Тема 8. Створення CSS-АНІМІЦІЇ.

Вам потрібні певні файли для виконання завдання. Завантажте архів на комп'ютер і перегляньте його вміст. Щоб не витрачати час на написання коду, який не відноситься безпосередньо до теми цього уроку, ми заздалегідь підготували для вас HTML-розмітку та базові стилі CSS. Вам потрібно буде додати тільки ті стилі, які стосуватимуться анімованих об'єктів.

План роботи

Для чотирьох точок на карті (рис.8.1) створити пульсуючу анімацію для привернення уваги.

При наведенні курсору на точку міста анімація має ставати на паузу. Коли курсор прибрано, анімація відновлюється.

Також при наведенні курсору на точку міста має плавно з'являтися невелика виноска з місцем та датою концерту, кнопкою для покупки квитків. Коли курсор прибрано, інформаційна виноска зникає.

Колір кнопки у виносці повинен змінюватися при наведенні курсору.

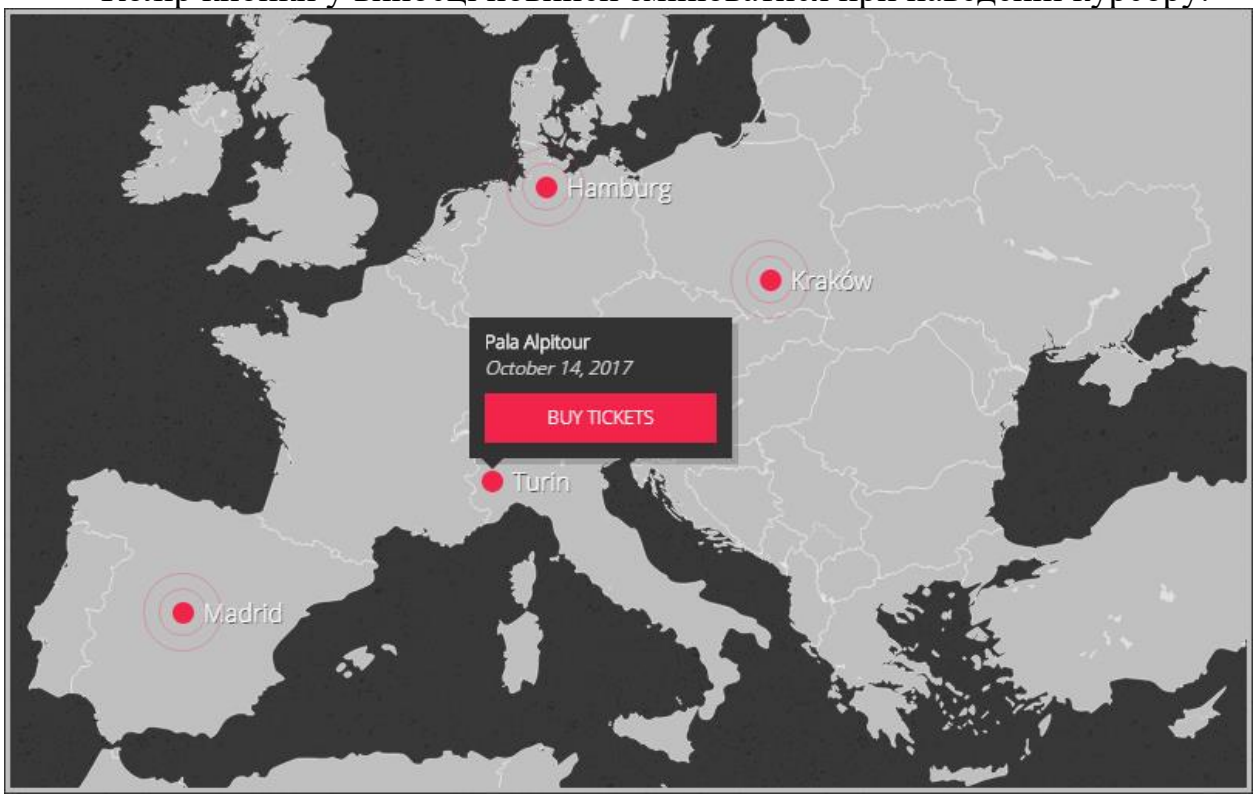

Рисунок 8.1 Пульсуюча анімація для привернення уваги

1. Анімація для позначок на карті

Відкрийте веб-сторінку зі скачаного архіву в браузері. Перед вами блок із картою, на якій нанесено чотири позначки із назвами міст. Наше завдання — створити ефект хвиль, що розходяться навколо кожної точки. Ми зробимо дві хвилі, у ролі яких виступатимуть псевдоелементи :before і :after:

Псевдоелементи зручно використовувати, коли потрібно додати якусь прикрасу до основного елементу. Це позбавляє необхідності додавати зайвий HTML-код, який не несе особливого сенсу.

У редакторі коду відкрийте документ style.css з папки css.. Знайдіть селектор .pin .popover і додайте над ним такі стилі для псевдоелементів елемента .pin:

```
.pin:before, .pin:after {
   content: '';
   position: absolute;
   left: 50%;
   top: 50%;
   display: block;
   border-radius: 50%;
   border: 1px solid #f3244a;
   width: 0;
   height: 0;
   margin-left: -2px;
   margin-top: -2px;
}
```

Ми не додаємо жодного контенту до цих псевдоелементів, а лише створюємо порожній блок з рамкою та закругленими кутами, щоб у результаті отримати кільце.

У початковому вигляді ширина та висота псевдоелементів дорівнює нулю. І за допомогою анімації ми збільшимо їх розміри, щоб створити ефект хвиль, що розходяться. Для цього потрібно створити ключові кадри для кожної хвилі.

Знайдіть на початку таблиці стилів коментар /* Keyframes */ і запишіть під ним два наступні правила @keyframes:

```
@keyframes pinBeforeWave {
   from {
      width: 0;
      height: 0;
      margin-left: -2px;
      margin-top: -2px;
   }
   to {
      width: 40px;
      height: 40px;
      margin-left: -21px;
      margin-top: -21px;
      opacity: 0;
   }
}
```

```
@keyframes pinAfterWave {
  from {
    width: 0;
    height: 0;
    margin-left: -2px;
    margin-top: -2px;
  }
  to {
    width: 66px;
    height: 66px;
    margin-left: -34px;
    margin-top: -34px;
    opacity: 0;
  }
}
```

Обидві анімації мають однаковий початок, але відрізняються закінченням. Анімацію pinBeforeWave ми застосовуємо до псевдоелементу: before, і в процесі її виконання він збільшиться до розмірів 40×40 пікселів (маленька хвиля). Анімація pinAfterWave призначена для псевдоелемента: after, до кінця якої він прийме розміри 66×66 пікселів (велика хвиля).

Крім цього, в обох анімаціях передбачено зміщення елементів на певну кількість пікселів ліворуч і вгору — це потрібно для того, щоб при збільшенні кільця позначка міста залишалася візуально в його центрі. І, нарешті, властивість opacity в останньому ключовому кадрі означає, що обидві хвилі будуть плавно зникати в процесі свого розширення.

Застосуємо створені анімації до псевдоелементів. Для цього додайте наступний код, розмістивши його під стилем для селектора .pin:before, .pin:after:

```
.pin:before {
  animation: pinBeforeWave 1s ease-in infinite;
}
.pin:after {
  animation: pinAfterWave 1s ease-in infinite;
}
```

Обидві анімації триватимуть одну секунду. Розподіл швидкості анімації протягом цієї секунди відбуватиметься за функцією ease-in. Анімація повторюватиметься нескінченно.

2. Пауза анімації при наведенні курсору

Коли користувач наводить курсор на елемент .pin, анімація псевдоелементів має ставати паузу. CSS дозволяє застосовувати стилі до дочірніх елементів, коли їхнього батька наведено курсор. Те саме працює і з псевдоелементами. Додати цей стиль під попереднім:

```
.pin:hover:before,
.pin:hover:after {
    animation-play-state: paused;
}
```

Цей код каже: коли елемент .pin наведений курсор (стан :hover), потрібно поставити на паузу анімацію псевдоелементів :before і :after.

3. Поява виноски

При наведенні курсору на позначку має відбуватися ще одна дія, а саме поява винесення з інформацією. У первісному стані всі виноски приховані від очей відвідувача за допомогою властивостей visibility: hidden та opacity: 0.

Знайдіть у файлі CSS селектор .pin .popover:before і додайте слідом за ним наступний стиль:

```
.pin:hover .popover {
   visibility: visible;
   opacity: 1;
}
```

Якщо прибрати властивість visibility і залишити тільки opacity: 0, то елемент не буде по-справжньому прихований: так, він не буде видно для очей, але його побачить скринрідер, що може заважати.

Якщо ж прибрати властивість прозорості і залишити тільки visibility: hidden, то ми не зможемо досягти плавності при появі елемента, оскільки властивість transition не діє властивість visibility. Саме тому ми використовували відразу дві властивості для приховування виноски.

До речі про плавність. Якщо ви зараз оновите сторінку і наведете курсор на точку, побачите, що виноска з'являється, але робить це різко. Настав час звернутися до якості transition. Знайдіть селектор .pin .popover і додайте до нього такий рядок:

```
transition: all 0.2s ease-in-out;
```

Тепер виноска з'являється плавно: властивість opacity переходить від значення 0 до значення 1 до 200 мілісекунд, а швидкість руху анімації відповідає функції ease-in-out.

Коли ми розповідали про властивість transition-delay, то згадували про те, що його зручно застосовувати для невеликої затримки меню, що випадає, перед його зникненням, щоб користувач встигав навести курсор на посилання меню. Не зайвим буде застосувати цей ефект для виноски, щоб вона зникала з легкою затримкою. Для цього додайте ще одне значення до властивості transition для селектора .pin .popover - 0.5s перед точкою з комою:

```
transition: all 0.2s ease-in-out 0.5s;
```

Але це ще не все. Зараз затримка появи винесення спрацьовує в обох напрямках — коли курсор наводиться і коли відводиться. Щоб залишити затримку лише для другого випадку, допишіть наступний рядок до селектора .pin:hover .popover:

transition-delay: 0s;

Збережіть зміни та оновіть сторінку у браузері. Тепер все працює правильно: винесення з'являється плавно і без затримки, а зникає так само плавно, але із затримкою в півсекунди.

Лишилося попрацювати ще над однією деталлю. На цей раз ми згадаємо властивість transform і його функцію обертання rotate, щоб додати ще один візуальний ефект з появою виноски. Вона ніби розвертатиметься обличчям до глядача в процесі появи, а потім так само зникатиме.

У стані невидимості винесення буде розгорнуто на 90 градусів по осі Y. Допишіть наступний стиль до селектора .pin .popover:

transform: rotateY(90deg);

При наведенні курсору на точку міста виноска має повертатися у своє нормальне становище. Для цього додайте рядок нижче до стилю для селектора .pin:hover .popover:

transform: rotateY(0deg);

Збережіть та оновіть веб-сторінку. Поспостерігайте тепер за поведінкою виносок: вони з'являються плавно та з ефектом розвороту, нагадуючи картки.

4. Колір кнопки у виносці

Останній пункт, плавна зміна кольору кнопки у виносці при наведенні курсору на неї. Для початку створимо стиль для кнопки в стані: hover. Знайдіть селектор .pin .popover .button і запишіть під ним наступний код:

```
.pin .popover .button:hover {
    background-color: #e10087;
}
```

І, нарешті, щоб перехід від початкового кольору до кінцевого відбувався плавно, додайте цей рядок до стилю селектора .pin .popover .button:

transition: all 0.2s ease-in-out;