Học Về FlexBox

Flexbox là gì?

Flexbox là hộp linh hoạt, nó giúp cho các bạn dễ dàng thay đổi định vị của các phần tử

Các thành phần của flexbox

Để tìm hiểu các thành phần của flexbox thì các bạn xem xét ví dụ sau:

```
<style>
    .box_container {
        display: flex;
        width: 500px;
        border: 2px dotted lightseagreen;
    }
    .box_container .box {
        width: 100px;
        height: 100px;
        background: #ccc;
        border: 1px solid brown;
    }
</style>
<h1>Các thành phần của Flexbox</h1>
<div class="box_container">
    <div class="box">One</div>
    <div class="box">Two</div>
</div>
```

Khi chọn phần tử có class là box_container và thêm thuộc tính css là display: flex thì ngay lập tức phần tử đó đóng vai trò là **flex container**. Các thành con trực tiếp của flex container (hay chính là các element có class box) trở thành các **flex items**

Câu lệnh **display: flex** sẽ thay đổi các phần tử **flex items** được sắp xếp trên cùng một dòng và vị trí bắt đầu sắp xếp từ bên trái và top của **flex container**

Tìm hiểu tại sao các **flex items** mặc định được xếp từ trái sang phải

Khi làm việc với flexbox bạn cần nghĩ đến 2 trục: main axit and cross axis. trục main axis được xác định bởi thuộc tính **flex-direction** và trục cross axis vuông góc với nó. Các items sẽ được sắp xếp theo trục chính, mặc định giá trị **flex-direction: row** nên các items được sắp xếp từ trái qua phải, từ trên xuống dưới

justify-content - căn chỉnh các items dọc theo trục chính

justify-content được sử dụng để căn chỉnh các mục items trên trục chính mà ở đó flex-direction đang được xét, nó bao gồm các giá trị như sau: (thực tế nó thể hiện cách sử dụng khoảng trống còn lại của container)

flex-start: sắp xếp các items ở cạnh bắt đầu của container flex-end: sắp xếp các items ở cuối của container center: sắp xếp các items ở giữa của container space-between: items will be positioned with equal space between them, but no extra space before the first or after the last elements space-around: items will be positioned with equal space before and after each item, resulting in double the space between elements space-evenly: các khoảng cách được chia đều cho cả 2 đầu

align-items - căn chỉnh các mục trên trục chéo

Giá trị khởi tạo ban đầu là `stretch` và đây là lí do tại sao các mục linh hoạt sẽ kéo dài theo chiều cao của thùng chứa linh hoạt theo mặc định. Điều này được quyết định bởi item cao nhất trong thùng chứa hoặc theo kích thước được đặt trên thùng chứa linh hoạt (trường hợp này ít dùng, thông thường lưu ý không fix chiều cao cố định)

Bạn có thể xem xét đoạn code sau để rõ hơn:

```
<style>
    .box_container_align {
        display: flex;
        width: 500px;
        height: 200px;
        border: 2px dotted brown;
    }
    .box_container_align .box {
        width: 100px;
        background: lightblue;
        border: 2px solid rgb(96, 139, 168);
    }
</style>
<h1>Học về align items</h1>
<div class="box_container_align">
    <div class="box">One</div>
    <div class="box">Two</div>
    <div class="box">
        Three<br/>
        Have<br/>
        Extra Text
    </div>
</div>
```

Bạn chạy lên sẽ thấy các item có cùng độ cao với box thứ 3, mặc dù 2 box kia có ít nội dung hơn

Một số giá trị khác để điều chỉnh các items dọc theo trục vuông góc:

`flex-start`: Tất cả các phần tử sẽ được định vị ở đầu vùng chứa `flex-end`: Tất cả các phần tử sẽ được định vị ở dưới cùng của vùng chứa `center`: Tâm của tất cả các phần tử sẽ được đặt ở giữa phần trên và phần dưới của vùng chứa `baseline`: Phần dưới cùng nội dung của tất cả các mục sẽ được căn chỉnh với nhau `stretch`: Nếu khả thi, các items sẽ được kéo dài từ trên xuống dưới thùng (Đây là giá trị mặc định, các phần tử có chiều cao cố định không được kéo dài, các phần tử có chiều cao tối thiểu hoặc không có chiều cao xác định sẽ được kéo dài)

flex-wrap

Mặc định nếu không đủ không gian trong container, các item cố gắng thu nhỏ lại để vừa vùng chứa. Chúng ta mong muốn các items sẽ chuyển đến dòng tiếp theo nếu không có đủ không gian. Điều này làm được bằng cách sử dụng thuộc tính `flex-wrap`

1. `wrap`: Các phần tử con của một thùng chứa linh hoạt không vừa với một hàng sẽ di chuyển xuống dòng tiếp theo
2. `nowrap`: prevents items from wrapping; this is the default value and is only necessary to override a wrap value set by a different CSS rule
3. `wrap-reverse`: Chức năng tương tự wrap nhưng thứ tự các hàng trong thùng chứa bị đảo ngược

Tại sao cần inline-block

Xem xet ví dụ sau:

```
    <style>
        span {
            padding: 20px;
            margin: 20px;
            background: lightcoral;
            width: 100px;
            height: 100px;
        }
    </style>
    <h1>Học về inline-block</h1>
    <p>
        Lorem ipsum dolor sit amet consectetur adipisicing elit.
        Nulla officiis, dolorem quia sint quidem commodi quasi velit distinctio?
        Ipsum excepturi eveniet necessitatibus voluptas atque eum laudantium
vitae. Laborum.
    </p>
    <p>
        Lorem ipsum dolor sit amet consectetur adipisicing elit.
        Nulla officiis, dolorem quia sint quidem commodi quasi velit distinctio?
        Ipsum excepturi eveniet necessitatibus <span>truong hoc</span> voluptas
atque eum laudantium vitae. Laborum.
    </p>
```

Với thẻ span inline thì hướng của nó sẽ được sắp xếp theo chiều ngang nên các chiều đứng, width, height không được tôn trọng vì vậy chúng ta sẽ chuyển sang inline-block để đảm bảo nó vẫn nằm trên 1 hàng nhưng lại có tính chất của block

css margin ứng dụng căn giữa