

Дата: 16.10.2023

Група: 33

Предмет: Інформатика

Урок 3

Тема: «Анімаційні ефекти»

Мета:

- ✓ **формування предметних компетентностей:** забезпечити засвоєння знань про анімаційні ефекти; формувати вміння використовувати анімаційні ефекти у створенні сайта;
- ✓ **формування ключових компетентностей:**
 - *спілкування рідною мовою:* уникати ненормативних іншомовних запозичень у спілкуванні на ІТ-тематику;
 - *спілкування іноземними мовами:* уміння читати іншомовні навчальні й науково-популярні тексти;
 - *уміння вчитися:* навички організовувати самоосвіту із веб-технологій;
 - *ініціативність і підприємливість:* уміння вибирати оптимальний спосіб вирішення практичної проблеми.

Хід уроку

I. Організаційний етап

- привітання
- перевірка присутніх

II. Актуалізація опорних знань і життєвого досвіду

Приєм «Асоціативний ряд»

Учням пропонується визначити свої асоціації до слова «анімація». Отримані слова аналізуються і визначається розуміння цього поняття.

Наприклад:

Анімація – мультфільм, рух, аніме, малювання, мультиплікація, кадр, пожвавлення, одухотворення.

Приєм «Міні-диспут»

Учитель організовує обговорення питання:

- Чи потрібна анімація для сайта?

III. Вивчення нового матеріалу

Анімація є одним із трендів у дизайні веб-інтерфейсів, вона вже давно стала невід'ємною частиною кожного сайта. Анімаційні ефекти можна застосовувати, як до окремих об'єктів, так і до зображень. Сучасні можливості анімації дозволяють зробити найрізноманітніші слайдери: на повний екран, з 3D-ефектами, адаптивні (які переглядаються з будь-яких мобільних пристроїв) та ін.

Анімація - це і крихітні, ледь помітні індикатори завантаження, і цілі сторінки, на яких вам наче показують фільм. Важко назвати галузь веб-дизайну, де б не використовувались анімаційні ефекти: від декоративних елементів, що просто прикрашають інтерфейс, до ефектів, які активно впливають на користувача.

Перший варіант (найпростіший), який ми розглянемо, - *анімовані об'єкти*. Це окремі зображення і об'єкти, які показуються користувачеві. Наприклад, курсор, що рухається, плаваюча кнопка «вгору», кнопки заклику до дії, зміна кольору тощо.

Другий варіант - *анімовані зображення* (наприклад, рекламні банери на сайті). Вони можуть вести на внутрішні сторінки, блог, перенаправляти відвідувача на інший сайт. Ще один приклад - слайд-шоу з фотографій. Анімаційні ефекти можна створювати як виключно засобами каскадних таблиць стилів, так і за допомогою JavaScript. CSS-анімація робить можливим анімацію переходів (transitions) з однієї конфігурації CSS-стилю до іншої. Анімація складаються з двох компонентів, а саме: зі стилю, котрий описує CSS-анімацію, та набору ключових кадрів (keyframes), які задають початковий і кінцевий стани стилю анімації (також є можливість задання точок проміжного стану).

Існує не менше сотні різноманітних плагінів та бібліотек для роботи з CSS-анімацією. Розглянемо лише декілька з них.

Animate.css - це фундаментальна бібліотека анімацій, сумісних з усіма браузерами та відповідних для безлічі завдань. Вона містить всі - від класичних підскакувань до останніх новинок і унікальних ефектів - і здатна задовольнити потреби практично будь-якого проекту.

Anime.js - це вражаючий набір функцій, які дозволяють пов'язувати безліч анімацій, синхронізувати етапи, малювати лінії, змінювати форму об'єктів, створювати власні анімації тощо.

CSS-Animate - це простий майданчик для написання робочого коду для будь-якої анімації. Достатньо задати ім'я, клас, властивості анімації і фрейму, і можна керувати часовою послідовністю і додавати маркери. Одне слово, налаштувати все необхідне для створення стандартної анімації, заснованої на ключовому кадрі.

- GIF-анімація;
- Флеш-анімація (дана технологія є застарілою);
- Веб-засоби для створення анімацій: CSS і JavaScript;
- Анімація за допомогою CSS;
- Анімація за допомогою JavaScript.

Не секрет, що анімація є досить популярним та широкоживаним явищем у сучасному веб-дизайні. Вона робить зображення живим, і, відповідно, краще запам'ятовується. І, як наслідок, той сайт, у дизайні якого вона була застосована, теж має більше шансів запам'ятися користувачеві та повернути його до себе знову. Це працює.

Анімація буває різна. По-перше, можна анімувати іконки сайту. При цьому, вони будуть «оживати» при наведенні курсору. Це інтуїтивно підказує користувачеві, що область, на котру він навів курсор миші, є клікабельною, тобто при натисненні на цю область відбудеться якась подія.

Анімувати можна зображення або світлину, що виступає основною на головній сторінці сайту – це також буде виглядати жваво і яскраво. В той же час, потрібно виділити основну інформацію та навігаційні елементи так, щоб користувач, під враженням від «привітання» не забув, за чим, власне, він прийшов на цей сайт.

Слід особливу увагу приділити використанню анімації в інфографіці. Погодьтеся, що це виглядає дуже динамічно і яскраво, коли графіки або діаграми неначе оживають, і наочно та зрозуміло демонструють суху цифрову інформацію.

Правильно підібрана швидкість та плавність анімації, підсвідомо викликає у користувача спокій і, як наслідок, довіру до веб-ресурсу.

Єдине, про що потрібно пам'ятати, використовуючи анімацію у веб-дизайні – це те, що користувач приходить на сайт, у першу чергу, за необхідною інформацією. Тому, під час розробки макету, необхідно врахувати всі деталі так, щоб різноманіття анімаційних ефектів не збило його з вірного шляху, а лише допомогло запам'ятати цей веб-ресурс. Інакше, вона тільки нашкодить і викличе негативне ставлення користувача.

Найпростіший спосіб “оживити” сайт за допомогою анімації – це використати рухомі gif-зображення. Додаються такі зображення так само як і звичайні (попередня лекція).

За допомогою стилів можна задавати не лише статичний вигляд сторінки та її елементів, але й їх динамічну зміну.

Так, можна задати стандартне оформлення для певного блоку (наприклад, **div**), а також його оформлення у момент наведення вказівника миші (**hover**) на цей блок. Можна переглянути html-код, опис стилів та результат застосування такого ефекту:

Приклад. Стандартно блок `div` має ширину 200 та висоту 50 пікселів, а при наведенні мишки (подія `hover`), ширина змінюється на 300 пікселів протягом 2 секунд і розмір шрифту збільшується. Якщо додати властивість `transition: 2s` до початкового стилю, то повернення до 200 пікселів теж відбуватиметься не миттєво, а протягом вказаного часу.

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
```

```

<title>Блокова верстка в три колонки</title>
<style type = "text/css">
div {
border: 4px double black;
background: yellow;
padding: 10px;
width: 200px;
height: 30px;
}
div:hover {
width: 300px;
transition: 2s;
font-size:xx-large;
}
</style>
</head>
<body>
<div>Я люблю Україну</div>
</body>
</html>

```

Для гнучкого керування процесом анімації можна задавати визначення окремих кадрів, **keyframes**. Для такої анімації необхідно вказувати ім'я – для використання при виклику, а також стилеві властивості у різні моменти часу:

<відсоток> значення властивостей у заданий момент часу (у відсотках);

<властивість> анімована властивість;

<значення> значення в момент, описаний селектором кадру.

Приклад. Блок div початково має червоний колір тла. Протягом 4 секунд відбувається анімація example з такими етапами: колір заливки змінюється з червоного до жовтого, до синього, до зеленого. У кінці відбувається повернення до червоного кольору, визначеного для блоку div.

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Анімація example </title>
<style type = "text/css">
div {
width: 200px;
height: 50px;
background-color: red;
-webkit-animation-name: example;
-webkit-animation-duration: 4s;
animation-name: example;
animation-duration: 4s;
}

```

```

@keyframes example {
0% {background-color: red;}
25% {background-color: yellow;}
50% {background-color: blue;}
100% {background-color: green;}
}
</style>
</head>
<body>
<div>Я люблю Україну</div>
</body>
</html>

```

Можна задавати зміну позиції в анімації: для цього у початкових властивостях слід вказати відносне позиціювання, а у кадрах анімації - задати параметри розташування об'єкту відносно верху та лівої межі сторінки.

Приклад. Змінений радіус фігури (border-radius) - таким чином можна отримати коло, чи еліпс. Цей атрибут може визначати форму фігури за параметрами округлення окремих кутів, наприклад border-radius:50% 0 0 0; означає, що округлений буде лише 1 верхній лівий кут.

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Блокова верстка в три колонки</title>
<style type = "text/css">
div {
width: 100px;
height: 100px;
background-color: red;
-webkit-animation-name: example;
-webkit-animation-duration: 4s;
animation-name: example;
animation-duration: 4s;
border-radius: 50%;
text-align:center;
position: relative;
}
@keyframes example {
0% { background-color: red; left: 0px; top: 0px;}
25% { background-color: yellow; left: 200px; top: 0px; border-radius: 50% 0 0 0;}
50% { background-color: blue; left: 200px; top: 100px; border-radius: 50% 50% 0 0}
75% { background-color: green; left: 0px; top: 100px; border-radius: 50% 50% 50% 0}
100% { background-color: red; left: 0px; top: 0px;}

```

```

}
</style>
</head>
<body>
<div>Я<br /> люблю<br /> Україну</div>
</body>
</html>

```

Серед анімованих властивостей можна виділити трансформацію transform, яка може набувати значень

matrix(n,n,n,n,n,n) - перетворення,

translate(x,y) - переміщення,

scale(x,y) - масштаб,

rotate(кут) - поворот,

skew(кут-x,кут-y) - нахилання,

perspective(n) - перспектива

```

<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Блокова верстка в три колонки</title>
<style type = "text/css">
div {
width: 60px;
height: 60px;
background-color: red;
animation-name: example;
animation-duration: 4s;
border-radius: 50%;
text-align:center;
}
@keyframes example {
50% {transform: rotate(360deg)}
}
</style>
</head>
<body>
<div>Я<br /> люблю<br /> Україну</div>
</body>
</html>

```

Серед додаткових параметрів анімації можна виділити наступні: animation-delay: 2s; - затримка початку анімації

animation-iteration-count: 3; - кількість повторів анімації (**infinite;** - нескінчене повторення)

animation-direction: reverse; - напрям анімації (повторити у зворотньому порядку). **alternate;** - повторювати в прямому та зворотньому порядку

animation-timing-function: linear; - прискорення анімації (відсутнє) (можливі

значення: **ease; ease-in; ease-out; ease-in-out;**)

Існує скорочений запис стильових властивостей.

```
div {
animation-name: example;
animation-duration: 5s;
animation-timing-function: linear;
animation-delay: 2s;
animation-iteration-count: infinite;
animation-direction: alternate;
}
```

можна записати:

```
div {
animation: example 5s linear 2s infinite alternate;
}
```

Властивості анімації можуть специфічно підтримуватись різними браузерами, тож для коректного відтворення слід вживати так звані префікси - додавати до стильових властивостей уточнення того, для якого браузера вони призначені.

Chrome & Safari: -webkit-

Firefox: -moz-

Opera: -o-

Internet Explorer: -ms-

```
div {-webkit-animation: example 5s linear 2s infinite alternate;-moz-animation: example 5s
linear 2s infinite alternate;-o-animation: example 5s linear 2s infinite alternate;-ms-
animation: example 5s linear 2s infinite alternate;animation: example 5s linear 2s infinite
alternate;}
```

Є вісім властивостей animation (в алфавітному порядку):

animation-delay – задає затримку перед початком анімації, в секундах

animation-direction – вказує на те, чи повинна анімація відтворюватися в зворотному або черговому циклах (reverse або alternate)

animation-duration – визначає те, скільки секунд або мілісекунд потрібно анімації для завершення одного циклу

animation-fill-mode – стиль для елемента, коли анімація не відтворюється.

Наприклад, коли анімація завершилася або у неї задана затримка

animation-iteration-count – задає кількість разів, яку має відтворюватися анімація (infinite – для нескінченної кількості разів)

animation-name – визначає ім'я @keyframes анімації

animation-play-state – вказує на те, чи запущена анімація або припинена (running або paused)

animation-timing-function – визначає криву швидкості анімації

Весь синтаксис властивостей анімації виглядає так:

```
div {
animation-name: тут назва для ключових кадрів;
animation-duration: тривалість анімації в секундах або мілісекундах;
```

```
animation-timing-function: ease-out;  
animation-delay: 0s;  
animation-direction: alternate;  
animation-iteration-count: infinite;  
animation-fill-mode: none;  
animation-play-state: running;  
}
```

А короткий запис властивостей анімації виглядає так:

```
div {  
animation: nazva-animaciyi 1.5s ease-out 0s alternate infinite none running;  
}
```

Властивості записуються без ком, бажано в порядку слідування властивостей. При цьому, якщо деякі властивості не вказати, будуть задіяні властивості за замовчуванням. Якщо потрібно присвоїти одному елементу кілька різних ключових кадрів, тоді між властивостями ставиться кома.

Зразок синтаксису ключових кадрів **@keyframes** виглядає так:

```
@keyframes [назва анімації] {  
0% { /* початковий кадр */  
[властивість CSS]: [значення властивості];  
}  
100% { /* фінальний кадр */  
[властивість CSS]: [значення властивості];  
}  
}
```

У правилах **@keyframes** можна використовувати більше двох (початкового і фінального) ключових кадрів (наприклад, проміжні в позначці 30%, 75% і т.д.). Також в кожному ключовому кадрі можна вказувати досить велику кількість властивостей CSS.

IV. Закріплення вивченого матеріалу

Прийом «Чесний самоаналіз»

Учитель пропонує учням відповісти на запитання і зробити певні висновки.

- Чи розумієте ви, для чого використовують анімаційні ефекти у веб-дизайні?
- Коли недоречно використовувати анімацію?
- Які існують способи створення анімації?
- Ви можете створювати анімацію?

V. Домашнє завдання

- 1) Опрацювати сторінки 109 підручника Інформатика для 11 класу (автори Руденко В. Д, Речич Н. В., Потієнко В. О.)
- 2) Основи веб-дизайну / О. Г. Пасічник, О. В. Пасічник, І. В. Стеценко : [Навч. посіб.]. — К.: Вид. група ВHV. — 2009.- розділ 3

- 3) Законспектувати матеріал уроку
- 4) За допомогою мережі інтернет знайти роз'яснення поняттю «Content management system» на сайті Вільної енциклопедії «Вікіпедія»
- 5) Провести аналіз інформаційної структури сайту коледжу та пояснити який CMS застосований там
- 6) Для зворотнього зв'язку e-mail: 2573562@ukr.net