M4.252 - HTML y CSS aula 1

PEC3

Estudiante: Ivan Lenin Schartun Brito

Primera parte

Pregunta 1. Sobre técnicas de posicionamiento y modos de diseño, responded a las siguientes preguntas:

* 1. **Explicad los siguientes tipos de posicionamiento: estático, relativo, absoluto y fijo.**

Respuesta. –

* *Posicionamiento estático*: Es el valor por defecto que poseen cada uno de los elementos HTML, básicamente indica que el ítem no se va a mover a ninguna posición distinta a la del flujo normal del documento:

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* *Posicionamiento relativo*: Cuando se asigna este valor a un elemento, este cambiara su posición con respecto a su ubicación inicial en el flujo del documento. Dicha traslación dependerá de los valores que se coloquen en las propiedades adicionales y relacionadas a *position:relative;* las cuales son: *top, left, right bottom.* Los contenidos adyacentes al ítem con la propiedad relativa no se verán afectados.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  | Relativa a la imagen de arriba |

* *Posicionamiento absoluto*: Cuando se asigna este valor a un elemento, este cambiara su posición con respecto al elemento ancestro del mismo. En el caso de que un elemento no tenga un ancestro se utilizara el bloque contenedor inicial. Al igual que en el caso anterior para que la propiedad tenga efecto se debe incluir *top, left, right bottom.*

Una forma de asignar un ancestro al elemento es indicar que el contenedor tiene una posición relativa.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  | La división interna se mueve dentro de la división externa que es su ancestro |

* *Posicionamiento fijo*: Cuando se asigna este valor a un elemento este se ubicará en relación con el *viewport.* Esto quiere decir que siempre permanece en el mismo lugar con respecto a la ventana incluso si se desplaza la página. De igual forma las propiedades *top, left, right bottom* son necesarias para lograr el posicionamiento.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* 1. **Explicad para qué sirven las siguientes propiedades del módulo CSS** **Grid: grid-template-columns, grid-template-rows, grid-gap y place-content.**

Respuesta. –

* *grid-template-columns*: Esta propiedad permite definir tanto el número, como ancho de las columnas que contendrá la cuadrícula. Los valores que se usan en la propiedad van separados por espacios; cada uno de ellos especifica el tamaño de la columna respectiva.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* *grid-template-rows :* Muy similar a la propiedad anterior, este elemento permite definir tanto el número, como el ancho de las filas que contendrá la cuadrícula. Los valores que se usan en la propiedad van separados por espacios; cada uno de ellos especifica el alto de la fila respectiva.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* *grid-gap:* Esta propiedad permite definir el tamaño del espacio entre cada fila y columna dentro de la cuadrícula. Es una abreviación de las propiedades hijas grid-row-gap y grid-column-gap, la primera referida a filas y la segunda a columna.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* *place-content:* Es una propiedad abreviada, del contenedor padre, que permite posicionar los elementos de la cuadricula según su alineación y justificación de contenido. El primer valor establece align-content y el segundo valor establece justify-content. Si solo se proporciona un valor, establece ambas propiedades.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* 1. **Explicad para qué sirven las siguientes propiedades del módulo CSS Flexbox: flex-direction, flex-wrap, order y align-self.**
* *flex-direction*: Es una propiedad que permite definir la dirección en la que se dispondrán los elementos dentro de un contender flexible (elementos flexibles), indicando el eje principal del layout. Dicha propiedad puede tomar valores como: row, row-reverse, column, columna reverse
  + row: organiza los elementos del contenedor horizontalmente de izquierda a derecha.
  + row-reverse: organiza los elementos del contenedor horizontalmente de derecha a izquierda.
  + column: organiza los elementos del contenedor verticalmente de izquierda a derecha.
  + column-reverse: organiza los elementos del contenedor verticalmente de derecha a izquierda.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* *flex-wrap*: Es una propiedad que indica si los elementos flexibles deben ajustarse o no al tamaño de su contenedor. Por defecto el valor es *nowrap*, sin embargo, puede tomar valores como *wrap y wrap-reverse*

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* *flex-order*: Es una propiedad que especifica el orden de un elemento flexible en relación con el resto de los elementos flexibles dentro del mismo contenedor, sobre el eje en que están dispuestos.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

* *align-self*: Es una propiedad que permita dar alineación a un elemento dentro del contenedor flexible. La misma puede manejar los siguientes valores:
  + flex-start: el elemento flexible se alineará verticalmente en la parte superior del contenedor.
  + flex-end: el elemento flexible se alineará verticalmente en la parte inferior del contenedor.
  + flex-center: el elemento flexible se alineará verticalmente en el centro del contenedor.
  + Stretch: el elemento flexible se alineará verticalmente de manera que ocupe todo el espacio vertical del contenedor.
  + baseline: el elemento flexible se alineará en la línea de base del eje transversal.

|  |  |  |
| --- | --- | --- |
| **HTML** | **CSS** | **Live** |
|  |  |  |

Pregunta 2. Sobre diseño responsivo, responded a las siguientes preguntas:

* 1. **Explicad en breves palabras qué es el diseño responsivo (responsive design).**

Respuesta. –

* El diseño web responsivo (RWD) es un enfoque implementado en el desarrollo de sitios web que considera los diferentes tipos de dispositivos que un usuario puede usar para acceder al sitio. La idea de este diseño es ajustar la forma en que se muestra el contenido de una página de acuerdo con las dimensiones de la pantalla del dispositivo, ofreciendo de esta forma una experiencia usuario adecuada al portal de presentación.
  1. **Enumerad al menos cuatro características y/o técnicas que ayudan a realizar un diseño responsivo.**

Respuesta. –

* *Fluid layout (diseño fluido):* Es una técnica en la que el diseño de la página cambia de tamaño a medida que cambia el tamaño de la ventana. Esto se logra mediante la definición de áreas de la página utilizando porcentajes en lugar de anchos de píxeles fijos.
* *Imágenes flexibles*: básicamente es una técnica que busca adaptar este tipo de contenido a la pantalla donde se presentan, ajustando sus dimensiones sin perder la relación de aspecto.
* *Meadia Queries*: es una técnica de CSS que permite que las propiedades de CSS dentro de la regla estén activas si se cumplen ciertas condiciones. Las condiciones pueden ser varias cosas, como la resolución de la pantalla, la profundidad del color, la relación de aspecto y mucho más.
* *Tipografía responsiva:* es una técnica de css en la cual mediante el uso de *media queries* o funciones de css como clamp(), cal() se puede hacer una adaptación del tamaño de una fuente según el dimensionamiento que sea necesario en el momento de presentación.
  1. **Explicad brevemente qué se entiende por "mobile first".**

Respuesta. –

* Es una estrategia de diseño que nos indica que al momento de crear un sitio web, se debe comenzar con un prototipado o esquematización enfocada en pantallas pequeñas inicialmente para luego dar paso al trabajo con *viewports* más grandes. Esencialmente, se trata de ofrecer la experiencia de usuario correcta al dispositivo correcto, tomando en consideración el gran auge que tiene los dispositivos portátiles de pequeñas dimensiones.