CSS GRID LAYOUT

¡HOLA!

Soy **Diana Aceves**

Frontend Developer

Twitter: **@diana_aceves_**

EJEMPLOS DEL CURSO

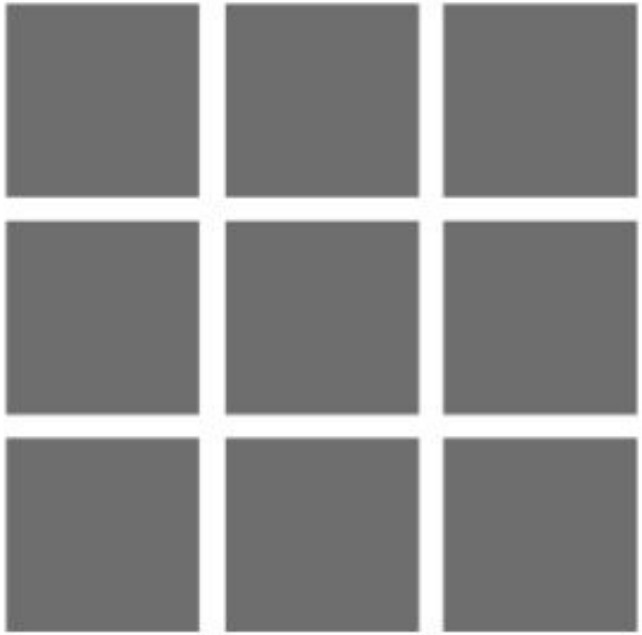
Colección Codepen:

“Escuela IT 2017:
Curso CSS GRID LAYOUT”

https://codepen.io/collection/712dfdc773b2532beaff9157a2ffe194/

¿QUÉ ES GRID LAYOUT ?

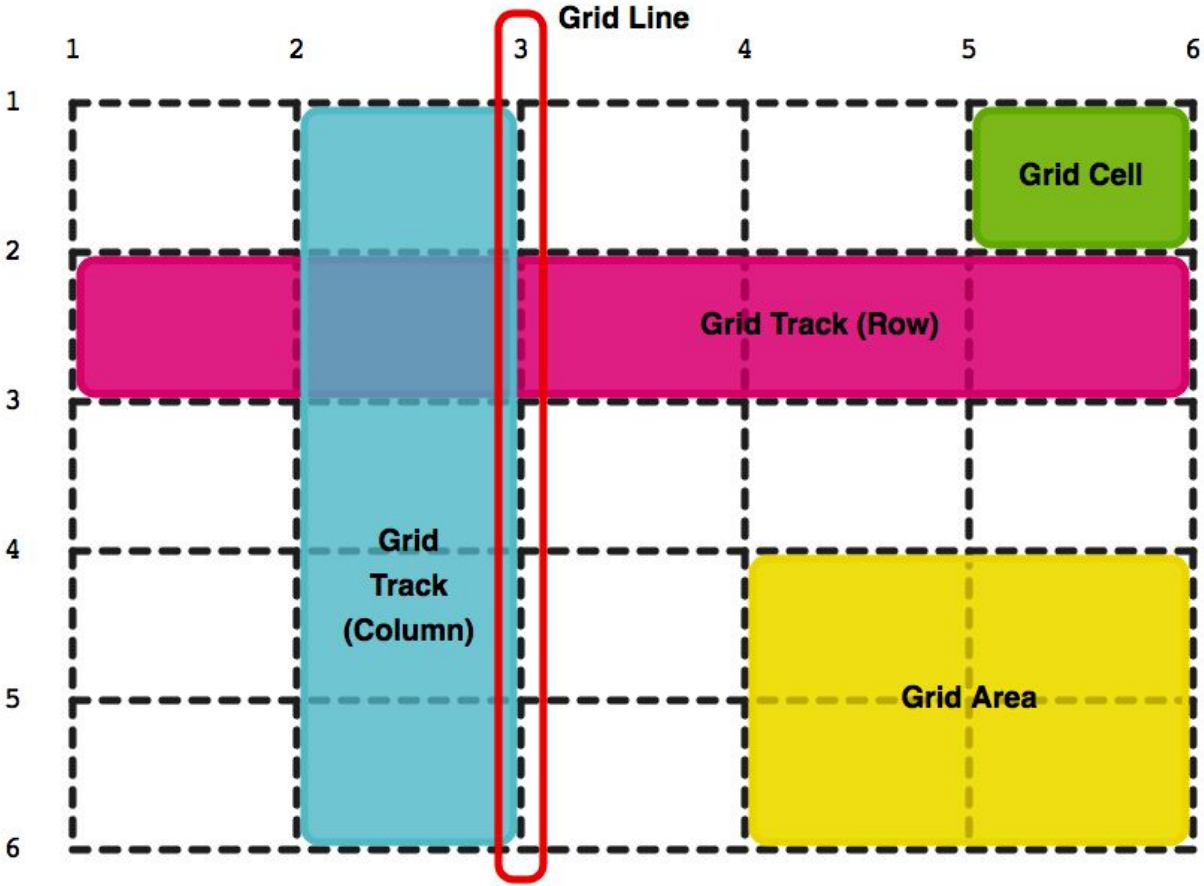
- Sistema de rejilla en
  2 DIMENSIONES



¿QUÉ TIENE DIFERENTE ?

- Voy a poder colocar los items **DONDE QUIERA…**
- **…PERO** habrá items que **SE COLOQUEN SOLOS** (AUTO-PLACEMENT)
- Puedo **HACER LO MISMO DE MUCHAS FORMAS…**
- **…PERO NO TODAS HACEN LO MISMO.**
- Controlo las 2 dimensiones, **NO ES FLEXBOX.**
- La colocación de los items es muy libre, **NO ES UNA TABLA.**
- Tiene una **EXTENSA SINTAXIS.**
- Nos va a volver un poco locos, pero **VA A CAMBIAR EL CSS PARA SIEMPRE.**

CONCEPTOS BÁSICOS



EMPECEMOS: DISPLAY GRID

En cuanto declaro

DISPLAY: GRID o

DISPLAY: INLINE-GRID

los hijos directos de ese elemento pasan a ser **GRID ITEMS.**

Vamos a verlo:
GRID LAYOUT - EJ.1

DEFINIENDO TRACKS EN EL GRID

- Creamos el grid **EXPLÍCITAMENTE.**
- Es decir, al **contenedor GRID**, le indico cómo deben ser las filas y las columnas:
- **EXISTEN MÚLTIPLES SINTAXIS Y FORMAS DE HACERLO.**

GRID LAYOUT - EJ.2

Funciones y keywords MUY ÚTILES

- Tengo que saber cada track? Cuánto mide? Cuántos hay? Escribirlos todos? **MUERO!!!**

NO

- repeat()
- minmax()
- auto-fill / auto-fit

GRID LAYOUT - EJ.3