**DISPOSIZIONE ELEMENTI CSS**

**Di base, tutti gli elementi vengono posti a sinistra della pagina web.**

**Consideriamo la seguente pagina web: come si nota, essa è molto semplice, ed è utile solo per fini didattici.**

**Considereremo un container di larghezza 980px ed altezza di 650 px. Al suo interno inseriremo tutti gli elementi.**

**Il nostro container sarà quindi il seguente:**

**Files:**

**05.disposizione-elementiA.html 05.disposizione-elementiA-css**

**CODICE HTML:**

<body>

    <div id="container">

    </div>

</body>

**CODICE CSS:**

#container{

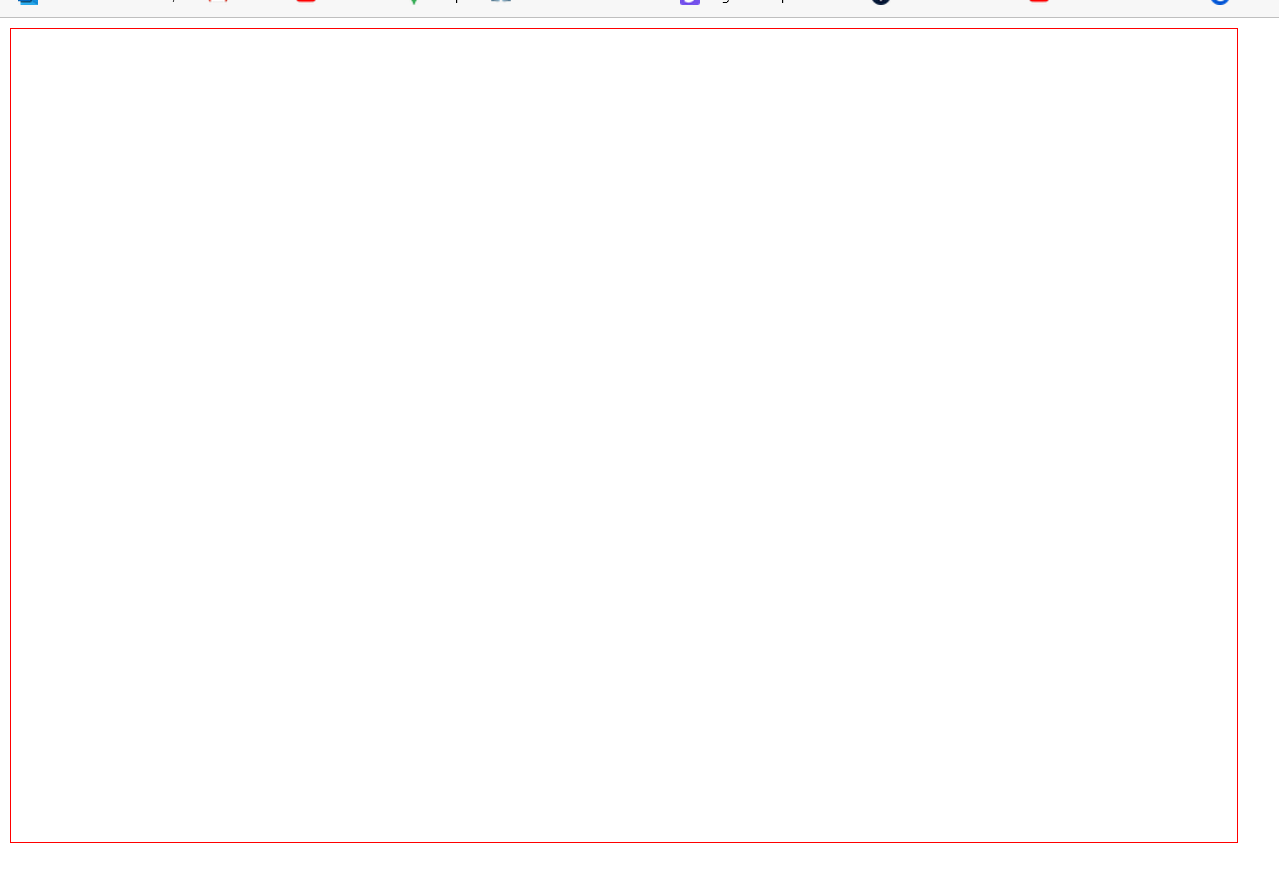
    border:1px solid red;

    width:980px;

    height:650px;

}

**Browser:**

****

**Il nostro container ha quindi un bordo rosso pari ad 1px.**

**Tutti gli elementi saranno messi al suo interno.**

**Inseriamo quindi alcuni elementi:**

**Files:**

**05.disposizione-elementiB.html 05.disposizione-elementiB-css**

**Codice HTML:**

<body>

    <div id="container">

        <div id="quadrato1"></div>

        <div id="quadrato2"></div>

        <div id="quadrato3"></div>

    </div>

</body>

**Codice CSS:**

#container{

    border:1px solid red;

    width:980px;

    height:650px;

}

div{

    width:15%;

    height: 15%;

}

#quadrato1{

    background-color:red;

}

#quadrato2{

    background-color:blueviolet;

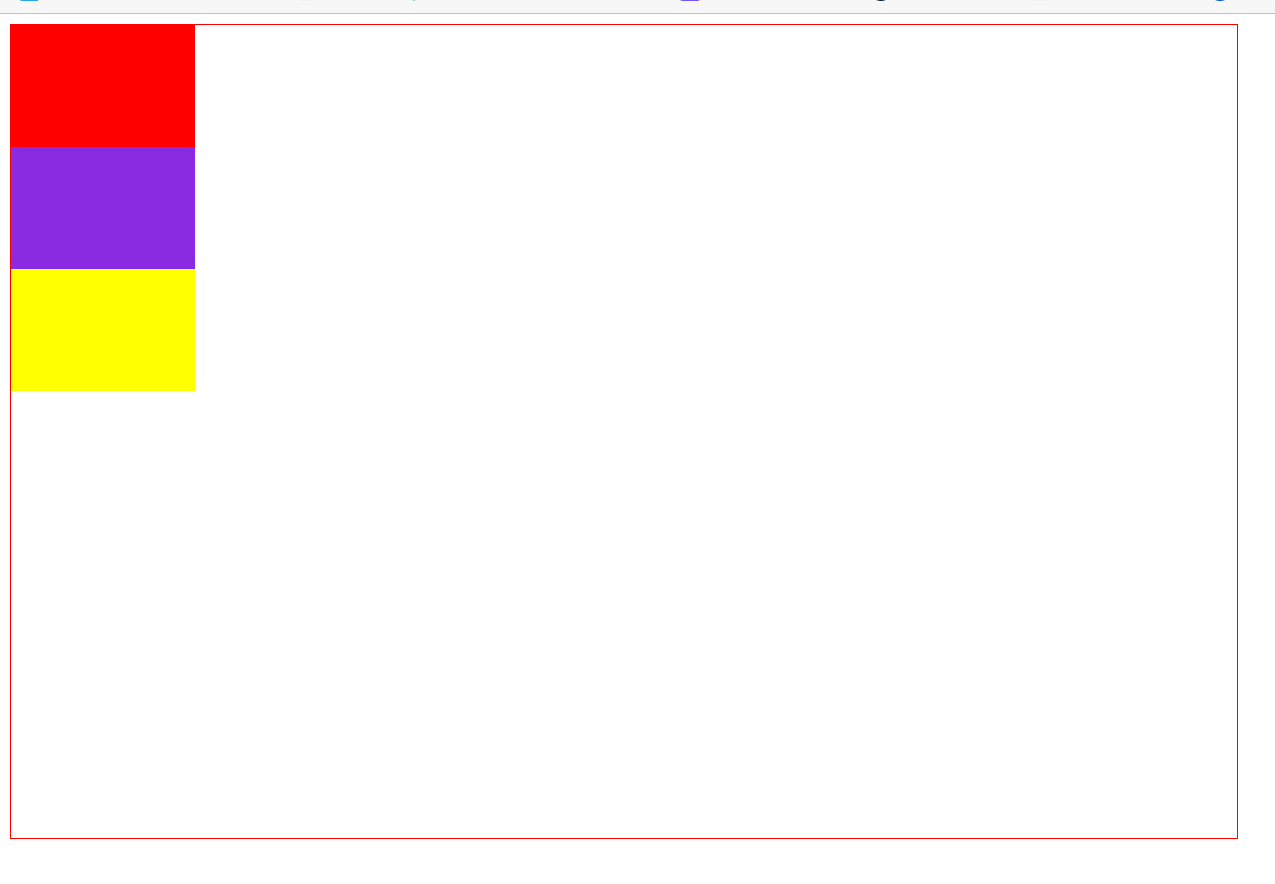
}

#quadrato3{

    background-color:yellow;

}

**BROWSER:**

****

**Si noti che, per gli elementi presenti nel container, abbiamo utilizzato le percentuali e non i pixel. Perché?**

**Perché si suppone che il container sia una pagina web. Questo significa che la posizione degli elementi al suo interno debba essere la stessa sia in relazione a tutti gli schermi di pc, sia in relazione a tutti i browser.**

**Quindi, le dimensioni e la posizione degli elementi presenti all’interno di un container non dovranno essere assolute, ma relative al container nel quale si trovano.**

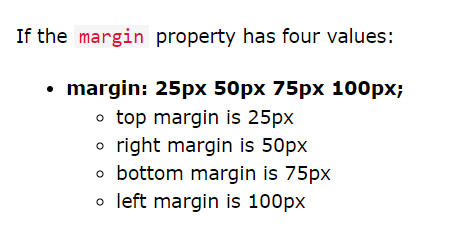
**Quindi, i div presenti nel nostro container, dovranno avere altezza e larghezza pari al 15% dell’altezza e della larghezza del container nel quale si trovano.**

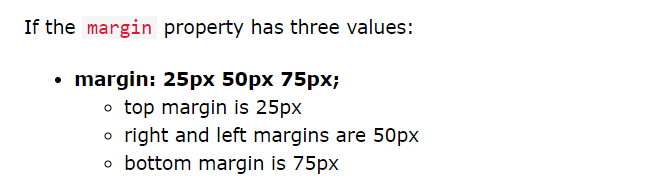
**Si noti che, di default, i nostri div si troveranno spiaccicati al bordo sx e superiore del nostro container. Inoltre, non ci sarà alcuno spazio tra l’uno e l’altro.**

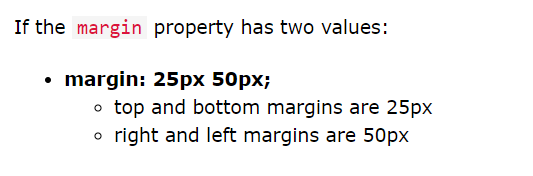
**Per prima cosa, quindi, è utile spostare i nostri div in una posizione più consona.**

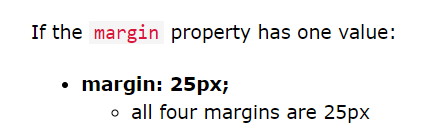
**Per spostare gli elementi utilizzeremo la proprietà margin**

**Vediamo quali sono i vari modi di scrivere questa proprietà.**

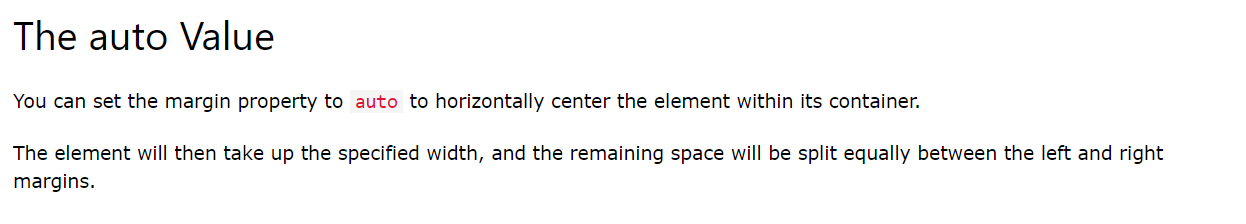
****

****

****

****

**Possiamo inoltre studiare la proprietà auto.**

****

**Prima di tornare al nostro esempio, dobbiamo notare che i div sono elementi di tipo block.**

**In altre parole, la loro proprietà display è, automaticamente, valorizzata con block.**

**Cosa è un elemento valorizzato con display:block?**

**Vediamolo in dettaglio.**

Caratteristiche e comportamenti di elementi di blocco

Le principlali caratteristiche di un elemento block sono:

1. Un elemento di blocco necessita l'attribuzione della larghezza mediante la proprietà width. Altrimenti l'elemento si estenderà per tutta la larghezza dell'elemento contenitore
2. Inserendo due o più elementi di blocco nel documento html, questi si disporranno uno sotto l'altro anche se la loro larghezza è inferiore a quella dell'elemento contenitore. Nel caso volessimo elementi di blocco affiancati, di deve ricorrere alla proprietà float
3. Ad un elemento di blocco è possibile assegnare altezza, larghezza, margini e padding
4. Un elemento di blocco può contenere altri elementi di blocco o elementi inline
5. Ad un elemento di blocco non è possibile assegnare la proprietà vertical-align
6. Molti elementi di blocco sono separati dalla proprietà margin. Questo margine dipende dall'elemento e dal browser in uso
7. I più comuni elementi di bloco sono: div, h1....h6, table, p, ol, ul ecc...

**Osserviamo il punto 6.I div sono elementi il cui margin pre-impostato è pari a 0.**

**Torniamo al nostro esempio, senza considerare le proprietà width inserita per tutti i div.**

**CODICE HTML:**

<body>

    <div id="container">

        <div id="quadrato1"></div>

        <div id="quadrato2"></div>

        <div id="quadrato3"></div>

    </div>

</body>

**CODICE CSS:**

#container{

    border:1px solid red;

    width:980px;

    height:650px;

}

div{

    height: 15%;

}

#quadrato1{

    background-color:red;

}

#quadrato2{

    background-color:blueviolet;

}

#quadrato3{

    background-color:yellow;

}

**BROWSER:**

****

**Di default, quindi, il valore width di un div è pari al valore della width del container nel quale esso si trova.**

**Ricordiamo infatti che abbiamo settato 980 come width del div container.**

**Utilizzando la freccetta, notiamo che il width dei div è, anch’esso, 980px:**

****

**Inseriamo quindi nuovamente la proprietà width ai nostri div e torniamo al nostro esempio.**

**Files:**

**05.disposizione-elementiB.html 05.disposizione-elementiB-css**

**Codice HTML:**

<body>

    <div id="container">

        <div id="quadrato1"></div>

        <div id="quadrato2"></div>

        <div id="quadrato3"></div>

    </div>

</body>

**Codice CSS:**

#container{

    border:1px solid red;

    width:980px;

    height:650px;

}

div{

    width:15%;

    height: 15%;

}

#quadrato1{

    background-color:red;

}

#quadrato2{

    background-color:blueviolet;

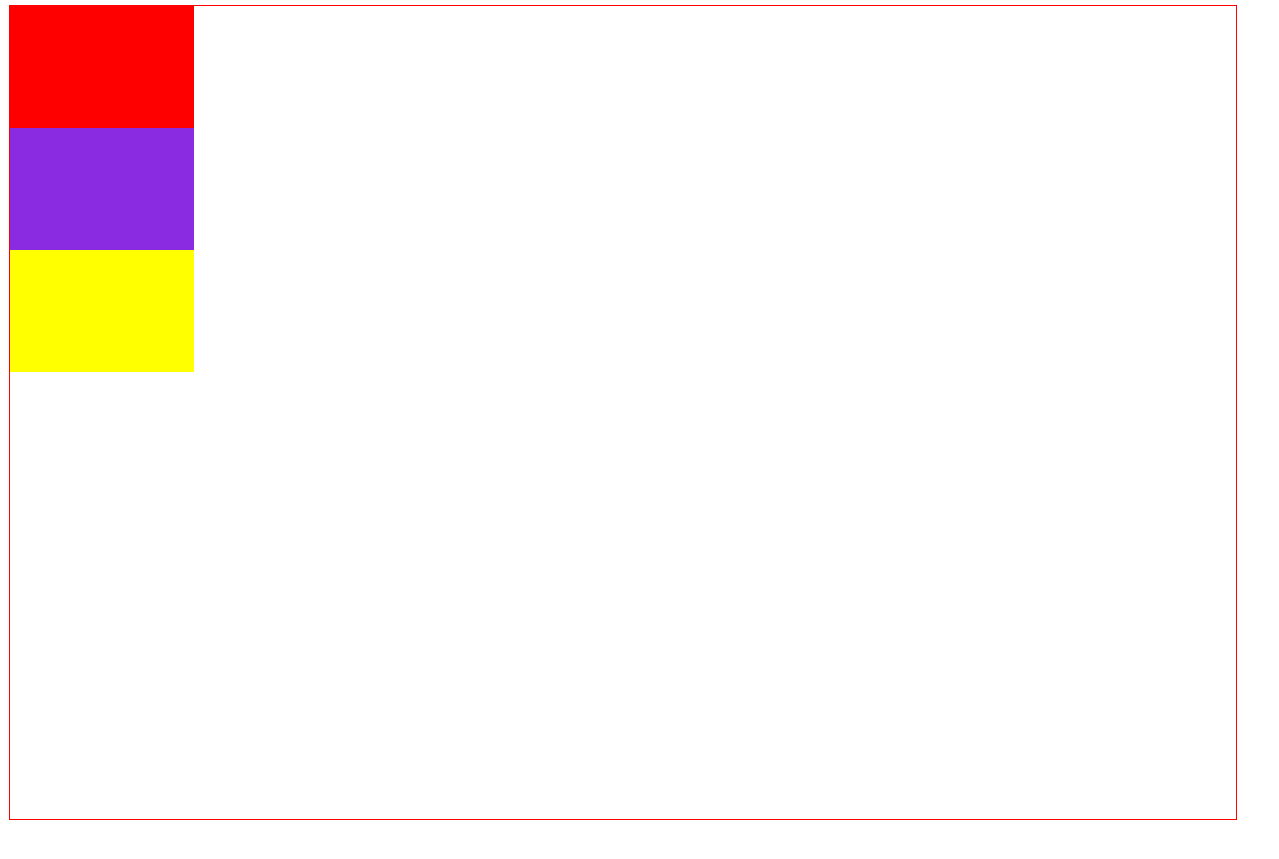
}

#quadrato3{

    background-color:yellow;

}

**BROWSER:**

****

**Applichiamo a questo punto la proprietà margin, in modo da staccare i div contenuti nel container dal bordo del container stesso.**

**In particolare, per mandare i div verso il basso possiamo anche scegliere di applicare la proprietà margin-top al solo primo div. Infatti, spingendo il primo quadrato verso il basso, in automatico spingeremo in basso anche tutti gli altri.**

**Per semplicità, comunque, applicheremo le proprietà margin-top e margin-left a tutti i div, ossia alla proprietà div presente nel nostro css.**

**FILES:**

**05.disposizione-elementiC.html 05.disposizione-elementiC.css**

**CODICE HTML:**

<body>

    <div id="container">

        <div id="quadrato1"></div>

        <div id="quadrato2"></div>

        <div id="quadrato3"></div>

    </div>

</body>

**CODICE CSS:**

#container{

    border:1px solid red;

    width:980px;

    height:650px;

}

div{

    width:15%;

    height: 15%;

    margin-left:12%;

    margin-top:12%;

}

#quadrato1{

    background-color:red;

}

#quadrato2{

    background-color:blueviolet;

}

#quadrato3{

    background-color:yellow;

}

**BROWSER:**

****

**AIUTO!!! E’ successo un casino. Abbiamo applicato margin-top e margin-left a tutti i div ed è uscito questo bordello?**

**Perché? Molto semplice.**

**Consideriamo la proprietà div prima della modifica.**

div{

    width:15%;

    height: 15%;

}

**Questo ci dice che la proprietà width ed height di tutti i div sarà del 15% rispetto al div container. Ma attenzione! Anche il container è un div! Quindi come mai width ed height vengono applicate solo ai div quadrati, ma non al div container?**

**Questo perché nel div container già abbiamo applicato delle width e delle height, e tali proprietà saranno specifiche di quel div:**

#container{

    border:1px solid red;

    width:980px;

    height:650px;

}

**Tuttavia, si noti che il nostro container non possiede una proprietà margin al suo interno.**

**Scrivendo quindi:**

div{

    width:15%;

    height: 15%;

    margin-left:12%;

    margin-top:12%;

}

**Avremo che le proprietà margin-top e margin-left verranno applicate a TUTTI i div, container compreso. Questo perché, come detto, il div avente id= container NON possiede già le proprietà margin-left e margin-top.**

**Il layout salta tutto.**

**Il modo migliore di procedere è quindi creare una classe a parte con i margini e inserirla nell’html:**

**Files:**

**05.disposizione-elementiD.html 05.disposizione-elementiC.css**

**Codice HTML:**

<body>

<body>

    <div id="container">

        <div id="quadrato1" class="margin-class"></div>

        <div id="quadrato2" class="margin-class"></div>

        <div id="quadrato3" class="margin-class"></div>

    </div>

</body>

**Codice CSS:**

#container{

    border:1px solid red;

    width:980px;

    height:650px;

}

.margin-class{

    margin-top:12%;

    margin-left:12%;

}

div{

    width:15%;

    height: 15%;

}

#quadrato1{

    background-color:red;

}

#quadrato2{

    background-color:blueviolet;

}

#quadrato3{

    background-color:yellow;

}

**Browser**

****

**Margin top sarà quindi applicato a tutti i quadrati 🡺 ogni quadrato sarà spinto verso il basso rispetto al precedente.**

**E’ arrivato il momento di studiare la proprietà FLOAT.**

**Torniamo al nostro esempio di partenza:**

**Files:**

**05.disposizione-elementiB.html 05.disposizione-elementiB.css**

**Codice HTML:**

<body>

    <div id="container">

        <div id="quadrato1"></div>

        <div id="quadrato2"></div>

        <div id="quadrato3"></div>

    </div>

</body>

**Codice CSS:**

#container{

    border:1px solid red;

    width:980px;

    height:650px;

}

div{

    width:15%;

    height: 15%;

}

#quadrato1{

    background-color:red;

}

#quadrato2{

    background-color:blueviolet;

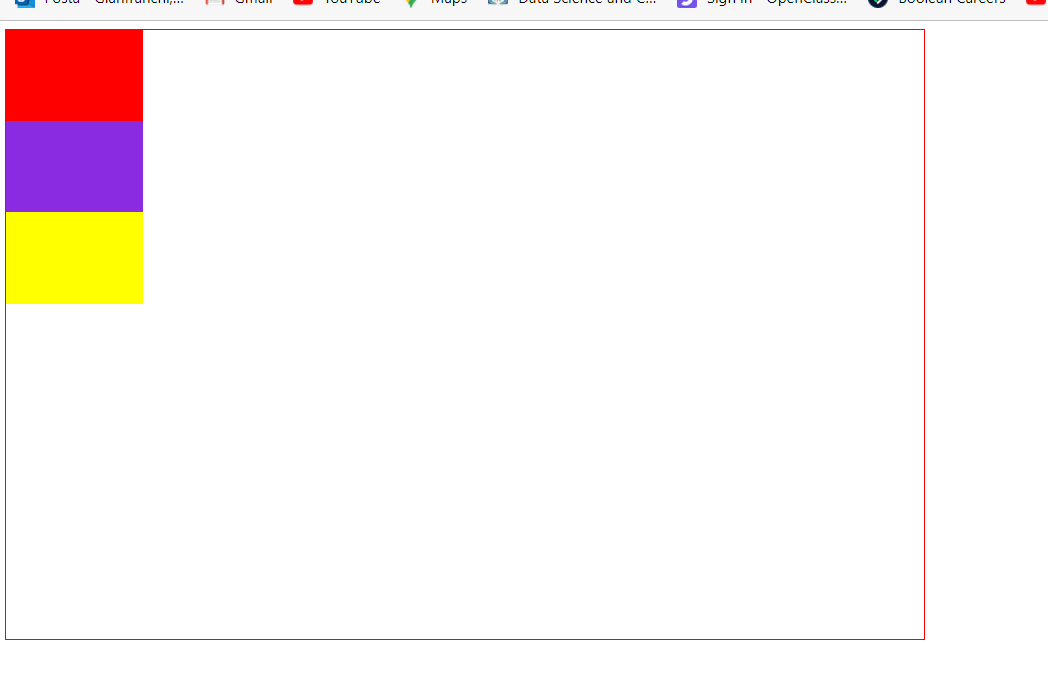
}

#quadrato3{

    background-color:yellow;

}

**Browser:**

****

**Ritorniamo a pagina 6 e rileggiamo il punto 2.**

**Ricordiamo che, in automatico, un elemento blocco ha una sua “riga”. Ciascun elemento di tipo blocco verrà inserito sotto al precedente.**

**Per mettere due o più elementi blocco su una stessa riga, dovremmo utilizzare la proprietà float.**

**Consideriamo quindi il quadrato rosso e il quadrato viola. Supponiamo di voler porre il quadrato viola alla destra del quadrato rosso.**

**Per rendere possibile ciò, dovremmo utilizzare la proprietà float sia per il quadrato rosso che per il quadrato viola.**

**Vediamo quindi come fare.**

**Files:**