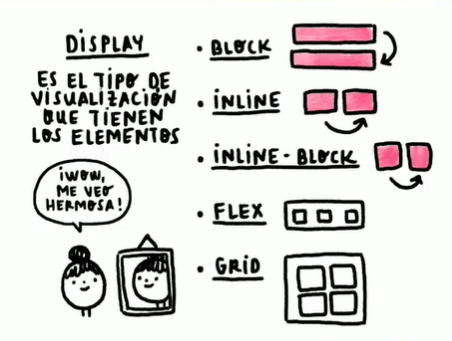
<https://htmlreference.io>



Existen etiquetas que por defecto su display ya está determinado, como la etiqueta <div>que tiene display block, <span> tiene display inline y <button> tiene display inline-block.

Abordaremos los tipos de display block, inline e inline-block a continuación.

**Visualización en bloque (block)**

El display block \*\*establece que un elemento ocupará todo el espacio disponible por defecto y el siguiente elemento a este se situará por debajo.

Es posible **añadir medidas** de anchura width y altura height a estos a elementos.

También es posible **agregar todas las propiedades del modelo de caja** (no te preocupes de este concepto, ya lo abordaremos).

* [Ejemplo de display block](https://codi.link/PGRpdj5Tb3kgZGlzcGxheSBibG9jazwvZGl2Pg0KPGRpdj5Tb3kgZGlzcGxheSBibG9jazwvZGl2Pg0KPGRpdiBjbGFzcz0iY29uX21lZGlkYXMiPlNveSBkaXNwbGF5IGJsb2NrPC9kaXY+DQoNCg==%7CLyogUXVpdGEgbG9zIGNvbWVudGFyaW9zIHkgb2JzZXJ2YSBlbCBjb21wb3J0YW1pZW50byAqLw0KZGl2ew0KICBiYWNrZ3JvdW5kLWNvbG9yOiBhcXVhOw0KICAvKiBtYXJnaW46IDEwcHg7ICovDQogIC8qIHBhZGRpbmc6IDEwcHg7ICovDQp9DQoNCi5jb25fbWVkaWRhcyB7DQogIC8qIHdpZHRoOiAyMDBweDsgKi8NCiAgLyogaGVpZ2h0OiAyMDBweDsgKi8NCn0NCg0KLyogSWdub3JhIGVzdG9zIGVzdGlsb3MsIHBvciBhaG9yYSAqLw0KKiB7DQogIGZvbnQtc2l6ZTogMS4ycmVtOw0KICBtYXJnaW46IDA7DQp9DQoNCg0KDQo=%7C)

**Visualización en línea (inline)**

El display inline **establece que un elemento ocupará el espacio del contenido del mismo y el siguiente elemento se situará a la derecha**.

**No es posible añadir medidas** de anchura width y altura height a estos a elementos.

También, **no es posible agregar todas las propiedades del modelo de caja, únicamente funcionará la propiedad margin en el eje horizontal** (no te preocupes de este concepto, ya lo abordaremos).

* [Ejemplo de display inline](https://codi.link/PHNwYW4+U295IGRpc3BsYXkgaW5saW5lPC9zcGFuPg0KPHNwYW4+U295IGRpc3BsYXkgaW5saW5lPC9zcGFuPg0KPHNwYW4+U295IGRpc3BsYXkgaW5saW5lPC9zcGFuPg0KPHNwYW4+U295IGRpc3BsYXkgaW5saW5lPC9zcGFuPg0KPHNwYW4gY2xhc3M9ImNvbl9tZWRpZGFzIj5Tb3kgZGlzcGxheSBpbmxpbmU8L3NwYW4+DQoNCg==%7CLyogUXVpdGEgeSBhZ3JlZ2EgbG9zIGNvbWVudGFyaW9zIHkgb2JzZXJ2YSBlbCBjb21wb3J0YW1pZW50byAqLw0Kc3BhbnsNCiAgYmFja2dyb3VuZC1jb2xvcjogYXF1YTsNCiAgLyogbWFyZ2luOiAyMHB4OyAqLw0KICAvKiBwYWRkaW5nOiAyMHB4OyAqLw0KfQ0KDQouY29uX21lZGlkYXMgew0KICAvKiB3aWR0aDogMjAwcHg7ICovDQogIC8qIGhlaWdodDogMjAwcHg7ICovDQp9DQoNCi8qIElnbm9yYSBlc3RvcyBlc3RpbG9zLCBwb3IgYWhvcmEgKi8NCiogew0KICBmb250LXNpemU6IDEuMnJlbTsNCiAgbWFyZ2luOiAwOw0KfQ0KDQoNCg0K%7C)

**Visualización de bloque y línea (inline-block)**

El display inline-block combina las ventajas de bloque de colocar medidas al elemento y propiedades del modelo de caja correctamente; con las ventajas de inline de color un elemento seguido de otro en el mismo espacio.

Si elemento excede el contenido total, se coloca en la siguiente línea por debajo.

* [Ejemplo de display inline-block](https://codi.link/PGJ1dHRvbj5Tb3kgZGlzcGxheSBpbmxpbmUtYmxvY2s8L2J1dHRvbj4NCjxidXR0b24+U295IGRpc3BsYXkgaW5saW5lLWJsb2NrPC9idXR0b24+DQo8YnV0dG9uPlNveSBkaXNwbGF5IGlubGluZS1ibG9jazwvYnV0dG9uPg0KPGJ1dHRvbj5Tb3kgZGlzcGxheSBpbmxpbmUtYmxvY2s8L2J1dHRvbj4NCjxidXR0b24gY2xhc3M9ImNvbl9tZWRpZGFzIj5Tb3kgZGlzcGxheSBpbmxpbmUtYmxvY2s8L2J1dHRvbj4NCg0K%7CLyogUXVpdGEgbG9zIGNvbWVudGFyaW9zIHkgb2JzZXJ2YSBlbCBjb21wb3J0YW1pZW50byAqLw0KYnV0dG9uew0KICAvKiBtYXJnaW46IDEwcHg7ICovDQogIC8qIHBhZGRpbmc6IDEwcHg7ICovDQp9DQoNCi5jb25fbWVkaWRhcyB7DQogIC8qIHdpZHRoOiAzMDBweDsgKi8NCiAgLyogaGVpZ2h0OiAxMDBweDsgKi8NCn0NCg0KLyogSWdub3JhIGVzdG9zIGVzdGlsb3MsIHBvciBhaG9yYSAqLw0KKiB7DQogIGZvbnQtc2l6ZTogMS4xcmVtOw0KICBtYXJnaW46IDA7DQp9DQoNCg0KDQo=%7C)

**Visualización nula (none)**

El display none **desactiva la visualización de un elemento**, como si el elemento no existiera.

* [Ejemplo de display none](https://codi.link/PGRpdj48L2Rpdj4NCjxkaXY+PC9kaXY+DQo8ZGl2IGNsYXNzPSJkZXNhcGFyZWNlciI+DQogIEVuIG1pIHNpZ3VpZW50ZSB0cnVjbywgwqF2b3kgYSBkZXNhcGFyZWNlciENCjwvZGl2Pg0KPGRpdj48L2Rpdj4NCjxkaXY+PC9kaXY+DQo=%%7C)

**FLEX Y GRID**

El display flex y grid son formas de visualización de elementos recientes y cada uno tienen sus propias características para crear interfaces de manera efectiva, a partir de un contenedor padre que dotará a los elementos hijos de superpoderes del posicionamiento.

Ambas son herramientas muy útiles en el desarrollo, especialmente para la creación de interfaces amigables al usuario y aptas para cualquier dispositivo, que este último se lo conoce como [responsive design](https://platzi.com/clases/2467-frontend-developer/40845-responsive-design/).

Sin embargo, ambas herramientas tienen temas muy extensos de entender, y como mi intención no es estresarte con demasiada información, simplemente ten presente de manera general en qué consisten.

**Qué es flexbox**

*Flexbox* consiste en el **ordenamiento de elementos hijos en un solo eje**, por defecto horizontalmente. El elemento padre o contenedor deberá contener la propiedad display con el valor flex. A partir de aquí, ya puedes ordenar los hijos según sea necesario.

* [Ejemplo de flexbox](https://codi.link/PGRpdiBjbGFzcz0iY29udGFpbmVyIj4NCiAgPGRpdj48L2Rpdj4NCiAgPGRpdj48L2Rpdj4NCiAgPGRpdj48L2Rpdj4NCiAgPGRpdj48L2Rpdj4NCiAgPGRpdj48L2Rpdj4NCjwvZGl2Pg0KDQoNCg0K%7CKiB7DQogIG1hcmdpbjogMDsNCiAgcGFkZGluZzogMDsNCiAgYm94LXNpemluZzogYm9yZGVyLWJveDsNCn0NCg0KLyogUXVpdGEgbG9zIGNvbWVudGFyaW9zIHkgb2JzZXJ2YSBlbCBjb21wb3J0YW1pZW50byAqLw0KLmNvbnRhaW5lcnsNCiAgLyogZGlzcGxheTogZmxleDsgKi8NCn0NCg0KLmNvbnRhaW5lciBkaXYgew0KICB3aWR0aDogMTAwcHg7IA0KICBoZWlnaHQ6IDEwMHB4Ow0KfQ0KDQouY29udGFpbmVyIGRpdjpudGgtY2hpbGQoMm4pew0KICBiYWNrZ3JvdW5kLWNvbG9yOiBhcXVhOw0KfQ0KDQouY29udGFpbmVyIGRpdjpudGgtY2hpbGQoMm4rMSl7DQogIGJhY2tncm91bmQtY29sb3I6IGJyb3duOw0KfQ0KDQoNCg0KDQoNCg0K%7C)

Cuando domines los temas básicos de CSS y las propiedades más usadas, revisadas en el contenido del curso, sigue con *Flexbox*. Puedes utilizar la guía y el curso en Platzi sobre el tema:

* [A Complete Guide to Flexbox](https://css-tricks.com/snippets/css/a-guide-to-flexbox/)
* [Curso de Diseño Web con CSS Grid y Flexbox](https://platzi.com/clases/flexbox-css-grid/)

**Qué es grid**

*Grid* consiste en el **ordenamiento de elementos hijos en dos ejes**, como si fuera una cuadrícula o tabla. El elemento padre o contenedor deberá contener la propiedad display con el valor grid y debes definir las medidas de las columnas y de las filas. A partir de aquí, ya puedes ordenar los hijos según sea necesario.

* [Ejemplo de grid](https://codi.link/PGRpdiBjbGFzcz0iY29udGFpbmVyIj4NCiAgPGRpdj48L2Rpdj4NCiAgPGRpdj48L2Rpdj4NCiAgPGRpdj48L2Rpdj4NCiAgPGRpdj48L2Rpdj4NCiAgPGRpdj48L2Rpdj4NCiAgPGRpdj48L2Rpdj4NCiAgPGRpdj48L2Rpdj4NCiAgPGRpdj48L2Rpdj4NCiAgPGRpdj48L2Rpdj4NCjwvZGl2Pg0KDQoNCg0K%%7C)

Cuando domines lo básico de *Flexbox*, sigue con *Grid*. Puedes utilizar la guía y el curso en Platzi que tenemos sobre el tema:

* [A Complete Guide to Grid](https://css-tricks.com/snippets/css/complete-guide-grid/)
* [Curso de CSS Grid Básico](https://platzi.com/cursos/css-grid/)

Y cuando domines lo básico de ambas, sigue con el [Curso de Diseño Web con CSS Grid y Flexbox](https://platzi.com/cursos/flexbox-css-grid/). Para que aprendas a emplear y combinar ambas herramientas correctamente.