**1st Video**

Units:

Px(pixel): fixed. Pretty self explanatory .Absolute length in pixel.

%: Relative to the value of parent element. 100% is the width of parent element.

Em: Relative to the font-size of the parent element

vh: equal to 1% of the height of the browser window size.

rem: Relative to font-size of the root element.

vw: equal to 1% of the width of the browser window size.

Absolute mane fixed mane egula responsibe hobena jototuku hobe tototuku dekhabe.

Relative % em rem use korbe

Absolute

Pixels (px)

Inches (in)

Centimeters (cm)

Millimeters (mm)

Points (pt)

Picas (pc)

Relative

Percentages (%)

Font sizes (em, rem)

Character sizes (ex, ch)

Viewport dimensions (vh,vw)

Viewport max (vmax)

Viewport min (vmin)

Relative Units

em

Relative to the font-size of the current element

ex

Relative to the font's x-height

%

Relative to the enclosing parent element in percent

ch

Relative to the width of the digit "0"

rem

Relative to the font-size of the root element

vw

Relative to 1% of the width of the viewport

vh

Relative to 1% of the height of the viewport

vmin

Relative to 1% of the viewport (smaller between vw & vh)

vmax

Relative to 1% of the viewport (bigger between vw & vh)

**2nd Video**

Div e box create kore display flex diye flexbox use kora jay. amra justify content use kore x axis e box gulake distribute korte pari ba boshaite pari.

Flex-start dile ekdom shurutei ba left e thake

Flex-end dile ekdom right e chole jay

Center dile middle e thake.

Space-between dile 3 item er moddhe ba shobgular moddhe gap dey.

Space-around dile majhe ebong left e right er element eo space dey. But majhe beshit hake

Space-evenly dile shobkhane space shomanvabe dey

align items diye y axis e item align kora jay. Flex-start, flex-end, center

flex direction o deya jay.

Row dile left theke left to right e div elements dekhabe after flex.

Row-reverse dile ekdom right theke item dekhabe and reverse order e mane 3 2 1.

Column kore dile top to bottom dekhabe and column-reverse dile ekdom nich theke dekhabe oi section ba box er and reverse order e dekhabe mane 3 2 1

Flex-wrap use kora jay ja diye jodi ekta view te ba box e item er jayga hoyna pashapashi taile oita wrap kore nicher line e chole jabe. Mane ekta box e 6ta element thakle and oita jodi choto kora hoy taile jodi box er baire chole jay, wrap dile choto korte thakle oita nicher ;line e chole jabe box eri

**3rd Video**

Child gulake justify content korte chaile oi child er immediate parent ke display flex kore justify content kora lage. Eki vabe align items.

Ekta nav er ekta item ke right and bakigula left e dekhaite chaile. Bakigulake ekta div/span e nibe and oi ekta alada. Then eitar parent ke display flex korbe and justify content space between dibe.

**4th Video**

Flex chole unidirectional mane ek direction e chole. Grid chole dui dike.

display: grid; just dile kono change dekhabena. Use korte hobe other styles.

/\* grid-template-columns: 250px; This will take 250px width of each child \*/

/\* grid-template-columns: 250px 250px; This will take 250px width of each child but in one row two child will be visible. \*/

/\* grid-template-columns: 250px 250px 250px 250px 250px; This will take 250px width of each child but in one row five child will be visible. \*/

/\* grid-template-columns: repeat(2, 250px); This will take 250px width of each child but in one row two child will be visible because of the repeat. \*/

/\* grid-template-columns: repeat(2, 250px) 450px; This will take 250px width of the first two child and the last child in one row will be 450px, means there will be 3child. \*/

/\* grid-template-columns: repeat(5, 1fr) 240px; This will take 1 fraction of total space available for each of the childs thus it will make it resposibe and 5 childs in a row. \*/

/\* grid-template-columns: 250px auto 250px; This will provide the first and last child in the row with 250px width and the middle one will take remaining whole space. \*/

grid-template-columns: 250px 250px 250px 250px;

/\* justify-content: space-between; This will provide space between each child but in row. \*/

row-gap: 25px;

column-gap: 25px;

/\* This will take gap in row and column with 25px \*/

**5th Video**

**8th Video**

Seven Things You Need To Do To Make A Website Responsive

1. Viewport Meta Tag
2. CSS Relative Unit (em, rem, %, vh, vw, vmin, vmax)
3. Body Max Width and Horizontal Center Align
4. Image Fluid (use % as width)
5. Two Column Flex and Use Media Query with Flex Direction Column
6. Multi Column: Grid Layout
7. Menu Responsive: Will Require JS

**9th Video**

Grid: Multi-Dimensional Layout

Flexbox: One Dimensional Layout (Column or Row Wise)

Flexbox: Focuses on content flow Content First. Widths (or heights) of flex items are determined by the content of the item.

Spacing and aligning items are flexible.

Grid: Focuses on content placement Layout First. Provides a mechanism to divide available viewport for layout into columns and rows.

Each item can be positioned anywhere in a grid cell.

When to use Flexbox

* You have a small design to implement
* You need to align elements
* You need a content-first design

When to use Grid

* You have a complex design to implement
* You need to have a gap between block elements
* You need to overlap elements
* You need a layout-first design

Media Query is-

* The key part of responsive web design
* Used to create different layouts depending on the size of the viewport
* Used to check width and height of the viewport
* Used to check orientation (is the tablet/phone in landscape or portrait mode?)
* Used to check resolution