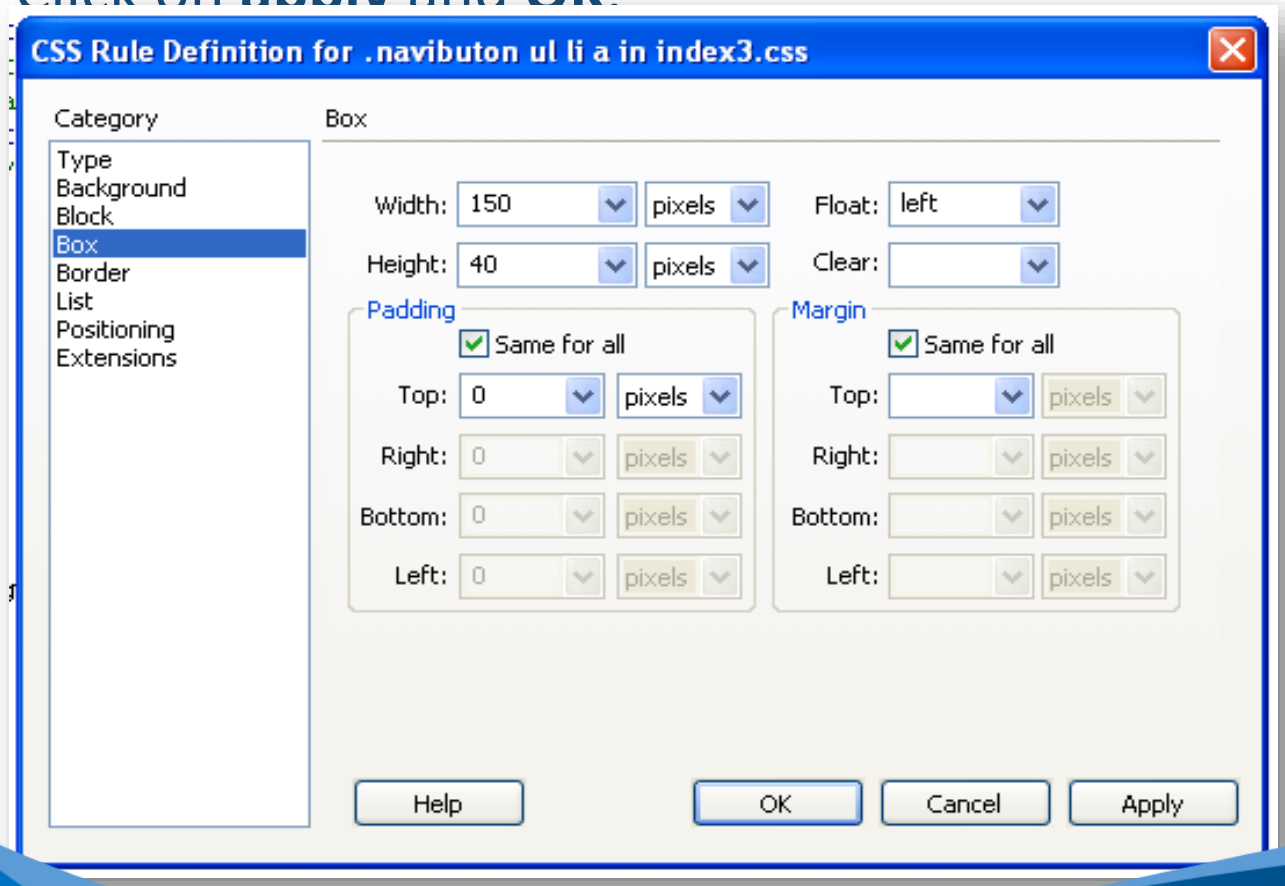
- Change back to **code view** and **right click** on the link name ,select **css style** and click on **new**.

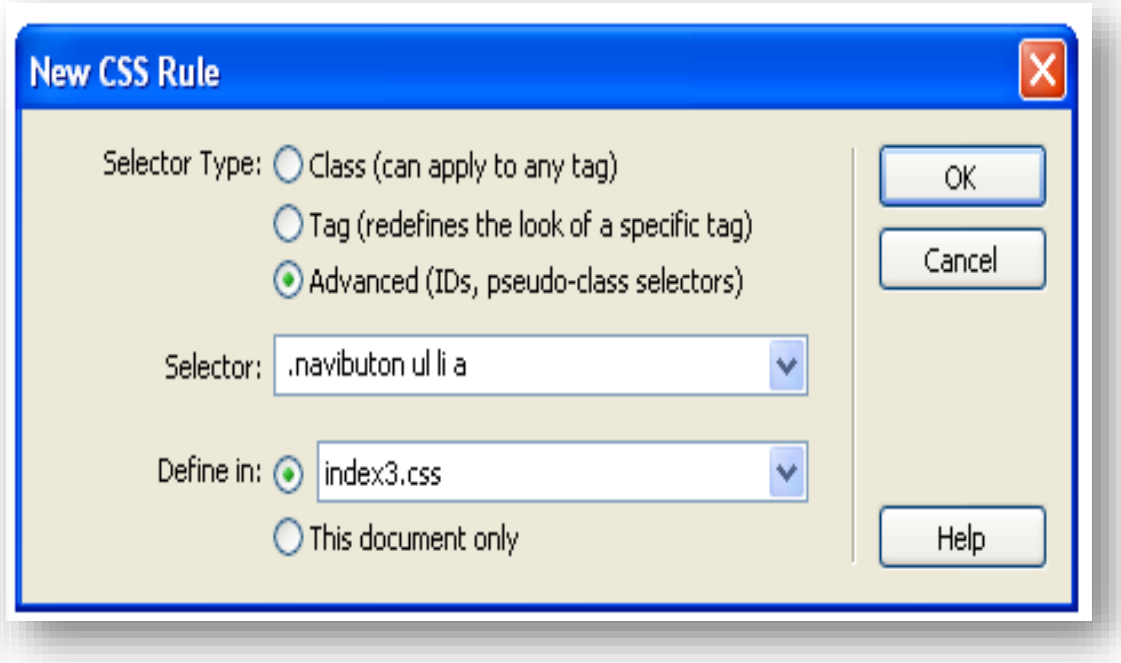
Creating a dropdown menu with css

Type **ul li a** in front of **navibutton** to look as shown

Under **navibutton** property, set it as shown below.

Click on **apply** and **OK**.



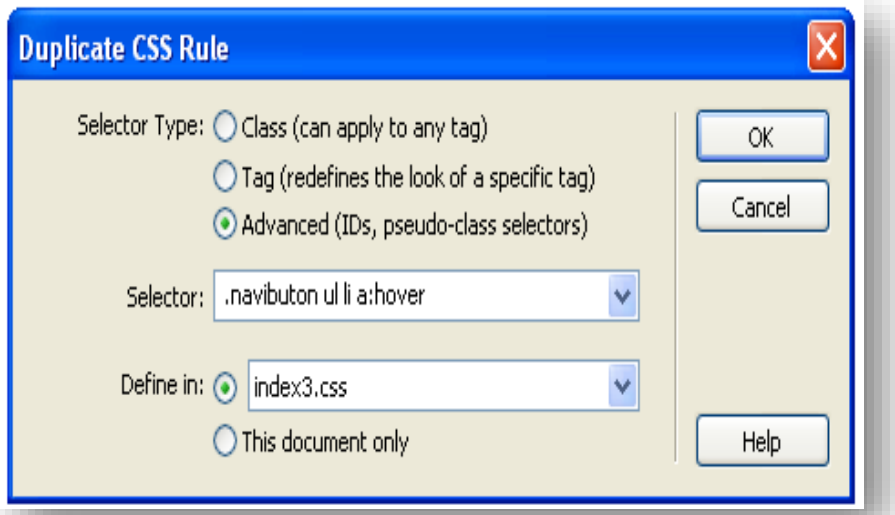


Creating a dropdown menu with css

Go to **list file** and right click on **navibutton ul li a** and click on **Duplicate.**

Type **navibutton ul li a:hover** as shown then Click on OK**.**

Change to **code view** and copy the ** code** and **paste** as shown below**.**




```
<div class="navi">
<div class="navibuton"><a href="#">homepage</a> <ul>
<li><a href="#">realhome</a></li>
<li><a href="#">realhome</a></li>
<li><a href="#">realhome</a></li>
<li><a href="#">realhome</a></li>
<li><a href="#">realhome</a></li>
<li><a href="#">realhome</a></li>
<li><a href="#">realhome</a></li>
</ul>
</div>
```

Finally**, save** changes and run to test the page.

CSS Animations

What are CSS Animations?
An animation lets an element gradually change from one style to another.
You can change as many CSS properties you want, as many times you want.
To use CSS animation, you must first specify some keyframes for the animation.
Keyframes hold what styles the element will have at certain times.

The @keyframes Rule
When you specify CSS styles inside the @keyframes rule, the animation will gradually change from the current style to the new style at certain times.
To get an animation to work, you must bind the animation to an element.
The following example binds the "example" animation to the element. The animation will last for 4 seconds, and it will gradually change the background-color of the element from "red" to "yellow":

Example

```
/* The animation code */
@keyframes example {
  from {background-color: red;}
  to {background-color: yellow;}
}
```

```
/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```

Note: The animation-duration property defines how long time an animation should take to complete. If the animation-duration property is not specified, no animation will occur, because the default value is 0s (0 seconds).
In the example above we have specified when the style will change by using the keywords "from" and "to" (which represents 0% (start) and 100% (complete)).
It is also possible to use percent. By using percent, you can add as many style changes as you like.
The following example will change the background-color of the element when the animation is 25% complete, 50% complete, and again when the animation is 100% complete:

Example

```
/* The animation code */
@keyframes example {
    0%   {background-color: red;}
    25%  {background-color: yellow;}
    50%  {background-color: blue;}
    100% {background-color: green;}
}

/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```

The following example will change both the background-color and the position of the element when the animation is 25% complete, 50% complete, and again when the animation is 100% complete:

Example

```
/* The animation code */
@keyframes example {
    0%   {background-color:red; left:0px; top:0px;}
    25%  {background-color:yellow; left:200px; top:0px;}
    50%  {background-color:blue; left:200px; top:200px;}
    75%  {background-color:green; left:0px; top:200px;}
    100% {background-color:red; left:0px; top:0px;}
}

/* The element to apply the animation to */
div {
    width: 100px;
    height: 100px;
    position: relative;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
}
```

Delay an Animation

The animation-delay property specifies a delay for the start of an animation.

The following example has a 2 seconds delay before starting the animation:

Example

```
div {
    width: 100px;
    height: 100px;
    position: relative;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
    animation-delay: 2s;
}
```

Negative values are also allowed. If using negative values, the animation will start as if it had already been playing for *N* seconds.

In the following example, the animation will start as if it had already been playing for 2 seconds:

Example

```
div {
    width: 100px;
    height: 100px;
    position: relative;
    background-color: red;
    animation-name: example;
    animation-duration: 4s;
    animation-delay: -2s;
}
```

Set How Many Times an Animation Should Run

The animation-iteration-count property specifies the number of times an animation should run.

The following example will run the animation 3 times before it stops:

Example

```
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: 3;
}
```

The following example uses the value "infinite" to make the animation continue for ever:

```
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: infinite;
}
```

Run Animation in Reverse Direction or Alternate Cycles

The animation-direction property specifies whether an animation should be played forwards, backwards or in alternate cycles.

The animation-direction property can have the following values:

normal - The animation is played as normal (forwards). This is default

reverse - The animation is played in reverse direction (backwards)

alternate - The animation is played forwards first, then backwards

alternate-reverse - The animation is played backwards first, then forwards

The following example will run the animation in reverse direction (backwards):

Example

```
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
  animation-direction: reverse;
}
```

The following example uses the value "alternate" to make the animation run forwards first, then backwards:

Example

```
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: 2;
  animation-direction: alternate;
}
```

The following example uses the value "alternate-reverse" to make the animation run backwards first, then forwards:

Example

```
div {
  width: 100px;
  height: 100px;
  position: relative;
  background-color: red;
  animation-name: example;
  animation-duration: 4s;
  animation-iteration-count: 2;
  animation-direction: alternate-reverse;
}
```

Specify the Speed Curve of the Animation

The animation-timing-function property specifies the speed curve of the animation.

The animation-timing-function property can have the following values:

- ease - Specifies an animation with a slow start, then fast, then end slowly (this is default)
- linear - Specifies an animation with the same speed from start to end
- ease-in - Specifies an animation with a slow start
- ease-out - Specifies an animation with a slow end
- ease-in-out - Specifies an animation with a slow start and end
- cubic-bezier(n,n,n,n) - Lets you define your own values in a cubic-bezier function

The following example shows the some of the different speed curves that can be used:

Example

```
#div1 {animation-timing-function: linear;}
#div2 {animation-timing-function: ease;}
#div3 {animation-timing-function: ease-in;}
#div4 {animation-timing-function: ease-out;}
#div5 {animation-timing-function: ease-in-out;}
```

CSS Animation Properties

The following table lists the @keyframes rule and all the CSS animation properties:

Property	Description
@keyframes	Specifies the animation code
animation	A shorthand property for setting all the animation properties
animation-delay	Specifies a delay for the start of an animation
animation-direction	Specifies whether an animation should be played forwards, backwards or in alternate cycles

animation-duration	Specifies how long time an animation should take to complete one cycle
animation-fill-mode	Specifies a style for the element when the animation is not playing (before it starts, after it ends, or both)
animation-iteration-count	Specifies the number of times an animation should be played
animation-name	Specifies the name of the @keyframes animation
animation-play-state	Specifies whether the animation is running or paused
animation-timing-function	Specifies the speed curve of the animation

Questions


Recycle Bin
a
c
b
chris
DIGITAL DREAMS
ICT ACADEMY
HOT LINE: 07064286765
4:04 PM