HTML

Hyper Text Markup Language

It is a program which is used to create the multipage of a website effectively by using graphics, Images, texts etc.

It is having two types of components =

i) Container Element = The Element which requires closing, and supports various content is turned as container Element.

ii) Empty Element = The Elements which does not supports any content, only has some effective work on the program and does not requires closing is turned as ~~container~~ Empty Element ↑

- Tags = The system which supports Elements is turned as tags.

Eg = <A HREF = " " ></A>

Tags

Elements

Flow Chart =

Elements
↓
Tags
↓
Content

Program

* Basic Format of HTML :-

<!DOCTYPE> ⇒ Empty Element having docu Ty
<HTML> ⇒ Duration of program by HTML.
<HEAD> ⇒ Head of the program
→ <META NAME = " " CONTENT = " " >
<TITLE> " " </TITLE>
</HEAD> → Since, head was container Element. It requires closing.
<BODY>
:
:
</BODY> ← Body is a container Element which main content of the program is shown.
</HTML>
↑ Closing Program

→ Mita is an Empty Element having n and content as tags. Generally, Informati Under mita is not displayd.
→ Title is a container Element which is used to give Title of the program but i displayd in title bar of the screen's Output.

Various Attributes used in program :-
i) <P> ⇒ It is a container Element which supports paragraph.
ii) ⇒ It is a container Element whi used to give bold texture to t
iii) <EM> ⇒ It is also the container Element used to give Italic texture to text.

iv) <U> ⇒ It is also the container Element Used
Underline the text.

v) <BR ⇒ It is Empty Element Used for a line
break. But not prefferd for Use.

vi) <HR> ⇒ It is Empty Element Used for horiz
tal line rule.

vii) <DIV> ⇒ It is a container Element Used for
line break. Preffered method. (Recommen

viii) <PRE> ⇒ Formatting will be visible.

Note ⇒ Before HTML5 the Attributes Used for
bold and Italic were <B> and <I>
which were Also the container Element and is
replaced by <STRONG> and <EM>.
⇒ But we can Use them in latest versions als
will work similarly without giving Error.

* HEADINGS =
⇒ There are Six heading tags which are all
a container Elements.
⇒ Starting from H1 to H6.
⇒ Size of the text differs as =

H1 > H2 > H3 > H4 > H5 > H6

* GENERATION OF LINKS =
⇒ We can generate links Using HTML program
to Open various websites.

Syntax = <A HREF = "HTTPS://WEBSITENAME.COM">
CLICK HERE TO OPEN WEBSITE </A>

But by the Above syntax we can Open One website at a time.

To Open multiple websites at a time we must Use target and blank after website name as given below =

Syntax = <A HREF = "HTTPS:// WEBSITE NAME.COM"
TARGET = "_BLANK"> TEXT </A>

* TO ATTATCH IMAGES =

⇒ To Attatch Images Online to a program we must Use the following syntax =

<IMG SRC = "IMAGE LINK" ALT = "TEXT>

⇒ It is empty Element. Hence, does not require closing

⇒ ALT="TEXT" is Used because If the Image from given link is not working then whatever is written at text part will be displayed.

* LISTS = There are two type of lists to Arrange various things Accordingly.

i) Unordered List = Arrangement in Unordered form Like in form of bullets, squares and discs.

Syntax = <UL TYPE = "CIRCLE / SQUARE / DISC"
<LI> TEXT
<LI> TEXT
</UL>

ii) Ordered List = Arrangment in Ordered form like number, capital Alphabits, small Alphabits

Syntax = <OL TYPE = "1/A/a" >

<LI> TEXT

<LI> TEXT

</OL>

⇒ Here, LI are list Items which is to be listed

*. TO FORM TABLES = Various Elements and Information which Arranged in Rows and Columns forms tables.

Syntax =

<TABLE>

<THEAD>

<TR> ⟶ <T R> ⟶ Represents Rows

<TH> TEXT </TH> ⟶ Represents heading

:

</TR>

</THEAD>

<TBODY> → Contains Body of the table

<TD> TEXT </TD>

:

</TBODY>

Very Important

FORM

→ Isle yaar, make a website form for taking the credentials by using various Attributes on a website.

Syntax =

```
<FORM ACTION = "BACKEND.PHP">
NAME <INPUT TYPE = " " NAME = " " ID = " ">
</FORM>
```

Here type could be of various types =

a) TEXT = Gives Alphabetical values.
b) NUMBER = Gives Numeric Values.
c) EMAIL = Stores mail Id.
d) RADIO = Used to click various options.
e) CHECKBOX = Used to select or Unselect the Option.
F) DATE = Used to select date.
g) TIME = Used to select time.
h) DATETIME = Used to select date and time.
i) TEXTBAR = Space to write discription According to columns (clms) and rows (ROWS) given to it

Note = To submit the credentials following syntax is Used =

```
<INPUT TYPE = "SUBMIT" VALUE = "SUBMIT">
```

To reset the credentials following syntax is Used =

```
<INPUT TYPE = "RESET" VALUE = "RESET">
```

Now to click and select the credentials following syntax is Used =

```
<LABEL VALUE = "ID NAME"> " "</LABEL>
```

Note = Here ID Value must be name of ID.

⇒ The name and Id mentioned is not going to be displayed on Output screen but, these are the terminologies which are going to be used under backend for the credintials.

⇒ 3 Note = To select One Option from multiple Options following Syntax is Used =

```
<LABEL VALUE = " ID " > CREDENTIAL </LABEL>
<SELECT ~~TYPE~~ NAME = " " ID = " ">
    < OPTION VALUE = " " > CREDENTIALI </OPTION>
    |
    |
    |
</SELECT>
```

* Inline And ~~Drop~~ Block =

⇒ To make the function shift in One line by the compiler so that to reduce function Overload and to reduce compile time, Inline terminologies / Elements are Used.

The syntax is as follows =

```
<SPAN> - - - - - - </SPAN>
```

⇒ Keyword Used for Inline is SPAN.

Ex = For the Program

```
<P> HII ! I AM HARSHIDA SHAILY. </P>
<P> AND I AM EIGHTEEN YEARS OLD. </P>
```

Output = HII! I AM HARSHIDA SHAILY.
AND I AM EIGHTEEN YEARS OLD.

⇒ Now, here without using
 or <DIV> tag the line break occured.
Which means this are block code.

But If the Program is like =

<SPAN> HII! I AM HARSHIDA SHAILY. </SPAN>
<SPAN> AND I AM EIGHTEEN YEARS OLD. </SPAN>

Output = HII I AM HARSHIDA SHAILY. AND I AM
EIGHTEEN YEARS OLD.

⇒ By using Inline concept, it shifts to One line
Now,

⇒ By using Inline concept the function Applies particular block of space but by using block codes they occupy full space.

* ~~CLAS~~ ID'S AND CLASSES =

⇒ The Id's of the particular Attribute is Unique.
But classes name can be used at multiple pla

⇒ In VS-CODE is represented by =

ID ⇒ Shortcut ⇒ #
CLASS ⇒ Shortcut ⇒ .

Syntax = <DIV ID="" CLASS="">

⇒ Giving various class names at same tag, u
use space and then write the name of th
class in general compilers

⇒ These are used to give shortcut to various symbols and gives shortcuts to reserved words and shortcuts to the elements which are not present on keyboard

*. SEMATIC ELEMENTS :-

⇒ These are used to give meaning to various keywords.

⇒ ~~Useful~~

Useful in ranking of multipage in cluster of websites ~~on~~ Search Engine Optimisation (SEO).

⇒ Differentiates header, footer, navigation, contents etc.

Syntax =< DETAILS >

TEXT

< SUMMARY > TEXT </ SUMMARY >

</ DETAILS >

~~3~~ Note = LOREM gives / provides dummy words to the program

Attributes =

⇒ Used to make program effective

Example = ID, Class,

Tags :-

< header > = Consists, logo, heading, weblinks.

< navigation > = Consists buttons

↳ < nav >

< main > = Content of multipage

i) < article > = Subsection

ii) < section > = Single piece of functionality

HTML 5.0 is the latest version of HTML.

HTML	HTML 5
→ Was not user-friendly.	→ User-friendly.
→ Not having Audio and Video tags.	→ Having Audio and Video tags.
→ Not supported by latest browsers.	→ Supported by All latest browsers such as Google Chrome, Opera Mini, Firfox, Safari Etc.
→ Device dipendent. (Only runs on particular device as per code.)	→ Device Indipendent. (Runs on each and every device as per code)
→ Syntax was not Organized.	→ Syntax was Organized.

Note = General Syntax = <tagname> Content of the tag </tagname>

*. Attributes = HTML Elements are having Information in it.
Hence, the Additional Information About HTML Elements is After the declaration of tag name is termed as Attribute.

Ex =
```
<HEAD
<META  NAME="  "  CONTENT=" ">
</HEAD>
```

Here HEAD is Element, META is tag, NAME and CONTENT is Attribute

*.

Various Tags of HTML

Tag	Working
<P>	<P> It is a paragraph tag which supports Sentences (Short or long)
<H1>	<H1> This is heading tag which ranges from 1 to 6, varies in size. </H1>

<A>	<A> This is Anchor tag which is Useful in combining various links through website </A>
<IMG>	This is Empty tag which is responsible in giving the Image to the text or website

Attributes in HTML

Attri	
HREF <A HREF = " " >_</A>	HREF is Used After Anchor tag to specify links of ~~this~~ website.
SRC <IMG SRC = " " >	SRC is Used to link Images to the webpage.
ALT <IMG SRC = " " ALT = " ">	If the link of the Image does not work properly then the text Under ALT is

~~4~~ 3 Note = The SRC Attribute is further divided into two parts =

a) Absolute URL = Gives and supports links to the Online Images.
b) Relative URL = Supports the Images within the computer system (Offline Images).

FORMATTING

<B>, <STRONG>	Useful in giving bold texture to the text.
<I>, <EM>	Useful in giving Italic texture to the text.
<U>	It is giving Underline beneath the text.
<mark>	To make highlighted text.
<small>	To make texts smaller than normal size.
<big>	To make texts Enlarged than normal size.

Additional formatting Tags

Tags	Uses
<STRIKE>	Strikes the text written Syntax = <STRIKE> ---- </STRIKE> Ex = <STRIKE> CIGRATTE </STRIKE> Output = ~~CIGRATTE~~
<SUB>	Uses to make text as Subscript Syntax = ₋₋₋₋₋ Ex = <P> A ₂ </P> Output = A_2
<SUP>	Makes the text as Superscript. Syntax = ⁻⁻⁻⁻ ~~Syntax~~ = <P> A ² </P> Ex Output = A^2
<BIG> <SMALL>	Enlarges Original text Enamalles the original text

* Text - Alignment = The Alignment or The position of text on the screen of multipage.

Syntax = <TEXT-ALIGN="POSITION">-</TEXT>

Or

<Selector TEXT-ALIGN="POSITION"> ---- </Selector>

Or

<Selector style="TEXT-ALIGN:POSITION;"> ---- </Selector>

→ Here position can be Right, Left, Center, Top, Bottom.

Tag	Uses and Syntax Formatting Effects
VSPACE	Gives Vertical spacing At both sides of the Image in pixels. Syntax = < IMG SRC=" " ALT=" " VSPACE = "—PX">
HSPACE	Gives horizontal spacing at both sides of the Image in pixels. Syntax = < IMG SRC=" " ALT=" " HSPACE = "—PX">
MARQUEE	It is a Container Element which is showing Used for moving text Syntax = <MARQUEE > TEXT_FOR_MOVEMENT </MARQUEE>

(Cascading Style Sheets)

Importance =

i) It gives style to ~~Raw~~ raw HTML code and makes it presentable and stylish

ii) It gives color, texture and graphics to the markup

iii) Gives Effectiveness to the viewers and customers

* MARKUP ⇒ The raw HTML code is turned as markup.

Syntax (General) = Selectors { ~~color~~ Attribute : Type ; }

or

Selectors { Property : Value }

Example = P { color : blue; }

* WAYS TO ADD COLOR STYLING TO MARKUP =

i) Inline Style = It is Added within the same line after the tag

But this is generally not recommended because it makes the markup confusing

Ex-: <P style = "color: RED; background-color : black"; HII </P>

ii) Internal Style = It is Added to the head Element in the markup.

Here is the Usage of general syntax

Ex-:

```
<HEAD>
<STYLE>
  P
    {
      color : RED;
      background -color: BLACK ; }
</STYLE>
</HEAD>
```

iii) External Style = We need to follow two steps in the same =

a) Create the styling to CSS shute.css
b) Add the styling shut to the head to Apply in markup.

Ex:-

Step 1 = Create a shut let's say HARSHIDA.CSS

```
p
{
  color: RED;
  background-color: BLACK;
}
```

Step 2 = Add to HTML's head Element

```
<HEAD>
<LINK REL=" " HREF="HARSHIDA.CSS">
```

Save the code and then run the program.

Order of the style colors =

Inline > Internal / External (According to discending Order positioning)

Selectors in HTML and CSS (Cascading Style Shuts)

⇒ Syntax = Selectors { Condition / Paramiter : Value ;
⇒ (These all functions in HEAD Element)
⇒ There are four kinds of Selectors =

i) Element Selector = This are normal CSS (Cascading style shuts) same as Internal style.

ii) Id Selector = The ID (which is Always Unique is given to selector and Attributes and it's values are set.

Syntax = #Id { Attribute / Paramiter : Value ; }

ii) Class Selector = The class name and value are given to tags which can be common and that is set with the parameter.

Syntax = .class_name

{

Parameter / Attribute : Value;

}

r) Grouping Selectors = They are used to add effects to multiple tags at a time.

Syntax = tag1, tag2

{

Attribute / Parameter : Value;

}

Example = FOOTER, SPAN, P

{

color: RED;
background-color: YELLOW;
Font-style: ITALIC;
border: 2PX SOLID BLACK;
Font-weight: bold;
line-height: 1.3em;
Font-Family: -;

}

↑ Below, these properties are Inherited to all parameters mentioned above.

5. DEVELOPER TOOLS = When we compile the code, it shows it's Output on the browser (Generally Google chrome).
Here by right clicking and Inspecting the various functions, we can chang different Attributes for effective display of the Output.

3 Note = These are temporary. When we reload the ~~when we~~ it, it will again show the same Output as previous One.

6. FONT - STYLES = These are used to give Effectiveness to fonts in various Aspects.

i) Font - Family = Used to give the texture i.e. Arial, Cursive etc. (list of font name)
It is of two types = a) Web Safe font family
b) Web font family

a) Web Safe font family = Which is Already present in our Computers library.

b) Web fonts = family
This is something which we need to extract and download from Internet.
Best Site to download/extract is Google Chrome font styles.

ii) Font - weight = This shows Bold, Italic, Underline etc texture.

iii) Font - size = Specifies the size of font in pixels.

) line-height = Makes distance between two lines according to the code.
Dimensions in em and rem.

font-style = Specifies Style of font (Bold, italic, underline)

HEIGHT, WIDTH, STYLE E = All things Under Head Element

ex = <STYLE>

{

color: rgb (31, 42, 26); (Rgb gives color and has a specific

background-color: #FFFFFF; Option to chose the colour.)

background-Image : url('LINK') ← Give Code of color

height: 490PX ; ← Describes the height of the box.

width: 460PX;

border: 2PX SOLID RED; ← Describes width of the box. Gives color to the border

border-radius: 2PX; ← Gives a curve to the border ends.

Further Specified Into =

border-bottom-left-radius;

border-bottom-right-radius;

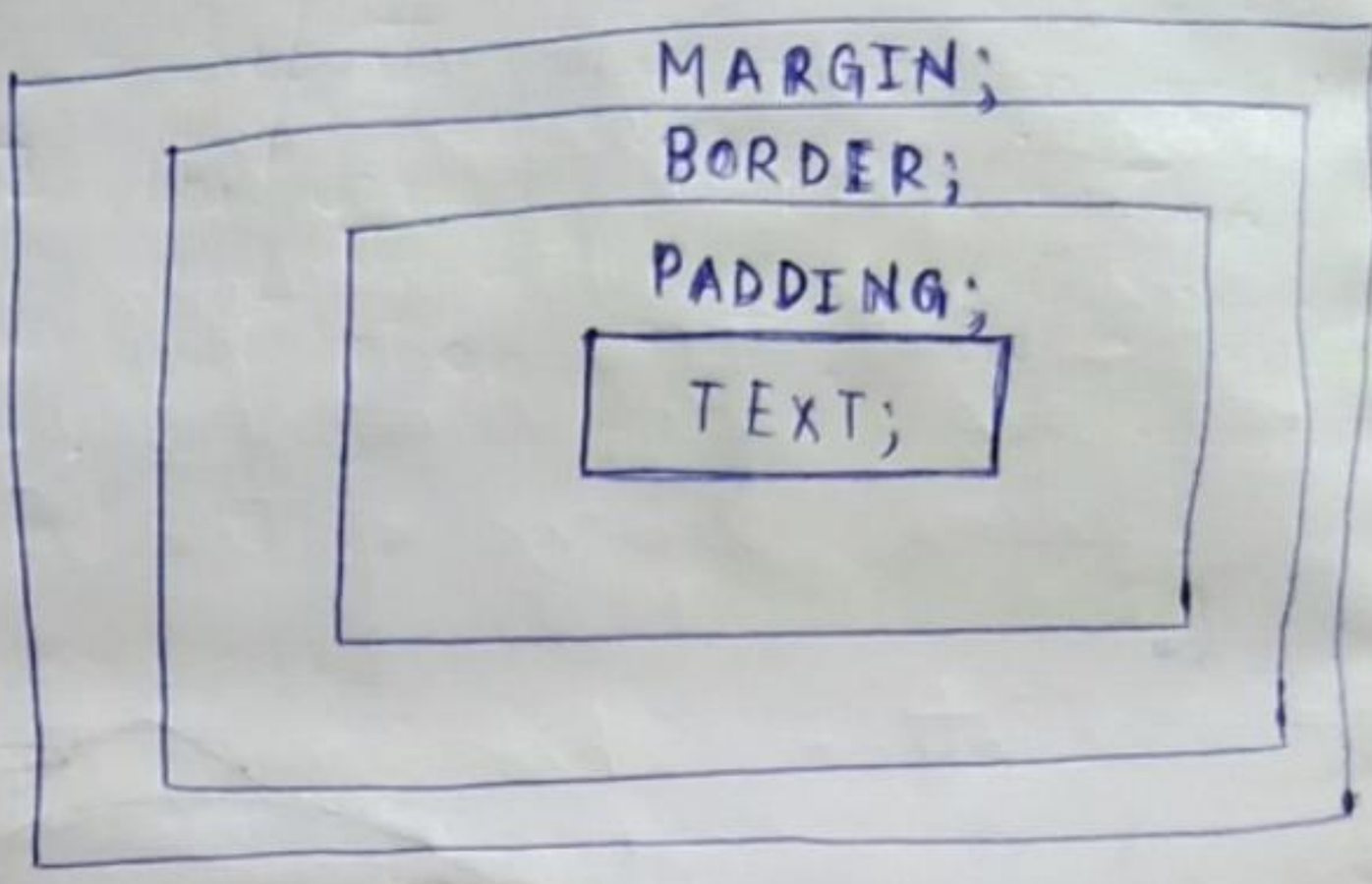
border-up-left-radius;

border-up-right-radius;

}

BOX - ELEMENTS = (V very V very Important)

Padding = The Inner Core After
xt / Charactor / Imag is
rmed as padding.



yntax = Selector
{
padding: Value (In PX);
}

The Value goes like = UP > RIGHT > BOTTOM > LEFT.

Border = Border is Used to give a box and
Outline to the padding.

yntax = Selector
{
border : Value ; (In PX)
}

It is of Various Attributes = border-left, border-right, border-up, border-bottom, border-radius.

) Margin = Margin is Used to give boundary
between two seperated lines.

Syntax = Selector
{
margin : Value (In Pixels);
}

It can Also vary from top, bottom, right,
ft according to requirment.

3 Note = In our browser we can inspect and
ist various Attributes to our code Acc
ingly so that the output comes out to be
ffective.

BACKGROUND PROPERTIES =

i) background-color = Specifies the background color.

Syntax = background-color : Value;

ii) background-Image = Specifies the background Imag
Or
Sets Image at the background
the text.

Syntax = background-Image : url ('LINK);

iii) background-position = Specifies Initial position of t
background Image.

Syntax = background-position : Value

→ The value of background position can be =
left top, left center, left bottom
right top, right center, right bottom
center top, center center, center bottom, x, Y, x%, Y

iv) background-attachment = This Indicates wheth
fixed or gets scrolled when the User scrolls t
page.

Syntax = background-attatchment: Va

→ The value can be fixed / Scroll.

v) background-repeat = This specifies whether the ba
ground Image will get repea
or not. Syntax = background-repeat: ~~repeat~~ value;

values can be a) Repeat = Repeats the Image on both page, b) Repeat-x = Repeats the Image in x-direction c) Repeat-Y = Repeats the Image in Y-direction d) - Repeat = Repetition / Iteration not, displaying only one Image.

FLOAT AND CLEAR =

FLOAT = It shifts the text according to the Instructions on Output screen.

Ex = <STYLE> ——— Class Selector

.HARSHIDA {

FLOAT : VALUE ; ← Here value is left, Right, bottom

</STYLE>

<DIV CLASS = "HARSHIDA">

<H1> THIS IS A HEADING </H1>

<P> THIS IS A PARAGRAPH. </P>

</DIV>

* CLEAR = This Attribute tag clears the effects of float in the Output area.

Ex =

<STYLE>

.MYCLASS {

FLOAT : VALUE ;

CLEAR : AUTO / NONE ;

}

</STYLE>

<BODY CLASS = "MYCLASS">

<H1> THIS IS HEADING. </H1>

<P> THIS IS A PARAGRAPH. </P>

</DIV>

</BODY>

... values ... a) Repeat = Repeats the ... page ... b) Repeat-x = Repeats the Image on ... x-direction ... c) Repeat-y = Repeats the Image in Y-direction ... d) - Repeat = repetition / Iteration not, displaying ... Image ...

FLOAT AND CLEAR =

FLOAT = It shifts the text According to the Instructions on Output screen.

Ex = <STYLE>

.HARSHIDA {

FLOAT : VALUE ; ← Here value is left, Right, bottom

</STYLE>

<DIV CLASS = "HARSHIDA">

<H1> THIS IS A HEADING </H1>

<P> THIS IS A PARAGRAPH. </P>

</DIV>

* CLEAR = This Attribute tag clears the effects of float in the Output area.

Ex = .MYCLASS {

FLOAT : VALUE ;

CLEAR : AUTO / NONE ;

</STYLE>

<BODY CLASS = "MYCLASS">

<H1> THIS IS HEADING. </H1>

<P> THIS IS A PARAGRAPH. </P>

</DIV>

</BODY>

STYLING LINKS AND BUTTONS = Used to Generate and chang colors buttons and links.

ex =

```
<STYLE>
 .MYCLASS {
          background-color : value;
            color : value;
          height : 160PX;
          width : 980PX;
          }
A:HOVER { background-color : ORANGE;
A {       }
    background-color : GREEN;
          color : WHITE;
    }     TEXT-DECORATION : NONE;
          CURSOR : POINTER;
A:VISITED { background-color : PINK;
<BODY>    }

<DIV  CLASS ="MYCLASS>
 <H3> THIS IS HEADING.</H3>
 <P> MY NAME IS HARSHIDA SHAILY.</P>
    CLASS="HARSHIDA
 <A  HREF = "HTTPS://GOOGLE.COM"> CLICK HERE
                                  </A>

 <BUTTON CLASS = "SHAILY"> PRESS HERE </BUTTON
 </DIV>
 </BODY>
```

Hover is Used to chang the property when Tab is Used pressed

Visited is Used to give effects to the links, buttons after the Usage.

CSS POSITION = It discribes the position relative to
ascading Style Sheets) the parent box.

is of various types =

Static = It is dfault position of Any tml Element

Absolute = It fixes its position According to mentioned.

Relative = Relative to the pag scroll, the position remains the same.

Sticky = It sticks somewhere on the screen After required movement According to commands

ISIBILITY AND Z INDEX =

isibility is responsible for displaying the command not.

f the visibility

Outcomes none / hidden then the command
d Outcome Inherit Just the Area will remain visi-
t If display Outcomes none, then the Area is also
sabled i.e; skipped.

ntax = Selector { visibility : Value ; }

Hidden ,
Visible

- Index is Used for the positioning of the command
The greater the z-Index, higher will be the
ition.

ue, negative and positive value not matters.

ex = -165 > 65 In According to z-Index.

t = Selector { z-index : Value; }

FLEX BOX (सुविधा के लिए बना है।)

Flexible Box

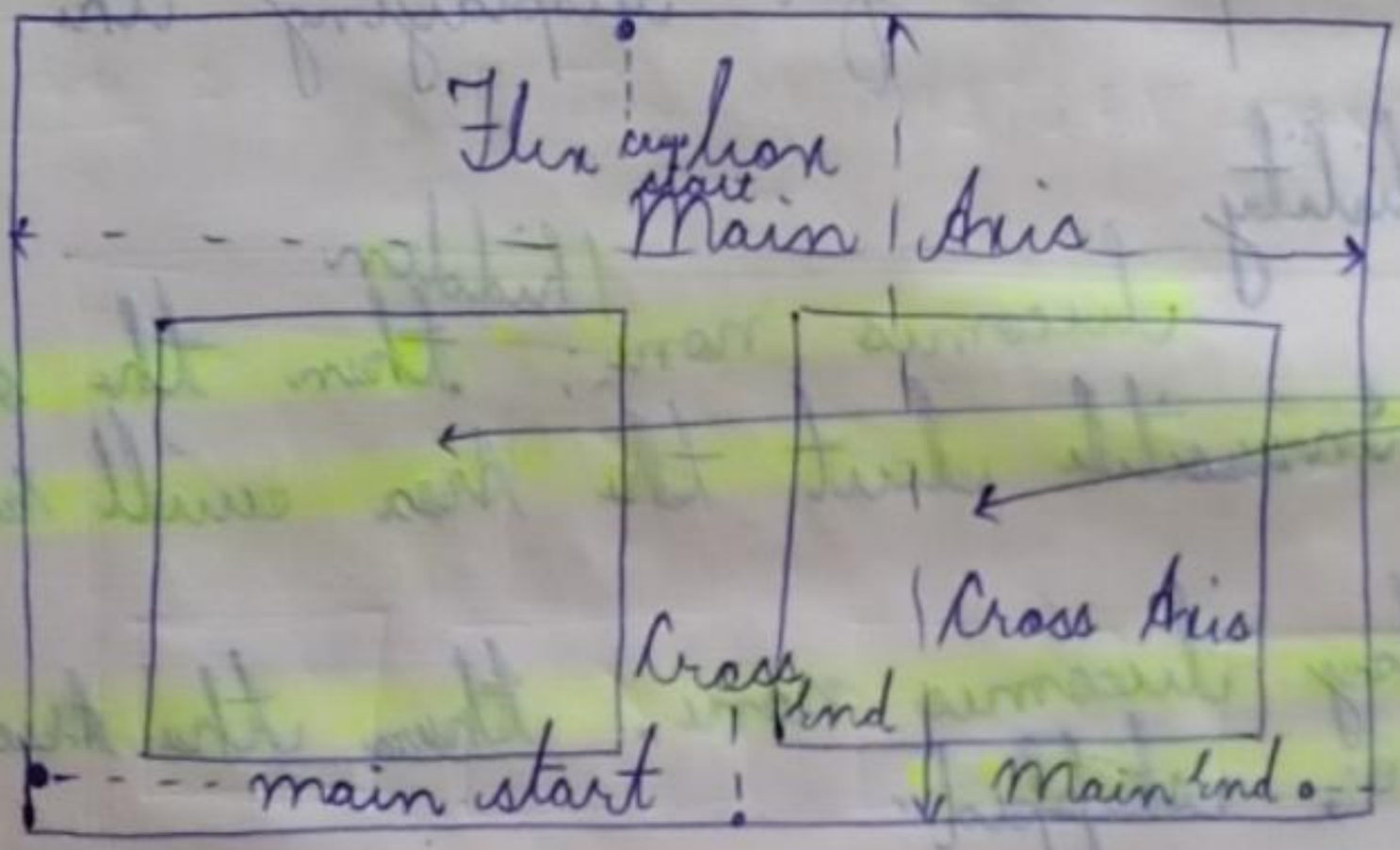
One - dimensional and Useful in Aligning Items in s and Columns

(Container)

ntax = Selector { Property : Value ; }

o display flex, the flex container gets created

Syntax = Selector { display : flex ; }



Note = x - Axis ⇒ Main Axis
y - Axis ⇒ Cross Axis

There are various properties in Flex-Box =

Flex - direction = Shows either the commands are going to stay =

They are of four types

a) Row = Align the commands in row.

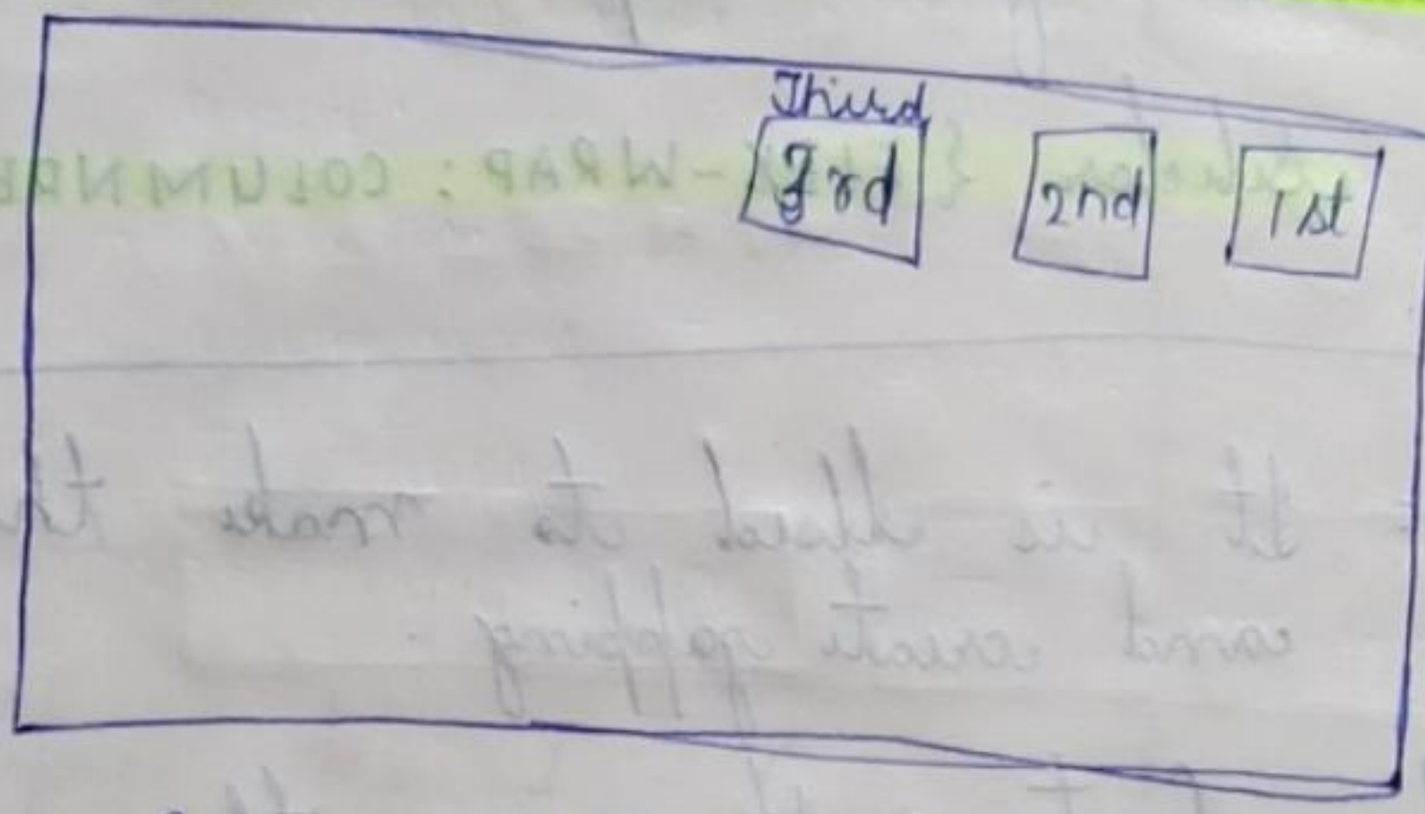
Syntax = Selector { FLEX-DIRECTION : ROW; }

b) Column = Align commands in column.

Syntax = Selector { FLEX-DIRECTION : COLUMN; }

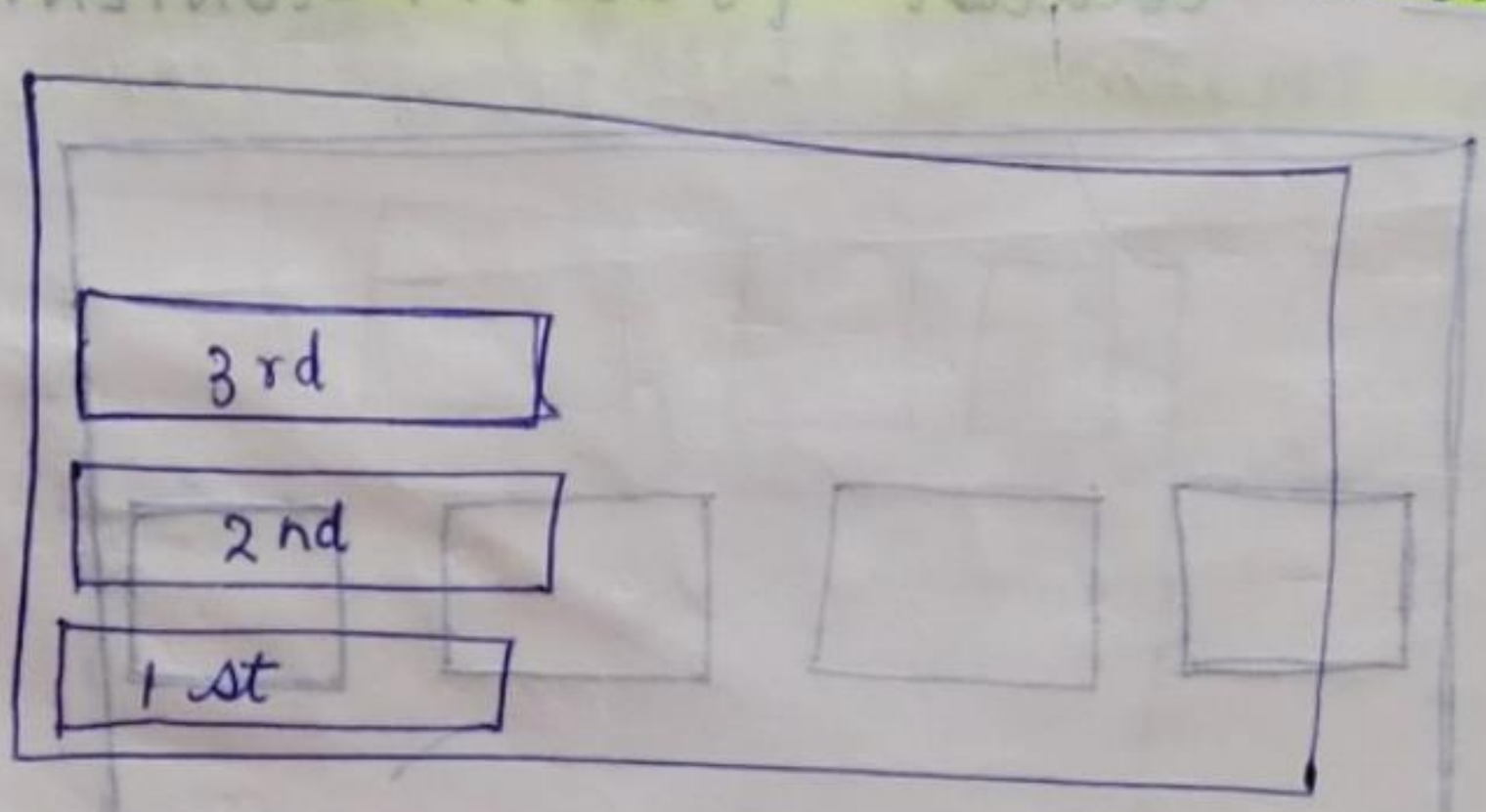
c) Row - Reverse = Reverse the alignment of the row.

Syntax = Selector { FLEX-DIRECTION: ROW-REVERSE; }



d) Column - Reverse = Reverse the column Alignment

Syntax = Selector { FLEX-DIRECTION: COLUMN-REVERSE; }



ii) Flex Wrap = The texture given to wrapping the text when screen is minimized.

→ The default value is no-wrap.

i) No-Wrap = No Wrapping occurs.

Syntax = Selector { FLEX-WRAP : NOWRAP; }

ii) Row-Reverse = If the flex direction is row or row reverse then row/row-reverse wrap

Syntax = Selector { FLEX-WRAP : ROW REVERSE ; }

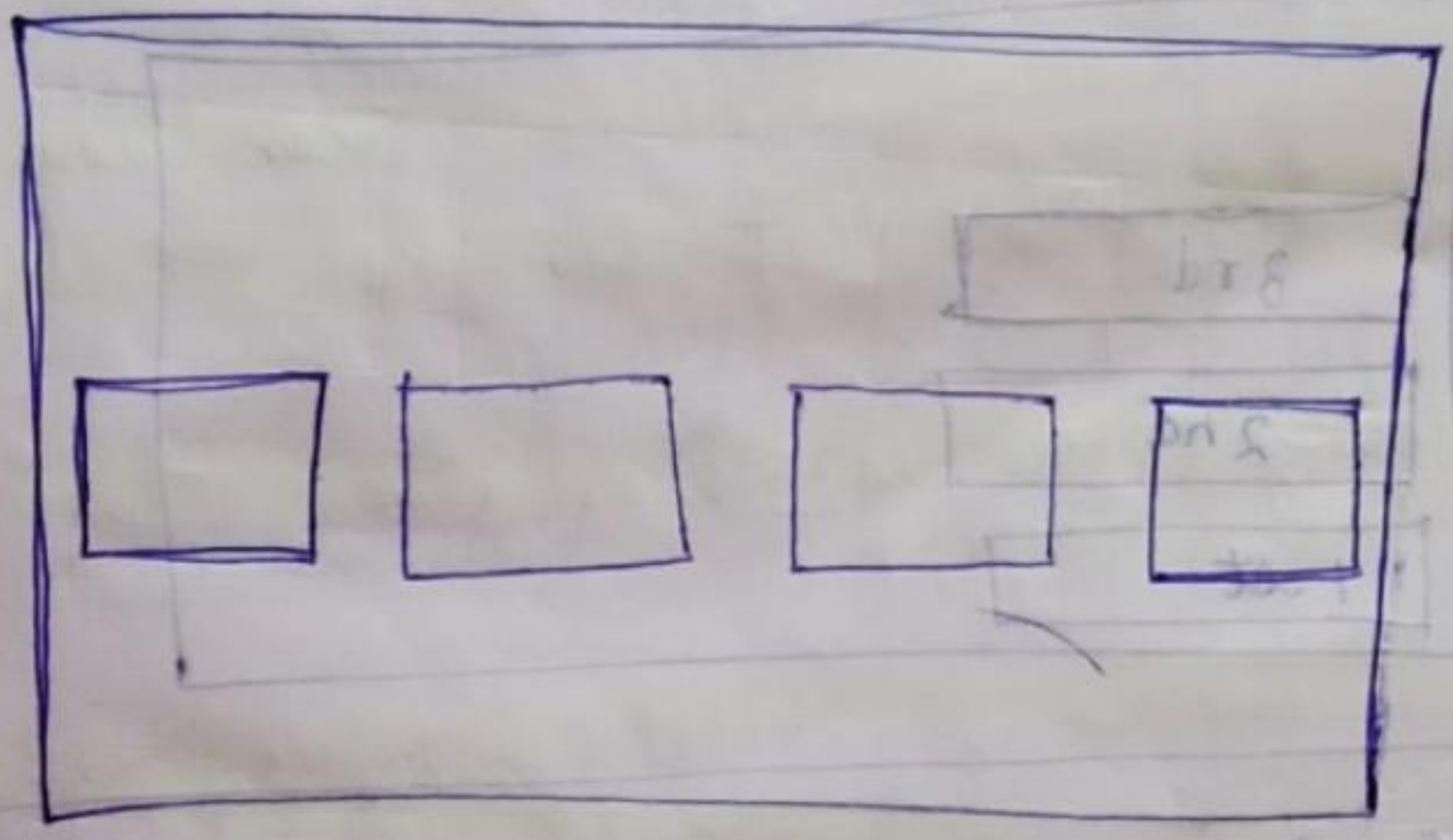
) Column - Reverse = When flex direction is column then column - reverse flex wrap is used.

Syntax = Selector { FLEX-WRAP : COLUMN REVERSE ; }

) Justify = It is used to make the text move and create gapping.

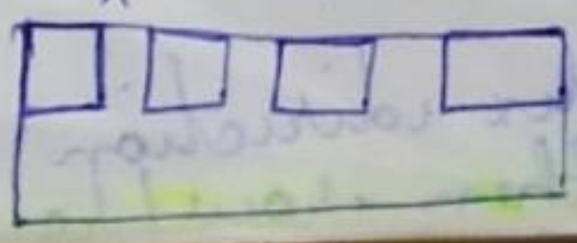
a) Justify - Center = It moves the commands and texts to center

Syntax = Selector { JUSTIFY-CONTENT : CENTER ; }



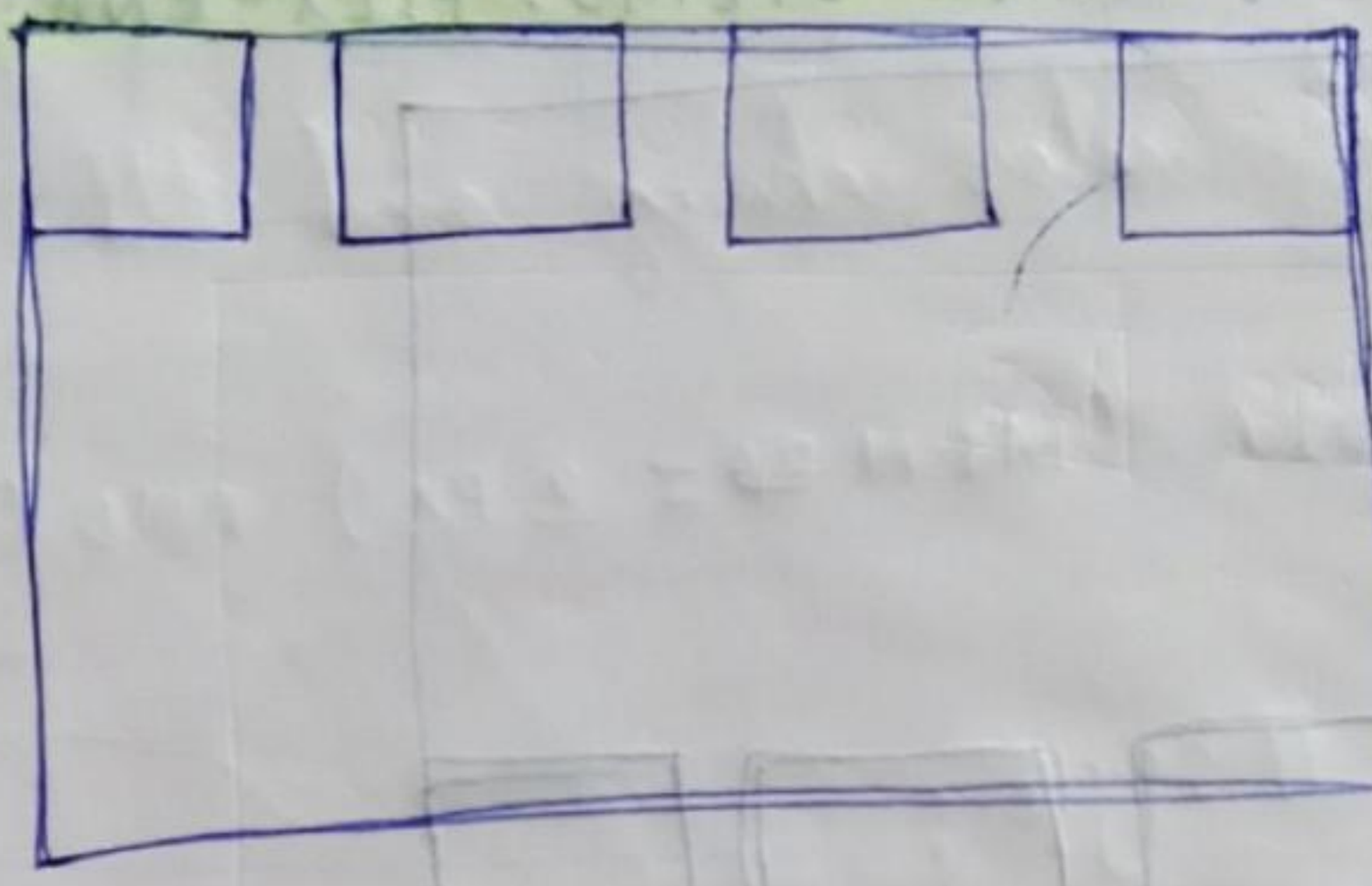
b) Justify Space between = It is responsible in giving Minimum spacing between two boxes but not at Right corner and left - corner

Syntax = Selector { JUSTIFY-CONTENT : SPACE-BETWEEN ; }



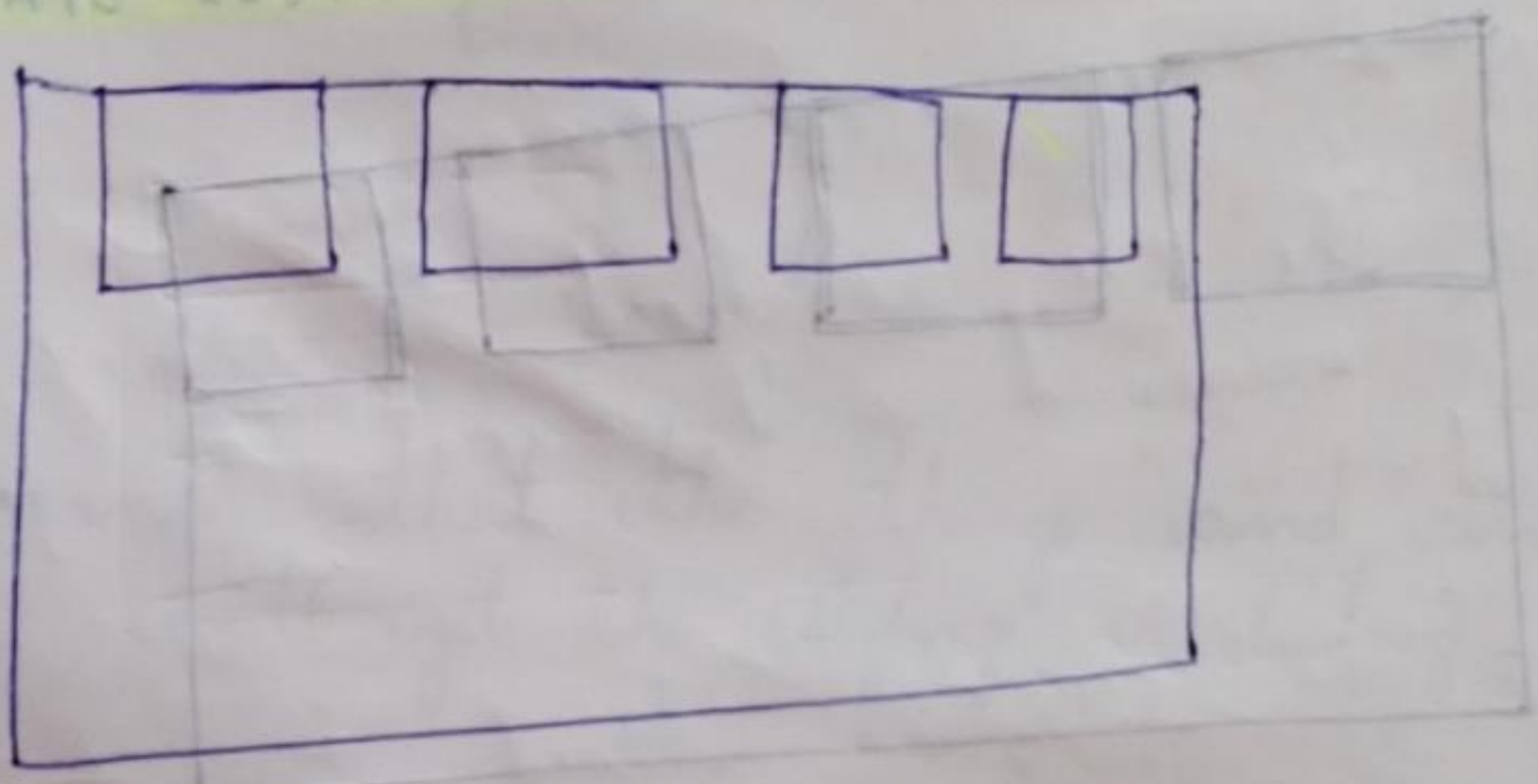
i) Justify Space Evenly = The space between the items becomes same.

Syntax = Selector { JUSTIFY-CONTENT: SPACE-EVENLY; }



iv) Justify Space Around = Gives space at each side of the items but not sure that the space is equal or not.

Syntax = Selector { JUSTIFY-CONTENT: SPACE-AROU



iv) ALIGN-ITEMS = Align and changes the position of Items.

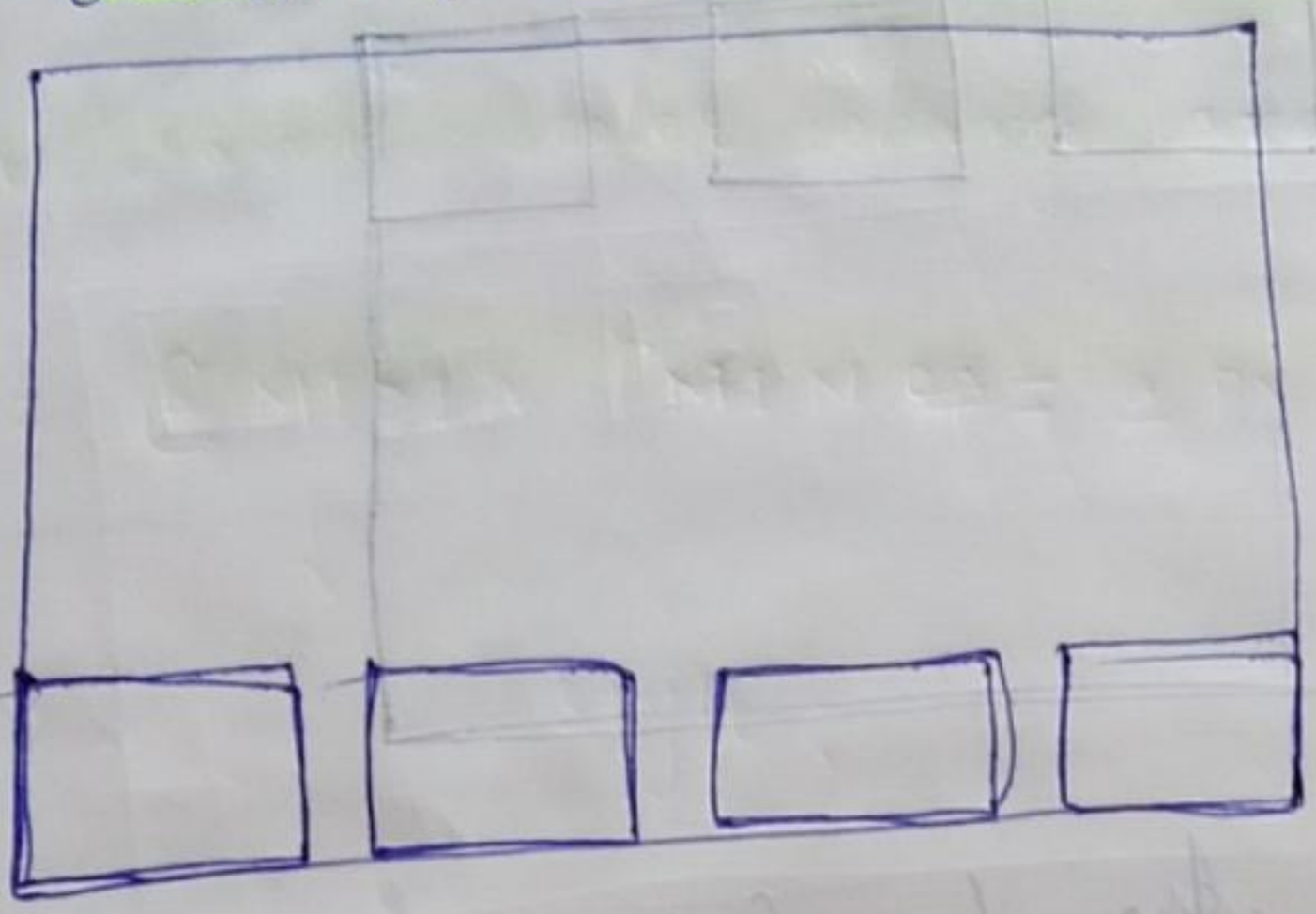
Various Types =

a) Align-Items Center = Alignment at Center.

Syntax = Selector { ALIGN-ITEMS: CENTER; }

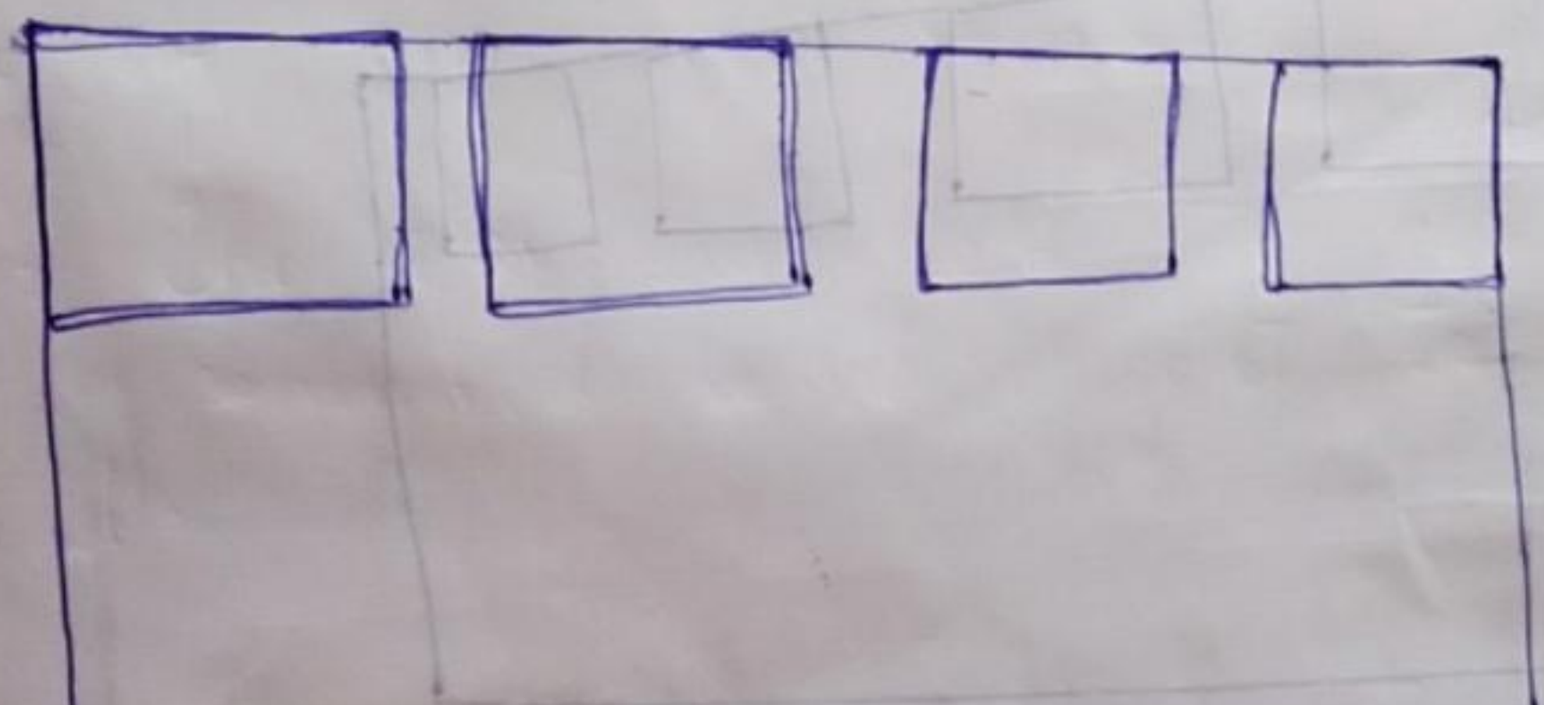
Align Items flex End ⇒ Display the items at (iii) End of the page.

ntax = Selector { ALIGN-ITEMS : FLEX-END ; }



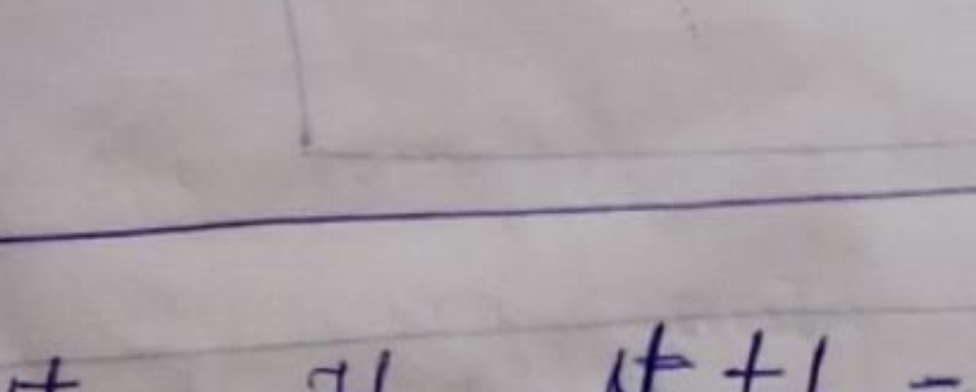
Align Items Flex Start = Align Items at the starting of the page.

Syntax = Selector { ALIGN- ITEMS : FLEX-START ; }



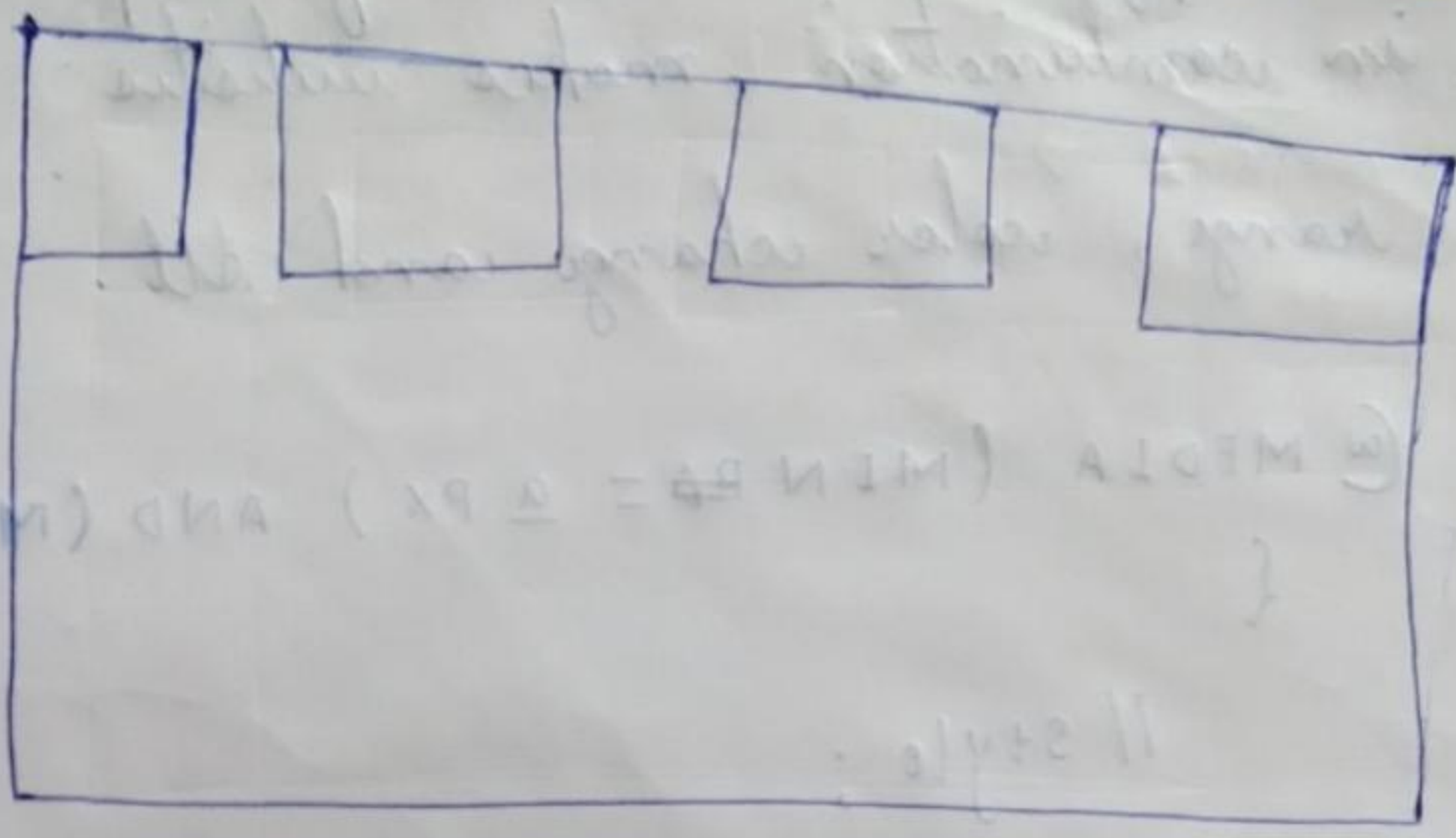
d) Align Items ~~Flex~~ Stretch = Align Items and stretchs it.

Syntax = Selector { ALIGN-ITEMS : STRETCH ; }



e) Align Items Shrink = Align Items get shrinked. (smallamud / thin)

Syntax = Selector {ALIGN_ITEMS:SHRINK;}



*. ADVANCED SELECTORS IN CSS (CASCADING STYLE SHEETS) =

i) Syntax = Selector 1 > Selector 2

{

}

This means that the things and commands of selector 1 will get the styling excluding ~~Selector~~ selector 2.

ii) Syntax = Selector 1 + Selector 2

{

}

This means that the selector which is past sibling of selector 1.