

Float CSS

In questa lezione studieremo la proprietà float dei CSS.

Questa proprietà consente di rimuovere un elemento dal suo normale flusso e quindi di poterlo spostare a destra oppure a sinistra.

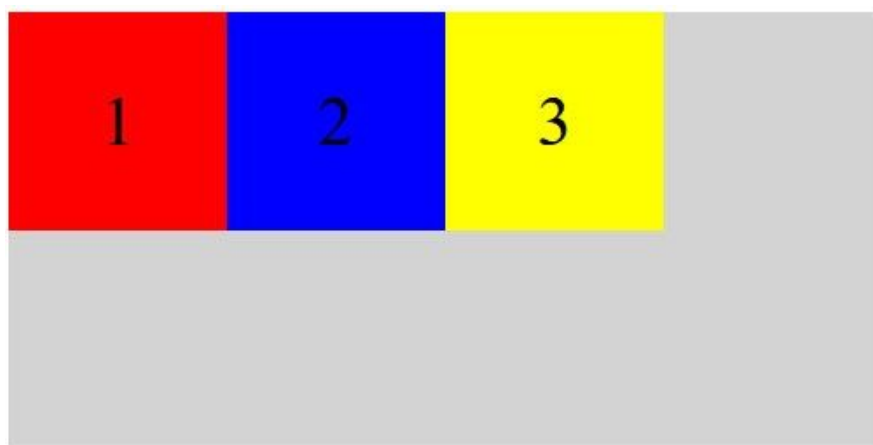
Se ad un elemento assegniamo la proprietà float, automaticamente diventa un block a cui poter assegnare larghezza, bordi, margini, ecc.

Proprietà float CSS

La proprietà float consente di impostare i seguenti valori: left, right, none, initial, inherit. Dove none è il valore di default.

Primo esempio con float left

Facciamo degli esempi per comprenderne il funzionamento. Supponiamo dunque di avere un elemento contenitore con degli elementi (div) uno accanto all'altro, come da figura sotto:



Uno dei modi per ottenere questo risultato è utilizzare quindi la proprietà float left dei CSS ed assegnarla a ciascun contenitore.

Ecco dunque la pagina html di esempio:


```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <link rel="stylesheet" href="css/style.css">
</head>
<body>
  <div class="container">
    <div class="box red">1</div>
    <div class="box blue">2</div>
    <div class="box yellow">3</div>
  </div>

</body>
</html>
```

Ed il codice CSS necessario per realizzare questo esempio:

```
.container {
  background-color: lightgrey;
  width: 200px;
  height: 100px;
}

.box {
  width: 50px;
  height: 50px;
  line-height: 50px;
  text-align: center;
  float:left;
}

.red {
  background-color: red;
}

.blue {
  background-color: blue;
}

.yellow {
```

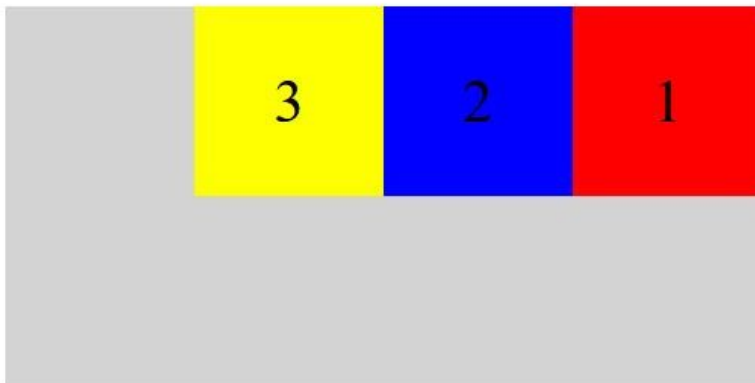


```
background-color: yellow;
}
```

Abbiamo scelto, per il momento, di dare un'altezza ed una larghezza fissa al contenitore esterno. Se non facciamo questa operazione il contenitore grigio non sarebbe visibile perché non prenderebbe in automatico l'altezza dei contenitori inseriti al suo interno. Nelle prossime lezioni vedremo come risolvere questo problema.

Secondo esempio con float right CSS

In questo secondo esempio ci prefissiamo di ottenere l'effetto come da figura, con gli elementi, a partire dal primo, allineati al margine destro.



Il codice html non cambia rispetto all'esempio precedente, mentre il codice CSS cambia solo per la classe float-right assegnata ai 3 contenitori.

```
.container {
background-color: lightgrey;
width: 200px;
height: 100px;
}
```

```
.box {
width: 50px;
height: 50px;
line-height: 50px;
text-align: center;
float:right;
}
```



```
.red {  
  background-color: red;  
}  
  
.blue {  
  background-color: blue;  
}  
  
.yellow {  
  background-color: yellow;  
}
```

Terzo esempio con float right CSS

In questo terzo esempio allineiamo il primo contenitore ed il secondo sulla destra, mentre il terzo lo posizioniamo sulla sinistra.



Quindi costruiamo la nostra pagina inserendo un contenitore principale e all'interno inseriamo 3 box come nell'esempio precedente.

Poi assegniamo al numero 3 la classe float left mentre agli altri 2 la classe float right.

Ecco dunque il codice di esempio:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="UTF-8">
```



```
<meta name="viewport" content="width=device-width,
initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
<link rel="stylesheet" href="css/style.css">
</head>
<body>
  <div class="container">
    <div class="box red float-right">1 float-right</div>
    <div class="box blue float-right">2 float-right</div>
    <div class="box yellow float-left">3 float-left</div>
  </div>

</body>
</html>
```

Nel css inseriamo una classe apposita che consente agli elementi di flottare a sinistra e un'altra che consente agli elementi di flottare a destra.

```
.container {
  background-color: lightgrey;
  width: 200px;
  height: 100px;
}

.float-left {
  float:left;
}

.float-right {
  float:right;
}

.box {
  width: 50px;
  height: 50px;
  text-align: center;
}

.red {
  background-color: red;
}

.blue {
```



```
background-color: blue;  
}
```

```
.yellow {  
background-color: yellow;  
}
```

