

Flexbox CSS

Introducción a flexbox

El módulo de flexbox o caja flexible establece un modelo unidimensional de diseño (de layout). Es decir, maneja el diseño de los elementos solo como filas (1 dimensión) o solo como columnas (1 dimensión).

El grid layout es la alternativa a flexbox, con un modelo bidimensional que controla columnas y filas a la vez.

Extracto de ejemplo grid layout:

```
.contenedor { display: grid; ... }
```

```
.elemento1{ grid-column: 1 / 3; grid-row: 1; ... }
```

Estructuración de flexbox

Flexbox necesita siempre de un contenedor (flex container) y los elementos contenidos en este (flex items).

Dentro de una página puede haber tantos contenedores como necesitemos, con sus correspondientes grupos de elementos.

Para trabajar con flexbox en el contenedor estableceremos:

```
display: flex;
```

Ejemplo flexbox - index.html

...

```
<body>
```

```
<div class="contenedor">
```

```
  <div class="elementos">1</div>
```

```
  <div class="elementos">2</div>
```

```
  <div class="elementos">3</div>
```

```
</div>
```

```
</body>
```

...

Ejemplo flexbox - estilo.css

```
*{  
    /* border y padding no suman */  
    box-sizing: border-box;  
    margin: 0;  
    padding: 0;  
}
```

Ejemplo flexbox - estilo.css

```
.contenedor{  
    display: flex;  
    width: 100%;  
    padding: 1%;  
    background-color: #CoCoCo;  
}
```

Ejemplo flexbox - estilo.css

```
.elementos {  
    width: 10%;  
    margin: 1%;  
    padding: 1%;  
    background-color: #000000;  
    color: #ffffff;  
}
```

Ejemplo flexbox - Resultado



Propiedades de flex container (contenedor)

1. `flex-direction`
2. `flex-wrap`
3. `flex-flow`
4. `justify-content`
5. `align-items`
6. `align-content`

Contenedor: flex-direction

Definición

Permite indicar la dirección y el orden de los elementos dentro del contenedor.

Contenedor: flex-direction

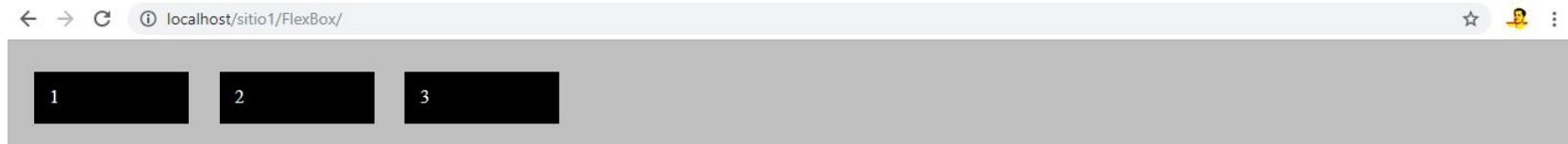
Sintaxis

```
.contenedor {  
    display: flex;  
    flex-direction: row | column | row-reverse | column-reverse;  
    width: 100%;  
    padding: 1%;  
    background-color: #CoCoCo;  
}
```

Contenedor: flex-direction

flex-direction: row;

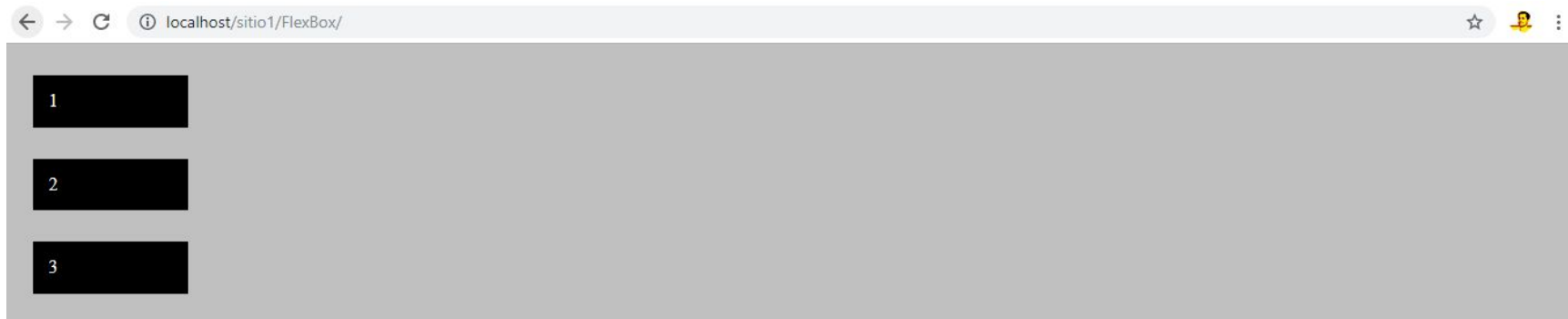
Valor por defecto. Alinea los elementos en fila, de izquierda a derecha.



Contenedor: flex-direction

flex-direction: column;

Alinea los elementos en columna, de arriba a abajo.



Contenedor: flex-direction

flex-direction: row-reverse;

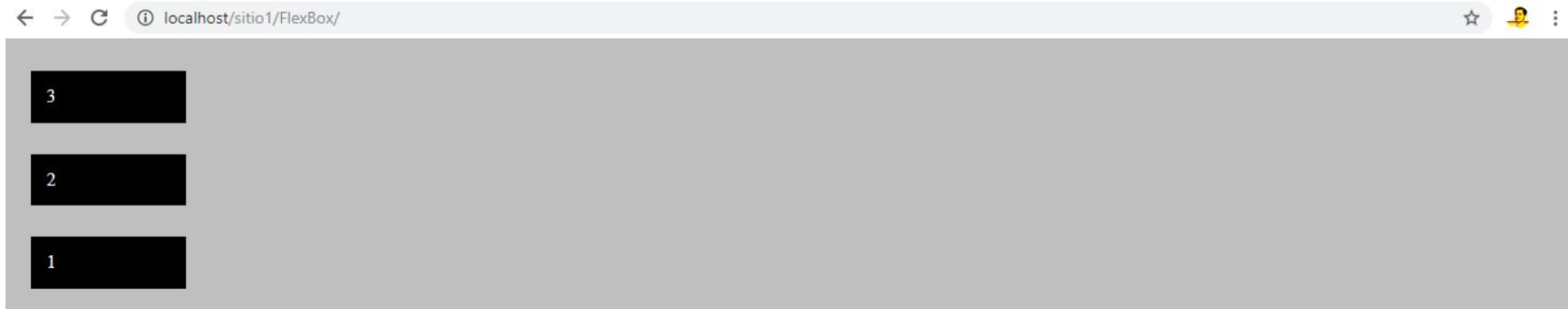
Alinea los elementos en fila, de derecha a izquierda.



Contenedor: flex-direction

flex-direction: column-reverse;

Alinea los elementos en columna, de abajo a arriba.



Contenedor: flex-wrap

Definición

Permite indicar cómo se van a distribuir los elementos dentro del contenedor y que tamaño tendrán.

Contenedor: flex-wrap

Sintaxis

```
.contenedor {  
    display: flex;  
    flex-wrap: nowrap | wrap | wrap-reverse;  
    width: 100%;  
    padding: 1%;  
    background-color: #CoCoCo;  
}
```

Contenedor: flex-wrap

flex-wrap: nowrap;

Valor por defecto. Los elementos se mantienen siempre dentro de la misma fila/columna (eje principal), aunque aumente su tamaño, hasta cierto límite.

Si en index.html, dentro del div que tiene la clase contenedor, añadimos nuevos elementos, se mantendrán dentro del eje principal, hasta cierto límite, aunque haya más de 10 y por tanto superen el 100%, al tener estos un ancho del 10%.

Contenedor: flex-wrap

flex-wrap: nowrap;

index.html

```
<div class="contenedor">  
  <div class="elementos">1</div>  
  <div class="elementos">2</div>  
  ...  
  <div class="elementos">11</div>  
  <div class="elementos">12</div>  
</div>
```

Contenedor: flex-wrap

flex-wrap: nowrap;

estilo.css

```
.contenedor{  
    display: flex;  
    flex-wrap: nowrap;  
    width: 100%;  
    padding: 1%;  
    background-color: #CoCoCo;  
}
```

Contenedor: flex-wrap

flex-wrap: nowrap;

Resultado



Contenedor: flex-wrap

flex-wrap: wrap;

Los elementos se van acomodando dentro de la misma fila/columna (eje principal), mientras quepan y cuando se cope todo el espacio pasarán a ocupar la siguiente fila/columna.



Contenedor: flex-wrap

flex-wrap: wrap-reverse;

Los elementos se van acomodando dentro de la misma fila/columna (eje principal), mientras quepan y cuando se cope todo el espacio pasarán a ocupar la siguiente fila/columna, pero de abajo a arriba.



Contenedor: flex-flow

Definición

Combina las cualidades de flex-direction y flex-wrap.

Contenedor: flex-flow

Sintaxis

```
.contenedor {  
    display: flex;  
    flex-flow: row | row-reverse | column | column-reverse | nowrap | wrap |  
    wrap-reverse | row nowrap | column wrap | column-reverse wrap-reverse;  
    ...  
}
```

<https://developer.mozilla.org/es/docs/Web/CSS/flex-flow>

Contenedor: flex-flow

flex-flow: row;

Equivale a flex-direction: row;

Valor por defecto. Alinea los elementos en fila, de izquierda a derecha.

flex-flow: row-reverse;

Equivale a flex-direction: row-reverse;

Alinea los elementos en fila, de derecha a izquierda.

flex-flow: column;

Equivale a flex-direction: column;

Alinea los elementos en columna, de arriba a abajo.

Contenedor: flex-flow

flex-flow: column-reverse;

Equivale a flex-direction: column-reverse;

Alinea los elementos en columna, de abajo a arriba.

flex-flow: nowrap;

Valor por defecto. Equivale a flex-wrap: nowrap. Los elementos se mantienen siempre dentro de la misma fila/columna (eje principal), aunque aumente su tamaño.

flex-flow: wrap;

Equivale a flex-wrap: wrap. Los elementos se van acomodando dentro del misma eje principal, mientras quepan y cuando se cope pasan al siguiente eje (fila/columna).

Contenedor: flex-flow

flex-flow: wrap-reverse;

Equivale a flex-wrap: wrap-reverse. Los elementos se van acomodando dentro del mismo eje principal, mientras quepan y cuando se cope pasan al siguiente eje (fila/columna), ordenado de forma inversa.

Son combinables entre sí:

flex-flow: row nowrap; = flex-direction: row + flex-wrap: nowrap

flex-flow: column wrap = flex-direction: column + flex-wrap: wrap

flex-flow: column-reverse wrap-reverse = flex-direction: column-reverse +
flex-wrap: wrap-reverse

Contenedor: justify-content

Definición

Permite indicar cómo se van a posicionar los elementos entre sí, **trabajando con el eje horizontal (izquierda-derecha)**.

Nota

justify-content trabaja con el eje horizontal (izquierda-derecha).

align-items trabaja con el eje vertical (arriba-abajo)

Contenedor: justify-content

Sintaxis

```
.contenedor {  
    display: flex;  
    justify-content: flex-start | flex-end | center | space-between |  
space-around;  
    width: 100%;  
    padding: 1%;  
    background-color: #CoCoCo;  
}
```

Contenedor: justify-content

justify-content: flex-start;

Valor por defecto. Sitúa los elementos desde el inicio principal (main-start).



Contenedor: justify-content

justify-content: flex-end;

Sitúa los elementos desde el final principal (main-end).



Contenedor: justify-content

justify-content: center;

Sitúa los elementos desde el centro.



Contenedor: justify-content

justify-content: space-between;

Sitúa los elementos separándolos entre ellos con el mismo espacio, sirviéndose de todo el espacio disponible.



Contenedor: justify-content

justify-content: space-around;

Sitúa los elementos separándolos a ambos lados con el mismo espacio cada uno, sirviéndose de todo el espacio disponible.



Contenedor: align-items

Definición

Permite indicar cómo se van a posicionar los elementos entre sí, **trabajando con el eje vertical (arriba-abajo)**.

Nota

justify-content trabaja con el eje horizontal (izquierda-derecha).

align-items trabaja con el eje vertical (arriba-abajo)

Contenedor: align-items

Advertencia

Para apreciar su comportamiento debemos asignar una altura al div con la clase contenedor, por ejemplo del 25% de la pantalla (`height: 25vh;`).

Contenedor: align-items

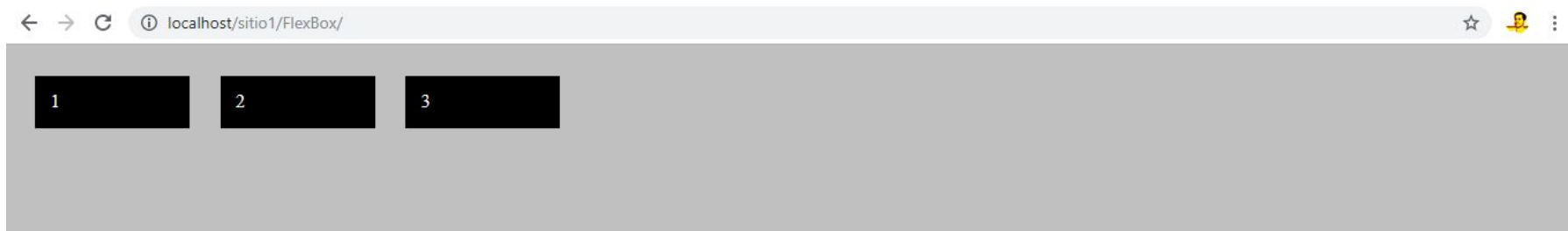
Sintaxis

```
.contenedor {  
    display: flex;  
    align-items: flex-start | flex-end | center | stretch | baseline;  
    width: 100%;  
    height: 25vh;  
    padding: 1%;  
    background-color: #CoCoCo;  
}
```

Contenedor: align-items

align-items: flex-start;

Valor por defecto. Sitúa los elementos desde la parte superior (main-start).



Contenedor: align-items

align-items: flex-end;

Sitúa los elementos desde la parte inferior (main-end).



Contenedor: align-items

align-items: center;

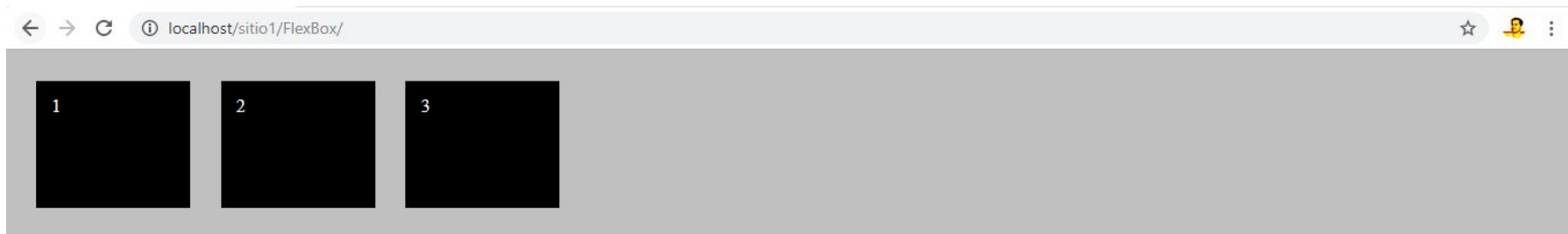
Sitúa los elementos en el centro del eje horizontal.



Contenedor: align-items

align-items: stretch;

Extiende los elementos para ocupar toda la altura del contenedor.



Contenedor: align-items

align-items: baseline;

Sitúa los elementos desde la parte superior, como flex-start, pero tomando como referencia la base del contenido del primer elemento, para alinear el resto horizontalmente a partir de este.



Contenedor: align-content

Definición

Similar a align-items, permite indicar cómo se van a posicionar los elementos entre sí, trabajando con el eje vertical (arriba-abajo).

align-content: alinea el contenido.

align-items: alinea los elementos.

Contenedor: align-content

Advertencia

Para apreciar su comportamiento debemos asignar una altura al div con la clase contenedor, por ejemplo del 25% de la pantalla (`height: 25vh;`).

El div contenedor también debe contar con la regla:

```
flex-wrap: wrap;
```

Esta regla indica que los elementos se van acomodando dentro de la misma fila/columna (eje principal), mientras quepan y cuando se cope todo el espacio pasarán a ocupar la siguiente fila/columna.

Contenedor: align-content

```
.contenedor {  
    display: flex;  
    flex-wrap: wrap;  
    align-content: flex-start | flex-end | center | space-between | space-around  
| stretch;  
    width: 100%;  
    height: 25vh;  
    padding: 1%;  
    background-color: #CoCoCo;  
}
```

Contenedor: align-content

Sintaxis

```
.elementos{  
    width: 10%;  
    margin: 1%;  
    padding: 1%;  
    background-color: #000000;  
    color: #ffffff;  
}
```

Contenedor: align-content

align-content: flex-start;

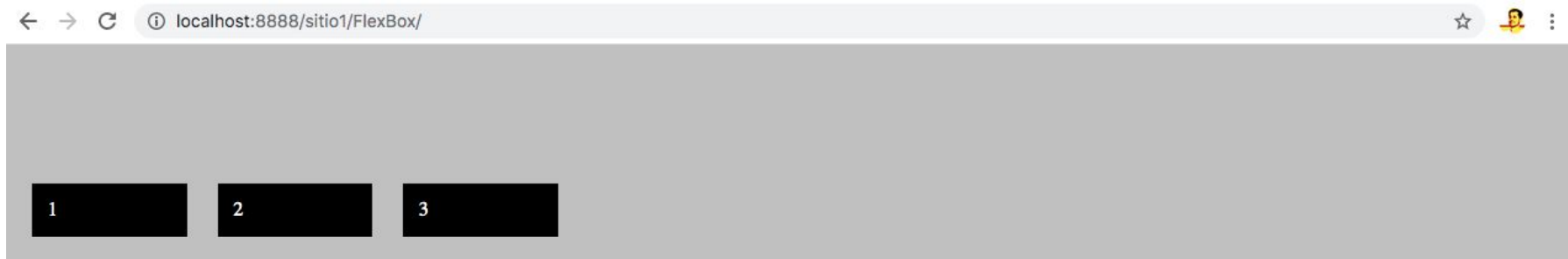
Valor por defecto. Sitúa los elementos desde la parte superior (main-start).



Contenedor: align-content

align-content: flex-end;

Sitúa los elementos desde la parte inferior (main-end).



Contenedor: align-content

align-content: center;

Sitúa los elementos desde la parte central (main-middle).



Contenedor: align-content

align-content: space-between;

Sitúa los elementos distribuidos con el mismo espacio entre si.



Contenedor: align-content

align-content: space-around;

Sitúa los elementos distribuidos en el espacio uniformemente.



Contenedor: align-content

align-content: stretch;

Extiende el contenido, haciendo que los elementos ocupen toda la altura/anchura del contenedor.



Resumen de lo visto y lo que queda por ver

Acabamos de ver las 6 propiedades de para flex container (contenedor)

1. flex-direction
2. flex-wrap
3. flex-flow
4. justify-content
5. align-items
6. align-content

Ahora vamos a ver las 5 propiedades para flex items (elementos)

Propiedades de flex items (elementos)

1. order
2. flex-grow
3. flex-shrink
4. flex-basis
5. flex

Propiedades de flex items (elementos)

Antes de explicar las propiedades de flex items (elementos), establecemos los valores de CSS de nuestro ejemplo como teníamos al principio.

Ejemplo flexbox - estilo.css

```
.contenedor{  
    display: flex;  
    width: 100%;  
    padding: 1%;  
    background-color: #CoCoCo;  
}
```

Ejemplo flexbox - estilo.css

```
.elementos {  
    width: 10%;  
    margin: 1%;  
    padding: 1%;  
    background-color: #000000;  
    color: #ffffff;  
}
```

Ejemplo flexbox - estilo.css

Nota

En estilo.css vamos a utilizar la pseudo clase **nth-child** para aludir mediante posición a un determinado elemento dentro de una misma clase.

Por ejemplo, si queremos aludir al **segundo** elemento de la clase “.elementos”, indicaremos:

```
.elementos:nth-child(2) {  
    ...  
}
```

Elementos: order

Definición

Permite ordenar los elementos dentro del contenedor. Pueden existir posiciones de orden repetidas. Por defecto todos elementos tienen el valor 0.

Los que tienen 0 irán en primer lugar por orden de establecimiento en HTML.

Los que tienen 1 irán a continuación de los que tienen 0.

Los que tienen 2 irán a continuación de los que tienen 1.

Y así sucesivamente.

Elementos: order

Sintaxis

```
<div class="contenedor">  
  <div class="elementos">1</div>  
  <div class="elementos">2</div>  
  <div class="elementos">3</div>  
  <div class="elementos">4</div>  
  <div class="elementos">5</div>  
  <div class="elementos">6</div>  
</div>
```

Elementos: order

```
.elementos:nth-child(1){ order:1; }
```

```
.elementos:nth-child(2){ order:2; }
```

```
.elementos:nth-child(3){ order:3; }
```

```
.elementos:nth-child(4){ order:1; }
```

Orden de salida: 5 - 6 - 1 - 4 - 2 - 3

← → ↻ ⓘ localhost/sitio1/FlexBox/



5

6

1

4

2

3

Elementos: flex-grow

Definición

Permite indicar el valor de crecimiento de los elementos dentro del contenedor. Por defecto los elementos tienen el valor 0, lo que suponen que mantendrán su tamaño, independientemente de que quede espacio libre en el contenedor.

Si asignamos 1 a flex-grow, el espacio disponible lo ocupará ese elemento. Y si varios elemento comparte 1 se repartirán a partes iguales el espacio disponible.

Si a dos elementos le damos el valor 1 y a otro el valor 2, los de valor 1 se repartirán una parte y el de valor 2 se quedará con dos partes. Y así sucesivamente con 3, 4, 5...

Elementos: flex-grow

Sintaxis

```
<div class="contenedor">  
  <div class="elementos">1</div>  
  <div class="elementos">2</div>  
  <div class="elementos">3</div>  
</div>
```


Elementos: flex-grow

```
.elementos:nth-child(1){ flex-grow: 1; }
```

```
.elementos:nth-child(2){ flex-grow: 1; }
```

```
.elementos:nth-child(3){ flex-grow: 2; }
```

← → ↻ ⓘ localhost/sitio1/FlexBox/



1

2

3

Elementos: flex-shrink

Definición

Permite indicar que el elemento se reducirá cuando cambie a menos el tamaño de la pantalla.

flex-shrink tiene como valor por defecto 1.

Si flex-shrink valor 0 el elemento no se reducirá aunque cambie a menos el tamaño de la pantalla.

Elementos: flex-shrink

Nota

Para qué flex-shrink funcione los elementos para width deberán tener un valor absoluto (px) en vez de relativo (%).

Elementos: flex-grow

```
.elementos {  
    width: 400px;  
    ...  
}  
.elementos:nth-child(1){ flex-shrink: 1; }  
.elementos:nth-child(2){ flex-shrink: 1; }  
.elementos:nth-child(3){ flex-shrink: 0; } /* No va a cambiar de tamaño */
```



localhost/sitio1/FlexBox/



1

2

3

Elementos: flex-basis

Definición

Permite indicar el tamaño que tendrá el elemento. Si el valor es auto el elemento tomará su tamaño de acuerdo con el contenido que el mismo albergue.

Elementos: flex-basis

```
.elementos {  
    width: 10%;  
    ...  
}  
.elementos:nth-child(1){ flex-basis: 20%; }  
.elementos:nth-child(2){ flex-basis: auto; }  
.elementos:nth-child(3){ flex-basis: auto; }
```

← → ↻ ⓘ localhost/sitio1/FlexBox/



1

2

3

Elementos: flex

Definición

Es la abreviatura de flex-grow, flex-shrink y flex-basis conjuntamente (en este estricto orden).

```
flex : 0 1 20%;
```

/* Equivale a: */

```
flex-grow : 0;
```

```
flex-shrink : 1;
```

```
flex-basis : 20%;
```