WEB INTERFACE DESIGN

HIGHER EDUCATION TRAINING CYCLE

WEB APPLICATIONS DEVELOPMENT

2024-2025

CHAPTER 4
USE OF STYLES

CLIL: Content and Language Integrated Learning

Teachers:
María Ángeles García Escrig
Alicia Fernández Catalán

Using Styles

Outline to get started

Web design encompasses many different skills and disciplines in the production and maintenance of websites.

The Importance of CSS in Web Development.

Cascading Style Sheets, commonly known as CSS, is an integral part of the modern web development process. It is a highly effective HTML tool that **provides easy control** over layout and presentation of website pages by separating content from design.

Objectives

1. Identify the possibilities of modifying HTML5 tags.
2. Being able to define styles directly. CSS3.
3. Being able to define and associate global styles in external sheets.
4. Being able to define alternative style sheets.
5. Identify the different properties of each element.
6. Being able to create style classes.
7. Being able to use style sheet validation tools.

Content Index

Related topics to research

- ***CLIL***
- Website builder
- CSS editor
- Selectors.
- **User experience design. UX**
- Web content Styles
- Web typography
- Web color

1. Introduction to CSS (cascading style sheet)

CSS (Cascading Style Sheet, Cascading Style Sheets)

Philosophy:
Use the HTML `<body>` tag to define the structures of the contents shown on the site (headings, paragraphs, bullet points, etc.) and then in another file or in the `<head>` of the HTML, the appearance of each page is defined using the CSS language.

Advantages:

- Change the contents that are placed in the `<body>` without affecting the appearance defined in the CSS file (or in the `<head>` of the HTML), or that the appearance in the CSS is changed without affecting anything included in the `<body>` of HTML.
- Adaptation of the websites to the devices with which it will be displayed.

2. SELECTORS: ONLINE STYLES BASED ON TAGS, CLASSES AND IDENTIFIERS

A CSS style sheet is made up of **rules** that indicate how the page will be displayed (style rules).

Each **rule** is composed of:

- One ***selector*** (or more) which indicate to which element or part of a page a certain style is applied.
- One or several `attribute: value` pair/s

```
selector{ attribute: value;}
```

2. 1. SELECTORS BASED ON TAGS(I)

```
selector{ attribute: value;}
```

The attribute refers to the characteristic that You want to modify the label, for example color.
The value refers to the instance of the attribute, for example, blue.

Example:

```
h1{ color: blue;}
```

2. 1. SELECTORS BASED ON TAGS(II)

CSS considers syntax for defining attributes for several selectors (1) and selectors with several attributes (2).

```
Selector1, selector2{ attribute1: value1; attribute2: value2;}
```

Examples:

- (1) If the same style is to be applied to two different selectors, the syntax would be:

```
h1, section, p{ color: blue;}
```

- (2) And if the same selector wants to apply different attributes:

```
div{ color: white;  background-color: blue;}
```

Activity 2.1

Examples:

1. Create an HTML5 file with three texts, the first one between <h1> </h1> tags, the second one between <h2> </h2> tags and finally, another between tags.
2. Later, include inside the <head> </head> section an style which affects to the <h1> and tags text color (color) and to the <h2> tags background color (background-color).
3. It should be shown as Figure 2.1 (text between <h1> and will be *navy*, and text inside <h2> will be *white* with *navy* background color).

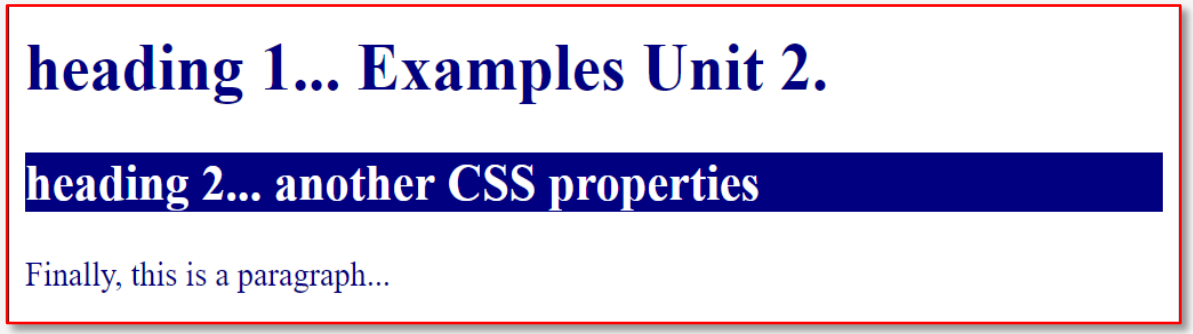


Figure 2.1

```
<head>
    <meta charset="utf-8" />
    <title>Act2.1</title>
    <style>
    h1, p{color:navy;}
    h2{color:white; background-color: navy}
    </style>
</head>
<body>
    <h1>heading 1... Examples Unit 2.</h1>
    <h2>heading 2... another CSS properties</h2>
    <p>Finally, this is a paragraph...</p>
</body>
```

2. 2. SELECTORS BASED ON CLASSES (I)

Classes associated with HTML tags are defined:

```
tagname.classname{ attribute1: value1; attribute2: value2; …}
```

This alternative is more powerful than the one we viewed with tag-based selectors, since it allows different styles to be applied to the same tags.
The most generic classes do not apply to any HTML tag, so in their description, the name of any tag is issued in the selector.

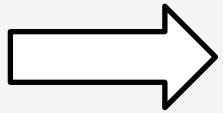
```
.classname{ attribute1: value1; attribute2: value2; …}
```

2. 2. SELECTORS BASED ON CLASSES (II)

Example:

```
<!DOCTYPE html>
<html lang="en">
     <head>
          <meta charset="utf-8" />
          <title>Act2.1</title>
          <style>
               h1.red {color: red;}
               h1.blue {color: blue;}
               h1.green {color:darkolivegreen; }
          </style>
     </head>

     <body>
          <h1 class="red"> A red heading </h1>
          <h1 class="blue"> now blue </h1>
          <h1 class="green"> And now green </h1>
     </body>
</html>
```



A red heading

now blue

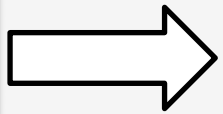
And now green

2. 2. SELECTORS BASED ON CLASSES (III)

Example:

```
<!DOCTYPE html>
<html lang="en">
<head>
     <meta charset="utf-8">
     <title>Example2.2.2</title>
     <style>
          .bluetext {color: #040195}
          .yellobackg{background-color: #FFE52A}
     </style>
</head>
<body>
     <h3 class = "bluetext yellobackg"> blue
     title, yellow background </h3>
     <p class = "bluetext"> blue paragraph;
     background:
     the one that inherits from the page. </p>
</body>
</html>
```

More than one class to the same tag



blue title, yellow background

blue paragraph; background: the one that inherits from the page.

Activity 2.2

Examples:

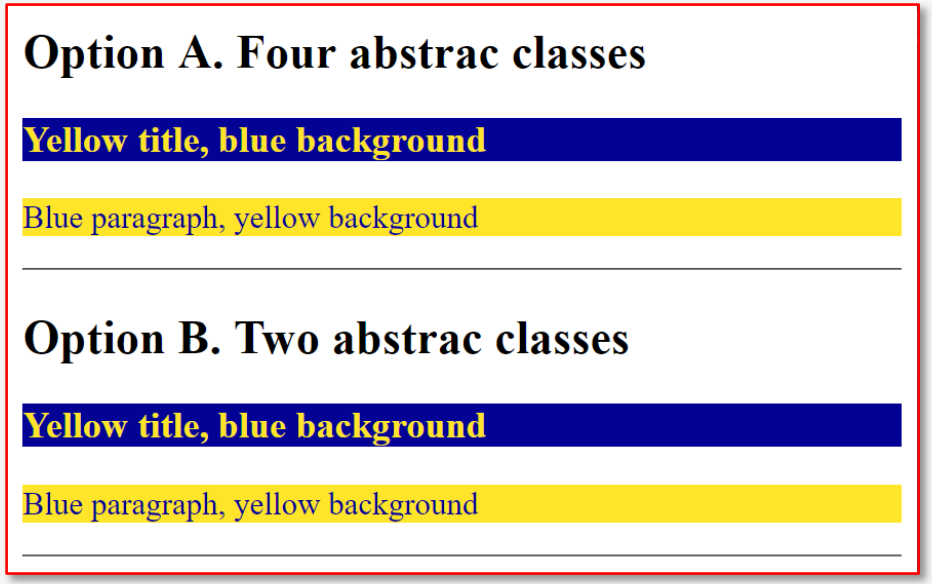
1. Create an HTML5 file with two texts, the first one between `<h3> </h3>` tags, and another between `<p> </p>` tags.
2. Then include, in the `<head>` section, styles with abstract classes that show: `<h3>` `with yellow` text and blue background and `<p>` `with` blue text and yellow background.

Solve the activity in two ways:

a) Using 4 abstract classes (2 for colors and two for backgrounds),

b) Using 2 classes: one called `yellowTblueB` and another called `blueTyellowB`

The result must be the following:



2. 3. SELECTORS BASED ON IDENTIFIERS(I)

Unlike classes, identifiers can only be used in a single element.

```
tagname#idname{ attribute1: value1; attribute2: value2; …}
```

Or just

```
#idname{ attribute1: value1; attribute2: value2; …}
```

As it can be seen, this statement is identical to that used to define class selectors. The only difference is that instead of using a dot "." a "#" pad is used. However, the meaning, the semantics of both expressions, is very similar.

2. 3. SELECTORS BASED ON IDENTIFIERS(I)

Example. Internal link to an identifier.

```
<style>
      section {
            background: #CCC;
            border: 1px solid #093;
            margin: 10px 12px 20px 15px;
            padding: 10px;
            height: 50vh;
      }

      #contentC{background: #EEE; color:#093;}
</style>
```

Selector based on identifier

Internal Link to an identifier. Anchor

```
<body>
      <a href="#contentC">Go to <b>Content C</b></a>
      <section>
            <h2>Content A </h2>
            <p>Here is some content A in a section tag</p>
      </section>
      <section>
            <h2>Content B </h2>
            <p>Here is some content B in a section tag</p>
      </section>
      <section id="contentC">
            <h2>Content C </h2>
            <p>Here is some content C in a section tag</p>
      </section>
</body>
```

Activity 2.3

Create an HTML file that defines a class selector (class) and a identifier (id).

Try using id and class in one and several HTML5 elements and check its effect.

We know that classes are used when the style has to be applied to more than one element, while identifiers are defined when "exclusivity“ is sought.

Assume that the specification is not met, and an identifier is repeated in various elements.

Answer these questions:

1. How does the browser interpret and respond?
2. And in another browser, …does it respond the same?
3. When we work with links and anchor points, what happens?
4. What happens when applying styles to elements with that identifier?

3. Grouping and nesting selectors

- ✓ Selectors can be grouped and nested to achieve CSS styles, on the one hand more concrete and defined, and on the other hand to have a CSS file more optimized and easy to understand by the development team.

3. 1. Grouping

Any selector can be grouped following this syntax, using "***comma***" between selectors:

```
Selector1, selector2{ attribute1: value1; attribute2: value2; …}
```

For example, if you want to apply exactly the same CSS properties to `<h1>, <p>` and `<article>` tags, the rule would be:

```
h1, p, article {font-family: arial; color: lightblue; }
```

The least optimized version of that same rule would be:

```
h1 {font-family: arial; color: lightblue; }
p {font-family: arial; color: lightblue; }
article {font-family: arial; color: lightblue; }
```

For tags, identifiers and classes would be:

```
#contentC, #welcome {font-style: italic}
span, .sport, #info{color: #FF0000}
```

3. 2. Nesting or contextual selectors (I)

Common nested selector: used to create rules over elements that are surronunded by other elements. The general syntax for two elements is as follows: using "***blank space***" between selectors:

```
selector1 selector2{ attribute1: value1; attribute2: value2; …}
```

Example **without nesting**:

HTML5:

```
<body>
    <h1><i><span> This h1 title in bold, italic and blue </span></i></h1>
    <section> <h2>The next "piece" of h2 title… <span> Does it also look blue? </span> </h2> </section>
</body>
```

CSS:

```
span {color: blue; }
```

Output:

This h1 title in bold, italic and blue

The next "piece" of h2 title… Does it also look blue?

3. 2. Nesting or contextual selectors (I)

Now, considering the general syntax for nested elements: using "***blank space***" between selectors:

```
selector1 selector2 selector3{ attribute1: value1; attribute2: value2; …}
```

For the same HTML5 code, consider the following ***rule for nested elements***:

HTML5:

```
<body>
 <h1><i><span> This h2 title in bold, italic and blue? </span></i></h1>
  <section> <h2> And the next "part" of h2 title... <span> Is it in blue? </span> </h2> </section>
</body>
```

CSS:

```
h1 i span {color: blue; } or even … h1 span {color: blue; } should work
```

New output:

This h1 title in bold, italic and blue?

And the next "part" of h2 title... Is it in blue?

3. 2. Nesting or contextual selectors (II)

Nesting of direct child selectors: to restrict the selection to those elements with a specific parent. Elements that are not **directly** a child of the specific parent are not selected

```
selector1 > selector2 > selector3{ attribute1: value1; attribute2: value2; …}
```

HTML5:

```
<body>
<h2><i><span> This h2 title in bold, italic and blue? </span></i></h2>
    <section> <h2> And this “part” of h2 title… <span> Is it in blue? </span> </h2>
    </section>
</body>
```

CSS:

```
h2 span {color: blue; }
```

Output:

This h2 title in bold, italic and blue?

And this "part" of h2 title... ¿Is it in blue?

```
h2 > span {color: blue; }
```

This h2 title in bold, italic and blue?

And this "part" of h2 title... ¿Is it in blue?

3. 2. Nesting or contextual selectors (III)

Nesting adjacent selectors. Siblings. This type of nesting is used when you want to apply a style to an element that is adjacent (next) to another element at the same level of HTML nesting.

```
Selector1 + selector2{ attribute1: value1; attribute2: value2; …}
```

Conditions:

- `selector1` and `selector2` must be siblings… having the same parent
- `selector2` must be inmediatly after `selector1`

3. 2. Nesting or contextual selectors (III)

Nesting adjacent selectors. Siblings.

Examples

```
Selector1 + selector2{ attribute1: value1; attribute2: value2; …}
```

```
<head>
<meta charset="utf-8">
<title>Example3.2.3.a</title>
<style>
            i+b {color: red;}
</style>
</head>


<body>
      <p> <i> Note </i>, this is a <b> warning </b> </p>
      <b> Read carefully </b>
</body>
```

Note , this is a **warning**

Read carefully

```
<head>
<meta charset="utf-8" />
<title>Example3.2.3.b</title>
<style>
            h1+h2 {color: green}
</style>
</head>
<body>
      <h1> Title1 </h1>
      <h2> Subtitle H2 in green </h2>
      <p> ... </p>
      <h2> Another subtitle H2 in black </h2>
      <p> ... </p>
</body>
```

Title1

Subtitle H2 in green

…

Another subtitle H2 in black

…

Activity 2.4

This activity is highly recommended.

You'll find the activity description and the necessary files to develop it in the virtual platform.

File: **activity2.4_EN.zip** (in the moodle/virtual class)

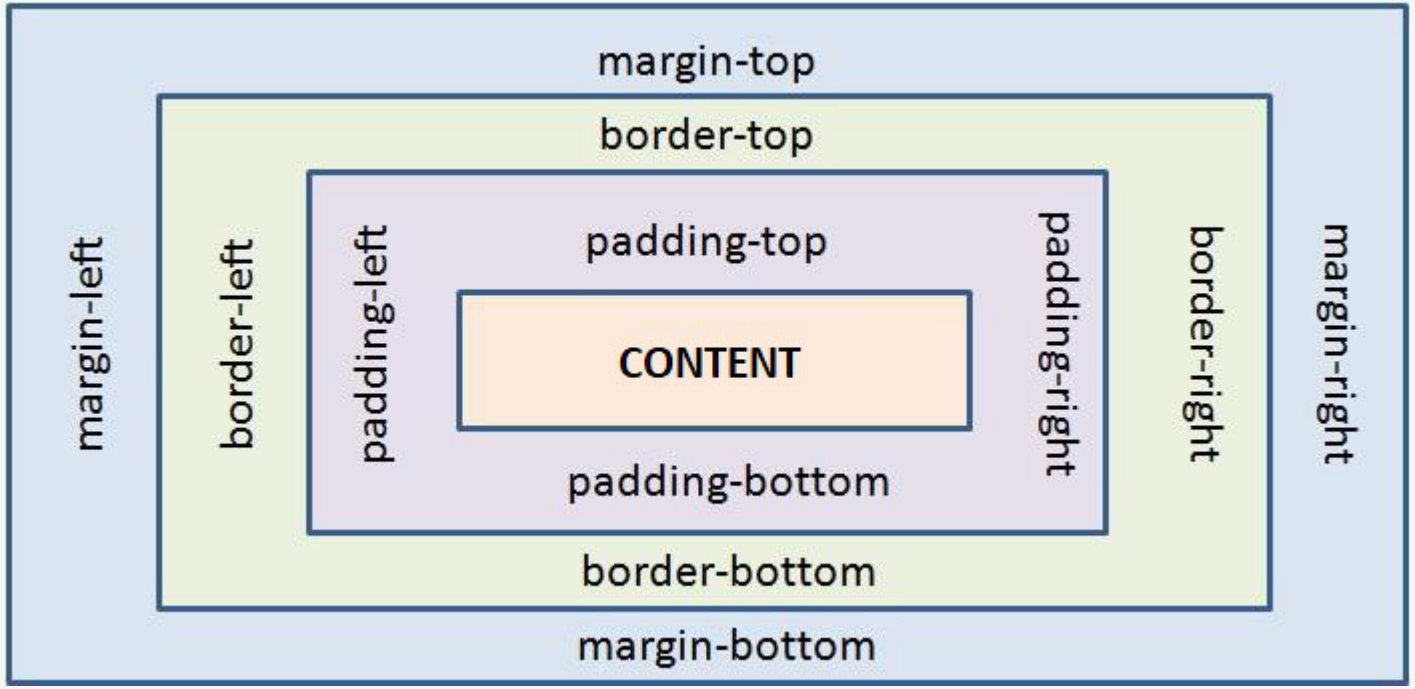
4. GOOD PRACTICES WHEN WRITING CSS

- Selectors are named in **lowercase**, never starting with special or numeric characters.
- The name of the selectors must be **specific and clear**, so that it has a greater expressive capacity.
- The name of classes and identifiers **should not describe a visual characteristic**, such as color, size or position.
- Names must follow more a **semantic** than a structural vision.
- **Separate** the words using hyphens or capital letters.
- Do **not** make **excessive** use of classes and identifiers.
- **Group** the rules according to your selector whenever possible.
- At the beginning of a CSS it is advisable to define the tag selectors.
- Visually **structure** the attributes.

5. ATTRIBUTES. BOX MODEL.

To better understand CSS styles, consider that any HTML5 element is a box whit the following attributes:

Simplifying, a web page is a **box of boxes**.



5. ATTRIBUTES. BOX MODEL.

Related LINKS

W3Schools CSS Box Model

W3Schools CSS Box Model Video-Tutorial

5. 1. ATTRIBUTES. MEASUREMENT UNITS.

Absolute:

- Inches (In). One inch = 2.54 cm.
- Centimeters (Cm).
- Milimeters (Mm).
- Points (Pt). One point = 1/72 of an inch.
- Pikes (pc). A pica = 12 points.

Relative:

- Pixels
- Em (height (font-size) of the letter of the element in which it is used)
- Viewport vw y vh.

Examples:

```
nav {font-size: 15px;}
p {font-size: 1.2em;}
```

Related LINKS:

CSS Viewport Units: A Quick Start

5. 1. ATTRIBUTES. MEASUREMENT UNITS.

Examples:
Consider these three rules:

```
1. p { font-size:14px; text-indent: 2em; }
2. div { font-size:20px; width:80vw; height:30vh; background: navajowhite;}
3. div p { font-size:1.2em;}
```

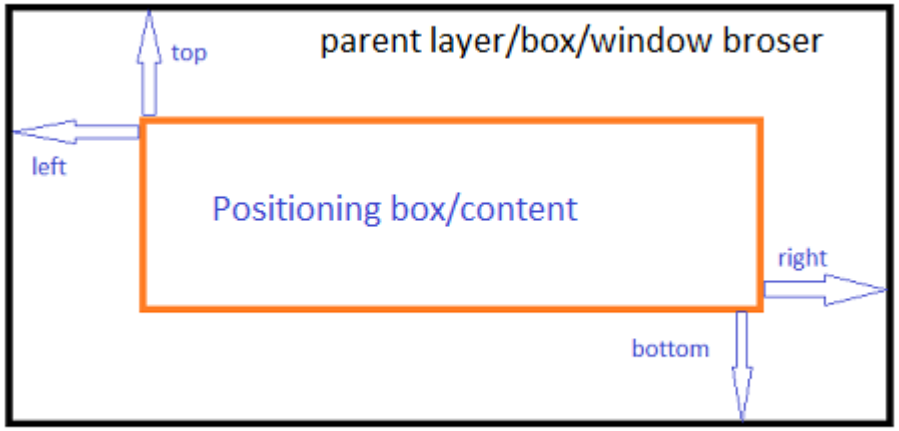
Analysis
(1) As `font-size` is `14px`, `text-indent` will be `28px (14px * 2 em)`.
(2) y (3) Now, consider `<p>` inside `<div>`,

- `p font-size` will be 24px, it is 20% bigger than `div font-size`
- `div width` is `80` parts of 100 from viewport width `vw`; `width:80vw;`
- `div height` is `30` parts of 100 from viewport height `vh`; `height:30vh;`

5. 2. ATTRIBUTES. POSITION.

`top`, `left`, `right` **and** `bottom`.

`top` (from above) and `bottom` (from below) indicate the vertical **distance** where the layer will be placed and `left` (from the left) and `right` (from the right) the horizontal **distance**.



5. 2. ATTRIBUTES. POSITION.

However, the "top" distance is subject to the positional attribute that defines the type of position of the layer. If the position attribute is:

1. fixed: top indicates the distance of the top edge of the layer from the top edge of the browser viewport. The fixed layer doesn't move in its position even if the vertical scroll moves.
2. absolute: top indicates the distance of the top edge of the layer from the top edge of the parent layer if it is not static, otherwise same as fixed is. But absolute layer moves its position if the vertical scroll does.
3. relative: top indicates the distance from where the page was currently being written to the top edge of the layer.
4. static: doesn't consider any positional attribute.
5. Sticky: top indicates the place where the sticky element will be fixed.

Activity 2.5

This activity is highly recommended.

You'll find the activity description and the necessary files to develop it in the virtual platform.

File: **Activity2.5_EN.zip**

5. 3. ATTRIBUTES. margin, padding, border.

margin	border	padding
marks the separation between another box and the border of which it is represented	defines the style and color of the edge of the box. size, style and color included in the same attribute	is used to generate space around an element's content, inside of any defined borders

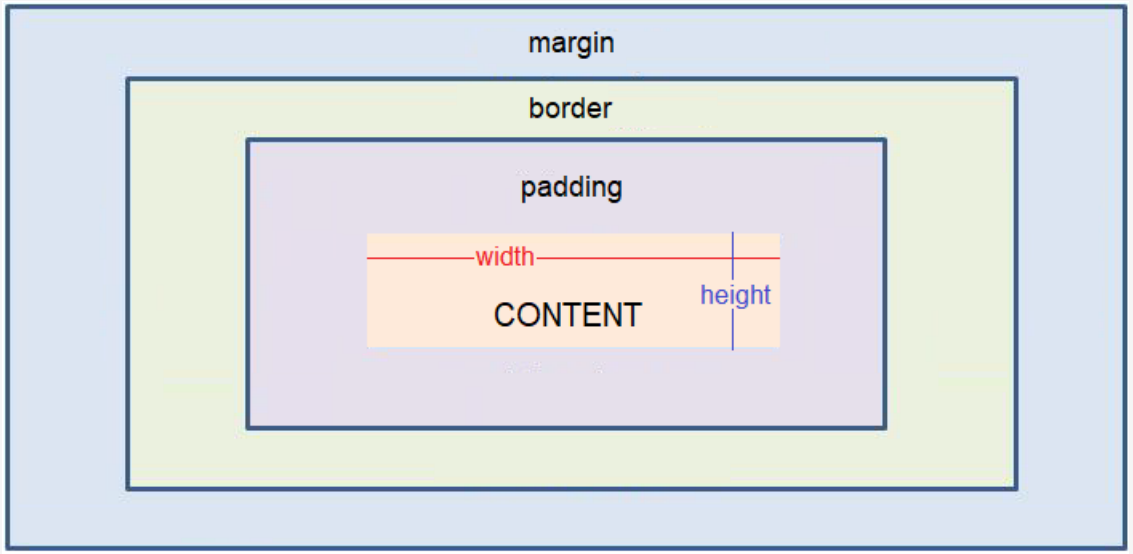
`margin-top: 100px;`
`margin-right: 50px;`
`margin-bottom: 10px;`
`margin-left: 100px;`
	`border-top: 3px dotted red;`
	`border-right: 2px solid blue;`
	`border-bottom: 3px double green;`
	`border-left: 3px groove red;`
		`padding-top: 50px;`
		`padding-right: 10px;`
		`padding-bottom: 25px;`
		`padding-left: 50px;`

`margin: 100px 50px 10px 100px;`	`border: 2px solid blue;`	`padding: 50px 10px 25px 50px;`
`margin: 100px;`		`padding: 100px;`
`margin: 100px 50px;`		`padding: 100px 50px;`
https://www.w3schools.com/css/css_margin.asp	https://www.w3schools.com/css/css_border.asp	https://www.w3schools.com/css/css_padding.asp

5. 4. ATTRIBUTES. Content.

`width` sets the distance between the limit of the padding-left and the padding-right

`height` the same but between padding-bottom and padding-top.



The `height` and `width` properties do not include padding, borders, or margins. It sets the height/width of the area inside the padding, border, and margin of the element.

Other attributes:

- ✓ All attributes available in CSS 2.1 of the W3C:
  http: //www.w3c.es/divulgacion/guiasreferencia/css21
- ✓ This other URL shows the new CSS3 additions, in which you can see border-radius.
  http: //www.css3.info/preview/

5. 5. CSS Property: *float*

The **float** CSS property specifies how an element should float.

The **float** CSS property places an element on the left or right side of its container, allowing text and inline elements to wrap around it. The element is removed from the normal flow of the page, though still remaining a part of the flow (in contrast to absolute positioning).

Download from the platform the following file: ***Annex2.float-and-clean.pdf (Spanish)***

Mandatory reading!!
Coming next fortnight (3)

Related LINKS

w3schools float property

float property - cleaning methods

Activity 2.6. float property

This activity is highly recommended.

It is very important and necessary to read the ***Annex2.float-and-clean.pdf (spanish)*** document before trying to develop this activity.

You'll find the activity description and the necessary files to develop it in the virtual platform.

File: **activity2.6.zip**

This content will be studied in more depth in the fortnight 3. Leave this activity for next fortnight.

6. ELEMENTS: BACKGROUND COLORS, TEXTS, LINKS, LISTS, TABLES, VISIBILITY, IMAGES

Font attributes	*color, font-size, font-family, font-weight, font-style…*
Paragraph attributes	*line-height, text-decoration, text-align, text-indent, text-transform.*
Background Attributes	*background-color, background-image.*
Tables attributes	*caption-side, table-layout, border-collapse, border-spacing, empty-cells,*
Visibility attributes	*overflow, clip, visibility, display.*
List Attributes	*list-style-type, list-style-image, list-style-position.*
Link attributes	•***Normal links*** *A:link {attributes}* •***Visited links*** *A:visited {attributes}* •***Hover links*** *A:hover {attributes}* (When the mouse is above them) •***Active links*** *A:active {attributes}* (The links are active at the precise moment you click on them) ▪ `text-decoration:none;` to define links without underlining

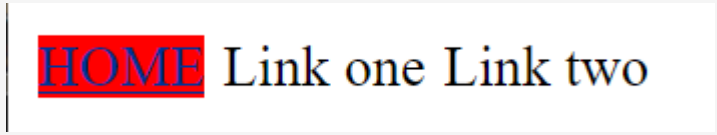
6. ELEMENTS: BACKGROUND COLORS, TEXTS, LINKS, LISTS, TABLES, VISIBILITY, IMAGES (II)

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Example6:links</title>
  <style>
   /*Link not visited */
   a:link {
     text-decoration: none;
     color: #000000;
     border: #FFFFFF 1px solid;
   }
   /*Link visited */
   a:visited {
     text-decoration: none;
     color: blue;
   }
   /*Mouse over the link */
   a:hover {
     text-decoration: underline;
     color: #003399;
     background: red;
     border: #FFFFFF 1px solid;
   }
   /*Active link */
   a:active {
     text-decoration: none;
     color: white;
     background: green;
     border: #FFFFFF 1px solid;
   }
</style>
</head>
```

Links

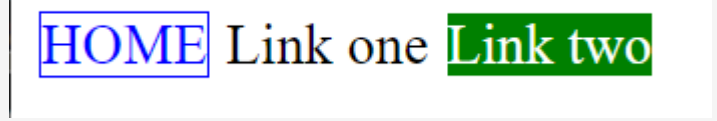
```
  <body>
      <header>
        <h1>Example 6: links</h1>
      </header>
      <nav>
        <a href="Example6.links.home.html"> HOME</a>
        <a href="Example6.links.wp1.html"> Link one </a>
        <a href="Example6.links.wp2.html"> Link two </a>
      </nav>
      <main>
        <h2>HOME MAIN CONTENT </h2>
      </main>
      <footer>IES CAMP 2019</footer>
  </body>
</html>
```

a:hover



a:visited a:hover

a:visited a:active



Activity 2.7. Float property. (II)

This activity is highly recommended.

It is very important and necessary to read the ***Annex2.float-and-clean.pdf (spanish)*** document before trying to develop this activity.

You'll find the activity description and the necessary files to develop it in the virtual platform.

File: **activity2.7.zip**

This content (***float*** property) will be studied in more depth in the fortnight 3. This activity **2.7** is the continuation of activity 2.6, therefore, we will also leave it for the next fortnight (3)

7. SUPERPOSITION AND PREVENTION OF STYLES

Controlling these aspects means:

- ✓ understanding the functioning of the CSS and
- ✓ achieving very precise and personal results.

We already know that:

1. HTML is a "language" interpreted by the browser
2. We have to have clear the "box model“:
   - ***inline*** boxes are placed next to each other, while they "fit" (horizontally) inside the container box.
   - ***block*** boxes are placed one below the other by introducing a line break.
3. This normal flow of box positioning can be "broken" using the position attributes in CSS: ***top, right, bottom y left***

7. 1. BOX OVERLAY (I)

CSS allows you to control the ***depth of the boxes*** by determining their order of superposition.

The ***z-index*** attribute allows you to define the depth level of a box.
Its value is an integer:

- The value 0 is usually taken as the lowest level.
- The higher the value, the closer the layer will be shown to the user on the web

The ***z-index*** attribute only has an effect if it is accompanied by a very important one to determine the position of one box relative to the others. This property is called ***position.***

7. 1. BOX OVERLAY (II). *Position* attibute

The ***position*** property defines the type of position of the boxes and can have these values:

- static
- relative
- absolute
- fixed and
- sticky.

Related LINKS

w3schools CSS position property

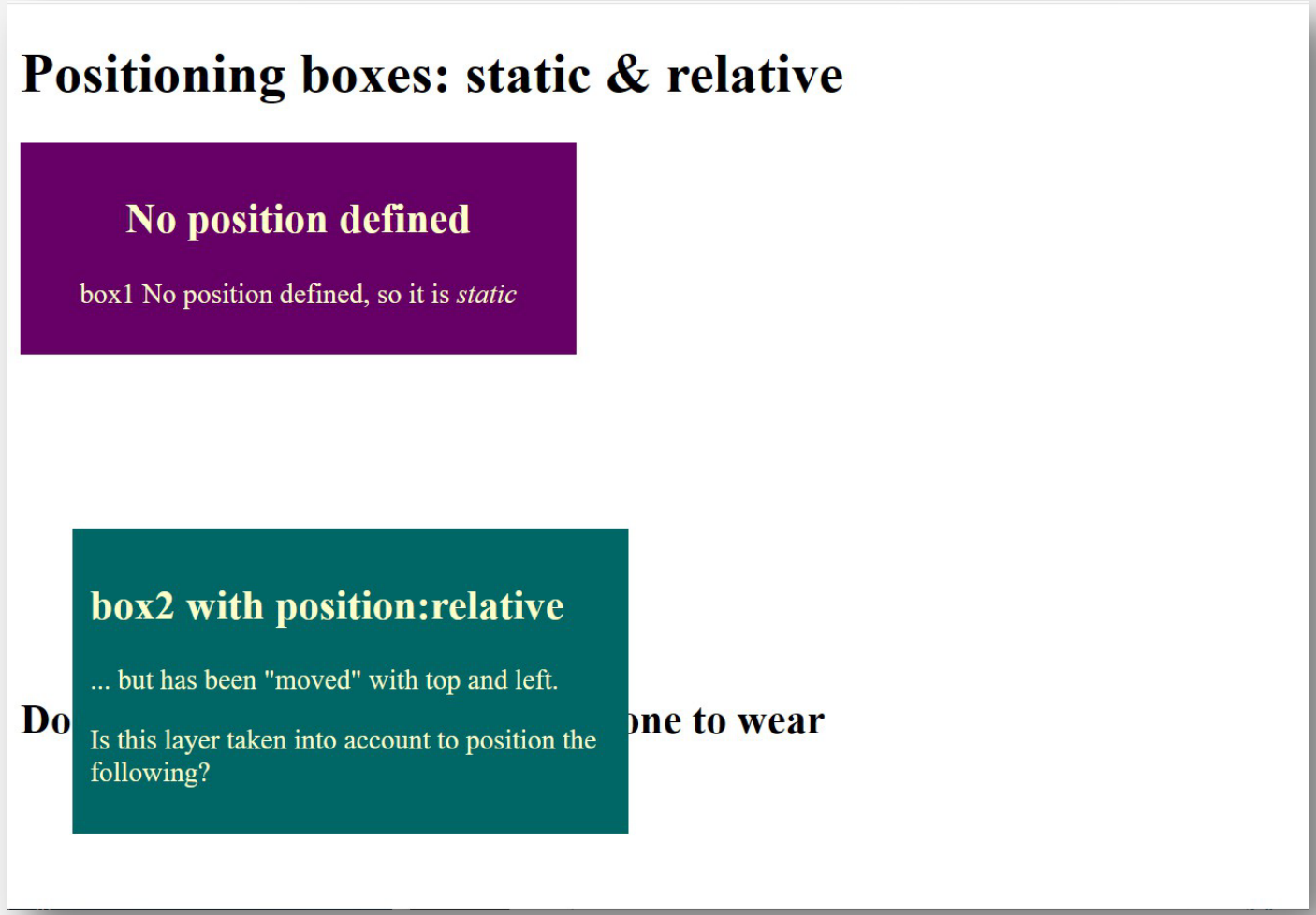
7. 1. BOX OVERLAY (III)

Example 7.1

Analyze Example7.1.a.position.html

1. `box1` and `box3` are out of the `body` "space" (border)
2. What happens if `body`'s property `position` is set to `position:relative;` ?

7. 1. BOX OVERLAY (IV)



Example 7.2

Analyze
Example7.2.a.position.html

Does the relative positioning of box2 influence the display of the next `h2` layer?

45

Activity 2.8. z-index property

This activity is highly recommended.

You'll find the activity description and the necessary files to develop it in the virtual platform.

File: ***Activity2.8.2019_ES.html***

Activity 2.9. z-index property

This activity is highly recommended.

You'll find the activity description and the necessary files to develop it in the virtual platform.

File: ***Activity2.9.2019_ES.html***

7. 2. PRECEDENCE OF STYLES

The precedence of styles is associated with the concept of specificity of a rule.
Specificity refers to the weight of each element of a style sheet:

- The more weight more specificity.
- The more specific a rule has the less problems when it comes to ensuring that it will be that and not the rule that applies to a certain content.

A simple procedure to calculate the specificity of a rule is to add the points according to the type of selectors it contains:

- ✓ A **tag selector** is given a value of **1 point** (for example, h1, p, div).
- ✓ A **class selector** is given the value of **10 points.**
- ✓ An **identifier selector** is given a value of **100 points.**
- ✓ A **style attribute** that is given a value of **1000 points.**

7. 2. PRECEDENCE OF STYLES (II)

Example:

```
<style>
    p {background: crimson;} /* Specificity of 1 points */
    .paragraph {background: pink;} /* Specificity of 10 points */
    p.paragraph {background: maroon;} /* Specificity of 11 points */
    #id-paragraph {background: orange;} /* Specificity of 100 points */
    p#id-paragraph {background: red;} /* Specificity of 101 points */
    p.paragraph#id-paragraph {background: green;} /* Specificity of 111 points */
</style>
```

8. CREATE AND LINK STYLE SHEETS

❑ The first alternative is to use the style attribute within HTML tags (integrated style rules). For example:

```
<p style = "background: black;"> The background of this paragraph will be black </p>
```

❑ Another alternative is to use the `<style>` tag within the same file (embedded style rules).

- ✓ Type
- ✓ Media
- ✓ Title

9. CREATE AND LINK STYLE SHEETS IN EXTERNAL CASCADE

- An external style sheet can be linked to an HTML document using the `<link>` tag that is placed on the `<head>` of the page
- The `rel` attribute is used to define the relationship between the linked file and the HTML document.
  - ✓ `rel="stylesheet"` specifies a persistent or preferred style.
  - ✓ `rel="alternate stylesheet"` defines an alternative style
- Another alternative to link external CSS is to use the @import rule. This rule is included within the <style> tags. An example of how to use this rule is as follows:

```
<style> @import url('/css/mystyle.css'); </style>
```

Recommended Activity 2.10. As this activity is based in activity **2.7** we will also leave it for the next fortnight (3)

10. TOOLS AND VERIFICATION TEST

- W3C: http://jigsaw.w3.org/css-validator/.
- Pug-in Firefox
- XHTML-CSS
- Firebug
- Epenule
- List-o-matic
- CSS Layout Generator
- CSS Text Wrapper

WEB INTERFACE DESIGN

HIGHER EDUCATION TRAINING CYCLE

WEB APPLICATIONS DEVELOPMENT

2024-2025

CHAPTER 4
USE OF STYLES

CLIL: Content and Language Integrated Learning

Teachers:
Alicia Fernández Catalán
María Ángeles García Escrig