

Mise en page avec Flexbox

Boîte flexible CSS

La propriété CSS `display: flex` définit un élément HTML comme un conteneur flexible au niveau du bloc qui prend toute la largeur de son conteneur parent. Tous les éléments enfants qui résident dans le conteneur flexible sont appelés éléments flexibles. Les éléments flexibles changent de taille et d'emplacement en réponse à la taille et à la position de leur conteneur parent.

```
div {  
  affichage : flexible ;  
}
```

`justify-content` Propriété

La propriété CSS `justify-content` flexbox définit la façon dont le navigateur distribue l'espace entre et autour des éléments de contenu le long de l'axe principal de leur conteneur. C'est lorsque les éléments de contenu n'utilisent pas tout l'espace disponible sur l'axe principal (horizontalement).

`justify-content` peut prendre les valeurs de :

- `flex-start`
- `flex-end`
- `center`
- `space-between`
- `space-around`

```
/* Éléments basés au centre du conteneur  
parent : */
```

```
div {  
  affichage : flexible ;  
  justifier-contenu : center ;  
}
```

```
/* Éléments basés sur le côté supérieur  
gauche du conteneur parent : */
```

```
div {  
  affichage : flexible ;  
  justifier-contenu : flex-start ;  
}
```

`flex` Propriété

La `flex` propriété CSS spécifie comment un élément flex grandira ou rétrécira afin de s'adapter à l'espace disponible dans son flex conteneur. Il s'agit d'une propriété abrégée qui déclare les propriétés suivantes dans l'ordre sur une seule ligne :

- `flex-grow`
- `flex-shrink`
- `flex-basis`

```
/* Trois propriétés déclarées sur trois  
lignes : */
```

```
.first-flex-item {  
  flex-croissance : 2 ;  
  flex-rétractable : 1 ;  
  flex-base : 150px ;  
}
```

```
/* Trois mêmes propriétés déclarées sur  
une seule ligne : */
```

```
.first-flex-item {
```

```
}
```

flex-direction

La `flex-direction` propriété CSS spécifie comment les éléments flexibles sont placés dans le conteneur flexible – soit verticalement, soit horizontalement. Cette propriété détermine également si ces éléments flexibles apparaissent dans l'ordre ou dans l'ordre inverse.

```
div {  
  display: flex;  
  flex-direction: row-reverse;  
}
```

align-content

The `align-content` property modifies the behavior of the `flex-wrap` property. It determines how to space rows from top to bottom (ie. along the cross axis). Multiple rows of items are needed for this property to take effect.

flex-grow

The CSS `flex-grow` property allows flex items to grow as the parent container increases in size horizontally. This property accepts numerical values and specifies how an element should grow relative to its sibling elements based on this value. The default value for this property is `0`.

```
.panelA {  
  width: 100px;  
  flex-grow: 1;  
}  
  
/* This panelB element will stretch twice  
wider than the panelA element */  
.panelB {  
  width: 100px;  
  flex-grow: 2;  
}
```

flex-shrink

The CSS `flex-shrink` property determines how an element should shrink as the parent container decreases in size horizontally. This property accepts a numerical value which specifies the ratios for the shrinkage of a flex item compared to its other sibling elements within its parent container. The default value for this property is `1`.

```
.container {  
  display: flex;  
}  
  
.item-a {  
  flex-shrink: 1;  
  /* The value 1 indicates that the item  
should shrink. */  
}
```

```
.item-b {
  flex-shrink: 2;

  /* The value 2 indicates that the item
  should shrink twice than the element
  item-a. */
}
```

Css flex-basis property

The `flex-basis` CSS property sets the initial base size for a flex item before any other space is distributed according to other flex properties.

```
// Default Syntax
flex-basis: auto;
```

The CSS flex-flow property

The CSS property `flex-flow` provides a shorthand for the properties `flex-direction` and `flex-wrap`. The value of the `flex-direction` property specifies the direction of the flex items and the value of the `flex-wrap` property allows flex items to move to the next line instead of shrinking to fit inside the flex container. The `flex-flow` property should be declared on the flex container.

// In this example code block, "column" is the value of the property "flex-direction" and "wrap" is the value of the property "flex-wrap".

```
.container {
  display: flex;
  flex-flow: column wrap;
}
```

CSS display: inline-flex property

The CSS `display: inline-flex` property sets an HTML element as an inline flex container which takes only the required space for the content. Any child elements that reside within the flex container are called flex items. Flex items change their size and location in response to the size and position of their parent container.

```
.container{
  display: inline-flex;
}
```

Flexbox Properties align-items

When working with CSS flexbox `align-items` is used to align flex items vertically within a parent container.

Css flex-wrap property

The `flex-wrap` property specifies whether flex items should wrap or not. This applies to flex items only. Once you tell your container to `flex-wrap`, wrapping become a priority over shrinking. Flex items will only

```
.container {
  display: flex;
  flex-wrap: wrap;
  width: 200px;
```

begin to wrap if their combined `flex-basis` value is greater than the current size of their flex container.

```
}
```

 Save  Print  Share ▼