

Sử dụng flexbox bố cục trang trong CSS

Thiết lập kích hoạt flexbox để dàn trang một cách mềm dẻo, thiết lập các cách hiện thị phần tử trong hộp flexbox

Sử dụng **flexbox** (Flexible Box Module), cho phép bố cục giao diện hết sức mềm dẻo, dễ tùy biến, đáp ứng cao về tính năng **Responsive**, **flexbox** được thiết kế như là mô hình layout một chiều, từ đó các phần tử được phân bổ, căn chỉnh phù hợp.

Thiết lập phần tử chứa - kích hoạt flex

Để sử dụng tính năng **flexbox** cần sử dụng theo mô hình cần có một phần tử chính thiết lập là phần tử chứa, sau đó là các phần tử con bên trong nó. Để kích hoạt phần tử chứa có tính năng **flexbox** chỉ cần thiết lập cho nó thuộc tính **display: flex;** hoặc **display: inline-flex;**. Ví dụ sau thiết lập lớp **.containerflex** có kích hoạt **flexbox**, sau đó triển khai HTML áp dụng nó

Khi phần tử đã kích hoạt **flexbox** thì điều khiển các phần tử con bằng các thuộc tính: **flex-direction** **flex-wrap** **justify-content** **align-items** **align-content**

```
.containerflex {
    display: flex;                /*Kích hoạt flexbox*/
    flex-direction: row;          /*Thiết lập hướng chính*/
    flex-wrap: nowrap;           /*Thiết lập chế độ wrap, khi các phần tử vượt
qua kích thước hướng chính*/
    justify-content: flex-end;    /*Căn chỉnh nội dung bên trong container có
flexbox*/
    align-items: stretch;        /*Căn chỉnh phần tử con theo hướng vuông góc
hướng chính*/
    align-content: stretch;      /*Căn chỉnh hàng (cột) theo hướng vuông góc
(chỉ hiệu lực khi nhiều hàng hoặc cột)*/

    padding: 5px;
    background: #9C27B0;
    min-width: 320px;
    height: 480px;
    margin: auto;
}

.flexitem {
    padding: 20px;
    background: lightblue;
    margin: 1px;
    font-weight: bold;
}

<div class="containerflex" title="container" data-toggle="tooltip" data-
placement="top">
    <div class="flexitem">1</div>
    <div class="flexitem">2</div>
    <div class="flexitem">3</div>
    <div class="flexitem">4</div>
    <div class="flexitem">5</div>
```

```
<div class="flexitem">6</div>
</div>
```

CSS phần tử cha (container):

```
.containerflex {
  display: flex;
  flex-direction: row;
  flex-wrap: nowrap;
  justify-content: flex-start;
  align-items: stretch;
  align-content: stretch;
}
```

Thuộc tính flex-direction

Khi kích hoạt **flex** cho một phần tử, thì các phần tử con chứa trong nó sẽ được bố trí liên tục theo một hướng được gọi là hướng chính. Hướng chính được thiết lập bằng thuộc tính **flex-direction** - khi đã có hướng chính thì hướng thứ hai sử dụng đến là hướng vuông góc với hướng chính (từ sau gọi là hướng vuông góc), **flex-direction** nó nhận các giá trị:

- **row** (mặc định) : Hướng chính nằm ngang, phần tử sắp xếp từ trái qua phải
- **row-reverse**: Hướng chính nằm ngang, phần tử sắp xếp từ phải qua trái
- **column**: Hướng chính thẳng đứng, phần tử xếp từ trên xuống
- **column-reverse**: Hướng chính thẳng đứng, phần tử xếp từ dưới lên

Thuộc tính flex-wrap

Nó nhận các giá trị

- **nowrap** (mặc định) : Các phần tử cứ xếp theo hướng chính, kể cả vượt ngoài khung container
- **wrap**: Các phần tử xếp theo hướng chính, nếu vượt qua kích thước khung chứa theo hướng chính sẽ ngắt tạo ra luồng xếp tiếp theo (hàng tiếp theo hoặc cột tiếp theo tùy thuộc đang là hướng chính nào).
- **wrap-reverse**: Tương tự **wrap** nhưng hàng (cột) bố trí ngược lại

Thuộc tính justify-content

Nó nhận các giá trị để đẩy các phần tử trôi về phía nào đó của hướng chính

- **flex-start** (mặc định) : đẩy các phần tử con trôi về khu vực bắt đầu hướng chính
- **flex-end** đẩy các phần tử về khu vực cuối hướng chính
- **center** đẩy các phần tử vào giữa hướng chính
- **space-around** không gian thừa theo hướng chính chia đều cho khoảng cách các phần tử
- **space-between** không gian thừa theo hướng chính chia đều cho khoảng cách các phần tử (phần tử đầu và cuối sát mép khung)

Thuộc tính align-items

Nó nhận các giá trị để điều khiển phần tử con theo hướng **Vuông góc** hướng chính

- **stretch** (mặc định) : phần tử con mở rộng kích thước tối đa (bằng với chiều cao hàng hoặc chiều rộng cột) của phần tử theo hướng vuông góc hướng chính
- **baseline** các phần tử con bám theo đường baseline của hàng (cột)
- **center** phần tử đẩy vào giữa hàng (cột)
- **space-around** không gian thừa theo hướng chính chia đều cho khoảng cách các phần tử
- **space-between** không gian thừa theo hướng chính chia đều cho khoảng cách các phần tử (phần tử đầu và cuối sát mép khung)

Thuộc tính align-content

Nó điều chỉnh vị trí các hàng (các cột) trong khung, nó nhận các giá trị như stretch, center, flex-end ...

Chỉ có hiệu lực khi có nhiều hàng hoặc nhiều cột

Thiết lập các phần tử con

Tính năng liên quan đến **flex** ngoài thiết lập với các thuộc tính từ phần tử chứa (cha - container), còn có thể thiết lập trực tiếp đến từng phần tử con, xem xét các thuộc tính

- flex-grow
- flex-shrink
- flex-basis
- flex
- align-self

Các ví dụ ở phần sau sẽ sử dụng đến CSS như sau

```
.exam-container {
  display: flex;
  background: #ab7bb0;
  padding: 4px;
}
.exam-element {
  background-color: orangered;
  margin: 1px;
  color: white;
  min-height: 40px;
  justify-content: center;
  display: flex;
  align-items: center;
}
```

Thuộc tính flex-grow

flex-grow gán các giá trị bằng số, nó cho biết phần tử này chiếm bao nhiêu phần trong không gian còn lại (trừ đi phần đã chiếm bởi các phần tử có flex-grow bằng 0) khung chứa theo hướng chính.

```
<div class="exam-container" id="flex-grow">
  <div class="exam-element flex-grow-ele1" style="flex-grow: 1">1</div>
  <div class="exam-element flex-grow-ele2" style="flex-grow: 2">2</div>
  <div class="exam-element flex-grow-ele3" style="flex-grow: 3">3</div>
</div>
```

Thuộc tính flex-shrink

Thuộc tính **flex-shrink** nhận giá trị là số, cho biết phần tử có thể co lại bao nhiêu phần khi cần thiết

Thuộc tính flex-basis

Thuộc tính gán chiều dài khởi tạo cho phần tử trong hệ thống flex, ví dụ **flex-basis:100px**

Thuộc tính flex

Thuộc tính này là cách viết tổng hợp cả ba thuộc tính **flex-grow**, **flex-shrink**, **flex-basis** trên một dòng

Ví dụ: **flex: 1 0 100px**

Thuộc tính align-self

Căn chỉnh một phần tử trong hệ thống flex, **align-self** nhận các giá trị: **stretch**, **flex-end**, **center**, **flex-start**

Cũng lưu ý có thể kết hợp với các thuộc tính margin để điều chỉnh vị trí

```
<div class="exam-container" style="height:200px;">
  <div class="exam-element" style="align-self: stretch">1</div>
  <div class="exam-element" style="align-self: flex-end">2</div>
  <div class="exam-element" style="align-self: center">3</div>
  <div class="exam-element" style="align-self: self-start; margin-left:
auto">1</div>
</div>
```