**APPUNTI**

Per prima cosa dobbiamo ricordare il funzionamento di un div container.

Esso, di default, ha una width definita, ed è uguale alla width del div contenitore. Nel caso in cui il container sia il div più esterno del nostro progetto, esso avrà width pari alla width dell’intera pagina web.

Discorso diverso per la height. La height di un div è pari a 0. Mano mano che il div si riempie, la sua height sarà pari alla somma delle height degli elementi che si trovano all’interno del nostro div.

Ovviamente possiamo aumentare la height del nostro container forzando la proprietà height ed impostarla ad un valore di nostro piacimento.

Per il momento non utilizziamo ne la proprietà height ne la proprietà width, ne sul div container ne sui div contenuti all’interno di esso. Consideriamo quindi una situazione iniziale di questo tipo:

HTML

<body>

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

}

p{

    margin:0;

}

#div1{

    border:1px solid black;

    background-color:white;

    font-size: 50px;

    font-weight: bold;

}

#div2{

    border:1px solid black;

    background-color:violet;

    font-size: 50px;

    font-weight: bold;

}

#div3{

    border:1px solid black;

    background-color:blue;

    font-size: 50px;

    font-weight: bold;

}

BROWSER



I div figli contenuti nel div container prendono quindi tutto lo spazio disponibile, ossia prendono tutto lo spazio già occupato dal div container.

Dal codice CSS si evince che il div container ha sfondo orangered, ma dato che i div figli prendono tutto lo spazio disponibile all’interno del div padre, tale sfondo non si vede.

Si vede invece lo sfondo dei div figli: div1 è bianco, div2 è rosa e div3 è blu.

Vediamo ora cosa succede se applichiamo la proprietà display:flex al nostro div container.

HTML

<body>

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

p{

    margin:0;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

#div3{

    border:1px solid black;

    background-color:blue;

}

BROWSER



E’ quindi successo che la lunghezza (width) di ogni div si è ridotta adattandosi al contenuto del div stesso. Dopodiché tutti i div sono stati inseriti sulla stessa riga. Lo spazio non riempito è la parte in rosso, ed è quanto rimane del div container. Guardando il CSS si nota che il div container ha sfondo rosso. Questo significa che la parte in rosso è proprio la parte del div container non occupata dai div figli.

C’è un modo per far sparire la parte in rosso? Certo. Basta modificare la proprietà display:flex in display:inline-flex.

HTML

<body>

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:inline-flex;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

#div3{

    border:1px solid black;

    background-color:blue;

}

BROWSER



**FLEX DIRECTION PROPERTY**

Torniamo ora indietro, al punto in cui avevamo display:flex

HTML

<body>

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

#div3{

    border:1px solid black;

    background-color:blue;

}

BROWSER



Si noti che tutti I div figli sono allineati alla sinistra del div container.

Quanto visto equivale ad aggiungere la proprietà flex-direction:row alla destra al nostro div container. Vediamolo qui sotto.

HTML

<body>

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

    flex-direction:row;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

#div3{

    border:1px solid black;

    background-color:blue;

}

BROWSER



CVD.

Torniamo ora un secondo alla situazione iniziale, ossia alla situazione in cui non è stato assegnata la proprietà display:flex al nostro container

HTML

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

#div3{

    border:1px solid black;

    background-color:blue;

}

BROWSER

Immagine che contiene testo, schermata, Carattere, Policromia

Descrizione generata automaticamente

Questa situazione di base equivale ad utilizzare display:flex e display-direction:column al nostro container:

HTML

<body>

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

    flex-direction:column;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

#div3{

    border:1px solid black;

    background-color:blue;

}

BROWSER

Immagine che contiene testo, schermata, Carattere, linea

Descrizione generata automaticamente

CVD

Torniamo alla situazione alla quale abbiamo display:flex per il nostro div container (Che ricordiamo coincide con la situazione in cui abbiamo display:flex e flex-direction:row)

HTML

<body>

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

#div3{

    border:1px solid black;

    background-color:blue;

}

BROWSER



In questo caso I nostri div figli sono allineati da sinistra a destra del nostro div container.

E’ possibile allinearli, al contrario, da destra a sinistra? Certo! Si aggiunge al nostro div container la proprietà flex-direction: row-reverse; al nostro div padre!

HTML

<body>

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

    flex-direction: row-reverse;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

#div3{

    border:1px solid black;

    background-color:blue;

}

BROWSER



In questo caso i div si sono ribaltati: Il primo div da sinistra diventa il primo da destra (quindi va per ultimo), il secondo div da sinistra diventa il secondo da destra (quindi resta al centro) ed il terzo div da sinistra diventa il terzo da destra (Quindi diventa il primo).

Torniamo all’esempio in cui abbiamo display:flex e display-direciton:column. Ricordiamo che l’inserimento di queste due proprietà corrisponde alla situazione in cui non abbiamo inserito la proprietà display:flex, in quanto di div sono inseriti per colonna. Ricordiamo che la situazione era la seguente.

Immagine che contiene testo, schermata, Carattere, Policromia

Descrizione generata automaticamente

Anche in questo caso, tuttavia, possiamo ribaltare i div: Il primo diventa l’ultimo, il centrale resta tale e l’ultimo diventa il primo. Per far ciò, dobbiamo utilizzare la solita proprietà display:flex; sul div padre ed aggiungere, sempre sul div padre, la proprietà flex-direction:column-reverse

HTML

<body>

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

    flex-direction: column-reverse;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

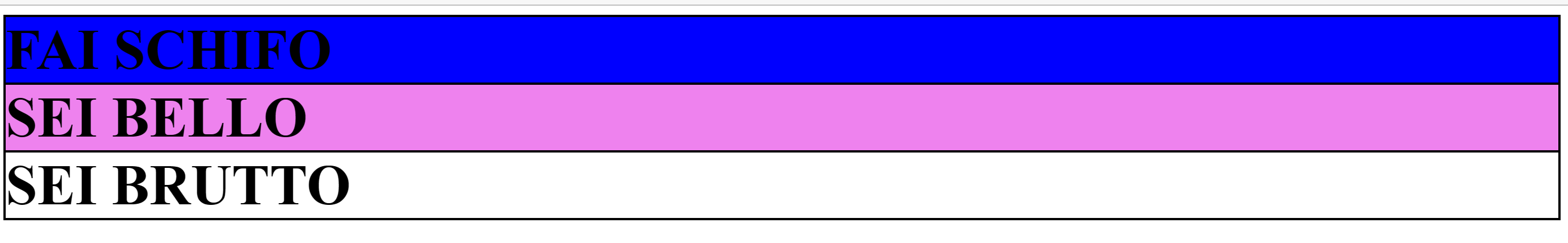
#div3{

    border:1px solid black;

    background-color:blue;

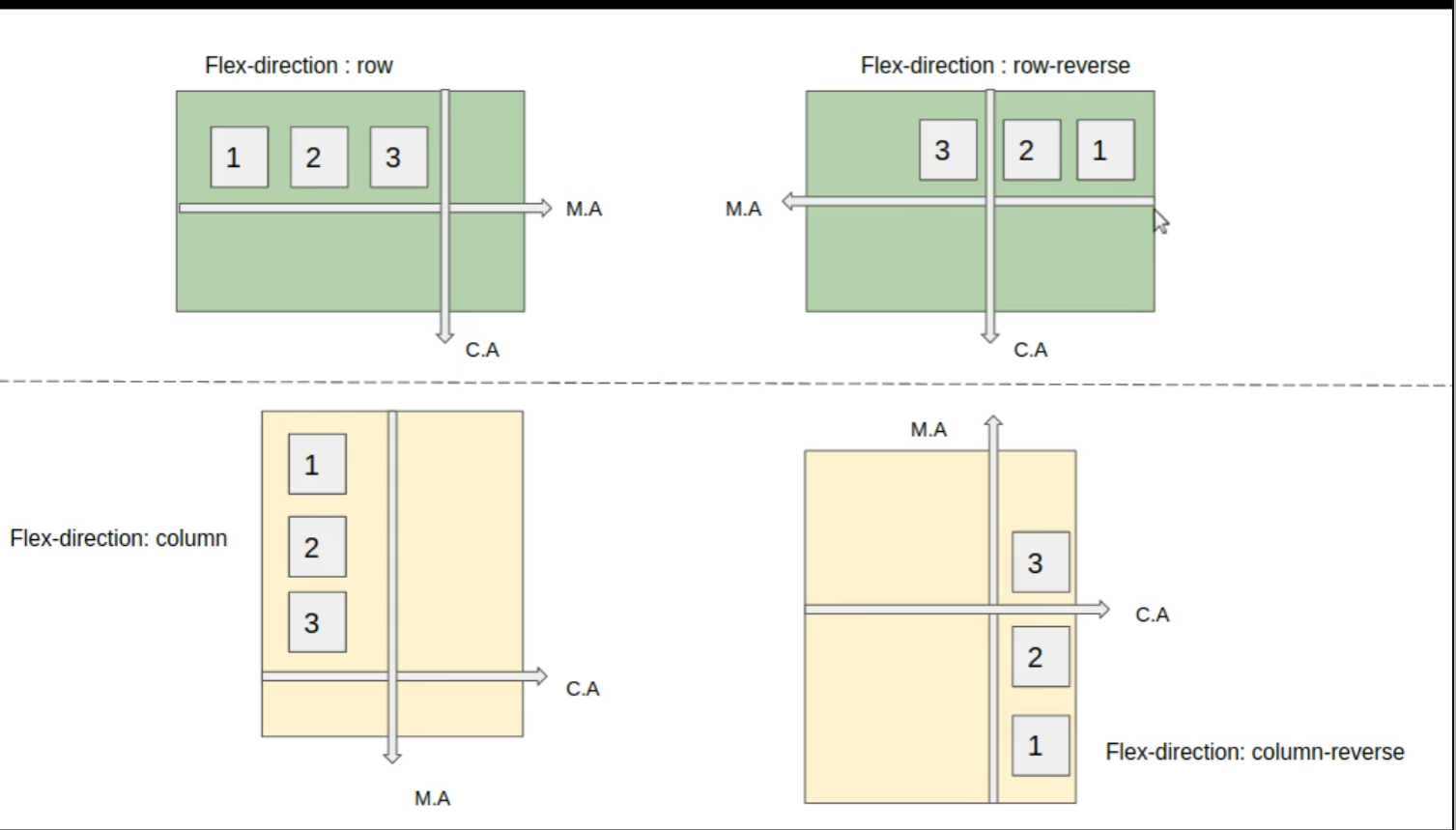
}

BROWSER



CVD

Quanto visto finora può essere riassunto dal seguente grafico.



**FLEX WRAP PROPERTY**

Così come flex-direction, anche la proprietà wrap viene inserita al div contenitore.

Il valore di default di wrap è nowrap. Questo significa che, se riduciamo la width del browser, il contenuto non si adatta ad essa automaticamente, ma scompare mano mano che il browser si restringe.

Vediamo quindi le proprietà di default di flex.direction e di wrap:

HTML

<body>

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

    flex-direction: column-reverse;

    flex-wrap: nowrap;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

#div3{

    border:1px solid black;

    background-color:blue;

}

BROWSER



E questo, ovviamente, equivale ad inserire la proprietà display:flex all’interno del nostro contenitore senza le proprietà flex-direction oppure wrap.

Cosa succede se proviamo a ridurre le dimensioni dello schermo? Succede che gli elementi presenti nel div contenitore, dopo un po che si restringe il browser, iniziano a scomparire

Immagine che contiene testo, Carattere, schermata, numero

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere, numero

Descrizione generata automaticamente

Per risolvere questo problema si utilizza flex-wrap:wrap:

HTML

<body>

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

    flex-direction: row;

    flex-wrap: wrap;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

#div3{

    border:1px solid black;

    background-color:blue;

}

BROWSER



Riducendo l’ampiezza del browser:

Immagine che contiene testo, schermata, Carattere, software

Descrizione generata automaticamente

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente



L’altezza del div container aumenta mano mano che si riduce la larghezza del browser.

Quindi, quando il browser si riduce in modo tale da ridurre la larghezza del div figlio, la larghezza di quest’ultimo non si riduce, ma va a capo su una nuova riga.

Oltre a flex-wrap: wrap esiste anche flex-wrap: wrap-reverse.

Quando il browser inizia a ridurre le dimensioni di un div figlio, esso verrà sbattuto in alto a sinistra nel container.

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

    flex-direction: row;

    flex-wrap: wrap-reverse;

}

etc

BROWSER









Quanto visto finora, ovviamente, vale anche utilizzando la proprietà display:flex e flex-wrap senza la proprietà flex-direction: row, in quanto quest’ultima è valorizzata a row di default

(FARE LA PROVA!!!)

Le proprietà flex-direction e flex-wrap possono essere inserite sulla stessa riga, sostituendole con la proprietà flex-flow.

Esempio.

Vediamo un paio di esempi per capire quale proprietà css si può utilizzare, senza esempi. Ormai dovrebbero essere chiari.

I css qui sotto sono modi diversi di scrivere la stessa cosa

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

    flex-wrap: wrap-reverse;

}

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

    flex-direction: row;

    flex-wrap: wrap-reverse;

}

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

    flex-flow: row wrap-reverse

}

**JUSTIFY CONTENT**

Ricordiamo le proprietà CSS viste finora.

* Flex-direction
* Flex-wrap

Di default, se display:flex, abbiamo che flex-direction:row e flex-wrap: nowrap.

Ricordiamo che flex-direction e flex-wrap possono essere raggruppate in una unica proprietà, chiamata flex-flow.

Riportiamo qui il nostro esempio di partenza con la proprietà flex-flow impostata al suo livello di default, ossia flex-flow: row nowrap:

HTML

<body>

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

    flex-flow: row nowrap;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

#div3{

    border:1px solid black;

    background-color:blue;

}

BROWSER



Introduciamo ora la proprietà justify-content. Questa proprietà serve ad allineare gli elementi figli sulle assi principali. Il valore di default di justify-content è flex-start. Riportiamo quindi il nostro codice qui sotto avente anche la proprietà justify-content scritta al suo valore di default.

HTML

<body>

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

    flex-flow: row nowrap;

    justify-content: flex-start;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

#div3{

    border:1px solid black;

    background-color:blue;

}

BROWSER



Tramite la proprietà justify-content e’ possibile modificare la posizione dei nostri elementi figli all’interno del nostro div container.

Supponiamo di voler spostare a destra i nostri elementi. Per farlo si utilizza justify-content:flex-ed;

HTML

<body>

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

    flex-flow: row nowrap;

    justify-content: flex-end;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

#div3{

    border:1px solid black;

    background-color:blue;

}

BROWSER



Si ricordi che abbiamo già visto una cosa simile. Avevamo inserito flex-direction:row-reverse, e gli elementi figli del div container venivano sbattuti alla destra del div container.

Tuttavia, oltre ad aver un movimento verso destra dei div figli, in quel caso avevamo anche un loro “rovesciamento”: Il primo div diventava l’ultimo, il secondo restava il secondo restava secondo e l’ultimo diventava il primo.



Invece, con justify-content:flex-end, abbiamo solo uno spostamento verso destra dei nostri div figli.



Nel nostro esempio abbiamo spostato i nostri div figli alla fine del nostro div container.  
Ovviamente i nostri div figli avrebbero potuto esse spostati al centro del div container.

Per farlo, si utilizza justify-content:center

HTML

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

    flex-flow: row nowrap;

    justify-content: center;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

#div3{

    border:1px solid black;

    background-color:blue;

}

BROWSER



Nei nostril esempi si nota che i div figli sono sempre attaccati l’uno all’altro.

Per staccarli si utilizza justify-content:space-between

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    display:flex;

    flex-flow: row nowrap;

    justify-content: space-between;

}

BROWSER



In questo modo abbiamo ottenuto uno spazio tra i vari div figli, ma non abbiamo alcuno spazio tra i bordi del div container ed il div figlio attaccato ad esso.

Per ottenere uno spazio tra il bordo del div container e quello del div figlio attaccato ad esso si usa la proprietà justify-content:space-evenly; :

CSS

#container{

    border:1px solid black;

    background-color:orangered;

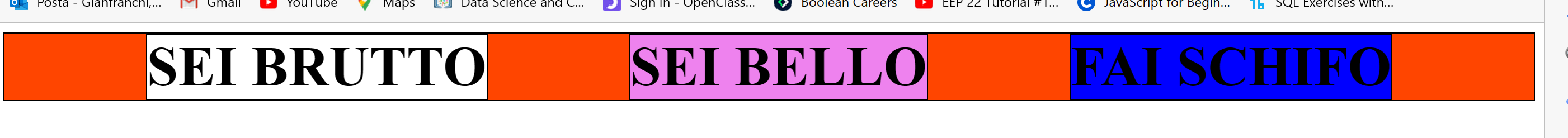
    display:flex;

    flex-flow: row nowrap;

    justify-content: space-evenly;

}

BROWSER



Si noti che lo spazio lasciato tra il bordo del container e il margine sinistro (o destro nel caso dell’ultimo elemento) e lo spazio lasciato tra un div e l’altro è identico.

**ALIGN ITEMS**

Per capire il funzionamento della proprietà align-items, supponiamo di voler aumentare l’altezza del nostro div container.

Finora, infatti, non abbiamo aumentato ne larghezza ne altezza del nostro container.

Supponiamo ora di voler impostare l’altezza del nostro div container a 600px. Vediamo cosa succede

HTML

<body>

    <div id="container">

        <div id="div1">SEI BRUTTO</div>

        <div id="div2">SEI BELLO</div>

        <div id="div3">FAI SCHIFO</div>

    </div>

</body>

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    height:600px;

    display:flex;

    flex-flow: row nowrap;

    justify-content: flex-start;

}

#div1,#div2,#div3{

    font-size: 50px;

    font-weight: bold;

}

#div1{

    border:1px solid black;

    background-color:white;

}

#div2{

    border:1px solid black;

    background-color:violet;

}

#div3{

    border:1px solid black;

    background-color:blue;

}

BROWSER

Immagine che contiene testo, schermata, Policromia

Descrizione generata automaticamente

La cosa strana è che non solo si sia modificata l’altezza del div container, ma anche l’altezza dei suoi div figli è diventata uguale a quella del div container.

Perché è successo ciò? Semplice! Di default, il div container presenta la proprietà **align-items:stretch!**

Infatti, aggiungendo tale proprietà align-items e valorizzandola a stretch abbiamo esattamente la situazione appena vista:

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    height:600px;

    display:flex;

    flex-flow: row nowrap;

    justify-content: flex-start;

    align-items:stretch;

}

BROWSER

Immagine che contiene testo, schermata, Rettangolo, Policromia

Descrizione generata automaticamente

CVD.

Come facciamo ad evitare che la lunghezza dei div si adatti a quella del div container?

Bisogna valorizzare la proprietà align-items a flex-start:

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    height:600px;

    display:flex;

    flex-flow: row nowrap;

    justify-content: flex-start;

    align-items:flex-start;

}

BROWSER

Immagine che contiene testo, schermata, Carattere, design

Descrizione generata automaticamente

Possiamo anche sbattere i nostri div figli in fondo al div container in seguito ad un aumento della lunghezza del div padre tramite la proprietà align-items valorizzata a flex-end:

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    height:600px;

    display:flex;

    flex-flow: row nowrap;

    justify-content: flex-start;

    align-items:flex-end;

}

BROWSER

Immagine che contiene testo, schermata, arancione, Carattere

Descrizione generata automaticamente

Ovviamente possiamo porre i nostri div figli al centro del nostro div padre, tramite align-items:center

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    height:600px;

    display:flex;

    flex-flow: row nowrap;

    justify-content: flex-start;

    align-items:center;

}

BROWSER

Immagine che contiene testo, schermata, Carattere, software

Descrizione generata automaticamente

Da quello che si è evinto finora, abbiamo che la proprietà **justify-content** riguarda lo spostamento dei nostri div figli lungo l’asse delle x, mentre la proprietà **align-items** riguarda lo spostamento dei nostri div figli lungo l’asse delle y.

Usate in sinergia, queste due proprietà ti permettono di inserire i nostri div figli in una qualsiasi posizione del div padre.

Ad esempio, possiamo inserire i nostri div figli esattamente al centro del nostro div padre utilizzando justify-content:center e align-items:center:

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    height:600px;

    display:flex;

    flex-flow: row nowrap;

    justify-content: center;

    align-items:center;

}

BROWSER

Immagine che contiene testo, schermata, Carattere, Rettangolo

Descrizione generata automaticamente

Cosa succede se riduciamo lo schermo? Ovviamente succede una schifezza

Immagine che contiene testo, schermata, software, Icona del computer

Descrizione generata automaticamente

Per far si che i div figli si adattino all’interno del div padre in caso di riduzione della larghezza del browser inseriamo come al solito il wrap all’interno del nostro flex-flow:

CSS

#container{

    border:1px solid black;

    background-color:orangered;

    height:600px;

    display:flex;

    flex-flow: row wrap;

    justify-content: center;

    align-items:center;

}

BROWSER

Immagine che contiene testo, schermata, Carattere, logo

Descrizione generata automaticamente