Rohit Roy

February 4th, 2024

Internet and Web Programming (CISC-3300-R01)

Fordham University

**Homework 3**

1. See hw3.css
2. See name.html
3. In web design, **margin**, **padding**, and **border** are fundamental CSS properties that play crucial roles in controlling the layout and appearance of elements. **Margin** determines the space outside an element's border, influencing its positioning in relation to neighboring elements. **Padding** dictates the space between an element's content and its border, shaping the internal spacing. **Border**, on the other hand, establishes a visible perimeter around an element. Margins and paddings can be adjusted individually for each side, providing directional control, while borders have separate properties for width, style, and color. Together, these properties enable designers to structure content, manage spacing, and define the visual boundaries of elements, contributing to the overall aesthetics and functionality of a webpage.   
   1. **display: flex**: This rule is applied to the container, and it establishes a flex container. The direct children of this container become flex items, and a flex context is created for them. This allows you to use flex properties to control the layout and alignment of the items.
   2. **flex-direction**: This property sets the main axis of the flex container, defining the direction in which the flex items are placed. It can have values like **row** (default), **row-reverse**, **column**, or **column-reverse**.
   3. **flex-wrap**: This property controls whether the flex container should wrap its content if it overflows the container. It can take values like **nowrap** (default), **wrap**, or **wrap-reverse**.
   4. **justify-content**: This property aligns the flex items along the main axis of the flex container. It determines how the extra space is distributed. Common values include **flex-start**, **flex-end**, **center**, **space-between**, and **space-around**.
   5. **align-items**: This property aligns the flex items along the cross axis of the flex container. It defines how the items are aligned when there's extra space in the cross axis. Common values are **flex-start**, **flex-end**, **center**, **baseline**, and **stretch**.
   6. **align-content**: This property is similar to **align-items**, but it applies to multiple lines of flex items. It defines how the lines are spaced along the cross axis. Common values are **flex-start**, **flex-end**, **center**, **space-between**, **space-around**, and **stretch**.
4. see cs\_webpage.html