



## Curso Profesional de CSS Grid Layout



### Aprobaste el curso

Felicitaciones, ya puedes acceder a tu [diploma digital](#)

# 9.5

Calificación

# 19 / 20

Aciertos

1. Evaluar el panorama actual con respecto a las herramientas que nos han facilitado el camino a lo largo de los años, nos sirve para:

Ver el impacto que tienen estas herramientas en la forma en la que diseñamos y desarrollamos



2. CSS Grid es una idea nueva

Falso



3. CSS Grid requiere una forma completamente nueva de pensar sobre el diseño en CSS

Verdadero



4. ¿Cuál de las siguientes propiedades no fue utilizada en la técnica de alineamiento con line-height?

table-cell



5. Una de las mayores limitantes de trabajar con las técnicas de alineamiento vistas en el curso es:

Propiedades lógicas



6. En latín, el Block Flow Direction va de:

Arriba hacia abajo



7. La propiedad lógica equivalente a la propiedad física margin-top es:

margin-block-start



8. caniuse.com es una página que nos permite:

Revisar la compatibilidad de las propiedades de CSS en los diferentes navegadores



9. Los elementos hijos de un contenedor también pueden ser padres

Verdadero



10. Uno de los tips dados en el curso para aprender a trabajar con CSS Grid es:

Identificar las relaciones de padre e hijos inmediatos para así saber qué propiedades implementar en cada caso



11. ¿Puedo darle width a un elemento inline? (Esto hace parte de una de las tareitas que te dejé)

Falso



12. A un elemento padre se le debe poner siempre la propiedad:

display: grid (o inline-grid)



13. La palabra "justify" hace alusión al eje:

inline



14. La palabra "align" hace alusión al eje:

block



15. place-items es una propiedad del elemento:

padre



16. justify-content es una propiedad del elemento:

padre



17. place-self es una propiedad del elemento:

hijo



18. ¿Qué propiedad usarías en el elemento padre, si quisieras alinear sus elementos hijos a la derecha?

justify-items: end;



19. Si queremos especificar el tamaño de los tracks de cuadrícula generados automáticamente, lo podemos hacer con la propiedad:

grid-auto-columns



20. Una de las cosas recomendadas para comenzar a trabajar con diseño responsivo es:

Establecer breakpoints dependiendo de los dispositivos que queremos/necesitemos abarcar



---

**REGRESAR**