

MODELO CAJA FLEXIBLE HTML5 y CSS3

MODELO CAJA FLEXIBLE

HTML5 y CSS3

CUAL ES SU FUNCION?

1. El principal objetivo de un modelo de caja es el proveer de un mecanismo con el que podamos dividir el espacio disponible en la ventana en varios cuadros o cajas y así crear las filas y columnas que forman parte de un diseño web estándar.
2. Con el contenedor flex, el tamaño de las cajas o elementos irán incrementándose o reduciéndose , siempre dependiendo del espacio disponible. Por ello, para saber la cantidad de espacio a distribuir necesitábamos saber el tamaño exacto del contenedor.
3. Un contenedor flexible es un elemento que permite a su contenido ser flexible. Los elementos flexibles tienen con contar con un elemento padre común para poder ser organizados.

MODELO CAJA FLEXIBLE

HTML5 y CSS3

Ejemplo:

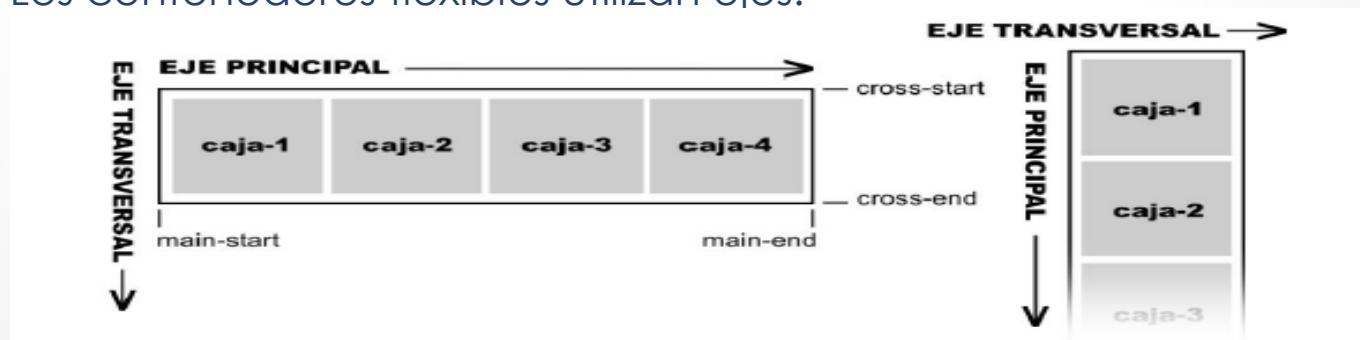
```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="utf-8">
    <title>Modelo de Caja Flexible</title>
    <link rel="stylesheet" href="prueba.css">
</head>
<body>
<section id="contenedor">
    <div id="caja-1">Caja 1</div>
    <div id="caja-2">Caja 2</div>
    <div id="caja-3">Caja 3</div>
    <div id="caja-4">Caja 4</div>
</section>
</body>
</html>
```

MODELO CAJA FLEXIBLE

HTML5 y CSS3

Propiedad display:

1. Tiene que ser declarado el atributo o propiedad en la hoja de estilo como **flex**. Como no esta estandarizado este valor en todos los navegadores, utilizaremos si es necesario los prefijos **-webkit-flex**, **-moz-flex**, **-ms-flex**, **flex**.
2. Los contenedores flexibles utilizan ejes:



MODELO CAJA FLEXIBLE

HTML5 y CSS3

Propiedad flex:

1. Las propiedades flex tienen que ser declaradas dentro de las hojas de estilo. Las propiedades de flex son las siguientes:
 - a. **Flex-grow:** declara la proporción de la caja hijo.
 - b. **Flex-shrink:** cuanto un hijo se va a encoger dentro del contenedor padre.
 - c. **Flex-basis:** tamaño inicial, el espacio libre a ser distribuido
2. El tamaño se calcula en proporción de la caja, por ejemplo 4 cajas de **flex: 1** con tamaño del contenedor 200px. El calculo seria el siguiente

200 x 1 / 4

MODELO CAJA FLEXIBLE

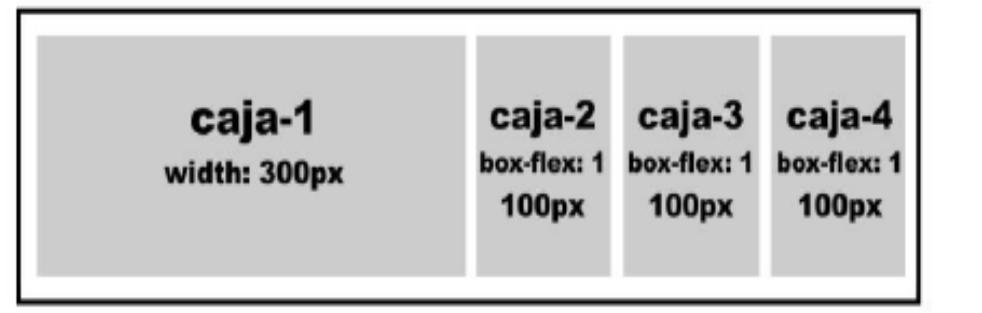
HTML5 y CSS3

Propiedad flex:

- Si alguno de los tamaños tuviera tamaño fijo, calcularíamos el tamaño restante dejado por la caja de tamaño fijo para ser distribuido al resto.

```
#contenedor{  
    display: flex;  
    width: 600px;  
}  
  
#caja-1{  
    width: 300px;  
}  
#caja-2{  
    flex: 1;  
}  
#caja-3{  
    flex: 1;  
}  
#caja-4{  
    flex: 1;  
}
```

contenedor principal - 600px



MODELO CAJA FLEXIBLE

HTML5 y CSS3

Propiedad flex:

1. Primer valor (**flex-grow**)
2. Segundo valor (**flex-shrink**)
3. Tercer valor (**flex-basis**)

El segundo valor y en el ejemplo de caja1 y en el resto de cajas 5, por eso el tamaño de caja en la primera caja es mayor.

```
#contenedor{  
    display: flex;  
}  
#caja-1{  
    flex: 1 1 200px;  
}  
#caja-2{  
    flex: 1 5 100px;  
}  
#caja-3{  
    flex: 1 5 100px;  
}  
#caja-4{  
    flex: 1 5 100px;  
}
```

MODELO CAJA FLEXIBLE

HTML5 y CSS3

Propiedad flex-direction:

1. **Row**: orientación de main-start a main-end.
2. **Row-reverse**: orientación inversa al anterior.
3. **Column**: orientación de columna.
4. **Column-reverse**: orientación inversa al anterior.
5. Tercer valor (**flex-basis**)

```
#contenedor{  
    display: flex;  
    flex-direction: row;  
}
```

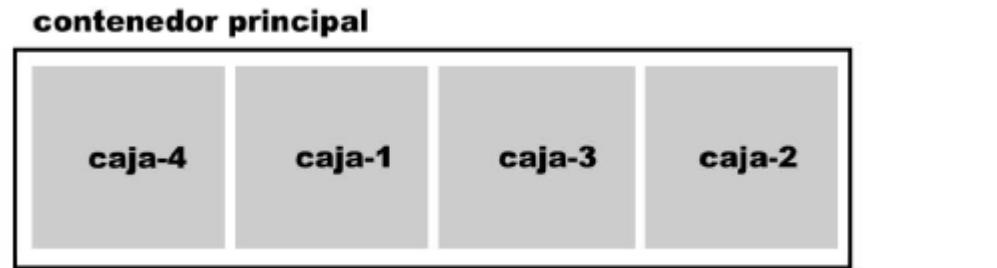
MODELO CAJA FLEXIBLE

HTML5 y CSS3

Propiedad flex-direction:

1. También podemos asignar la orden de aparición de las cajas. Siendo el 0 el orden asignado por efecto, el cual no se asigna.

```
#caja-1{
    flex: 1;
    order: 2;
}
#caja-2{
    flex: 1;
    order: 4;
}
#caja-3{
    flex: 1;
    order: 3;
}
#caja-4{
    flex: 1;
    order: 1;
}
```



MODELO CAJA FLEXIBLE

HTML5 y CSS3

Propiedad **justify-content**:

1. Con esta propiedad podremos asignar la distribución de las cajas dentro del contenedor padre, cuando esta sea de una distribución de ancho superior a las cajas.
 - a. **Justify-content: flex-start;**
 - b. **Justify-content: flex-end;**
 - c. **Justify-content: center;**
 - d. **Justify-content: space-between;**
 - e. **Justify-content: space-around;**

MODELO CAJA FLEXIBLE

HTML5 y CSS3

Propiedad align-items:

1. Con esta propiedad distribuimos el espacio entre las cajas.
Funciona como la justify-content pero alinea las cajas en el eje transversal.
 - a. **align-items: center;**
 - b. **align-items: stretch;** (estirando las cajas)
 - c. **align-items: flex-start;**
 - d. **align-items: flex-end;**
 - e. **align-items: baseline;** (alinea las cajas por la línea de base)

MODELO CAJA FLEXIBLE

HTML5 y CSS3

Propiedad align-self:

1. Permite alinear las cajas independientemente del alineamiento declarado en el contenedor.

```
#contenedor{
    display: flex;
    align-items: flex-end;
    width: 600px;
    height: 200px;
}
#caja-1{
    flex: 1;
    height: 100px;
}
#caja-2{
    flex: 1;
    height: 100px;
    align-self: center;
}
#caja-3{
    flex: 1;
    height: 100px;
}
#caja-4{
    flex: 1;
    height: 100px;
}
```

MODELO CAJA FLEXIBLE

HTML5 y CSS3

Propiedad **flex-wrap**:

1. Con esta propiedad podemos hacer que las cajas se coloquen en una simple línea o en varias líneas.
 - a. **flex-wrap: nowrap;** (línea simple)
 - b. **flex-wrap: nowrap; wrap;** (líneas multiples, ordena siempre desde cross-start a cross-end)
 - c. **flex-wrap: nowrap; wrap-reverse;** (como la anterior pero en orden inverso)

MODELO CAJA FLEXIBLE

HTML5 y CSS3

Propiedad align-content:

1. Esta propiedad sirve para alinear las líneas múltiples y trabaja como align-items.
 - a. **align-content: flex-start;**
 - b. **align-content: flex-end;**
 - c. **align-content: center;**
 - d. **align-content: space-between;**
 - e. **align-content: space-around;**
 - f. **align-content: stretch;**