

**Мова HTML.  
Створення галереї  
засобами CSS.**

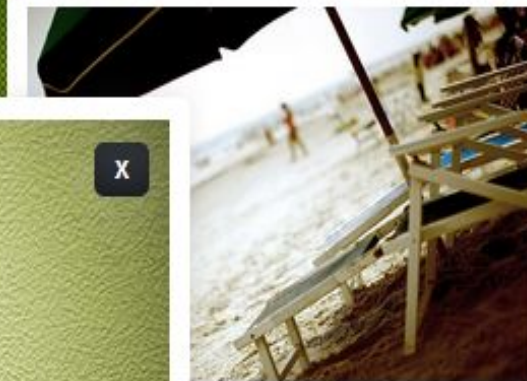
# Створення галереї

- Вставляти великі зображення на сайт - це не практично, так як це займає досить багато місця, а якщо занадто маленькі, то не буде зрозуміло, що на зображенні, тому доводиться йти на компроміс.
- На занятті ми розглянемо, як створити просте збільшення зображень за допомогою **CSS**.

## Створення галереї

- Необхідно, щоб після натискання на зображення воно збільшувалося, крім цього додамо кнопку закриття зображення

# Галерея



# Крок 1. Мова HTML

## СТРУКТУРА:

- Є загальний контейнер, який містить всю структуру, в середину цього контейнера поміщені дочірні елементи, як зображення і класи закриття розгорнутих зображень.
- Крім цього ми використовуємо клас: `target` який забезпечить працездатність вікон.



# Крок 1. Мова HTML

```
<div id="images-box">
  <div class="holder">
    <div id="image-1" class="image-lightbox">
      <span class="close"><a href="#">X</a></span>
      
      <a class="expand" href="#image-1"></a>
    </div>
  </div>
  <div class="holder">
    <div id="image-2" class="image-lightbox">
      <span class="close"><a href="#">X</a></span>
      
      <a class="expand" href="#image-2"></a>
    </div>
  </div>
  <div class="holder">
    <div id="image-3" class="image-lightbox">
      <span class="close"><a href="#">X</a></span>
      
      <a class="expand" href="#image-3"></a>
    </div>
  </div>
</div>
```

## Крок 2. CSS Базові стилі

```
/* * Общая ширина изображения контейнера, в основном, для центрирования * */
width: 850px;
margin: 0px auto;
position: relative;
top: 70px;
}

.image-lightbox img {
width: inherit;
height: inherit;
z-index: 3000;
}

(holder {
/* Ширина и высота, вы можете их изменить * /
width: 250px;
height: 166px;
float: left;
margin: 0 20px 0 0;
}
```

# position static

```
.static {  
  position: static;  
}
```

CSS

`<div class="static">`

`static` является значением по умолчанию. Элемент с `position: static;` отображается как обычно и не будет позиционироваться каким-либо специальным способом, он просто плюхнется вниз.

`</div>`



# position

## relative

```
.relative1 {  
  position: relative;  
}  
.relative2 {  
  position: relative;  
  top: -20px;  
  left: 20px;  
  background-color: white;  
  width: 500px;  
}
```

CSS

```
<div class="relative1">
```

`relative` ведет себя в точности как `static` но лишь до тех пор, пока вы не добавите некоторые дополнительные свойства.

```
<div class="relative2">
```

Добавление свойств `top`, `right`, `bottom`, и `left` изменяет позицию элемента, сдвигая его в ту или иную сторону относительно его начального расположения. Другой контент не будет перестраиваться, чтобы заполнить любое свободное пространство, оставленное этим элементом.

```
</div>
```

# Position fixed

- Фіксоване позиціювання елемента щодо вікна браузера таке, що він завжди займає одне і теж місце навіть при прокручуванні сторінки, немов він приклеєний до екрану. Також як і для **relative**, додаткові властивості **top**, **right**, **bottom**, і **left** для даного позиціонування працюватимуть.

# Position fixed

- ```
fixed {  
  position: fixed;  
  bottom: 0;  
  right: 0;  
  width: 200px;  
  background-color: white;  
}
```
- Фіксований елемент не залишає за собою вільний простір на сторінці, де він був розташований.
- У мобільних браузерів досить хитка підтримка фіксованого позиціонування.

```
<div class="fixed">
```

Привет! Не обращай  
на меня внимания.

```
</div>
```

# Position absolute

- absolute поводитьься як fixed за винятком того, що позиціювання відраховується щодо найближчого розташованого родича, а не щодо вікна браузера.
- Якщо елемент абсолютно позиціонується і не має родича, то він використовує тіло документа і продовжує рухатись разом з усією сторінкою при скролінгу.
- Пам'ятайте, що "позиційований" елемент - це елемент з будь-яким значенням властивості position крім значення static.

# Position absolute

CSS

```
.relative {  
  position: relative;  
  width: 600px;  
  height: 400px;  
}  
.absolute {  
  position: absolute;  
  top: 120px;  
  right: 0;  
  width: 300px;  
  height: 200px;  
}
```



# Position absolute

```
<div class="relative">
```

Этот элемент относительно позиционированный. Если бы элемент имел значение `position: static;` то его абсолютно позиционированный ребенок убежал бы и стал позиционироваться уже относительно тела документа.

```
<div class="absolute">
```

Этот элемент абсолютно позиционированный. Он позиционирован относительно его родителя.

```
</div>
```

```
</div>
```

# position у дії

```
.container {  
  position: relative;  
}  
nav {  
  position: absolute;  
  left: 0px;  
  width: 200px;  
}  
section {  
  /* position is static by default */  
  margin-left: 200px;  
}  
footer {  
  position: fixed;  
  bottom: 0;  
  left: 0;  
  height: 70px;  
  background-color: white;  
  width: 100%;  
}  
body {  
  margin-bottom: 120px;  
}
```

CSS

# position у дії

```
<nav> class="container" >
```

- Home
- Taco Menu
- Draft List
- Hours
- Directions
- Contact

```
</nav>
```

```
<section>
```

Использование `margin-left` для `section`s освобождает место для размещения `nav`. В противном случае элементы со значением `absolute`

```
<section>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.

```
<section>
```

Обратите внимание на то, что произойдет, если вы измените размер вашего браузера. Это прекрасно работает!

# float (поплавок)

- CSS властивість використовується для розмітки.
- Поплавок призначений для обгортання тексту навколо зображень, наприклад:

```
img {  
  float: right;  
  margin: 0 0 1em 1em;  
}
```



# float (поплавок)

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, nisi lorem egestas odio, vitae scelerisque enim ligula venenatis dolor. Maecenas nisl est, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante ligula, facilisis sed ornare eu, lobortis in odio. Praesent convallis urna a lacus interdum ut hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed nunc venenatis imperdiet sed ornare turpis. Donec vitae dui eget tellus gravida venenatis. Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo purus. Mauris quis diam velit.





# Приклад float шаблону

**nav**

{

float: left; width: 200px;

}

**section**

{

margin-left: 200px;

}

# Приклад float шаблону

```
<nav class="clearfix">
```

- Home
- Taco Menu
- Draft List
- Hours
- Directions
- Contact

```
</nav>
```

```
<section>
```

Этот пример работает точно так же, как предыдущий. Обратите внимание, мы применяем `clearfix` бы `nav` был больше чем не всплывающий контент.

```
<section>
```

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Phasellus imperdiet, nulla et dictum interdum, ultrices nec congue eget, auctor vitae massa. Fusce luctus vestibulum augue ut aliquet. Mauris ante lig hendrerit risus congue. Nunc sagittis dictum nisi, sed ullamcorper ipsum dignissim ac. In at libero sed Integer fringilla congue eros non fermentum. Sed dapibus pulvinar nibh tempor porta. Cras ac leo pur

# z-index

- Якщо в одному місці сторінки виявляються кілька «абсолютних» блоків, то вони перекривають один одного. За замовчуванням вище виявляється той блок, який розташований далі в коді сторінки.

За допомогою CSS-властивості `z-index` можна керувати тим, як перекриваються блоки. Значенням властивості може бути ціле число. Чим більше `z-index`, тим вище розташовується блок.

Властивість `z-index` працює для елементів, у яких `position` задана як `absolute`, `fixed` і `relative`.

- Таким чином, «відносний» елемент може перекривати «абсолютний».

# z-index

- `<!DOCTYPE html>`
- `<html lang="ru">`
- `<head>`
- `<meta charset="utf-8">`
- `<title>z-index или кто кого перекроет</title>`
- `</head>`
- `<body>`
- `<div class="block block-1">Блок 1</div>`
- `<div class="block block-2">Блок 2</div>`
- `<div class="block block-3">Блок 3</div>`
- `</body></html>`

# z-index

- `body { margin: 0; padding: 20px;}`
- `.block { padding: 10px; text-align: right; color: white; opacity: 0.8;}`
- `.block-1 { width: 350px; height: 350px; background: #3a78a1;}`
- `.block-2 { position: absolute; top: 20px; width: 250px; height: 250px; background: #e74c3c;}`
- `block-3 { position: absolute; top: 20px; width: 150px; height: 150px; background: #27ae60;}`



# z-index

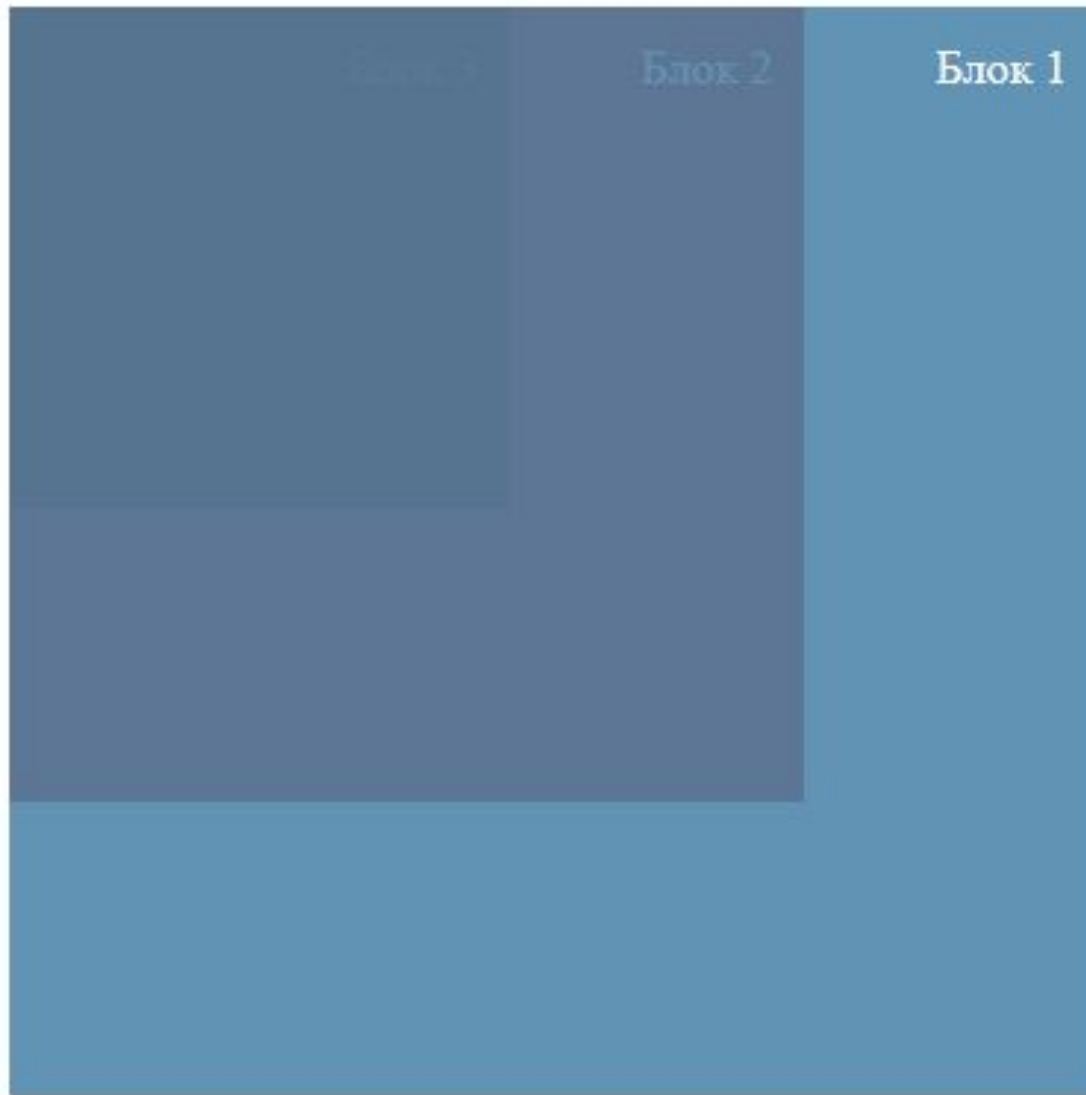


Цель 1 Задайте блоку 2 `z-index` со значением `100`.

Цель 2 Задайте блоку 1 относительное позиционирование

Цель 3 и `z-index` со значением `200`.

# z-index



## Крок 2. CSS Базові стилі

```
.image-lightbox {  
  
width: inherit;  
height: inherit;  
padding: 10px;  
/* Тени */  
box-shadow: 0px 0px 10px rgba(0,0,0,0.1);  
background: #fff;  
border-radius: 5px;  
position: absolute;  
top: 0;  
font-family: Arial, sans-serif;  
/* Переходи */  
-webkit-transition: all ease-in 0.5s;  
-moz-transition: all ease-in 0.5s;  
-ms-transition: all ease-in 0.5s;  
-o-transition: all ease-in 0.5s;  
}
```

# Значення inherit

```
15
16 html, body, div, span, applet, object, iframe
17 h1, h2, h3, h4, h5, h6, p, blockquote, pre,
18 a, abbr, acronym, address, big, cite, code,
19 del, dfn, em, font, ins, kbd, q, s, samp,
20 small, strike, strong, sub, sup, tt, var,
21 dl, dt, dd, ol, ul, li,
22 fieldset, form, label, legend
23 {
24     border: 0;
25     font-family: inherit;
26     font-size: 100%;
27     font-style: inherit;
28     font-weight: inherit;
29     line-height: normal;
30     margin: 0;
31     outline: 0;
32     padding: 0;
33     vertical-align: baseline;
34 }
```



# Значення inherit

- З англ. перекладається як “наслідувати”. Властивості, яким буде присвоєно це значення, будуть наслідувати значення тієї-ж властивості його родича.

```
<div style = "border: 1px solid # 999; padding: 10px;">
```

```
<div style = "border: inherit;"> Дочірній блок
```

```
</ div>
```

```
</ div>
```

# Властивість box-shadow

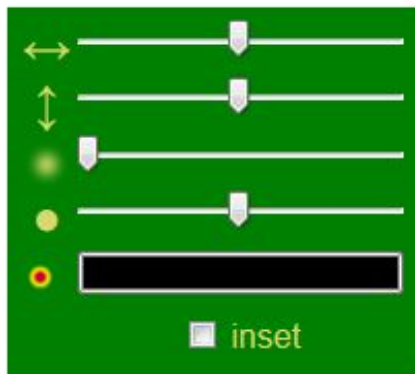


# Властивість box-shadow

# Властивість box-shadow

# Властивість box-shadow

# Властивість box-shadow



```
box-shadow: none;
```

смещение по горизонтали      размытие      цвет

***box-shadow: inset 8px 8px 10px 3px #000000;***

внутри элемента      смещение по вертикали      толщина

## Крок 2. CSS

- Встановили `<span>` елементи `display: none`, для того, щоб зробити кнопку закриття, яка з'являється, коли користувач натискає на зображення.



# Крок 2. CSS

```
. image-lightbox span {  
display: none;  
}  
.image-lightbox .expand {  
width: 100%;  
height: 100%;  
position: absolute;  
top: 0;  
z-index: 4000;  
background: rgba(0,0,0,0); /* Фиксы для IE */  
left: 0;  
}
```

```
.image-lightbox .close {  
position: absolute;  
width: 20px; height: 20px;  
right: 20px; top: 20px;  
}  
.image-lightbox .close a {  
height: auto; width: auto;  
padding: 5px 10px;  
color: #fff;  
text-decoration: none;  
background: #22272c;  
box-shadow: inset 0px 24px 20px -15px rgba(255, 255, 255, 0.1), inset 0px 0px 10px  
            rgba(0,0,0,0.4), 0px 0px 30px rgba(255,255,255,0.4);  
border-radius: 5px;  
font-weight: bold;  
float: right;  
}  
.image-lightbox .close a:hover {  
box-shadow: inset 0px -24px 20px -15px rgba(255, 255, 255, 0.01), inset 0px 0px 10px  
            rgba(0,0,0,0.4), 0px 0px 20px rgba(255,255,255,0.4);};
```

## Крок 3

- Необхідно додати позиціонування для зображень, якщо ви хочете додати нові, тоді необхідно внести доповнення до цього розділу:

```
div#image-1 { left: 0; }  
div#image-2 { left: 290px; }  
div#image-3 { left: 580px; }
```

# При відкритті зображення відбувається його позиціювання на сторінці

```
div[id^=image]:target {  
  width: 450px;  
  height: 300px;  
  z-index: 5000;  
  top: 50px;  
  left: 200px;  
}  
div[id^=image]:target .close {  
  display: block;  
}  
  
div[id^=image]:target .expand {  
  display: none;  
}  
  
div#image-1:target, div#image-2:target, div#image-3:target { left: 200px; }
```

## **Ітог**

**Жодного плагіну не  
було використано, це  
дозволяє збільшити  
швидкість  
завантаження  
сторінки**