ACTIVITY 3: FLEXBOX

Presentation

Flexbox is a CSS layout model designed to create more flexible and responsive layouts by aligning items in a container with ease. It allows items to be arranged in rows or columns, adjusting their size dynamically to fill available space, even when the container size changes. The flex container defines how its children (flex items) are positioned, using properties like flex-direction, justify-content, and align-items. Flexbox simplifies alignment, spacing, and distribution of elements, making it ideal for building modern, adaptive web layouts. Unlike traditional CSS layouts, Flexbox is one-dimensional, working in either a horizontal or vertical direction, offering much better control over the positioning and size of elements.

Skills

Generals

- Structure the content of a web page and define the visual design of an HTML page and layout

Competencies of the *disseny d'interfices web* subject

- CP 02: Apply modern CSS styles, CSS preprocessors and frameworks to develop web content in valid and coherent documents

Objectives

The objectives of this activity are:

- **Create Responsive Layouts**: Flexbox allows elements to adjust their size and position dynamically, making it ideal for creating layouts that adapt seamlessly to different screen sizes and devices.
- **Simplify Alignment and Spacing**: Flexbox makes it easier to center, align, and distribute space between elements, reducing the need for complex CSS rules and making layout development more efficient.

Description of the activity

This activity consists of carrying out 4 activities.

Exercise 1. Froggy (30')

Practice Flexbox basics with Froggy (24 Exercises)

To do the rest of the exercises, download the file flexbox.rar

Exercise 2. Flexbox Boxes (1h)

Given the file flexbox-boxes.html and flexbox-boxes.css, add the CSS needed to get the layout that you can see in the file captures/flexbox-boxes.png

Requirements

1. **Do not change the HTML!**
2. Use **flexbox CSS properties** whenever it's possible

Tricks:

- Use :nth-child() to identify the boxes individually if needed
- Use flex-grow or flex-shrink if needed

Exercise 3. Flexbox Cards (45')

Given the file **flexbox_cards.html** and **flexbox-cards.css**, add the CSS needed to get the layout that you can see in the captures (captures folder) and the behavior you can see in this video.

Requirements

1. Do not change the HTML and CSS, just add the CSS lines where indicated.
2. Use flexbox CSS properties wherever it's possible

3. The images are aligned in the top of the card
4. The buttons are aligned in the bottom of the card

Tricks:

- Use padding
- Use border-box if needed

Exercici 4. Flexbox Holy Grail (30’-1h)

Given the file **flexbox-holygrail.html** modify the CSS inside the file to achieve a layout for **desktop** (1200px) and **mobile** (Samsung S9) available in the captures folder.

Requirements

1. Do not change the HTML and CSS, just add the CSS lines where indicated.
2. Use flexbox CSS properties wherever it’s possible
3. The articles must be show in the same order as in the image
4. Remember to use *box-sizing* property if necessary!

Related readings and resources

Classroom Resources:

- CSS flexbox

Evaluation criteria

1. Use of Flexbox PropertiesThe html structure has not been modified
    - Flexbox properties are applied consistently and correctly to achieve the desired layout in all exercises. The visual replica is almost identical to the original design
2. Adherence to InstructionsPosition is used correctly
    - Follows instructions precisely without modifying HTML or CSS files other than where specified. The code is well commented, easy to read, with proper indentation and descriptive class names.
3. Layout Accuracy
    - Final layout closely matches the images or video provided in the exercise.
4. Use of Advanced Flexbox Features
    - Proper use of advanced properties like flex-grow, flex-shrink, :nth-child(), and box-sizing.
5. Responsiveness and Behavior
    - Layout adapts perfectly to different screen sizes (desktop/mobile) as required.

Format and date of delivery.

The final delivery will be made in "Classroom".

Delivery format: in a zip, rar… with the files HTML and CSS

Nomenclature: SurnamesNom_DIW_UT2A3.pdf

Note: Intellectual property

It is often unavoidable, when producing a multimedia work, to make use of resources created by third parties people. It is therefore understandable to do it within the framework of a practice of the *desenvolupament d'aplicacions web*, as long as this is clearly documented and does not involve plagiarism in practice.

Therefore, when presenting a practice that makes use of external resources, it must be presented along with it a document detailing all of them, specifying the name of each resource, its author, the place where it was obtained and its legal status: whether the work is protected by copyright or under some other license of use (Creative Commons, GNU license, GPL...). The student must ensure that the license does not prevent specifically its use within the framework of practice. If the information is not found corresponding must assume that the work is protected by copyright.