



# JavaScript: Лекція 2

## Тонкощі CSS

---

ДІАНА СІРЕНКО

[diana@4enjoy.com](mailto:diana@4enjoy.com)

АНДРІЙ ЧАЙКА

[chaika@4enjoy.com](mailto:chaika@4enjoy.com)

# HTML5 vs HTML4

---

HTML5 це

- HTML4 (95%)
- Нові теги для семантики (section, header, aside, article)
- Нові теги для UI (progress, meter, time)
- Нові атрибути ~25шт
- Нові функція в JavaScript ~20шт
- Прибрані старі і нікому непотрібні теги та атрибути
- Ще близько 10 незначних змін

Вся різниця: <https://www.w3.org/TR/html5-diff/>





# Абсолютне позиціювання

---

ДІАНА СІРЕНКО

[diana@4enjoy.com](mailto:diana@4enjoy.com)

АНДРІЙ ЧАЙКА

[chaika@4enjoy.com](mailto:chaika@4enjoy.com)

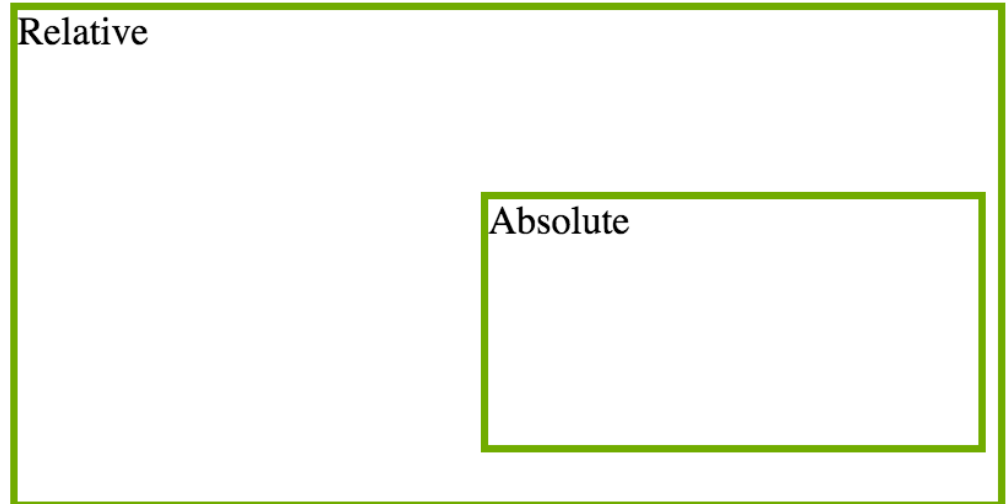
# Абсолютне позиціювання

```
.relative {  
  position: relative;
```

```
  width: 400px;  
  height: 200px;  
}
```

```
.absolute {  
  position: absolute;  
  bottom: 20px;  
  right: 5px;
```

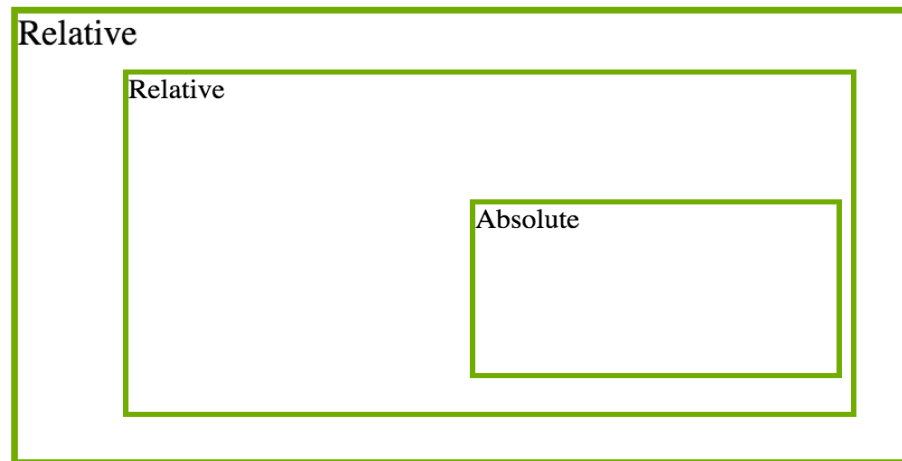
```
  width: 200px;  
  height: 100px;  
}
```



# position: absolute

---

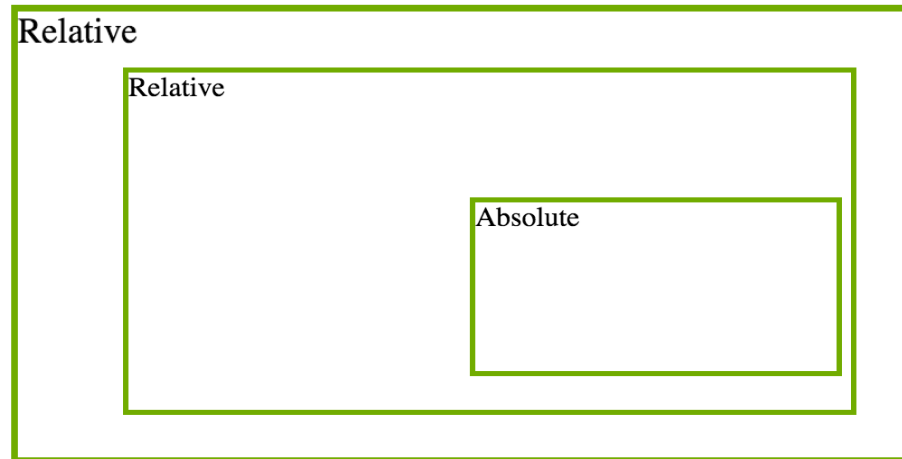
- Забирає елемент зі сторінки і розміщує у вільне плавання.
- Елемент розміщується поверх інших елементів на сторінці
- Координати елемента задаються за допомогою top, left, right, bottom
- Координати задаються відносно першого батька в якого встановлено position або **absolute** або **relative**. Якщо такого батька немає, тоді координати встановлюються відносно body.



# position: relative

---

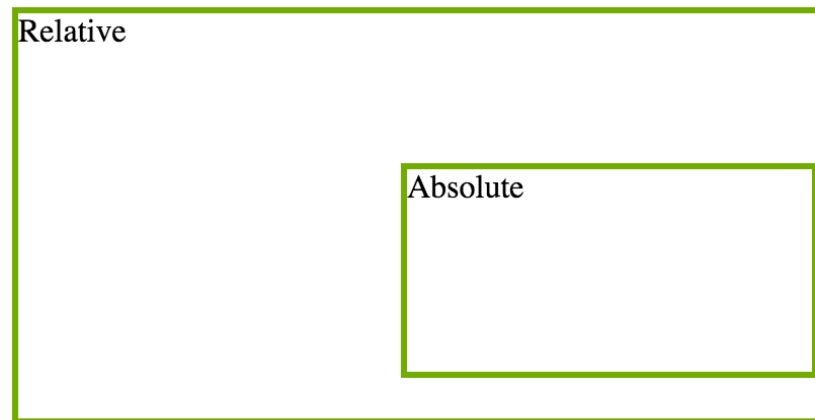
- НЕ змінює позицію елемента
- НЕ розміщує елемент поверх інших
- На елемент ніяк НЕ діють атрибути top, left, right, bottom
- **Він лише вказує, що всі синівські елементи з position: absolute; позицінуються відносно нього.**



# Top, Left, Right, Bottom

---

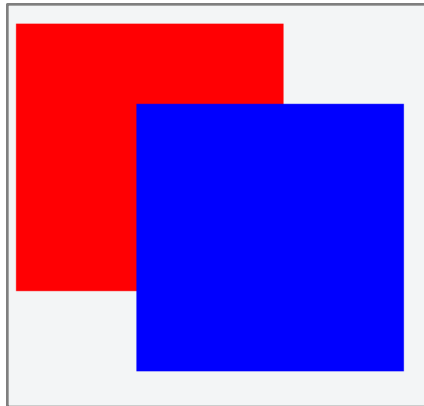
1. Не працюють якщо в елемента не стоїть `position: absolute`
2. Задаються в px, % та будь-яких інших одиницях
3. Можуть бути від'ємними
4. % залежать від висоти та ширини батьківського елемента із `position: relative`;



# Позиція по осі Z

Елементи із `position: absolute;` стають один на один, в тому порядку як вони були задані.

```
<div class="red"></div>
<div class="blue"></div>
```



```
.red, .blue {
  position: absolute;
  width: 100px;
  height: 100px;
  top: 40px;
  left: 50px;
  background: blue;
}

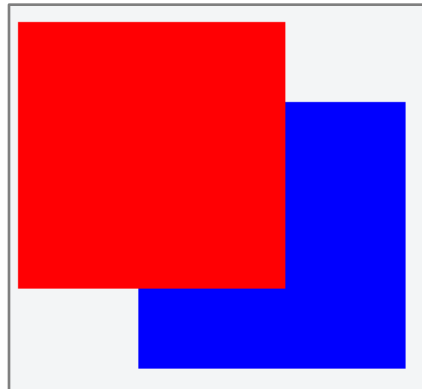
.red {
  background: red;
  top: 10px;
  left: 5px;
}
```



# Позиція по осі Z

z-index дозволяє міняти порядок в якому відображаються елементи

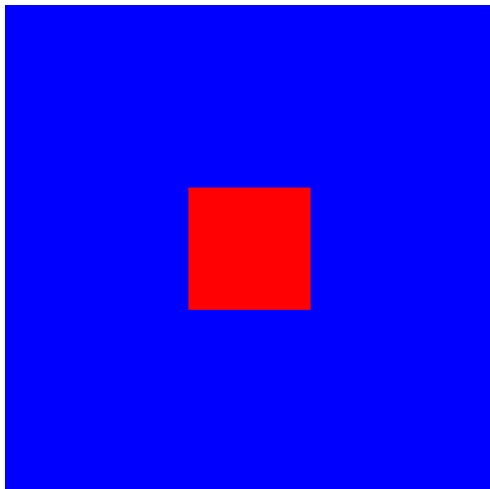
```
<div class="red"></div>
<div class="blue"></div>
```



```
.red, .blue {
  position: absolute;
  width: 100px;
  height: 100px;
  top: 40px;
  left: 50px;
  background: blue;
}
.red {
  background: red;
  top: 10px;
  left: 5px;
  z-index: 3;
}
```

# Позицювання по центру

```
<div class="blue">
  <div class="red"></div>
</div>
```



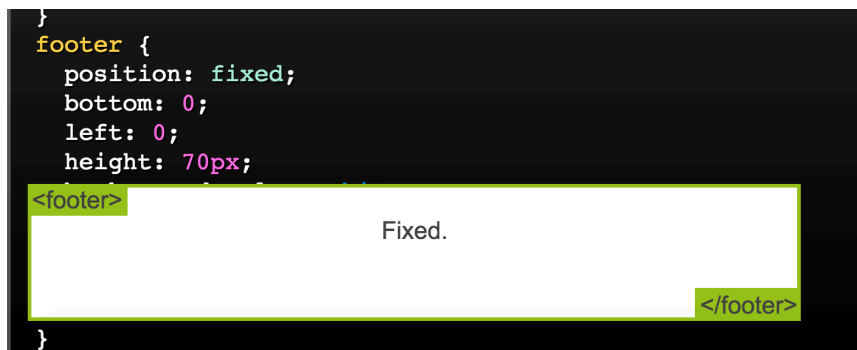
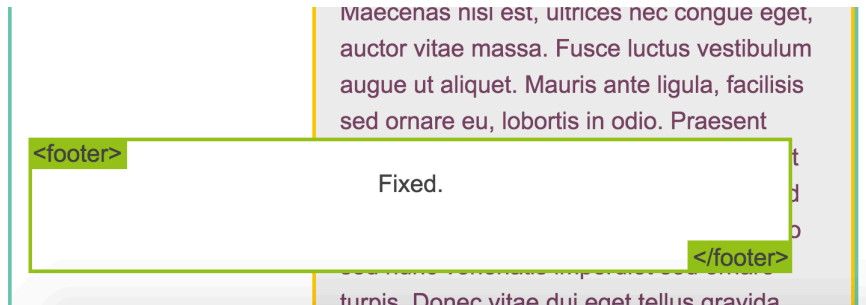
```
.blue{
  position: relative;
  width: 200px;
  height: 200px;
  background: blue;
}
```

```
.red {
  position: absolute;
  width: 50px;
  height: 50px;
  background: red;

  top: 0;
  bottom: 0;
  right: 0;
  left: 0;
  margin: auto
}
```

# position: fixed;

`position: fixed`, схожий на `position: absolute`, але прикріплює елементи відносно вікна а не сторінки. Такі елементи залишаються на екрані навіть під час скролу.



```
.footer {  
  position: fixed;  
  bottom: 20px;  
  left: 20px;  
  
  height: 70px;  
  background-color: white;  
  width: 85%;  
}
```



# Псевдо елементи ::before, ::after

---

ДІАНА СІРЕНКО

[diana@4enjoy.com](mailto:diana@4enjoy.com)

АНДРІЙ ЧАЙКА

[chaika@4enjoy.com](mailto:chaika@4enjoy.com)

# ::before, ::after

<h1>Sed ut perspiciatis</h1>

```
h1 {
  color: blue;
}
```

 Sed ut perspiciatis 

Iste natus error sit voluptatem accusantium  
doloremque laudantium, totam rem aperiam,  
eaque ipsa quae ab illo inventore veritatis

```
h1::before {
  content: "";
  display: inline-block;
  background: url("../img_before.png");
  height: 20px;
  width: 12px;
  margin-right: 5px;
}
h1::after {
  content: "";
  display: inline-block;
  background: url("../img_after.png");
  height: 20px;
  width: 12px;
  margin-left: 5px;
}
```

<http://pica-ae.deviantart.com/art/CSS-Tricks-Before-and-After-427267080>

# ::before, ::after

<h1>Sed ut perspiciatis</h1>

```
▼ <h1>
  ::before
  "Sed ut perspiciatis"
  ::after
</h1>
```

h1::before | 12 × 20



```
h1::before {
  content: "";
  display: inline-block;
  background: url("../img_before.png");
  height: 20px;
  width: 12px;
  margin-right: 5px;
}

h1::after {
  content: "";
  display: inline-block;
  background: url("../img_after.png");
  height: 20px;
  width: 12px;
  margin-left: 5px;
}
```

<http://pica-ae.deviantart.com/art/CSS-Tricks-Before-and-After-427267080>



# Псевдо класи

---

ДІАНА СІРЕНКО

[diana@4enjoy.com](mailto:diana@4enjoy.com)

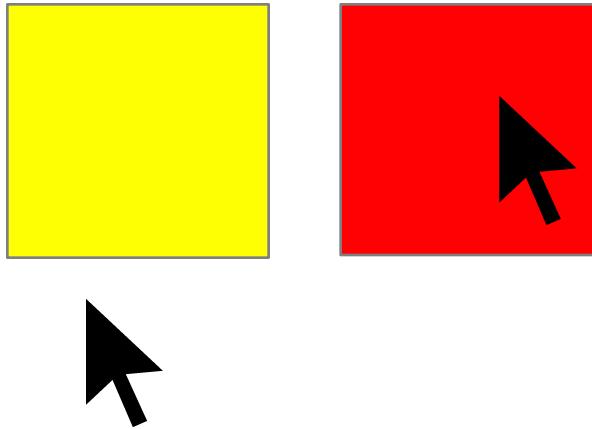
АНДРІЙ ЧАЙКА

[chaika@4enjoy.com](mailto:chaika@4enjoy.com)

# :hover

---

```
<div class="click-me"></div>
```



```
.click-me {  
  width: 100px;  
  height: 100px;  
  background: yellow;  
}
```

```
.click-me:hover {  
  background: red;  
}
```



# Псевдо класи

---

Псевдо класів 20+. Приклади:

[http://www.w3schools.com/css/css\\_pseudo\\_classes.asp](http://www.w3schools.com/css/css_pseudo_classes.asp)

## Найкращі

- :hover
- :active
- :focus
- :first-child / :last-child
- :nth-child(n)



# Clearfix

---

ДІАНА СІРЕНКО

[diana@4enjoy.com](mailto:diana@4enjoy.com)

АНДРІЙ ЧАЙКА

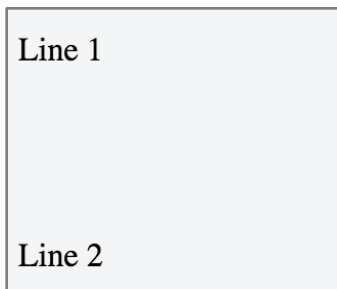
[chaika@4enjoy.com](mailto:chaika@4enjoy.com)

# Невидимі елементи

```
<p>Line 1</p>
<div class="hidden">Text</div>
<p>Line 2</p>
```

## display:none;

```
.hidden {
  height: 50px;
  background: red;
  display: none;
}
```



## visibility:hidden;

```
.hidden {
  height: 50px;
  background: red;
  visibility: hidden;
}
```



# Проблема

```
<div class="item">
   Text text text
</div>
```

```
img {
  float: right;
}
```

<div>

О нет... эта картинка больше чем элемент содержащий ее, поэтому она вышла за пределы своего контейнера, буквально переполнив его снаружи!



<http://ru.learnlayout.com/clearfix.html> Картинка на сайті класна, але clearfix поганий

# .clearfix

---

```
.clearfix::after {  
  visibility: hidden; //Робить елемент невидимим  
  display: block;  
  content: ""; //Порожнім  
  height: 0; //Без висоти  
  clear: both; //clear забороняє обтікати (float)  
                  //цей елемент з обох сторін  
}
```

# .clearfix

```
<div class="item clearfix">
   Text text text
</div>
```

```
img {
  float: right;
}
```

<div>

О нет... эта картинка больше чем элемент содержащий ее, поэтому она вышла за пределы своего контейнера, буквально переполнив его снаружи!





# box-sizing міняємо box model

---

ДІАНА СІРЕНКО

[diana@4enjoy.com](mailto:diana@4enjoy.com)

АНДРІЙ ЧАЙКА

[chaika@4enjoy.com](mailto:chaika@4enjoy.com)

# Чому не в один рядок?

```
<div class="A">AAAA</div>
<div class="B">BBBB</div>
```

```
.A, .B {
  width: 60%;
  float: left;
  background: red;

  padding: 10px;
}
```

```
.B {
  width: 40%;
  background: green;
}
```





# box-sizing

```
<div class="A">AAAA</div>
<div class="B">BBBB</div>
```

```
.A, .B {
  width: 60%;
  float: left;
  background: red;
```

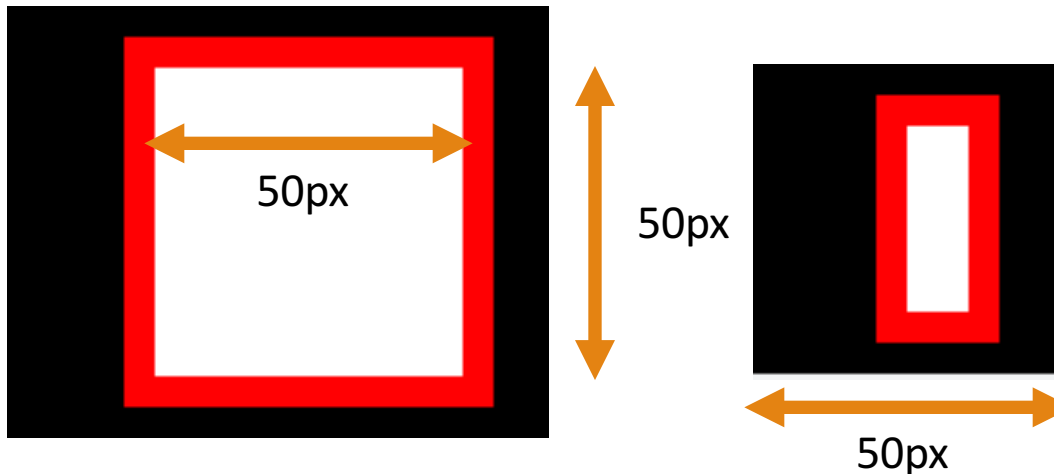
```
  padding: 10px;
  box-sizing: border-box;
}
```

```
.B {
  width: 40%;
  background: green;
}
```



# box-sizing

```
<div class="A"></div>
<div class="B"></div>
```



```
.A, .B {
  width: 50px;
  height: 50px;
  border: 5px solid black;
  border-left-width: 20px;
  border-right-width: 10px;
  padding: 5px;
  background: red;
}

.B {
  box-sizing: border-box;
}
```



# Media Queries

---

ДІАНА СІРЕНКО

[diana@4enjoy.com](mailto:diana@4enjoy.com)

АНДРІЙ ЧАЙКА

[chaika@4enjoy.com](mailto:chaika@4enjoy.com)

# @media

---

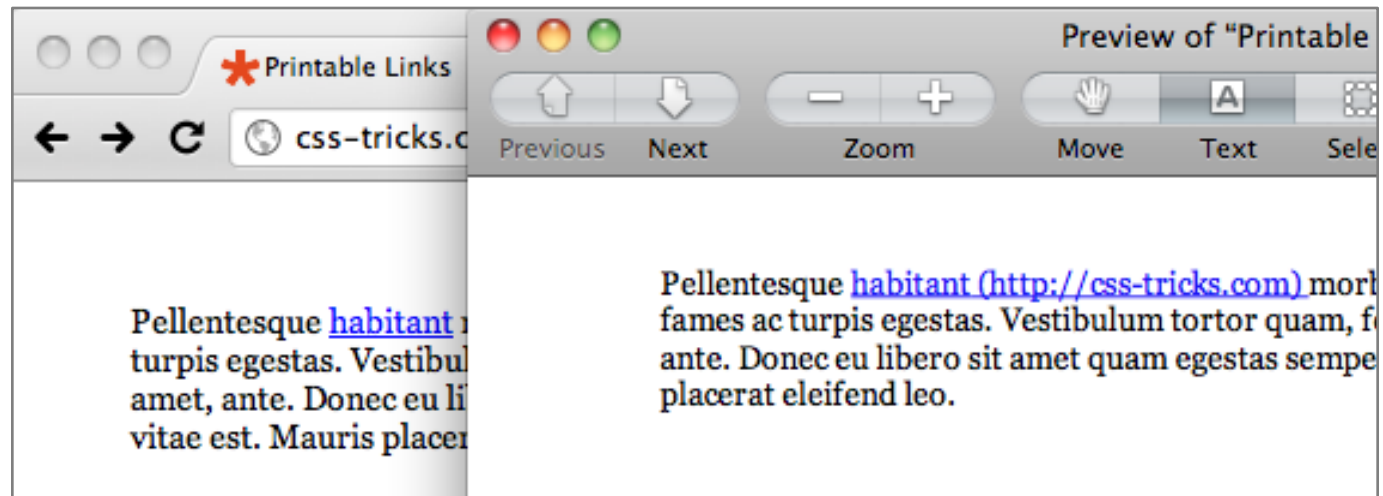
```
.top-panel {  
  font-size: 20px; //Базовий розмір шрифту  
}
```

```
@media (max-width: 800px) {  
  .top-panel {  
    font-size: 15px; //Розмір для екранів менше 800px  
  }  
}
```

```
@media (max-width: 400px) {  
  .top-panel {  
    font-size: 10px; //Розмір для екранів менше 400px  
  }  
}
```

# @media print

```
@media print {
  a::after {
    content: " (" attr(href) ") ";
  }
}
```





# CSS Transform

---

ДІАНА СІРЕНКО

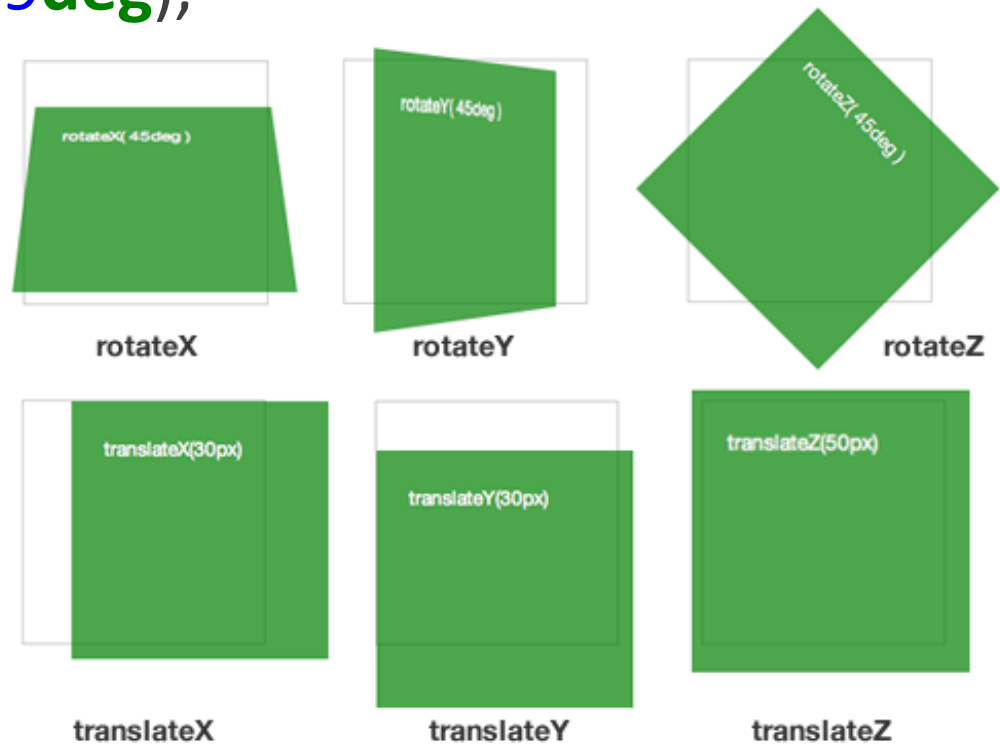
[diana@4enjoy.com](mailto:diana@4enjoy.com)

АНДРІЙ ЧАЙКА

[chaika@4enjoy.com](mailto:chaika@4enjoy.com)

# Transform

```
div {  
  transform: rotate(9deg);  
}
```



# Transform

---

**Не правильно!**

```
div {  
  transform: rotate(9deg);  
  transform: scale(1.2);  
}
```

**Правильно!**

```
div {  
  transform: rotate(9deg) scale(1.2);  
}
```

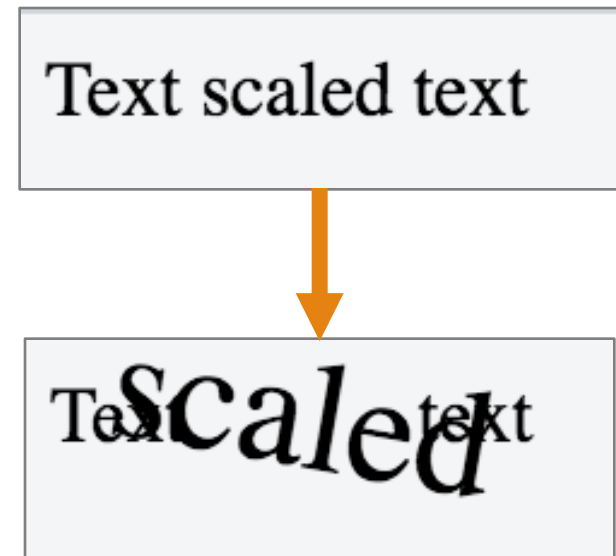


# Transform не міняє Layout

Text <span>scaled</span> text

```
span {  
  display: inline-block;  
  transform: scale(2) rotate(10deg);  
}
```

Зміна transform не потребує  
перерахунку layout на всій  
сторінці (margin, padding), через  
це це найкращий інструмент  
для анімацій





# CSS Transition

---

ДІАНА СІРЕНКО

[diana@4enjoy.com](mailto:diana@4enjoy.com)

АНДРІЙ ЧАЙКА

[chaika@4enjoy.com](mailto:chaika@4enjoy.com)

# transition

---

```
<div></div>
```

```
div {  
  background: blue;  
  width: 200px;  
  height: 200px;  
  transition: all 1s ease;  
}
```

```
div:hover {  
  background: orange;  
  border-radius: 100px;  
}
```

