

**ОСНОВИ ПРОГРАМУВАННЯ  
ДЛЯ ШКОЛЯРІВ**

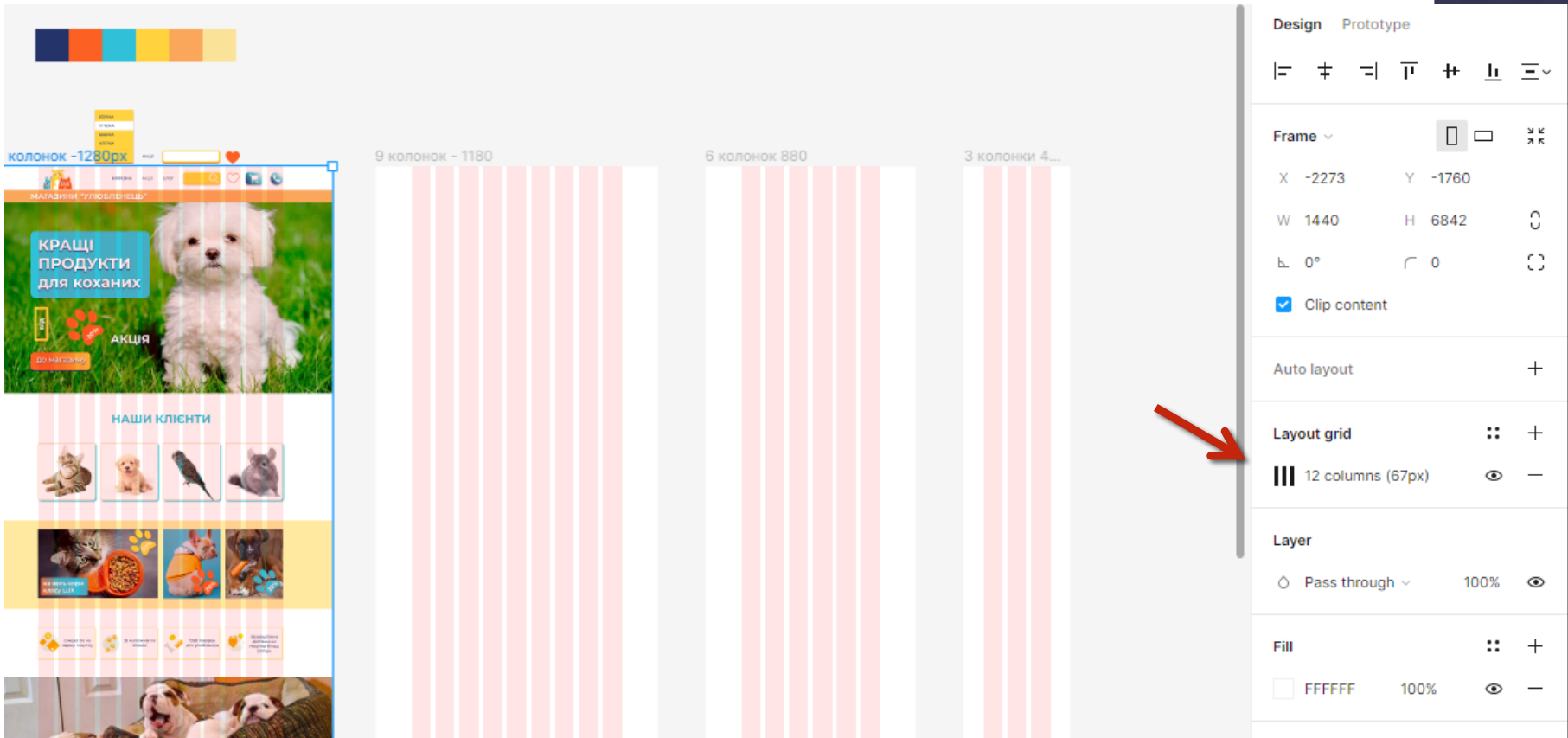
**СТВОРЕННЯ САЙТУ**

**HTML ТА CSS**

**ГРУПА  
PRO**

**BOOTSTRAP  
3 ЧАСТИНА  
СЕТКИ**

Ми знаємо, як позиціонувати елементи в html за допомогою flex технології. В Bootstrap все відбувається за допомогою системи сіток. Будь яка розмітка в англійській мові називається **Layout**. Ви можете побачити також таку назву в **Figma**. Такі сітки потрібні дизайнерам для того, щоб вирівнювати всі елементи. Так як макети верстаються для декількох адаптивних версій, як правило, їх верстають для 12, 9, 6 та 3 колонок. І зазвичай, після верстки макету, дизайнер прибирає видимість таких сіток.



**Bootstrap** підтримує верстку колонками. Тобто вмiє розставляти елементи за вертикальними колонками. Відмінність в тому, що він використовує всі колонки для адаптивної верстки автоматично.

Ми пам'ятаємо, що зазвичай у сайту використовується **.container** який обмежує весь контент сайту певною шириною, яка дозволяє підлатувати сайт для користувача. В **Bootstrap** також є такий клас **.container**, який відповідає звичному нам контейнеру. То ми можемо додавати його там, де він необхідний.

В разі, коли нам не потрібно обмежувати контент контейнером, але треба вказати контейнер для подальшої адаптивної верстки застосовується клас – **.container-fluid** Тобто контент розтягнеться на всю ширину.

- **Але запам'ятайте, якщо ви хочете використовувати строки, обов'язково треба використовувати .container**

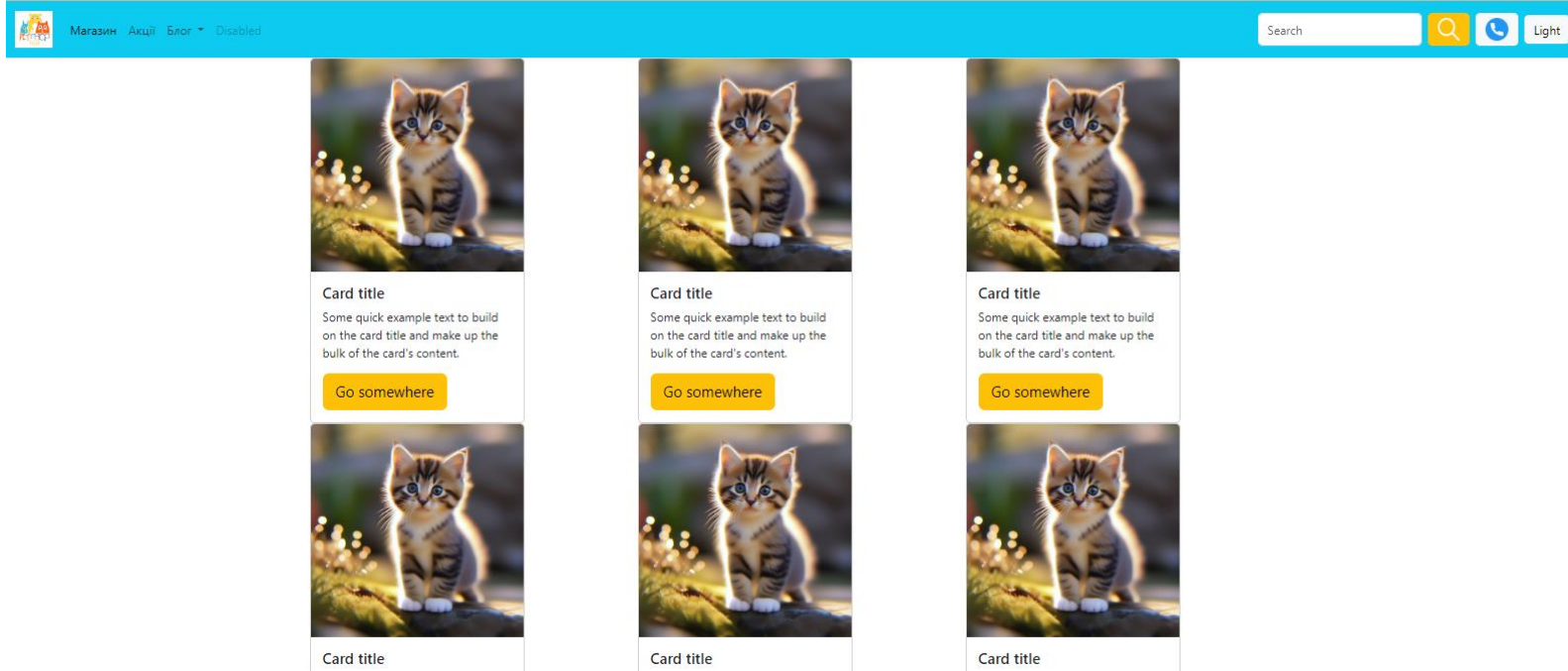
Далі, ми бачимо, що картки розташовані в строках. Для того, щоб розташувати картки в двох строках, там треба додати дві строки **.row**

Тепер, в залежності від того, скільки колонок (карток) вам потрібно, ви вставляєте кількість **.col**

Тепер в сітку, яку ми створили, ми можемо додати наші картки.

```
<div class="container">
  <div class="row">
    <div class="col"></div>
    <div class="col"></div>
    <div class="col"></div>
  </div>
  <div class="row"></div>
</div>
```

Після того, як ми помістили наші картки в колонки та строки, ми бачимо, що вони розташовуються строго по ситці, яку ми створили, та в контейнері.

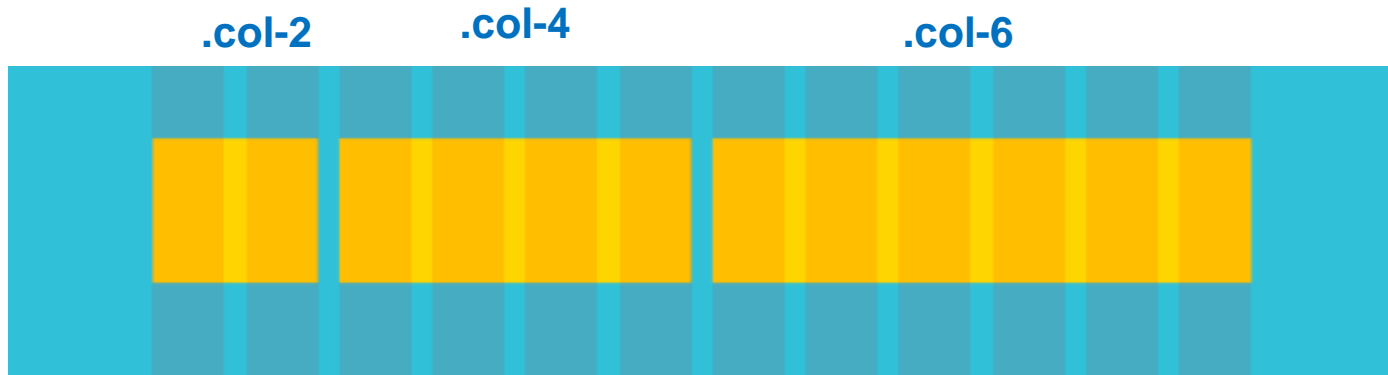


Запам'ятайте, що тільки при такому шаблоні у вас буде працювати така сітка. Якщо ви обернете колонку або строку в якійсь інший div ця технологія не спрацює.

```
<div class="container">
  <div class="row">
    <div class="col"></div>
    <div class="col"></div>
    <div class="col"></div>
  </div>
  <div class="row"></div>
</div>
```



Тепер згадаємо, що у нас всього 12 колонок. І **Bootstrap** просто автоматично поділив всю ширину на три рівні частини. Однак, іноді ми формуємо різні за шириною блоки і нам потрібна різна ширина блоків. Для цього ми можемо застосувати наступну технологію: після **.col** ми вказуємо скільки колонок буде займати наш блок. І блоки будуть займати стільки місця, скільки ви вказали

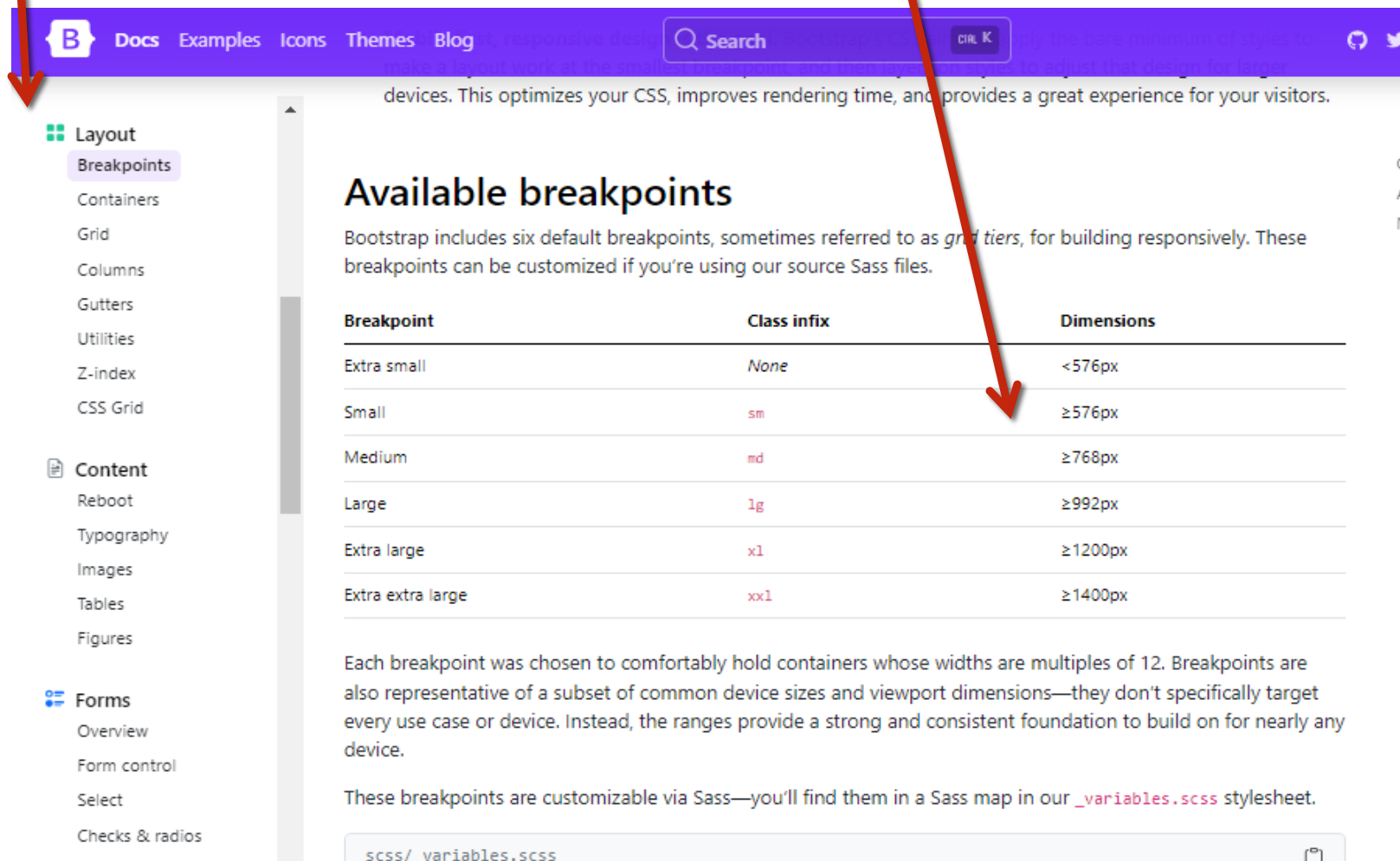


А потім вже ми вставляємо необхідний контент в ці блоки. Це також допоможе нам з адаптаційною версткою, так як медіазапити вже розраховані для кожного класу колонок. Для **.row** вже прописані якості **display:flex; flex-wrap:wrap;**

\*В разі, якщо вам не треба підключати Bootstrap, але хочеться створити таку саму зручну верстку сіткою, ми можете прописати таку сітку на CSS, використовуючи такі ж самі принципи побудови = **container >row>col**. Тільки, щоб ваші блоки не ламали цю структуру, не забудьте для всіх блоків задати **box-sizing; border-box;**

Як працюють медіазапити в Bootstrap ви можете побачити в розділі **Layout > Breakpoints**.

Чому саме такі точки перелому – це було прораховано спеціалістами виходячи з статистики використання сайтів з різною шириною екрана. То можливо наслідувати ці точки в інших проектах.



The screenshot shows the Bootstrap documentation page for 'Available breakpoints'. The page has a purple header with navigation links: Docs, Examples, Icons, Themes, Blog, and a search bar. A sidebar on the left lists categories: Layout (Breakpoints, Containers, Grid, Columns, Gutters, Utilities, Z-index, CSS Grid), Content (Reboot, Typography, Images, Tables, Figures), and Forms (Overview, Form control, Select, Checks & radios). The main content area is titled 'Available breakpoints' and includes a table with three columns: Breakpoint, Class infix, and Dimensions. A red arrow points to the 'sm' class infix in the table. Below the table, there is explanatory text and a code editor snippet showing the Sass variable for the small breakpoint.

devices. This optimizes your CSS, improves rendering time, and provides a great experience for your visitors.

## Available breakpoints

Bootstrap includes six default breakpoints, sometimes referred to as *grid tiers*, for building responsively. These breakpoints can be customized if you're using our source Sass files.

Breakpoint	Class infix	Dimensions
Extra small	<i>None</i>	<576px
Small	<i>sm</i>	≥576px
Medium	<i>md</i>	≥768px
Large	<i>lg</i>	≥992px
Extra large	<i>xl</i>	≥1200px
Extra extra large	<i>xxl</i>	≥1400px

Each breakpoint was chosen to comfortably hold containers whose widths are multiples of 12. Breakpoints are also representative of a subset of common device sizes and viewport dimensions—they don't specifically target every use case or device. Instead, the ranges provide a strong and consistent foundation to build on for nearly any device.

These breakpoints are customizable via Sass—you'll find them in a Sass map in our `_variables.scss` stylesheet.

```
scss/ variables.scss
```

# Надання margin та padding

Для встановлення margin для карток, або для будь якого елемента, існує додатковий клас **.m** та **.p**

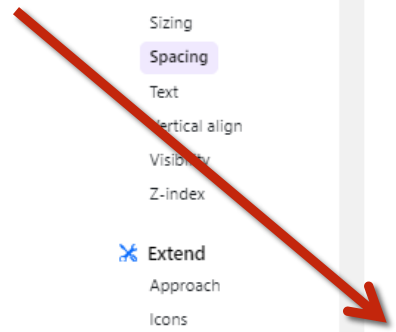
Синопис запису буде виглядати як додатковий клас, наприклад: **mt-5**  
 Де **mt** – margin-top, а **5**-вказувати на величину відступу.

```
<div class="container">
  <div class="row">
    <div class="col">
      <div class="card mt-5" style="width: 18rem;">
```



В стандарті бібліотеки існує декілька значень, які відповідають наступним значенням:

- 0px;
- ~10px;
- ~20px;
- ~30px;
- ~40px;
- ~50px;



- Font
- Interactions
- Link
- Object fit
- Opacity
- Overflow
- Position
- Shadows
- Sizing
- Spacing**
- Text
- Vertical align
- Visibility
- Z-index
- ✖ Extend
- Approach
- Icons
- 🌐 About
- Overview
- Team
- Brand
- License
- Translations

remaining breakpoints, however, do include a breakpoint abbreviation.

The classes are named using the format `{property}{sides}-{size}` for `xs` and `{property}{sides}-{breakpoint}-{size}` for `sm`, `md`, `lg`, `xl`, and `xxl`.

Where *property* is one of:

- **m** - for classes that set **margin**
- **p** - for classes that set **padding**

Where *sides* is one of:

- **t** - for classes that set **margin-top** or **padding-top**
- **b** - for classes that set **margin-bottom** or **padding-bottom**
- **s** - (start) for classes that set **margin-left** or **padding-left** in LTR, **margin-right** or **padding-right** in RTL
- **e** - (end) for classes that set **margin-right** or **padding-right** in LTR, **margin-left** or **padding-left** in RTL
- **x** - for classes that set both **\*-left** and **\*-right**
- **y** - for classes that set both **\*-top** and **\*-bottom**
- **blank** - for classes that set a **margin** or **padding** on all 4 sides of the element

Where *size* is one of:

- **0** - for classes that eliminate the **margin** or **padding** by setting it to `0`
- **1** - (by default) for classes that set the **margin** or **padding** to `$spacer * .25`
- **2** - (by default) for classes that set the **margin** or **padding** to `$spacer * .5`
- **3** - (by default) for classes that set the **margin** or **padding** to `$spacer`
- **4** - (by default) for classes that set the **margin** or **padding** to `$spacer * 1.5`
- **5** - (by default) for classes that set the **margin** or **padding** to `$spacer * 3`
- **auto** - for classes that set the **margin** to `auto`