**Grid**

**Загальна розмітка стіки:**

**grid-template-column**: 100px 100px 100px …; - ширина колонки і їх кількість

**grid-template-row**: 50px 50px 50px; - висота колонок і їх кількість

**grid-template**: 100px 100px 100px / 50px 50px 50px; **(row/ column) – короткий запис**

**grid-auto-rows / grid-auto-column** – дозволяє контролювати розміри динамічної сітки, впливає на розміри, але не на структуру

**grid-template-areas** – властивість для промальовування структури сітки в вигляді строки задаючи назву блоку **“header header**

**side content”**

**grid-area:** blockName **–** задає назву блоку для взаємодії із властивістю grid-template-area

grid-auto-flow (row, column, dense) – визначаємо напрямок свторення нових колонок. row – звична поведінка створення нових колонок, column – створення нових колонок лише по ширині контейнера, dense – заповнення доступного простору ігноруючи послідовність

**Задання розміру колонкам:**

**fr – одиниця виміру, що** вирівнює елементи в рівній степені відштовхуючись від ширини чи висоті контейнера.

**repeat(**count, value**) –** повторяє вписаний запис вказану к-сть разів

* + 1. **Grid-template-column: repeat(3, 50px 1fr) –** відбудеться чергування значень 3 рази
    2. **Grid-template-column: repeat(3, 1fr) 20px –** 3 контейнера по 1fr і 1 на 20px
    3. **Auto-fill –** чтобы он уместил как можно больше колонок с учетом указанной длины. але при розміщенні усіх колонок в в ряд, залишається вільне місце для потенційно нових

**(grid-template-column: repeat(auto-fill, minmax(min px, 1fr))) – часта формула адаптивності шаблону на сторінці**

Iv**. Auto-fit –** діє так само, лише при повному розтягненні колонок не залишає пустого місця, а розтягує колонки і відступи для заповнення усієї строки елементами рівномірно

1. **Задання імен лініям**: ***grid-template-column: [start] size [nextEl start] size [col-end]***, так само і до grid-template-row: - к-сть елементів вираховується автоматично від заданих парметрів, як і при звичайному утворені сітки

**Пізніше дані імена ми використовуємо для динамічного позиціонування через grid-column / grid-row**

**minmax(min, max)** - Встановлення мінімальної\максимальної ширини/висоти колонки. min значення не може бути в одиницях виміру fr

**auto** – розширяє колонку максимально по її наповненню

**fit-content(**100px**)** – працює, як auto з обмеженням по розширенню колонки вказаній, як параметр

**Відступи між елементами сітки, окрім крайніх сторін першого і останнього:**

1) column-gap: відступи по горизонталі між колонками

2) row-gap: відступи між строками

gap: row-gap column-gap

**Зміна напрямку відображення явних і не явних елементів:**

1. Grid-auto-flow: row / column / dense(розміщення не по порядку, а по розміру)

**Розміщення не формальних блоків на Grid сітці:**

1. Grid-column: index\_column\_line\_start / index\_column\_line\_end(number)
2. Grid-row: index\_row\_line-start(number) / index\_row\_line-end

*! Якщо елемент займає лише одне місце, то вказуємо лише один параметр старту*

*!Замість індекса можуть використовуватись імена колонок задані при створені сітки в grid-template-…, індекс є завжди, але назви може і не бути*

**Властивості для вирівнювання елементів по гололовній і другорядній осі:**

1. align-selft: start / end /stratch / center / content - **До конкретного елемента grid** сітки можна використати дану властивість, для позиціонування окремого блоку в межах доступного йому місця, елемент приймає
2. justify-self: вирівнювання **конкретного елемента по** головній осі
3. justify-items: глобально для кожного елемента контейнера задає властивість вище
4. align-items: start / end / stratch / center / content – властивість задається контейнеру для глобального оголошення властивості вище для усіх елементів контейнера
5. justify-content: вирівнювання усіх елементів контейнера (контенту) по головній осі в середині контейнера
6. align-content: вирівнювання усього контенту по другорядній осі
7. order – для зміни позиції елементів
8. margin: auto - застосовується для центрування окремого елемента

**Комбінація вищезаписаних властивостей**

1. place-content: justify-content align-content – комплексна властивість, яка задається контейнеру для позиціонування всього контенту по головній і другорядній осі
2. place-items: justify-items align-items: - комплексна властивість, яка задаєтсья контейнеру для глобального прописування align-self justify-self кожному елементу
3. place-self: align-self justify-self - комплексна властивість, яка задається конкретному елементів для його вирівняння по осям

**Позначення імені для окремих grid елементів**

1. grid-area: name – позначення імені конкретного grid element і застосовується в конкретному елементі
2. grid-template-areas: - властивість в якій ми описуємо структуру grid сітки через об’єявлення імен grid елементів в строці, яка відображає одну строку в grid елементі
3. (grid-template-areas: ‘header header header’

‘blog blog message’

‘footer footer footer’)

– сітка 3 на 3 складається із header, розтягнутого на 3 колонки, blog на 2 перші , message на третю в центральному рядку і footer розтягнутий на 3 нижні колонки

! якщо нам потрібно зробити пропуск в конструкції, то потрібно поставити ‘.’, пропуск залишати не можна

!блок має обов’язково бути прямокутна по структурі, якщо елемент буде буквою Г – структура зруйнується

**Фічі:**

1. Якщо не вказувати розмір елемента в середині грід сітки, то він повністю прийме розміри елемента сітки в якому знаходиться