Teach Me Skills

Lesson 12

Анимация элементов

Препроцессоры и сборщики

План занятия

- Линейная анимация: transition
- Цикличная анимация: @keyframes
- Понятие Pixel Perfect

Линейная анимация

Изучение линейной анимации: transition

CSS3-переходы позволяют анимировать исходное значение CSS-свойства на новое значение с течением времени, управляя скоростью смены значений свойств.

Смена свойств происходит при наступлении определенного события, которое описывается соответствующим псевдоклассом. Чаще всего используется псевдокласс :hover. Данный псевдокласс не работает на мобильных устройствах, таких как iPhone или Android.

Переходы применяются ко всем элементам, а также к псевдоэлементам :before и :after. CSS3-переходы могут применяться не ко всем свойствам и их значениям.

Подробный перечень

Линейная анимация

transition-property

Содержит название CSS-свойств, к которым будет применен эффект перехода. Значение свойства может содержать как одно свойство, так и список свойств через запятую.

Синтаксис:
transition-property: none | all | <свойство>

Линейная анимация

Продолжительность перехода transition-duration

Задаёт промежуток времени, в течение которого должен осуществляться переход. Если разные свойства имеют разные значения для перехода, они указываются через запятую.

Синтаксис:

transition-duration: <время>

Линейная анимация

Функция перехода transition-timing-function

Свойство задаёт временную функцию, которая описывает скорость перехода объекта от одного значения к другому. Если вы определяете более одного перехода для элемента, например, цвет фона элемента и его положение, вы можете использовать разные функции для каждого свойства.

Синтаксис:

transition-timing-function: ease, ease-in, ease-out, ease-in-out, linear, step-start, step-end, steps, cubic-bezier

Линейная анимация

Задержка перехода transition-delay

Необязательное свойство, позволяет сделать так, чтобы изменение свойства происходило не моментально, а с некоторой задержкой.

Синтаксис:

transition-delay: <время>

Линейная анимация

Краткая запись перехода

Все свойства, отвечающие за изменение внешнего вида элемента, можно объединить в одно свойство transition:

transition: transition-property transition-duration transition-timing-function transition-delay;

```
div {
  transition: all 1s ease 0s;
}
```

Синтаксис:

transition: <переход>

Линейная анимация

Плавный переход нескольких свойств

Для элемента можно задать несколько последовательных переходов, перечислив их через запятую. Каждый переход можно оформить своей временной функцией.

```
div {
  transition: background 0.3s
ease, color 0.2s linear;
}
```

Цикличная анимация

Изучение цикличной анимации: @keyframes

CSS3-анимация придаёт сайтам динамичность. Она оживляет веб-страницы, улучшая взаимодействие с пользователем. В отличие от CSS3-переходов, создание анимации базируется на ключевых кадрах, которые позволяют автоматически воспроизводить и повторять эффекты на протяжении заданного времени, а также останавливать анимацию внутри цикла.

CSS3-анимация может применяться практически для всех html-элементов, а также для псевдоэлементов :before и :after.

Список анимируемых свойств

Линейная анимация

Ключевые кадры

Ключевые кадры используются для указания значений свойств анимации в различных точках анимации.

Ключевые кадры определяют поведение одного цикла анимации; анимация может повторяться ноль или более раз.
Ключевые кадры указываются с помощью правила @keyframes, определяемого следующим образом:

@keyframes имя анимации { список правил }

```
@keyframes shadow {
  from { text-shadow: 0 0 3px black; }
  50% { text-shadow: 0 0 30px black; }
  to { text-shadow: 0 0 3px black; }
}

@keyframes move {
  from, to { top: 0; left: 0; }
  25%, 75% { top: 100%; }
  50% { top: 50%; }
}
```

Цикличная анимация

Ключевые кадры создаются с помощью ключевых слов from и to (эквивалентны значениям 0% и 100%) или с помощью процентных пунктов, которых можно задавать сколько угодно. Также можно комбинировать ключевые слова и процентные пункты.

Если 0% или 100% кадры не указаны, то браузер пользователя создает их, используя вычисляемые (первоначально заданные) значения анимируемого свойства.

После объявления правила @keyframes, мы можем ссылаться на него в свойстве animation:

```
animation: shadow  2s infinite ease-in-out;
```

Цикличная анимация

Название анимации: свойство animation-name

Свойство animation-name определяет список применяемых к элементу анимаций. Каждое имя используется для выбора ключевого кадра в правиле, которое предоставляет значения свойств для анимации. Если имя не соответствует ни одному ключевому кадру в правиле, нет свойств для анимации, отсутствует имя анимации, анимация не будет выполняться.

Рекомендуется использовать название, отражающее суть анимации, при этом можно использовать одно или несколько слов, перечисленных через дефис - или символ нижнего подчеркивания _.

Синтаксис: animation-name: none | <идентификатор>

```
animation-name: none;
animation-name: test-01;
animation-name: -sliding;
animation-name: moving-vertically;
animation-name: test2;
animation-name: test3, move4;
animation-name: initial;
animation-name: inherit;
```

Цикличная анимация

Продолжительность анимации: свойство animation-duration

Свойство animation-duration определяет продолжительность одного цикла анимации. Задаётся в секундах s или миллисекундах ms. Если для элемента задано более одной анимации, то можно установить разное время для каждой, перечислив значения через запятую.

Синтаксис:

```
animation-duration: <время>
animation-duration: .5s;
animation-duration: 200ms;
animation-duration: 2s, 10s;
```

Цикличная анимация

Временная функция: свойство animation-timing-function

Свойство animation-timing-function описывает, как будет развиваться анимация между каждой парой ключевых кадров. Во время задержки анимации временные функции не применяются.

Синтаксис:

animation-timing-function: ease, ease-in, ease-out, ease-in-out, linear, step-start, step-end, steps, cubic-bezier

Цикличная анимация

Повтор анимации: свойство animation-iteration-count

Свойство animation-iteration-count указывает, сколько раз проигрывается цикл анимации. Начальное значение 1 означает, что анимация будет воспроизводиться от начала до конца один раз. Это свойство часто используется в сочетании со значением alternate свойства animation-direction, которое заставляет анимацию воспроизводиться в обратном порядке в альтернативных циклах.

Синтаксис:

```
animation-iteration-count: infinite | <число>

animation-iteration-count: infinite;
animation-iteration-count: 3;
animation-iteration-count: 2.5;
animation-iteration-count: 2, 0, infinite;
```

Цикличная анимация

Направление анимации: свойство animation-direction

Свойство animation-direction определяет, должна ли анимация воспроизводиться в обратном порядке в некоторых или во всех циклах.

Когда анимация воспроизводится в обратном порядке, временные функции также меняются местами. Например, при воспроизведении в обратном порядке функция ease-in будет вести себя как ease-out.

Синтаксис:

animation-direction: normal, alternate, reverse, alternate-reverse

Цикличная анимация

Проигрывание анимации: свойство animation-play-state

Свойство animation-play-state определяет, будет ли анимация запущена или приостановлена. Можно останавливать анимацию при наведении курсора мыши на объект - состояние :hover.

Синтаксис:

animation-play-state: running, paused

Цикличная анимация

Задержка анимации: свойство animation-delay

Свойство animation-delay определяет, когда анимация начнется. Задается в секундах s или миллисекундах ms.

Синтаксис:

```
animation-delay: <время>

animation-delay: 5s;
animation-delay: 3s, 10ms;
```

Цикличная анимация

Состояние элемента до и после воспроизведения анимации: свойство animation-fill-mode

Свойство animation-fill-mode определяет, какие значения применяются анимацией вне времени ее выполнения. Когда анимация завершается, элемент возвращается к своим исходным стилям.

По умолчанию анимация не влияет на значения свойств animation-name и animation-delay, когда анимация применяется к элементу.

Кроме того, по умолчанию анимация не влияет на значения свойств animation-duration и animation-iteration-count после ее завершения. Свойство animation-fill-mode может переопределить это поведение.

Синтаксис:

animation-fill-mode: none, forwards, backwards, both

Цикличная анимация

Краткая запись анимации: свойство animation

Все параметры воспроизведения анимации можно объединить в одном свойстве - animation, перечислив их через пробел:

animation: animation-name animation-duration animation-timing-function animation-delay animation-iteration-count animation-direction;

Для воспроизведения анимации достаточно указать только два свойства - animation-name и animation-duration, остальные свойства примут значения по умолчанию.

Порядок перечисления свойств не имеет значения, единственное, время выполнения анимации animation-duration обязательно должно стоять перед задержкой animation-delay.

Цикличная анимация

Множественные анимации

Для одного элемента можно задавать несколько анимаций, перечислив их названия через запятую.

```
div {
  animation: shadow 1s ease-in-out
0.5s alternate, move 5s linear 2s;
}
```

Pixel Perfect

Pixel Perfect верстка - это особая техника создания структуры html-кода, которая позволяет сверстанному html-шаблону максимально точно совпадать с оригинальным PSD-макетом пиксель в пиксель. При наложении html-шаблона на макет PSD должно произойти полное совпадение графических элементов, изображений и текста.

Применение Pixel Perfect верстки

Техника работы с Pixel Perfect версткой осуществляется за счет применения особых плагинов, созданных специально для определенных браузеров, а также с помощью специализированных скриптов. Основные этапы работы с Pixel Perfect версткой включают в себя такие процедуры:

1. Оригинальный макет необходимо сохранить в формате .png.
2. Html-шаблон, сверстанный по данному макету, открывается в браузере, после чего с помощью применения плагина копию в формате .png необходимо наложить на сверстанную страницу.
3. После наложения становится видна разница между расположением элементов на png-копии и html-макете. Далее проводится коррекция значений для точного совпадения.

Ссылки на дополнительные материалы

Transition

Animation

Pixel Perfect