

**ОСНОВИ ПРОГРАМУВАННЯ
ДЛЯ ШКОЛЯРІВ**

**ГРУПА
PRO**

**СТВОРЕННЯ САЙТУ
HTML ТА CSS**

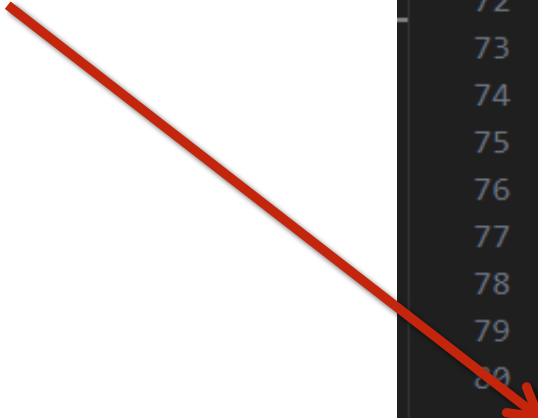
**АНІМАЦІЯ
ЧАСТИНА 2**

animation-delay

Властивість затримує відтворення анімації. Значенням може бути будь-яке число, як негативне, і позитивне.

Якщо значення позитивне, буде затримка перед початком анімації. Якщо значення негативне, то анімація почнеться за кадром.

Нехай анімація квадрата розпочнеться із затримкою -2.5 секунди. Так вона розпочнеться з середини, бо вся наша анімація триває 5 секунд.



```
69  .circle {
70      animation-name: circle-to-patrat;
71      animation-duration: 5s;
72      animation-iteration-count: infinite;
73      animation-direction: alternate;
74  }
75
76  .patrat {
77      animation-name: circle-to-patrat;
78      animation-duration: 5s;
79      animation-iteration-count: infinite;
80      animation-direction: alternate-reverse;
81      animation-delay: -2.5s;
82  }
83
```

animation-play-state

Властивість, що дозволяє ставити анімацію на паузу та запускати знову.

Доступні значення:

running - анімація програватиметься (значення за промовчанням).

paused – анімація ставиться на паузу. При повторному запуску анімації вона продовжується з того місця, де була зупинена.

Додамо можливість взаємодіяти із нашою анімацією. Нехай за наведенням курсору анімація ставиться на паузу, а якщо курсор прибраний, то продовжує програватися.

```
}  
.circle2:hover {  
  animation-play-state: paused;  
}
```

Таким чином ми можемо зробити декілька цікавих ефектів при взаємодії з елементом.

animation - це мега-шорткат, в якому можна за раз вказати значення для всіх перерахованих властивостей, що починаються на **animation-**.

Значення вказуються через пропуск. Порядок переліку значень не є важливим. Через те, що значення цих властивостей дуже різні, браузер сам здогадується, яке значення якої властивості належить. Важливо тільки пам'ятати, що перше значення часу буде сприйнято як **значення animation-duration (тривалість анімації)**, а друге — **animation-delay (затримка відтворення)**.

Для роботи анімації зовсім не обов'язково перераховувати всі значення. Достатньо вказати ім'я анімації та її тривалість. Для інших властивостей буде встановлено значення за замовчуванням.

```
.patrat {
  animation-name: circle-to-patrat;
  animation-duration: 5s;
  animation-iteration-count: infinite;
  animation-direction: alternate-reverse;
  animation-delay: -2.5s;
}
```



```
.patrat {
  animation: circle-to-patrat 5s infinite
  alternate-reverse -2.5s;
}
```

Декілька анімацій.

Є можливість застосувати до одного елемента кілька анімацій. Для цього потрібно перерахувати кілька значень через кому. Можна вказати будь-яку кількість значень для будь-якої з властивостей анімації. Анімації відтворюватимуться одночасно.

Наприклад, зробимо, щоб наш круг змінював розмір та змінював колір окремо. Для цього одна анімація буде написана для зміни розмірів, а друга для зміни кольору. Обі дві анімації присвоюються об'єкту через кому. При чому час виконання анімацій різний.



```
.circle2 {
  animation:
    circle2-to-block 10s infinite alternate 1s,
    color-change 5s alternate linear infinite;
}
```

```
91
92 .circle2 {
93   width: 100px;
94   height: 100px;
95   background-color: palevioletred;
96   border-radius: 50%;
97 }
98 @keyframes circle2-to-block {
99   from {
100     width: 50px;
101     height: 100px;
102     border-radius: 50%
103   }
104   50% {
105     width: 100%;
106     height: 100px;
107     border-radius: 50px;
108   }
109   to {
110     width: 100%;
111     height: 100%;
112     border-radius: 10px;
113   }
114 }
115 @keyframes color-change {
116   from {
117     background-color: #F498AD;
118   }
119   50% {
120     background-color: #7F6EDB;
121   }
122   to {
123     background-color: #2E9AFF;
124   }
125 }
```

animation-timing-function

CSS-анімації за замовчуванням програваються лінійно, змінні властивості елемента на рівні долі в рівні проміжки часу. Така поведінка рідко зустрічається в реальному житті. Якщо кинути м'яч на підлогу, то він спочатку повільно впаде, а потім досить швидко відскочить. Тобто його рух прискориться.

За допомогою властивості **animation-timing-function** можна задати, як буде розвиватися анімація між ключовими кадрами: рівномірно, або спочатку швидко, потім повільно, або за якими-то складними внутрішніми законами.

Доступні значення:

linear

Анімація програється рівномірно, без коливань швидкості.

ease

Значення за умовчанням. Анімація починається повільно, потім швидко розгоняється і знову замирає до кінця.

ease-in

Анімація починається повільно і плавно прискорюється до кінця.

ease-out

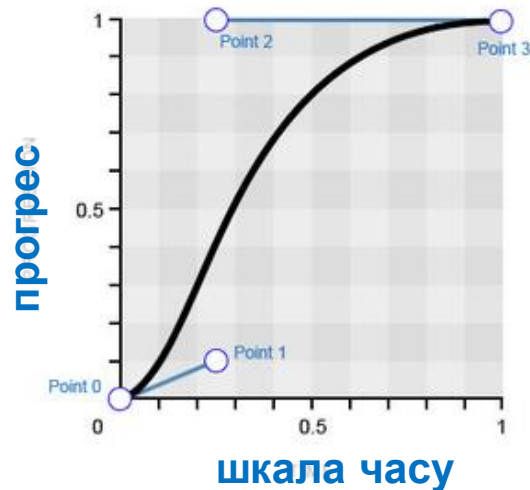
Анімація починається швидко і плавно замирає до кінця.

ease-in-out

Анімація починається і закінчується повільно, ускоряючись в середині.

cubic-bezier(x1, y1, x2, y2)

Часова функція, що описує тип прискорення у вигляді кривої Без'є.



По осі **x** розміщена **шкала часу** анімації, а по осі **y** — **прогрес анімації**. Це дуже потужний інструмент для створення різноманітних анімацій зі складними внутрішніми законами.

Значення x_1 і x_2 повинні знаходитися в діапазоні від 0 до 1 включно. Якщо присвоїти значення y_1 і y_2 менше 0 або більше 1, можна отримати ефект пружини.

Рідко, коли розробники пишуть цю функцію з голови. Частіше всього використовується інструмент візуалізації, який дозволяє змінювати значення і відразу бачити, як буде виглядати анімація.

Один із самих популярних інструментів — cubic-bezier.com.

step-start

Задає пошагову анімацію, розбиваючи її на відрізки, зміни проходять на початку кожного кроку.

step-end Пошагова анімація, зміни відбуваються в кінці кожного кроку.

steps(кількість кроків, положення кроку)

Функція, яка вказує на те, що анімація повинна відтворюватися кроками, які різко переходять від одного стану до іншого.

Перший параметр вказує, на скільки відрізків потрібно розбити анімацію. Значенням повинно бути ціле позитивне число більше 0.

Другий параметр є необов'язковим і вказує позицію кроку, моменту, коли починається анімація. **Можливі значення:**

jump-start — перший крок відбувається при значенні 0.

jump-end — останній крок відбувається при значенні 1.

jump-none — всі кроки проходять в межах від 0 до 1 включно.

jump-both — перший крок відбувається при значенні 0, останній — при значенні 1.

start — поводить себе як **jump-start**.

end — поводить себе як **jump-end**.

Зі значенням **start** анімація починається на початку кожного кроку, за значенням **end** — в кінці кожного кроку із затримкою. Затримка вираховується як результат ділення часу анімації на кількість кроків. Якщо другий параметр не вказано, використовується значення за замовчанням **end**.

animation-fill-mode

Властивість, яка повідомляє браузеру, чи потрібно застосовувати стилі ключових кадрів до або після програвання анімації.

Доступні значення:

none — стилі анімації не застосовуються до і після відтворення анімації (значення за замовчуванням).

forwards – після закінчення анімації елемент збереже стилі останнього ключового кадру.

backwards — після закінчення анімації до елемента буде використано стилі першого ключового кадру.

both — до початку анімації до елемента застосовується перший ключовий кадр, а після закінчення анімації елемент залишиться у стані останнього ключового кадру.