John Dudas

1. What are CSS grid and Flexbox? When might you use one over the other?

A grid is a set of intersecting lines both horizontally and vertically. They define the columns and rows. Elements are placed into the grid with these lines. You can create a grid with fixed sizes or flexible sizes using percentages. Items can be placed into a precise location on the grid using line numbers, names, or by targeting an area of the grid. This gives you more precise control and spacing of the lines.

Grids are held within a Grid Container by declaring *display: grid* on an element. When this is done, all direct children of that element become grid items. Grids can have rows and columns defined with template rows or columns.

Flexbox is more like working with two axes. The main axis and the cross axis. Flexbox uses lines appearing one under the other. The area the flexbox uses is called the flex container. Change the area’s *display* property to *flex* and all the children of that container become flex items. You can control the container lines, and area by using inline flex or block flex.

The difference between the two is that Flexbox is a one-dimensional layout while the grid layout is a two-dimensional layout. Flex tries to line everything up with each other irrelevant to the placement of items. Grid allows us to control the layout in both rows and columns. The majority of the sizing is done on the container, in flexbox, you create a flex container and set the direction at that level. Which one you select is really up to you, there is no right or wrong.

<https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_flexible_box_layout/Relationship_of_flexbox_to_other_layout_methods>

1. Explore the CSS box model and its significance in determining the layout and spacing of elements on a webpage. Research the components of the box model, including content, padding, border, and margin.

The CSS box model is used by the browser’s rendering engine to represent each element as a rectangular box. The CSS determines the main factors of these boxes as size, position, and properties. There are four main parts, the Content Edge, Padding Edge, Border Edge and Margin Edge. The Content Area has the real element content such as images or text. The Padding area contained by the Padding Edge extends content to include the element’s padding. Thickness is determined by the padding-top, padding-right and other padding properties. The Margin area is the empty area within the margin edge to separate the element from its neighbors.

Using the Box model will help developers place their elements within the page relevant to how they want them to look with other elements. It defines spaces, and margins between boxes and margins of the webpage. Placing each from the outside Margin Edge to Border Edge and then Padding edge will move the text, image or media for the developer.

<https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_box_model/Introduction_to_the_CSS_box_model>