CSS + BEM + RESPONSIVE

Quique Fdez. Guerra
efernandez@plainconcepts.com
@CKGrafico

TRANSFORM

- Permite aplicar transformaciones 2D o 3D a un elemento:
  - translate(*x,y*)
  - scale(*x,y*)
  - rotate(*angle*)
  - skew(*x-angle,y-angle*)
  - perspective(n)
  - …

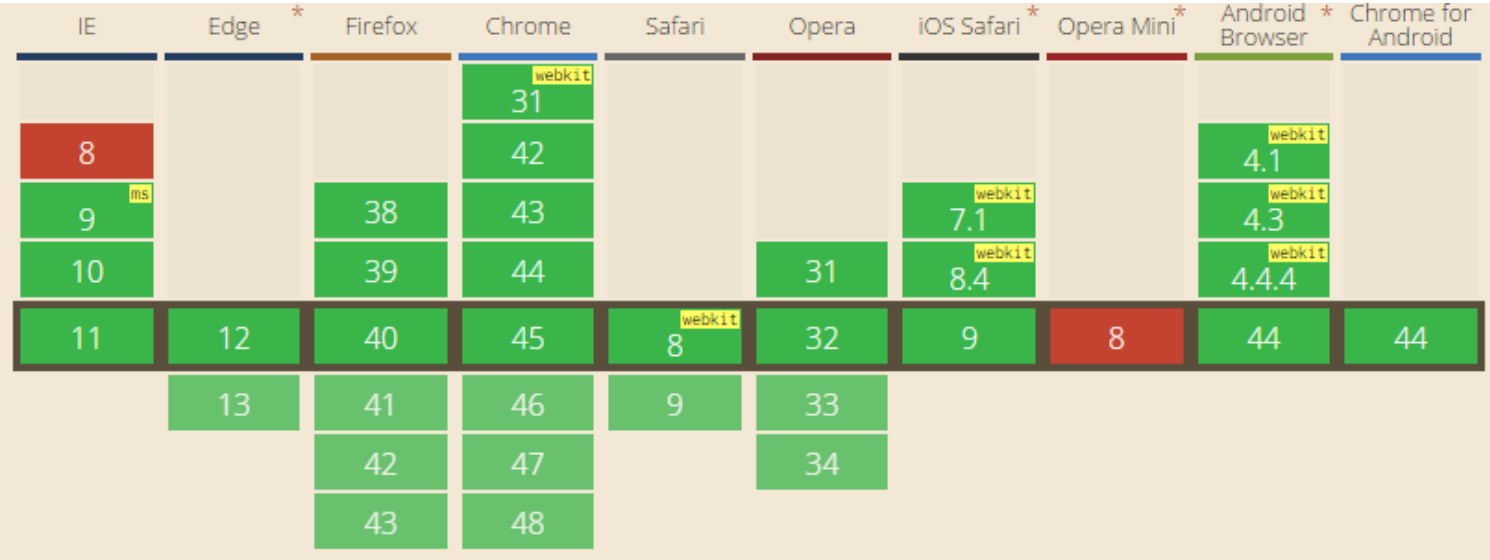
```
transform: scale(0.5, 0.5) rotate(180deg);
```

TRANSFORM-ORIGIN

- Permite establecer la posición de rotación de un elemento.
- Se usa como complemento de la propiedad *transform* con un valor de tipo rotación.

```
transform-origin: bottom left;
```

TRANSFORM


IE
Edge *
Firefox
Chrome
Safari
Opera
iOS Safari *
Opera Mini *
Android Browser *
Chrome for Android
8
9 ms
10
11
12
13
38
39
40
41
42
43
31 webkit
42
43
44
45
46
47
48
8 webkit
9
31
32
33
34
7.1 webkit
8.4 webkit
9
8
4.1 webkit
4.3 webkit
4.4.4 webkit
44
44

TRANSITION

- Permite añadir efectos cuando se transiciona a un estilo/estado, sobre unas determinadas propiedades de dicho estilo.

transition: [property] [duration] [timing-function] [delay]

```
.box {
    background: red;
    transition: background .5s linear; }
.box:hover {
    background: black; }
```

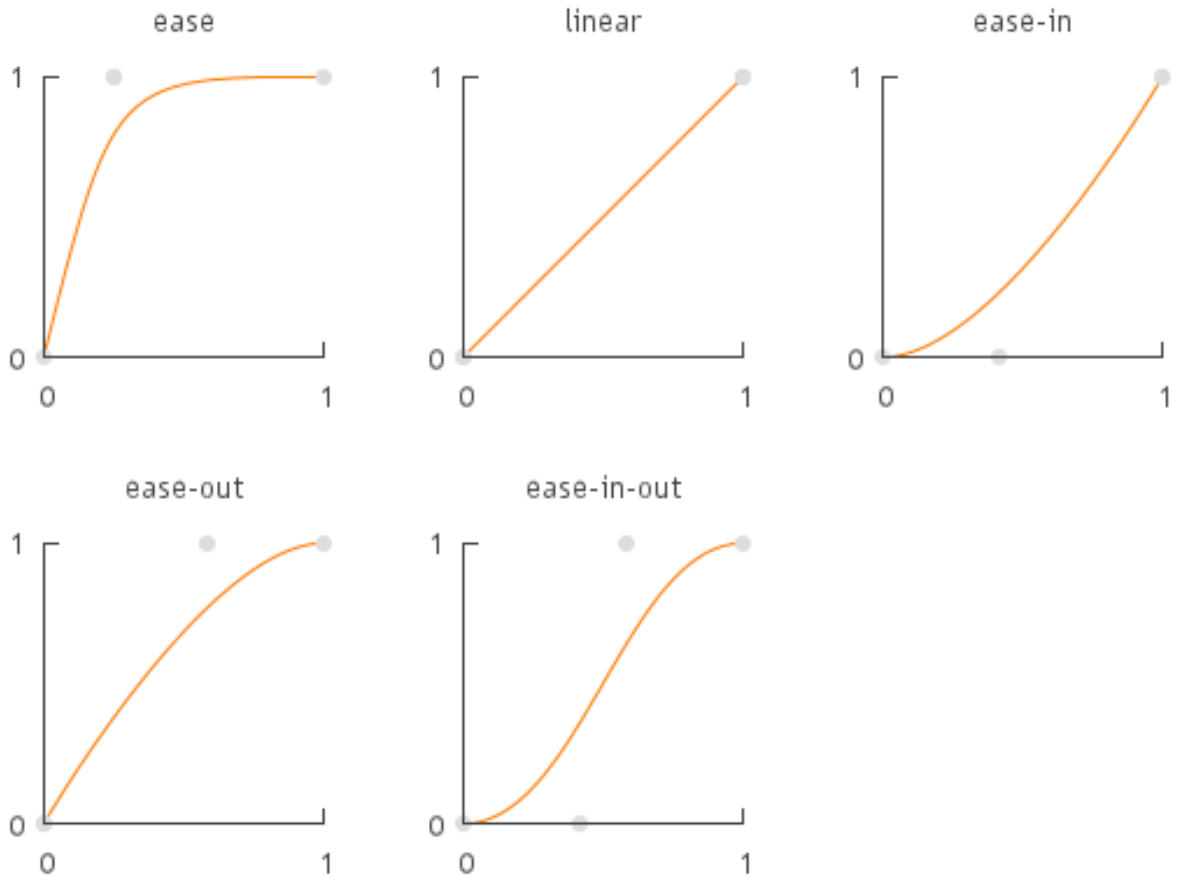
transition-property: all, <property>

transition-duration: <time>

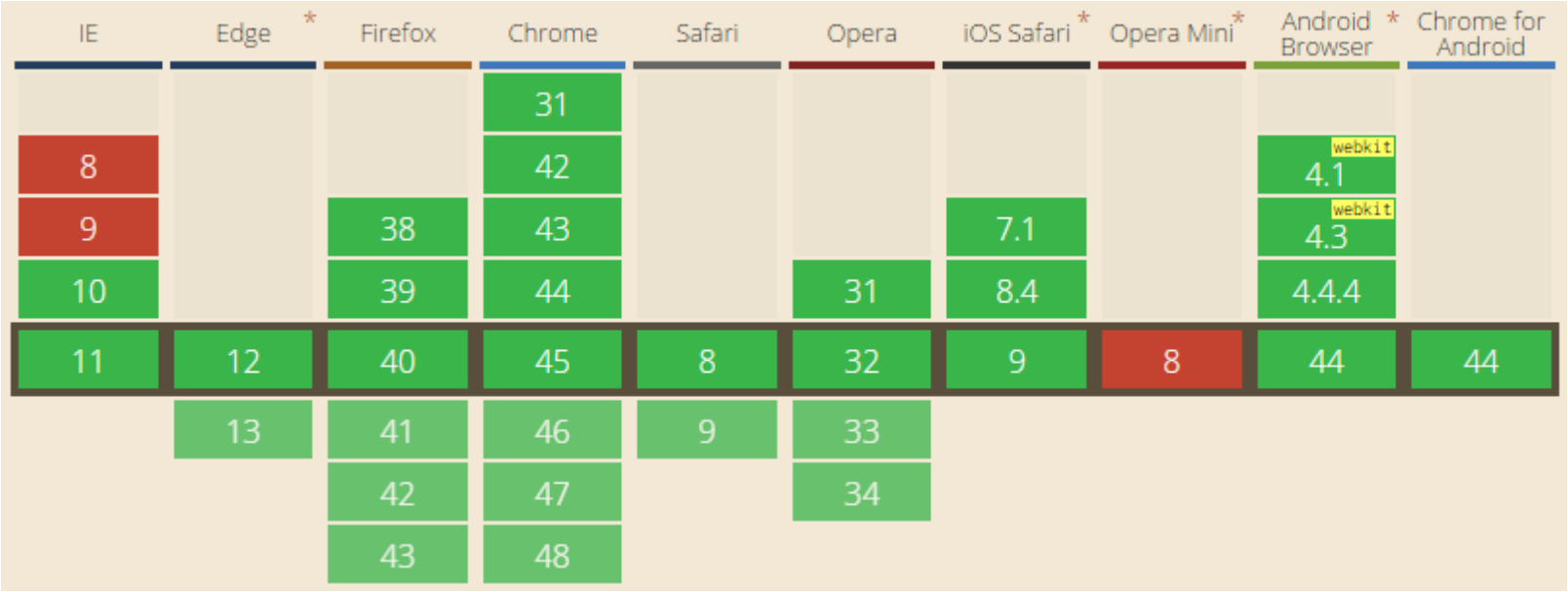
transition-timing-function: ease, linear, ease-in, ease-out, ease-in-out, cubic-bezier, steps.

transition-delay: <time>

TIMING-FUNCTION



TRANSITION



IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			31						
8			42					webkit 4.1	
9		38	43			7.1		webkit 4.3	
10		39	44		31	8.4		4.4.4	
11	12	40	45	8	32	9	8	44	44
	13	41	46	9	33				
		42	47		34				
		43	48						

ANIMATION

- Permite establecer animaciones a un elemento:

animation: [name] [duration] [timing-function] [delay] [iteration-count] [direction] [fill-mode]

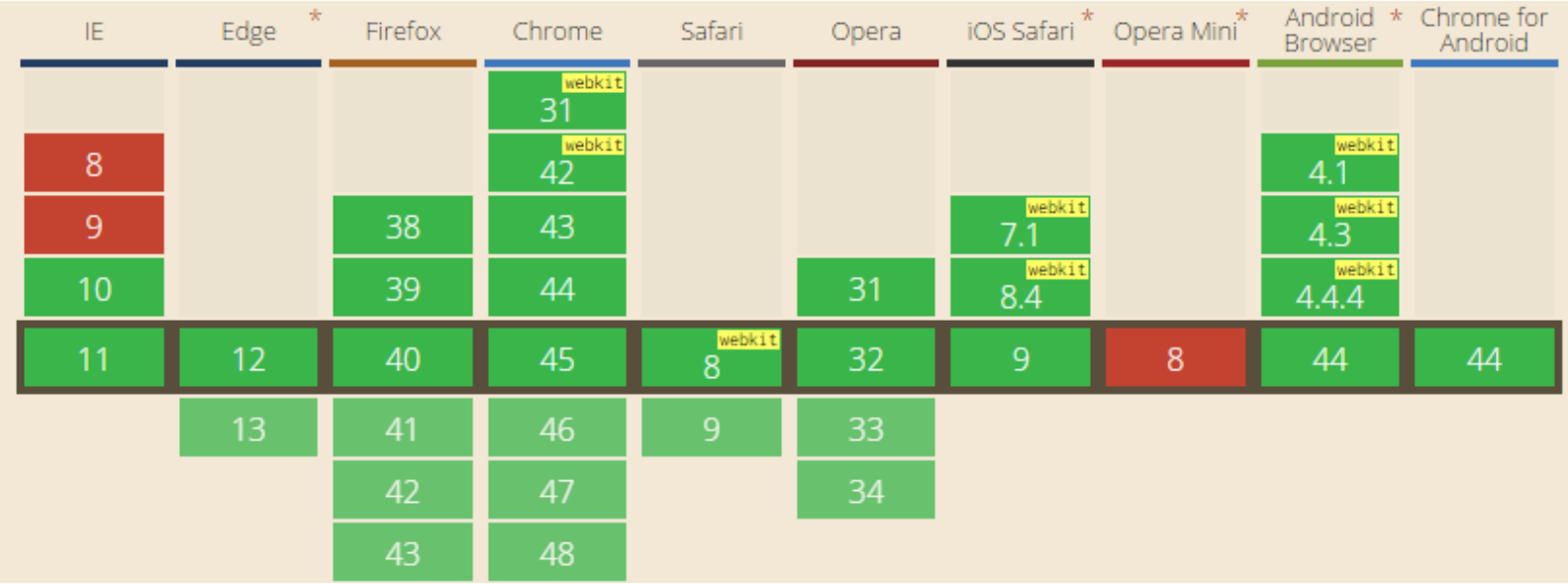
animation-name: <name>
animation-duration: <time>
animation-timing-function: ease, linear, ease-in, ease-out, ease-in-out, cubic-bezier, steps
animation-delay: <time>
animation-iteration-count: <number>, infinite
animation-direction: normal, alternate, reverse, alternate-reverse
animation-fill-mode: none, forwards, backwards, both

ANIMATION

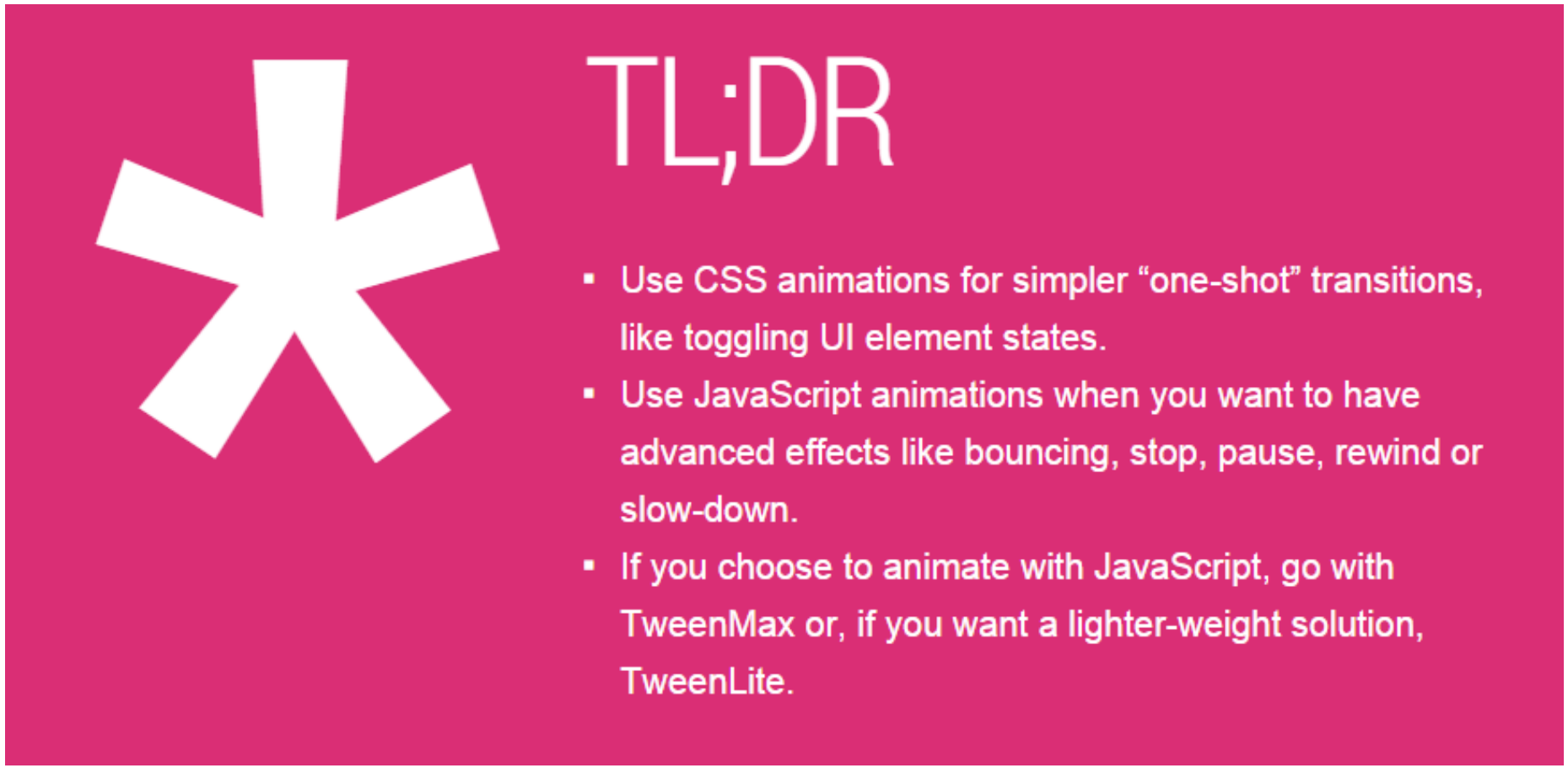
```
.mouth {
    position: absolute;
    top: 85px;
    animation: mouth-up-down 0.15s linear 9s infinite normal forwards;
}
```

```
@keyframes mouth-up-down {
  50% {
    top: 80px;
  }
  100% {
    top: 85px;
  }
}
```

ANIMATION


IE
Edge *
Firefox
Chrome
Safari
Opera
iOS Safari *
Opera Mini *
Android Browser *
Chrome for Android
8
9
10
11
12
13
38
39
40
41
42
43
webkit 31
webkit 42
43
44
45
46
47
48
webkit 8
9
31
32
33
34
webkit 7.1
webkit 8.4
9
8
webkit 4.1
webkit 4.3
webkit 4.4.4
44
44

ANIMACIONES CSS VS JS



https://developers.google.com/web/fundamentals/look-and-feel/animations/css-vs-javascript

FLEXBOX

- Viene a solucionar uno de los mayores problemas de los desarrolladores web:

Crear componentes flexibles, fluidos y dinámicos utilizando CSS

FLEXBOX

- Es un nuevo modelo de caja:

```
display: flex;
```

```
display: inline-flex;
```

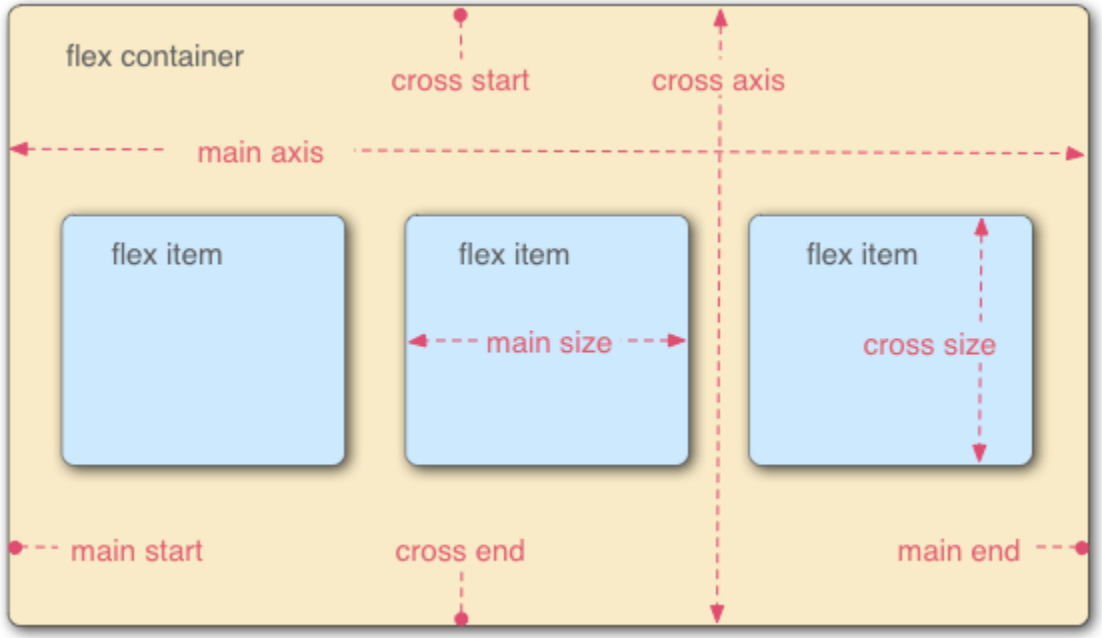
- Que permite **ajustar tamaños y disposición de los elementos que se encuentran dentro**, de tal manera que **se adapten** siempre **al espacio disponible**.

DISPLAY: FLEX;

- Permite definir un contenedor flexible.
- Sus elementos hijos se convierten en “elementos flexibles”.

DISPLAY: FLEX;

- Dispone de dos ejes:
  - Principal (main axis): Dirección en la que posicionan los elementos flexibles.
  - Transversal (cross axis): Perpendicular al principal.



https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Flexible_boxes

FLEX-DIRECTION

- Permite establecer el eje principal (main axis), que definirá la dirección en la que los hijos son posicionados:

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

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

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

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

- Lo define el contenedor padre.

FLEX-WRAP

- Permite indicar si el contenedor flexible contiene una sola línea o múltiples.
- Si tienes varias, también se puede indicar la dirección en la que se colocan las nuevas líneas en el eje transversal.
- Lo define el contenedor padre.

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

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

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

FLEX-FLOW

- *Shorthand* de `flex-direction` y `flex-wrap`.
- Lo define el contenedor padre.

flex-flow: [flex-direction] [flex-wrap]

```
flex-flow: row wrap;
```

ORDER

- Permite modificar el orden en el que aparecen los elementos flexibles en su contenedor.
- Por defecto, los elementos se ordenan en base a su aparición.
- Admite números negativos.
- Lo define cada elemento hijo flexible.

```
order: 4;
```

FLEX-GROW

- Define la habilidad de los elementos flexibles para crecer en caso de que sea necesario.
- Acepta valores sin unidad, que actúan a modo de proporción.
- No admite números negativos.
- Lo define cada elemento hijo flexible.

```
flex-grow: 2;
```

JUSTIFY-CONTENT

- Permite distribuir el espacio libre a lo largo del eje principal.
- Lo define el contenedor padre.

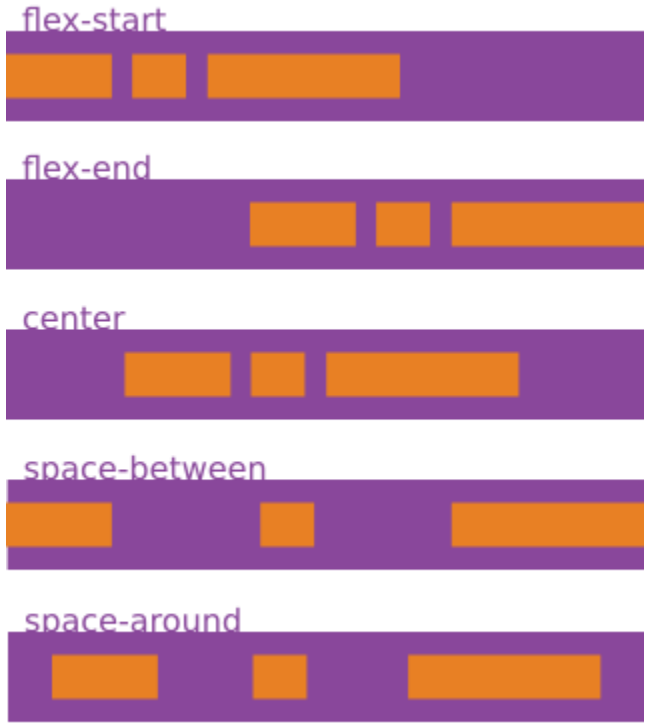
```
justify-content: flex-start;
```

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

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

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

```
justify-content: space-around;
```



ALIGN-ITEMS

- Permite definir como se colocan los elementos a lo largo del eje transversal.
- Lo define el contenedor padre.

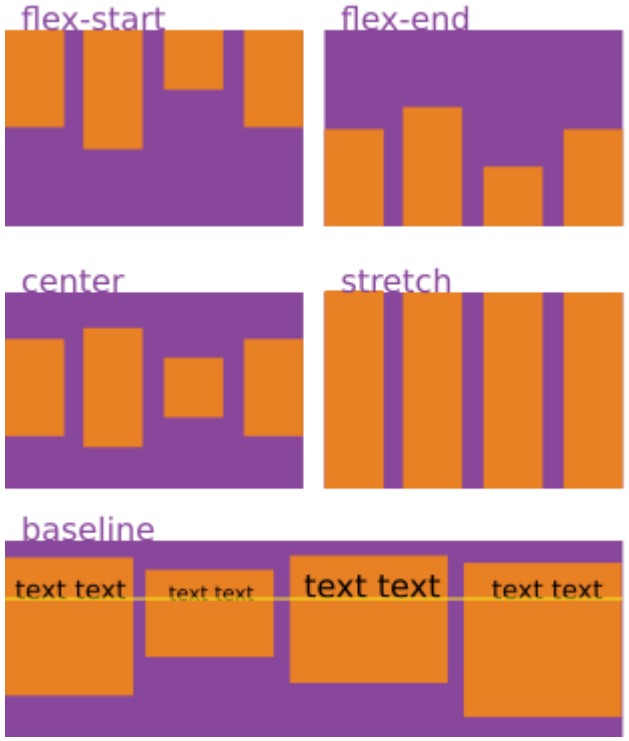
```
align-items: flex-start;
```

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

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

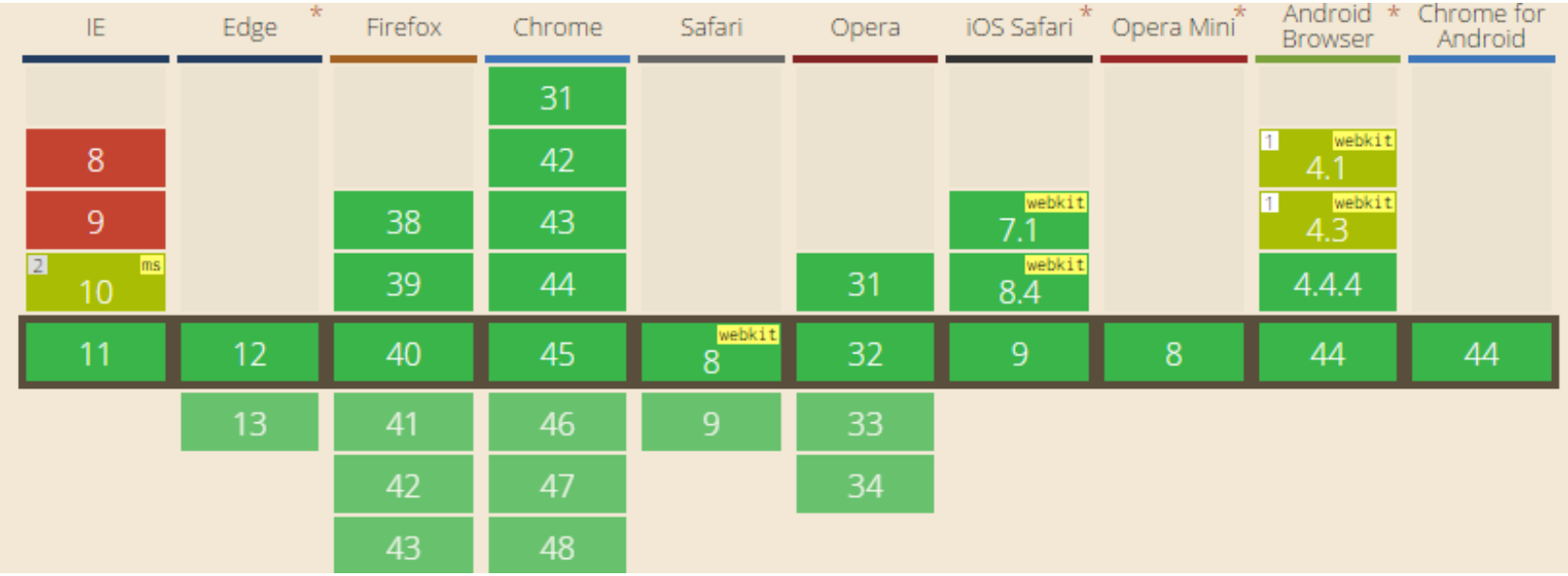
```
align-items: stretch;
```

```
align-items: baseline;
```



plain concepts

FLEXBOX


IE
Edge *
Firefox
Chrome
Safari
Opera
iOS Safari *
Opera Mini *
Android Browser *
Chrome for Android
8
9
2 ms 10
11
12
13
38
39
40
41
42
43
31
42
43
44
45
46
47
48
webkit 8
9
31
32
33
34
webkit 7.1
webkit 8.4
9
8
1 webkit 4.1
1 webkit 4.3
4.4.4
44
44

LECTURAS RECOMENDADAS.

- Castellano:
  - http://ksesocss.blogspot.com/
  - http://supercss.net/
  - http://www.frontazo.com/

- Inglés:
  - http://css-tricks.com/
  - http://css-weekly.com/
  - http://web-design-weekly.com/

RECURSOS

- Centering:
  - http://css-tricks.com/centering-css-complete-guide/
  - http://ksesocss.blogspot.com/2012/05/centrando-al-centro-con-css-16-maneras.html
- Display grid:
  - https://developer.mozilla.org/en-US/docs/Web/Guide/CSS/Using_multi-column_layouts
  - http://codepen.io/HugoGiraudel/pen/c3558b7c99737b3787fded4774de37a5
- CrossBrowser testing:
  - http://www.browserstack.com/

FUENTES

- http://www.w3c.es
- http://css-tricks.com
- http://inserthtml.com
- http://www.paulirish.com
- http://librosweb.es
- http://code.tutsplus.com
- http://www.w3schools.com
- http://stackoverflow.com
- http://www.cssbasics.com
- http://es.wikipedia.org
- http://ksesocss.blogspot.com
- http://www.emenia.es/
- http://www.cssneuse.net