

學習要點:

1. 新版本

本章主要探討 HTML5 中 CSS3 提供的用來實現未來響應式彈性伸縮佈局方案，這裡做一個初步了解。

一、新版本

新版本的 Flexbox 模型是 2012 年 9 月提出的工作草案，這個草案是由 W3C 推出的最新語法。這個版本立志指定標準，讓新式的瀏覽器全面兼容，在未來瀏覽器的更新時代中實現了統一。

首先，設置伸縮盒的 display 有如下兩個屬性值：

屬性值	說明
flex	將容器盒模型作為塊級彈性伸縮盒顯示(新版本)
Inline-flex	將容器盒模型作為內聯級彈性伸縮盒顯示(新版本)

//大部分不需要前綴

```
div {  
    display: flex;  
}
```

屬性	IE	Firefox	Chrome	Opera	Safari
帶前綴	無	無	21 ~ 28	無	7.0
不帶前綴	11+	20+	29+	12.1	無

從這張表可以看出，只有 webkit 引擎的瀏覽器 Chrome 和 Safari 的版本需要添加-webkit-，而 Chrome 瀏覽器 29 版本以後可以省略了。

1. flex-direction

flex-direction 屬性和舊版本 box-orient 屬性一樣，都是設置伸縮項目的排列方式。

//設置從上往下排列

```
div {  
    flex-direction: column;  
}
```

屬性值	說明
row	設置從左到右排列
row-reverse	設置從右到左排列
column	設置從上到下排列
column-reverse	設置從下到上排列

2. flex-wrap

`flex-wrap` 屬性類似於舊版本中的 `box-lines`，但是 `box-lines` 我們沒有講解，原因是沒有瀏覽器支持它。

//設置無法容納時，自動換行

```
div {  
    -ms-flex-wrap;  
}
```

屬性值	說明
<code>nowrap</code>	默認值，都在一行或一列中顯示
<code>wrap</code>	伸縮項目無法容納時，自動換行
<code>wrap-reverse</code>	伸縮項目無法容納時，自動換行，方向和 <code>wrap</code> 相反

3. flex-flow

`flex-flow` 屬性是集合了排列方向和控制換行的簡寫形式

//簡寫形式

```
div {  
    flex-flow: row wrap;  
}
```

4. justify-content

`justify-content` 屬性和舊版本中的 `box-pack` 一樣，設置伸縮項目的對齊方式。

//按照中心點對齊

```
div {  
    justify-content: space-around;  
}
```

屬性值	說明
<code>flex-start</code>	伸縮項目以起始點對齊
<code>flex-end</code>	伸縮項目以結束點對齊
<code>center</code>	伸縮項目以中心點對齊
<code>space-between</code>	伸縮項目平局分布
<code>space-around</code>	同上，但兩端保留一半空間

5. align-items

`align-items` 屬性和舊版本中的 `box-align` 一樣，處理伸縮項目容器得額外空間。

//處理額外空間

```
div {
```

`align-items: center;`

`}`

屬性值	說明
<code>flex-start</code>	伸縮項目以頂部為基準，清理下部空間
<code>flex-end</code>	伸縮項目以底部為基準，清理上部空間
<code>center</code>	伸縮項目以中部為基準，清理上下部空間
<code>baseline</code>	伸縮項目以基線為基準，清理額外的空間
<code>stretch</code>	伸縮項目填充整個容器，默認

6. align-self

`align-self` 和 `align-items` 一樣，都是清理額外空間，但它單獨設置某一個伸縮項目的。所有的值和 `align-items` 一致。

//單獨設置清理額外空間

```
p:nth-child(2) {  
    align-self: center;  
}
```

7. flex

`flex` 屬性和舊版本中的 `box-flex` 類似，用來控制伸縮容器的比例分配。

//設置比例分配

```
p:nth-child(1) {  
    flex: 1;  
}  
p:nth-child(2) {  
    flex: 3;  
}  
p:nth-child(3) {  
    flex: 1;  
}
```

8. order

`order` 屬性和 `box-ordinal-group` 屬性一樣控制伸縮項目出現的順序。

//設置伸縮項目順序

```
p:nth-child(1) {  
    order: 2;  
}
```