

**ОСНОВИ ПРОГРАМУВАННЯ
ДЛЯ ШКОЛЯРІВ**

**ГРУПА
PRO**

**СТВОРЕННЯ САЙТУ
HTML ТА CSS**

**BURGER
СХОВАНЕ МЕНЮ**

Певно ви бачили, як на маленьких екранах меню перетворюється на маленькі точки або лінії. Це сховане меню. При його натисканні відкривається сховане меню

Спочатку давайте розберемо, логіку.

При досягненні екраном брекпоінту (наприклад 800px), коли вже меню не вміщується, його треба приховати. А на екран вивести маленький бургер.

Дії з головним меню:

1. Для цього **за замовчуванням** головному меню надається властивість **display: block;**
2. При медіазапросі у 800px пишемо, що меню **не** відображається **display: none;** а burger з'являється

```
/* BURGER */  
.burger {  
  display: none;  
}
```

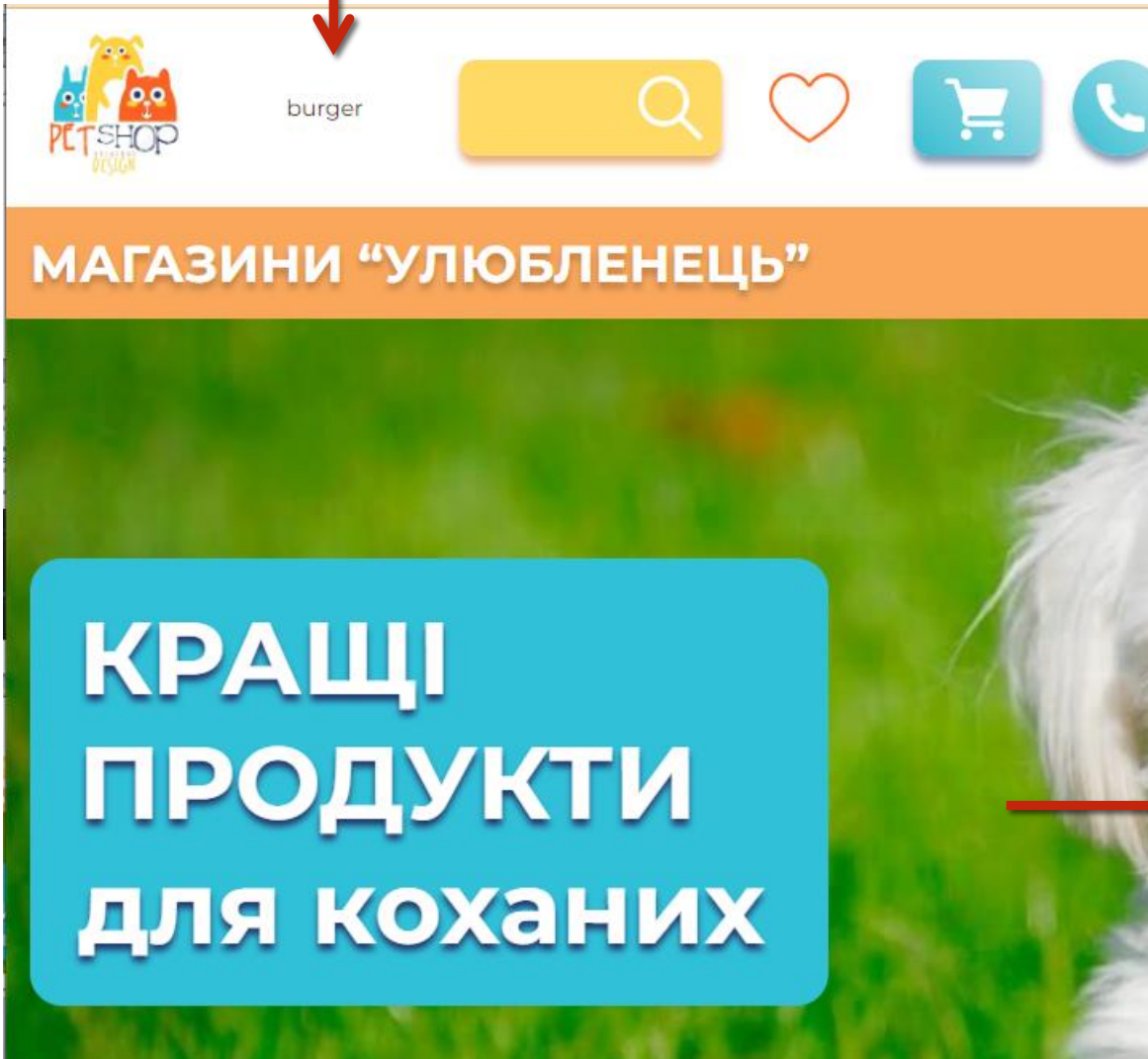
```
462  
463 @media screen and (max-width: 800px){  
464  
465   .nav {  
466     display: none;  
467   }  
468 }  
469 .burger {  
470   display: block;  
471 }  
472 }
```

```

</div>
<button class="burger">
  burger
</button>
</div>
</div>

```

3. Тепер у файлі html, після основного меню формуємо burger. Формуємо його за допомогою тегу button – щоб натискалося все, що знаходиться всередині.



```

<!DOCTYPE html>
<html lang="en">
<head>
<body>
  <!--HEADER-->
  <header class="header">
    <div class="container">
      <div class="header_inner">
        <a href="index.html" class="logo" type="button" id="main">
      </a>
      <nav class="nav">
      <div class="burger"> burger
      </div>
      <div class="bloc_btn">
        </div>
      </div>

```

Стили

```

@media screen and (max-width: 800px)
.nav {
  display: none;
}
.nav {
  display: flex;
}
.nav {
  display: block;
}

```

Так як сам `burger` у нас створений за допомогою тегу `button`, в нього є параметри за замовчуванням. То, щоб не показувалися ці параметри ми пишемо:

`background: none;` - прибираємо колір.

`border: 0;` - прибираємо обведення.

`padding: 10px 3px;` - задаємо трохи відступів, щоб натискання мало більшу площу, а не тільки полоски.

`cursor: pointer;` - змінили курсор на руку.

```
489 .burger {  
490     display: block;  
491     background: none;  
492     border: 0;  
493     padding: 10px 3px;  
494     cursor: pointer;  
495 }
```

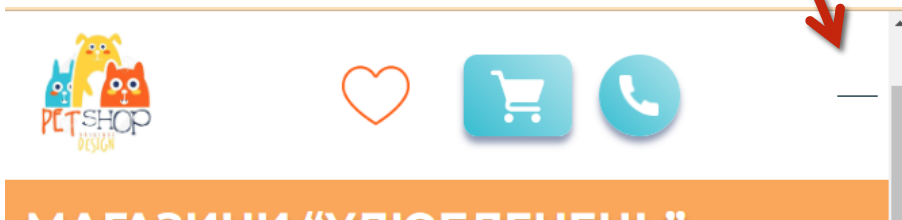

Формувати сам бургер можливо, як просто вставивши іконку такого меню, так і сформувавши його за допомогою універсального тегу ``.
Універсальний рядковий елемент `` (від англ. span - охоплювати) призначений для виділення окремих рядків, символів або інших малих елементів для подальшої зміни їхнього оформлення за допомогою стилів.

Наприклад, всередині абзацу (`<p>`) можна змінити колір і розмір першого слова, якщо його виділити за допомогою елемента `` і задати йому клас, та бажаний стиль через `css`.

Для того, щоб створити смужки бургера через `span` робимо новий клас `burger_item`, який вже кастомізуємо. В середині необхідно прописати якийсь текст, який ми зробимо прозорим (щоб не забути, що це таке).

```
<button class="burger">
  <span class="burger_item">Menu</span>
</button>
</div>
/div>
```

```
497 }
498 .burger_item {
499   display: block;
500   width: 30px;
501   height: 1px;
502   background-color: #001f30;
503   font-size: 0;
504   color: transparent;
505   position: relative;
506 }
```



Тепер нам потрібно зробити ще дві смужки: зверху нашого burger-Item та знизу. Можливо це зробити прописавши в `html` такі самі `span` елементи, але ми використаємо **псевдо-елементи**.

Псевдоелемент (pseudo-element) - елемент, який створюється не за допомогою HTML-тегу в розмітці сторінки, а за допомогою CSS.

Для цього до селектора приписуються конструкції `::before` або `::after`. Для фактичного відображення псевдоелемента на сторінці обов'язково потрібно вказати властивість вмісту в рамках блоку правил для вищеописаного селектора.

The image shows a browser window with a pet shop website. The website header includes a logo for 'PET SHOP' and several icons: a heart, a shopping cart, and a phone. Below the header is a large banner with the text 'МАГАЗИНИ "УЛЮБЛЕНЕЦЬ" КРАЩІ ПРОДУКТИ ДЛЯ КОХАНИХ'. The browser's developer tools are open, showing the HTML structure. A `span` element with class `'burger_item'` is highlighted. The CSS rule for this element is shown as follows:

```

...
::before == $0
    "Menu"
    ::after
    </span>
  </button>
</div>

```

Для того, щоб текст всередині span не відображався пишемо:

font-size:0;
color: transparent;

Для того, щоб наші псевдо елементи можливо було позиціонувати відносно до основного елемента ми задаємо їм

position: absolute;
z-index: 1;

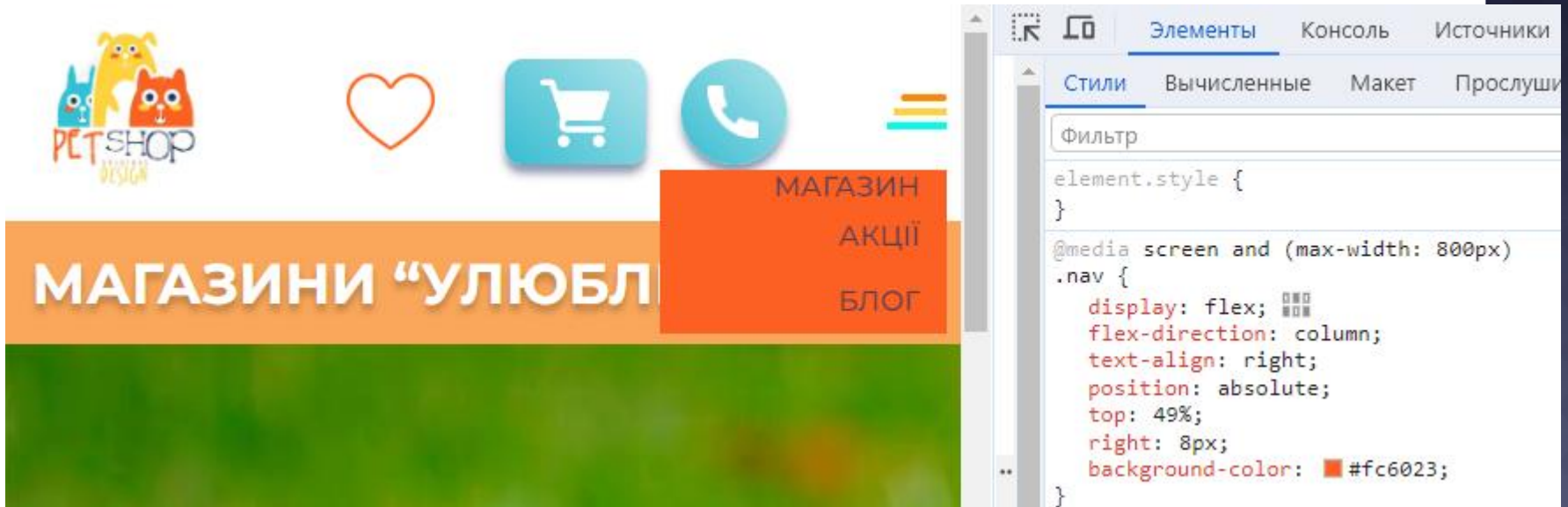
Та не забуваємо при цьому надати основному елементу **position: relative;**

```

498 .burger__item {
499     display: block;
500     width: 30px;
501     height: 4px;
502     background-color: #FFD23E;
503     font-size: 0;
504     color: transparent;
505     position: relative;
506 }
507 .burger__item::before {
508     content: "#";
509     width: 26px;
510     height: 100%;
511     top: -8px;
512     position: absolute;
513     right: 0;
514     z-index: 1;
515     background-color: #ff8801;
516     border-radius: 15px;
517 }
518 .burger__item::after {
519     content: "#";
520     width: 34px;
521     height: 100%;
522     bottom: -8px;
523     position: absolute;
524     right: 0;
525     z-index: 1;
526     background-color: #00ffea;
527 }

```

Тепер давайте застилізуємо саме випадаюче меню, яке буде нам показано при натисканні на бургер. Для цього, спочатку, покажемо це меню при медіазапиті (бо зараз воно сховане). Та додамо йому деяких властивостей.



- Для того, щоб меню встало вертикально → **display: flex;**
- Вирівнювання в середині блоку. → **flex-direction: column;**
- Позиція абсолют відносно header → **text-align: right;**
- Вирівнювання відносно header → **position: absolute;**
- **top: 50%;**
- **right: 15px;**
- **background-color: #ee5656;**


```

521
522 .nav_link {
523     padding: 9px 15px;
524 }
525

```

Обов'язково треба допрацювати **nav_link** тобто самі пункти меню, для того, щоб ними було зручно користуватися в мобільній версії. Тобто в них має бути відступ при один пункт меню від іншого при медіа запиті.

Після того, як ви зробили випадаюче меню, його потрібно сховати **display: none**

Та зробити так, щоб при натисканні воно відображалося. Робиться це за допомогою JS та методів JQuery.

Для початку в html надаємо ID нашій навігації та кнопці, за допомогою якого ми будемо відстежувати натискання на цю кнопку



```

<nav class="nav" id="nav">
  <div class="dropdown">
    <a class="nav_link" href="#" >Магазин</a>
    <div class="dropdown_content">
      <a href="#" >КОРМА</a>
      <a href="#">ГИГИЕНА</a>
      <a href="#">ОДЕЖА</a>
      <a href="#">КЛИТКИ</a>
    </div>
  </div>
</nav>

```



```

<button class="burger" type="button" id="navToggle">
  <span class="burger_item">Menu</span>
</button>

```

Відкриваємо новий розділ в Java Script, пишемо коментар та відкриваємо функцію.

Далі, зберігаємо першу варіаційну для того, щоб зафіксувати **ID nav**

```
let nav = $("#nav");
```

Та другу для

```
let navToggle = $("#navToggle");
```

Потім для змінної **navToggle**

встановлюємо дію по **on("click")**

Також, на випадок, якщо кнопка у вас прописана не через **button** а через тег **<a>**, **прибираємо** поведінку за замовчуванням.

```

28
29      /*Navigation*/
30      $(function () {
31          let nav = $("#nav");
32          let navToggle = $("#navToggle");
33
34          navToggle.on("click", function (event) {
35              event.preventDefault();
36
37              nav.toggleClass("show");
38          });
39      });
40

```

Наступний рядок:

Ми хочемо, щоб при натисканні у нас показувалося меню, а при другому натисканні – прибиралося. Ми застосуємо метод **ToggleClass**. При цьому методі ми встановимо клас для події. Тепер при натисканні на бургер у класу **nav** з'явиться додатковий клас **show** Це можливо побачити в інспекторі коду. Знайдіть клас та натискайте.

Залишилося тільки прописати поведінку для **.nav.show** в медіа запиті **css**.

```

518     .nav.show {
519         display: flex;
520     }
521

```