



FLEXBOX

“

Flexbox o “sistema de cajas flexibles” es un set de **propiedades** que nos entrega **CSS** para **construir** un sitio web utilizando una **estructura** de **filas** y **columnas**.

Nos ayudará a que los **elementos de bloque** compartan una misma línea.

”



FLEXBOX - COMPATIBILIDAD CON NAVEGADORES

<http://caniuse.com/#search=flexbox/>

CSS Flexible Box Layout Module - CR

Method of positioning elements in horizontal or vertical stacks. Support includes all properties prefixed with `flex`, as well as `display: flex`, `display: inline-flex`, `align-content`, `align-items`, `align-self`, `justify-content` and `order`.

Current aligned

Usage relative

Date relative

Filtered

All



Usage

% of all users














?

Global

97.18% + 1.54% = 98.72%

unprefixed:

97.01% + 1.04% = 98.05%

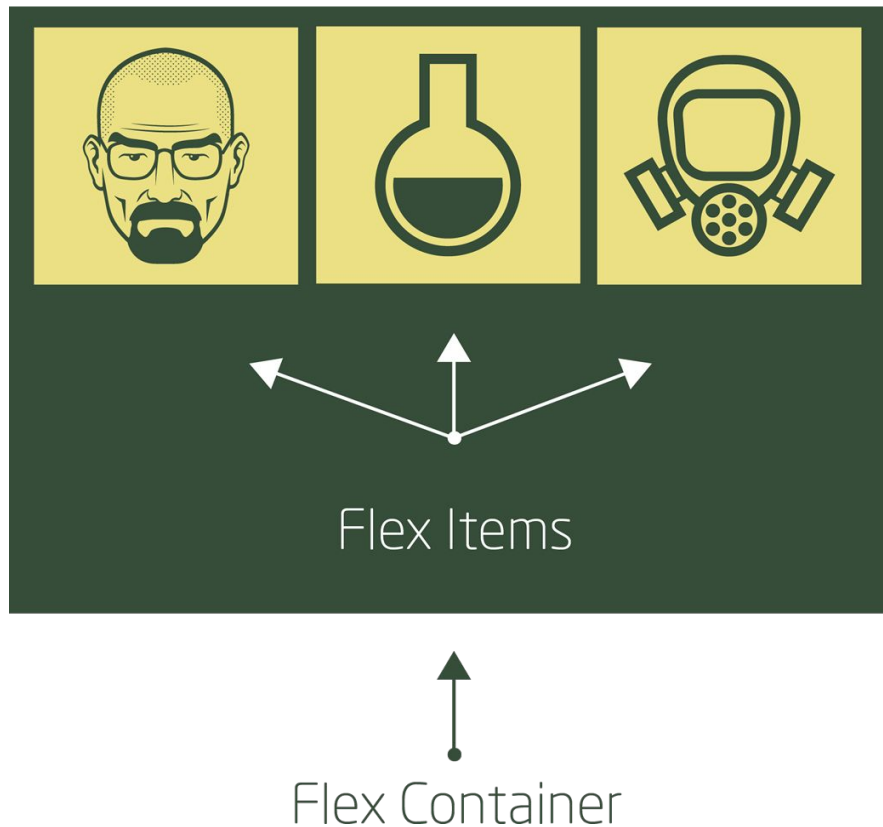
IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet
					10-11.5								
		1 2-21 	1 4-20 	1 3.1-6 	12.1	1 3.2-6.1 							
6-9		3 22-27 	21-28 	6.1-8 	15-16 	7-8.4 		1 2.1-4.3 	12				
2 4 10 	12-87	28-84	29-87	9-13.1	17-71	9-13.7		4.4-4.4.4	12.1				4-12.0
1 11	88	85	88	14	72	14.4	all	81	59	88	83	12.12	13.0
		86-87	89-91	TP									

ESTRUCTURA BÁSICA DE FLEXBOX

ESTRUCTURA BÁSICA DE FLEXBOX

Flexbox propone una estructura basada en la creación de un **contenedor padre** (*Flex-container*) que administra la ubicación de sus **elementos hijos** (*Flex-items*).

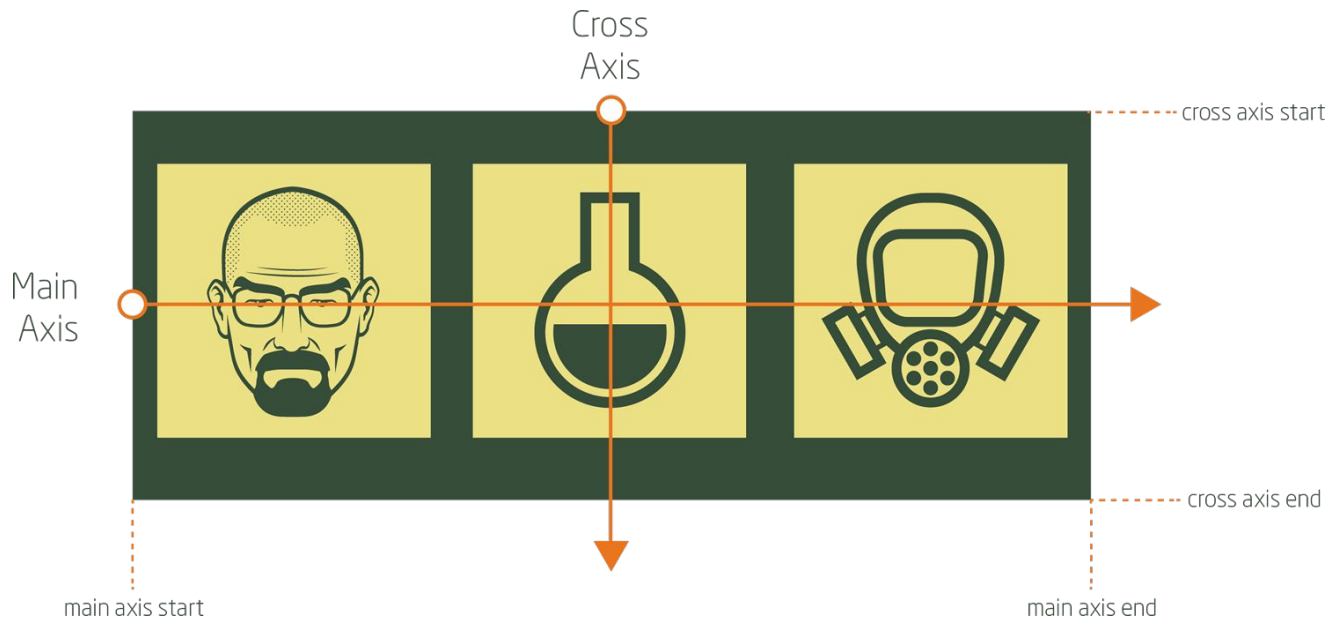
```
display: flex;
```



ESTRUCTURA BÁSICA DE FLEXBOX

Para organizar los elementos dentro del contenedor Flexbox trabaja con dos ejes:

- **Main axis.**
- **Cross axis.**

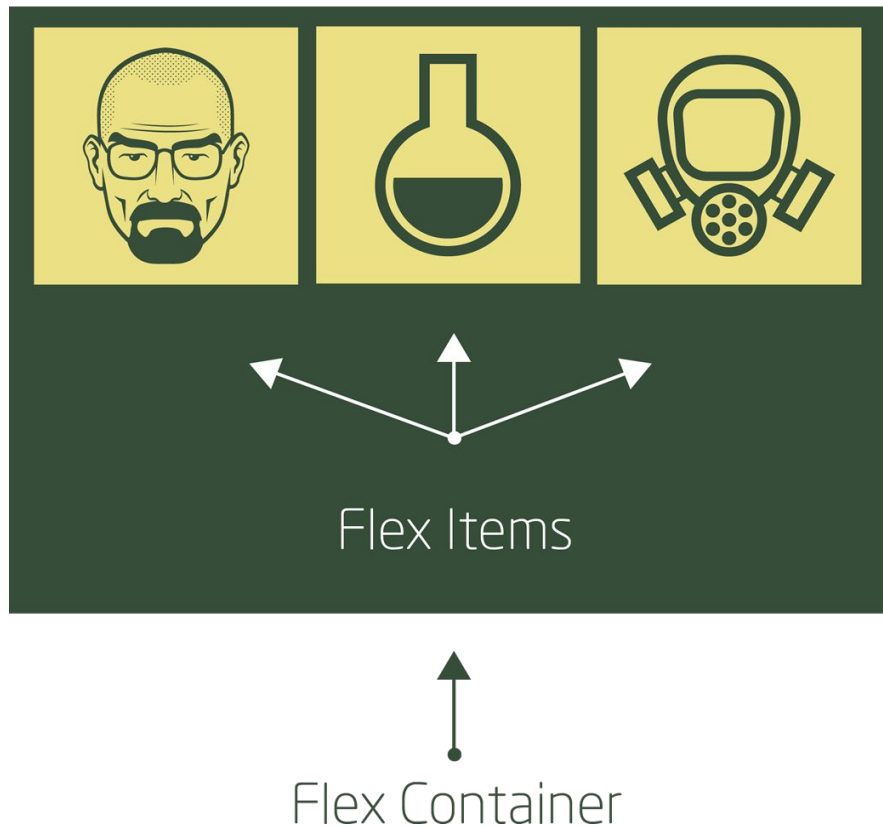


TRABAJANDO CON FLEXBOX

TRABAJANDO CON FLEXBOX

El primer paso será definir al contenedor padre como un **contenedor flexible**.

```
display: flex;
```



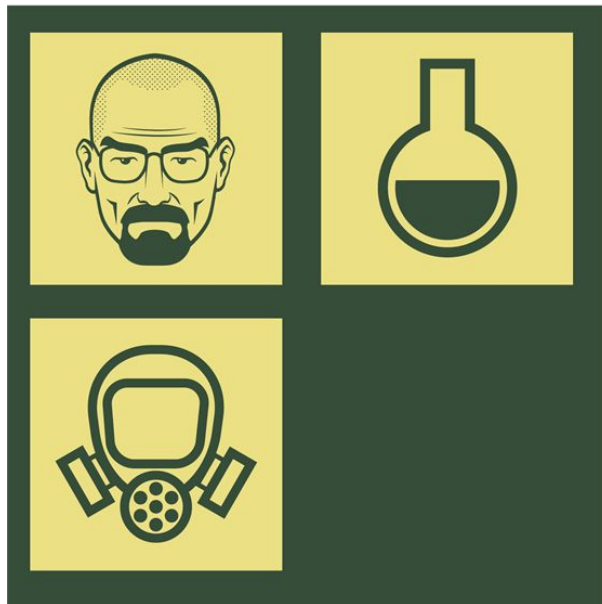
El segundo paso será hacer que el container respete los anchos de sus hijos.

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

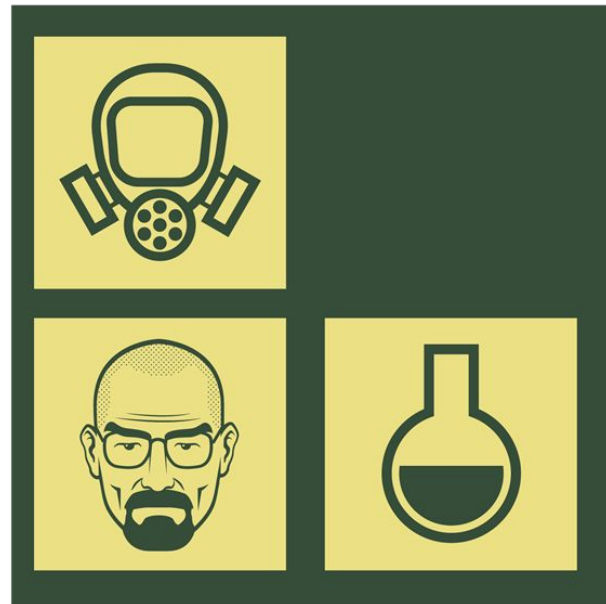
o

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

flex-wrap: wrap;



flex-wrap: wrap-reverse;



Si no declaramos **explícitamente** el valor de la propiedad ***flex-wrap*** por defecto será ***nowrap*** y los elementos dentro del contenedor padre se **ubicarán todos** en **una** misma **fila** haciéndose **cada vez más angostos** si el contenedor se achica o aparecen más elementos.

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



CREANDO FILAS O COLUMNAS

TRABAJANDO CON FLEXBOX

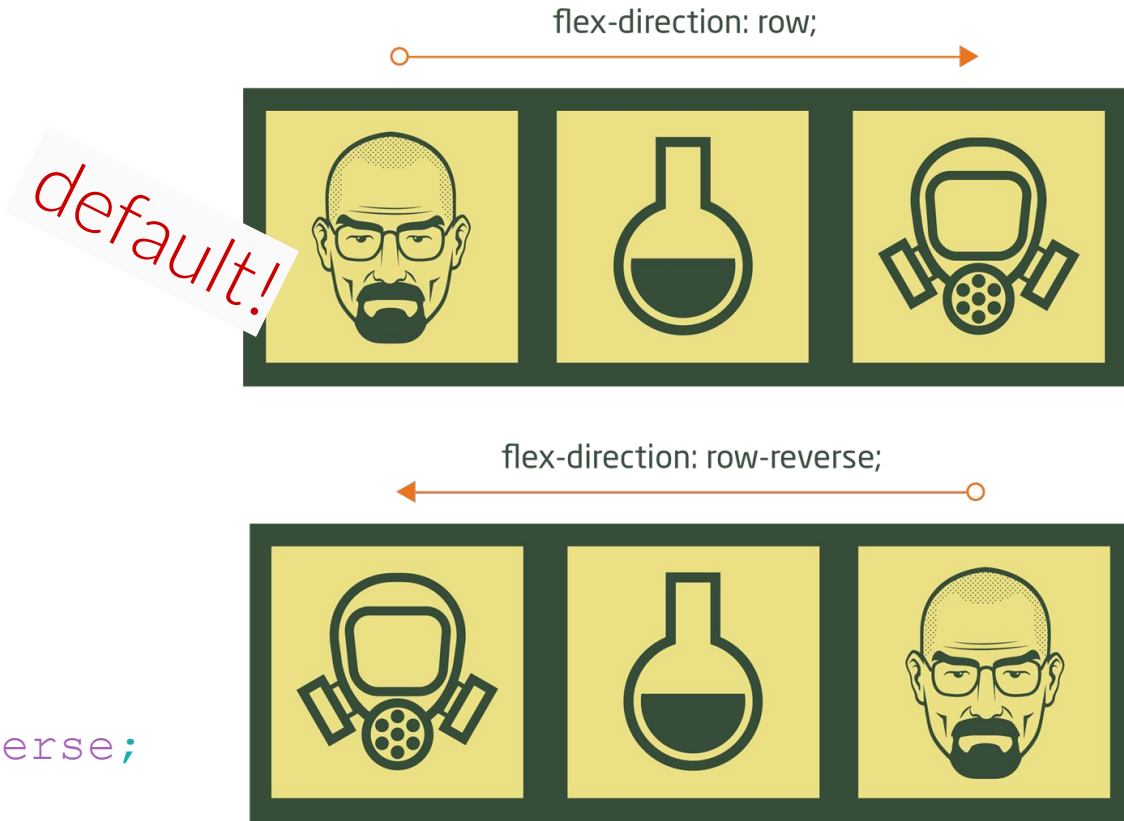
Para crear filas o columnas tenemos la propiedad ***flex-direction*** cuyo valor por defecto será ***row***.

La propiedad estará aplicada al contenedor padre aún cuando no la declaremos explícitamente.

```
flex-direction: row;
```

o

```
flex-direction: row-reverse;
```



TRABAJANDO CON FLEXBOX

Para crear columnas tendremos que declarar explícitamente la propiedad ***flex-direction*** con valor ***column*** ó ***column-reverse***.

```
flex-direction: column;
```

o

```
flex-direction: column-reverse;
```

flex-direction: column;



flex-direction: column-reverse;



ALINEANDO CONTENIDO EN EL MAIN AXIS

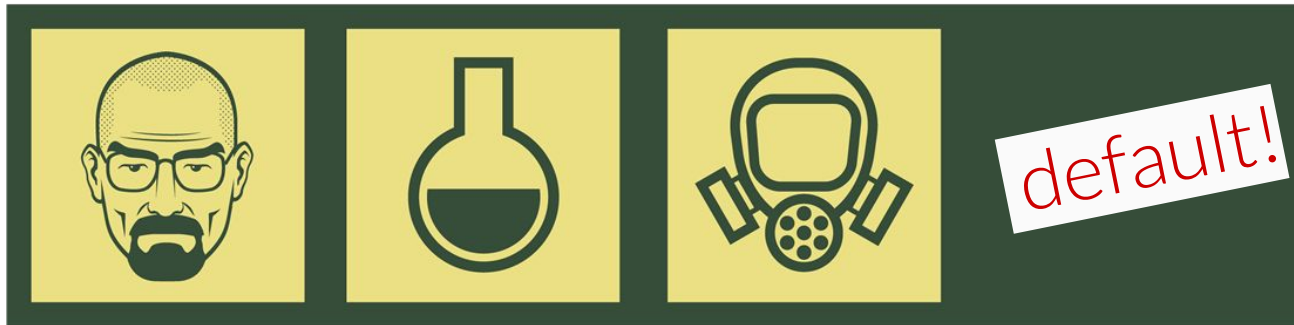
TRABAJANDO CON FLEXBOX - MAIN AXIS

Para alinear contenido sobre el **main axis** contamos con la propiedad ***justify-content***. Su valor default implícito será ***flex-start*** y puede tomar los valores:

```
justify-content:  
flex-start;
```

```
justify-content:  
flex-end;
```

`justify-content: flex-start;`



`justify-content: flex-end;`



TRABAJANDO CON FLEXBOX - MAIN AXIS

```
justify-content:  
center;
```

justify-content: center;



```
justify-content:  
space-between;
```

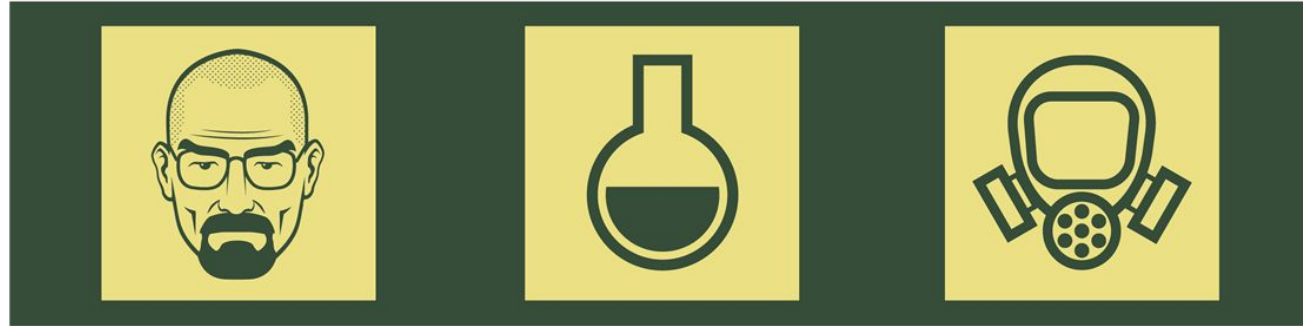
justify-content: space-between;



TRABAJANDO CON FLEXBOX - MAIN AXIS

`justify-content: space-around;`

`justify-content:
space-around;`



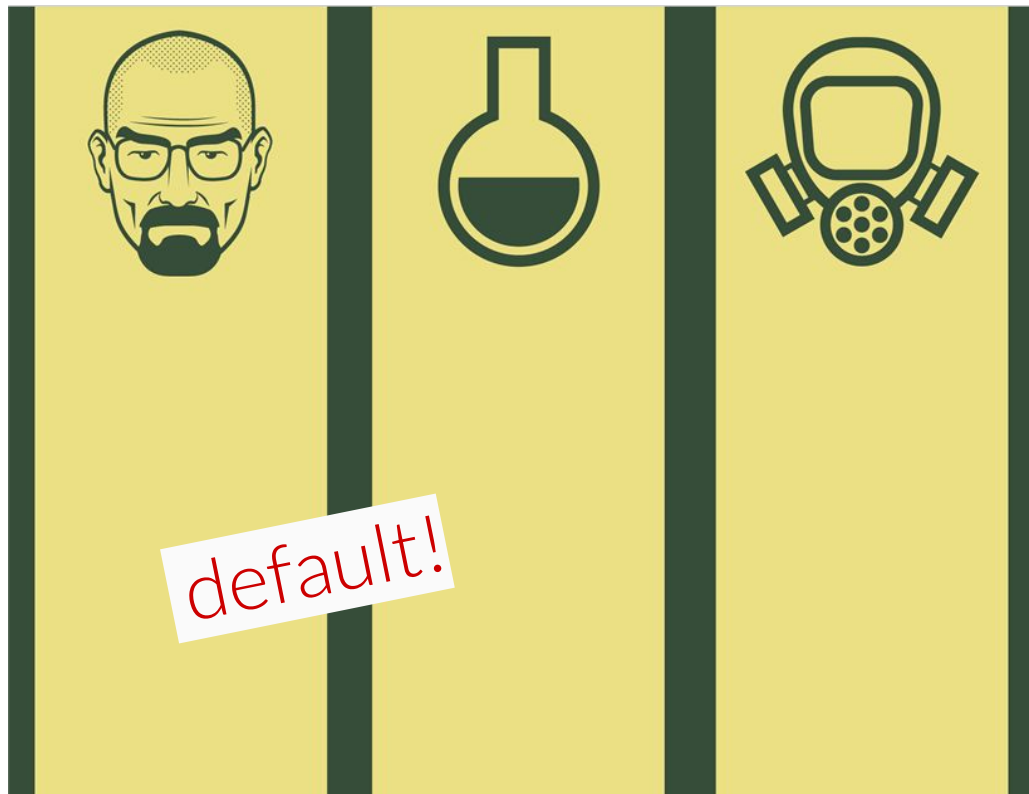
ALINEANDO CONTENIDO EN EL CROSS AXIS

TRABAJANDO CON FLEXBOX - CROSS AXIS

`align-items: stretch;`

Para alinear contenido sobre el **cross axis** contamos con la propiedad ***align-items*** cuyo valor por default es ***stretch*** pero que además puede tomar los valores:

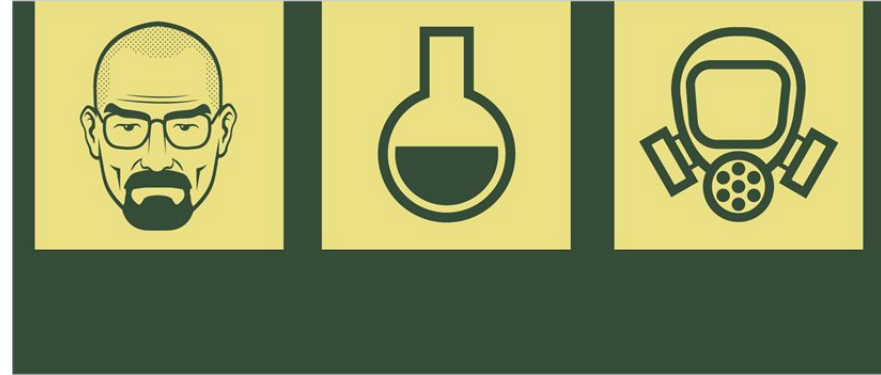
```
align-items: stretch;
```



TRABAJANDO CON FLEXBOX - CROSS AXIS

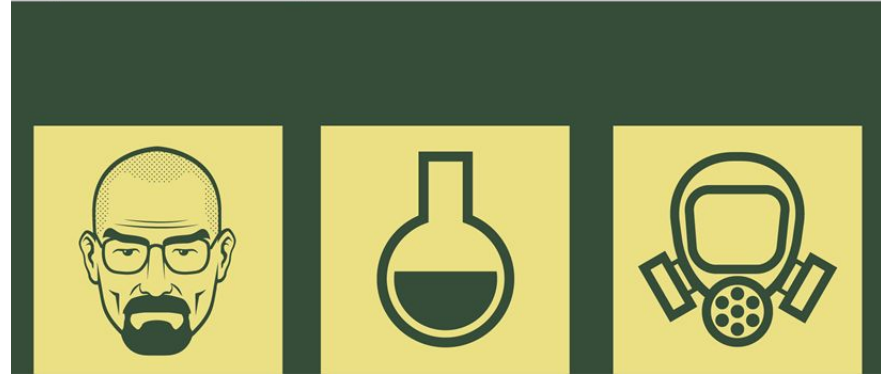
```
align-items: flex-start;
```

align-items: flex-start;



```
align-items: flex-end;
```

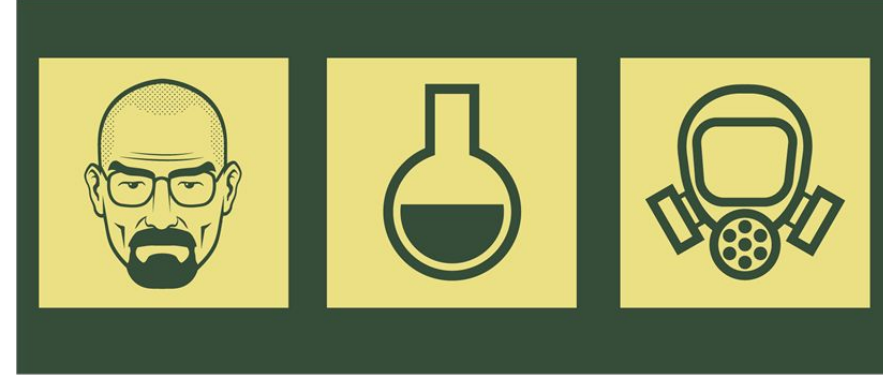
align-items: flex-end;



TRABAJANDO CON FLEXBOX - CROSS AXIS

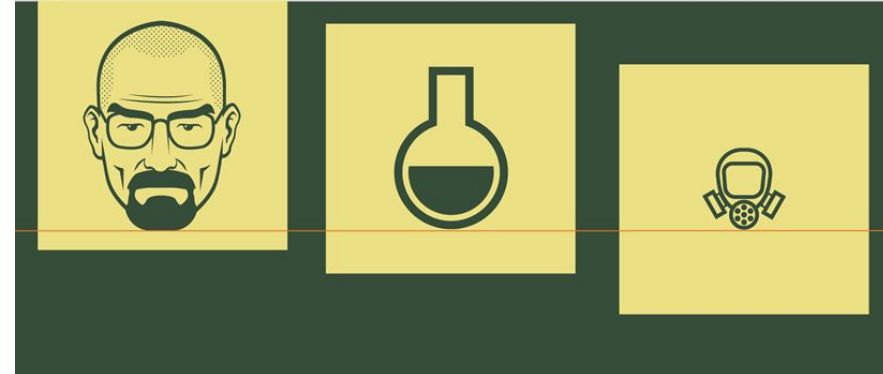
```
align-items: center;
```

align-items: center;



```
align-items: base-line;
```

align-items: base-line;



EXCELENTE

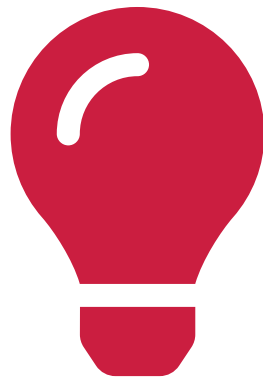
¿Y qué pasa con los flex-items?

Flexbox se la re banca.



Flexbox complementa sus funcionalidades con un **set de propiedades** que permiten **gestionar** a los ***flex-items*** directamente.

Recordemos que siempre será dentro de un ***flex-container***.



TRABAJANDO CON FLEXBOX - FLEX ITEMS: ORDER

La propiedad **order** permite modificar el orden natural de los elementos dentro del *flex-container*.

Para que la propiedad funcione correctamente todos los elementos deben estar identificados y tener asignado un orden de manera explícita.

ORDER



```
.walt{order: 3;}  
.pipe{order: 1;}  
.mask{order: 2;}
```



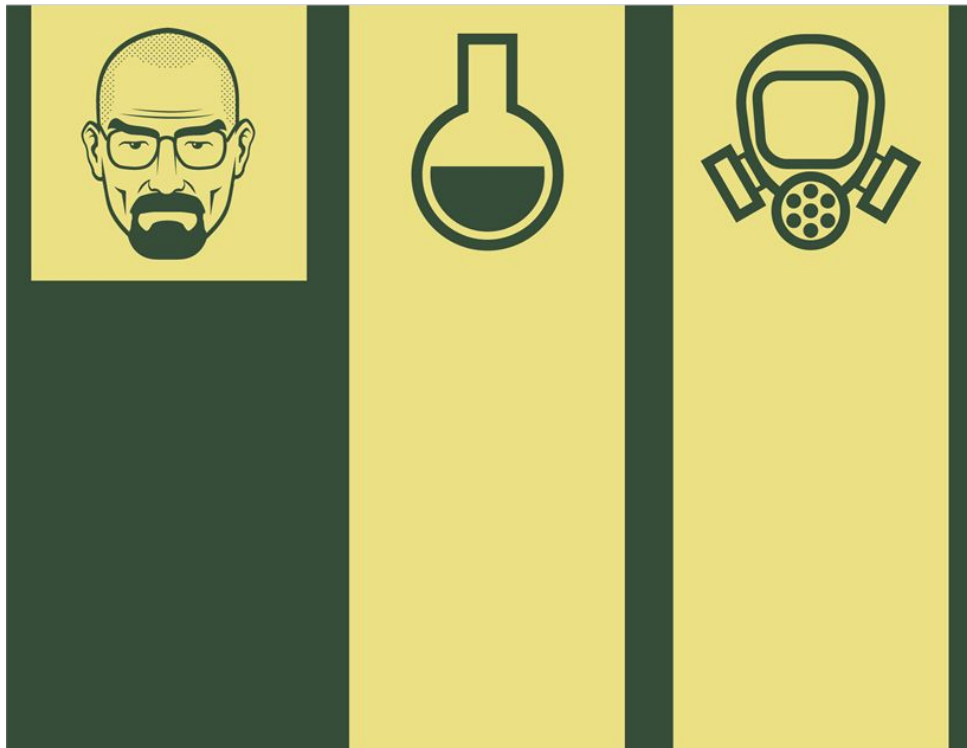
TRABAJANDO CON FLEXBOX - FLEX ITEMS: ALIGN-SELF

También podemos alinear los elementos individualmente dentro del ***flex-container*** utilizando la propiedad ***align-self*** con los valores:

```
.walt {  
  align-self: flex-start;  
}
```

ALIGN-SELF

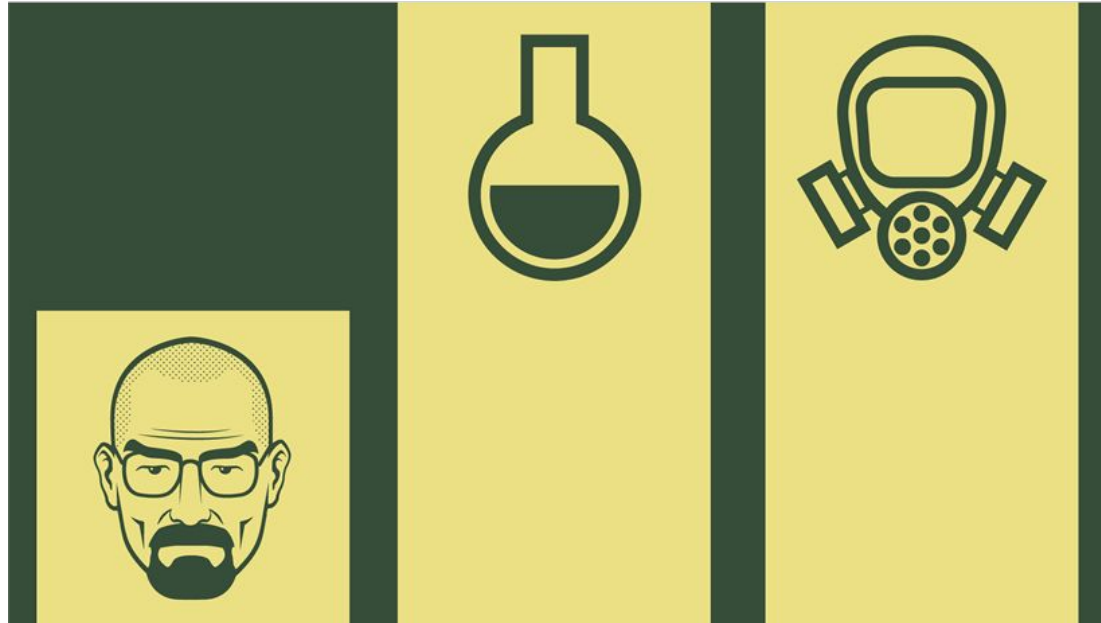
.walt{align-self: flex-start;}



TRABAJANDO CON FLEXBOX - FLEX ITEMS: ALIGN-SELF

```
.walt {  
  align-self: flex-end;  
}
```

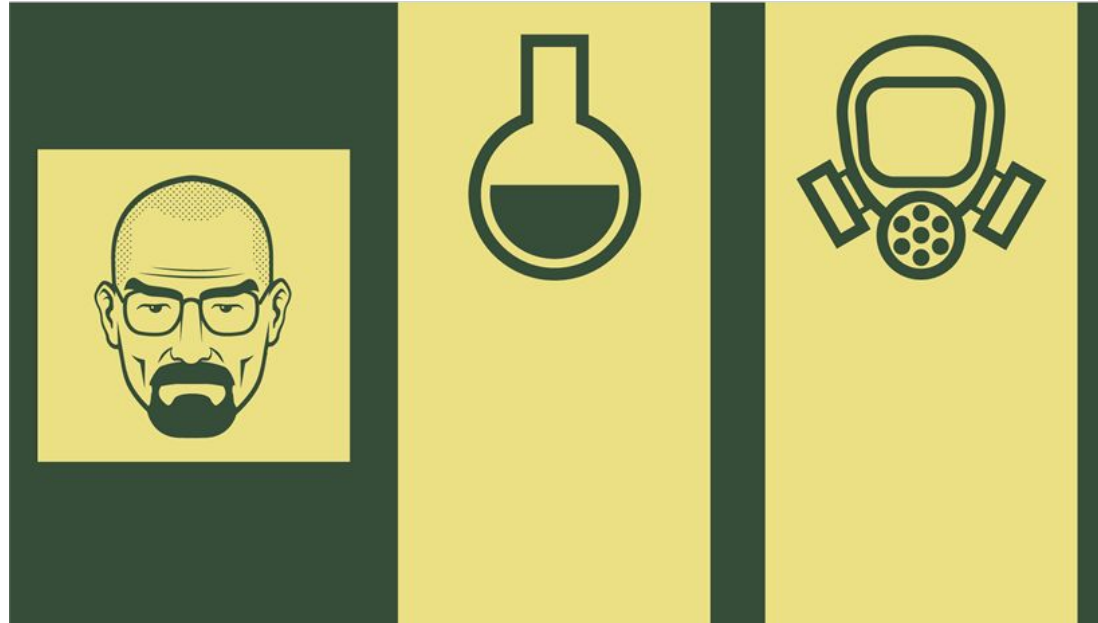
```
.walt{align-self: flex-end;}
```



TRABAJANDO CON FLEXBOX - FLEX ITEMS: ALIGN-SELF

```
.walt {  
  align-self: center;  
}
```

```
.walt{align-self: center;}
```



LINKS ÚTILES



DigitalHouse >
Coding School

/University
/Schools
/Corporate Training
/Academy

Juegos para poner en práctica Flexbox

[Flexbox Froggy](#)

[Flexbox Adventure](#)

Más detalles sobre cómo funciona flexbox.

[Flexbox explicado con gifs animados - Parte 1](#)

[Flexbox explicado con gifs animados - Parte 2 \(la venganza de Adrian :D\)](#)

PRACTIQUEMOS



DigitalHouse >
Coding School



¡NOS VEMOS LA PRÓXIMA!