

A decorative graphic on the left side of the slide consisting of two overlapping parallelograms. The front one is blue and the back one is a light green. They are positioned diagonally, with the blue one partially covering the green one.

HTML5 & CSS3

Awesome. Innovative. Different



Html5. Semantic

`<header>` хедер сайта, блок, в котором могут быть меню\лого\контакты и прочая информация

`<nav>` блок навигации (часто внутри - ссылки)

`<aside>` - дополнительный второстепенный контент

`<main>` блок с главным контентом документа

`<section>` секция документа

`<article>` - содержимое сайта по типу статьи (блог, новость и т.п.)

`<details>` в связке с `summary` - информация, которую пользователь может посмотреть\скрыть



Html5. Semantic

`<summary>` в связке с `<details>` - общий итог секции, которую можно показать\скрыть

`<figure>` Самодостаточный контейнер для группирования в себе контента (чаще изображения\диаграммы и т.п)

`<figcaption>` Подпись к элементу `<figure>`

`<mark>` - для обозначения выделенного текста

`<footer>` Футер сайта (нижняя часть)



New html features

`<audio src="url" autoplay volume="0.9">` -вставить аудио на страницу

`<video src="url" autoplay >` - вставить видео на страницу

`<dialog>` - семантически - модальное окно

`<progress>` - элемент отображения прогресса (загрузки и т.п.)

Управление этими элементами - с помощью JS. Поддержка не во всех браузерах, не забывайте проверять на сайтах подобных `canIUse`



Css transform

Css transform - группа свойств, которые позволяют изменять (сдвигать, поворачивать и масштабировать элементы) не затрагивая при этом остальные элементы на странице.

`transform: translate(x,y) or translateX(x) | translateY(y)` - перемещаем элемент относительно текущей позиции

`transform: scale(x,y) or scaleX(x) | scaleY(y)` - увеличиваем (1+)\уменьшаем((от 0 до 1), отражаем зеркально (минусовое значение) элемент

`transform: rotate(deg)` - поворачиваем элемент

`transform: skew(x-deg. y-deg) or skewX(deg) | skewY(deg)` - наклон элемента



Css transition

Transition - переход между 2 состояниями свойств элементов

transition-property- Свойство css (width, color etc), к которому применяем анимацию. Есть исключения, свойства, у которых нет промежуточного состояния (например display)

transition-duration - Длительность анимации.

transition-timing-function - математическая функция, которая указывает на то, как быстро по времени меняется указанное через transition-property значение свойства: ease, linear, ease-in-out, ease-in, ease-out, cubic-bezier

transition-delay - Задержка анимации.

Shorthand: transition: transition-property | transition-duration | transition-timing-function | transition-delay



Css Animations

Css Animations - переход между любым количеством состояний свойств элементов (в отличие от transition). Позволяет делать более сложные переходы, эффекты и т.п.

Особенность: описываем анимацию в специальном блоке с ключевым словом - keyframes:

```
@keyframes animationName{  
  
  /* animation goes here */  
  
}
```

Внутри этого блока пишем селекторы и группы свойств. Селекторами являются либо ключевые слова (from/to), либо же проценты(0%, 12% и т.п.). В данном случае - селекторы - условные пропорции временных промежутков, на которых мы хотим задавать определенные свойства. Чем больше таких селекторов, тем более плавным будет переход



Css Animations

Пример:

```
@keyframes animationName {  
  from {  
    color: red;  
    font-size: 14px;  
  }  
  10% {  
    color: rgba(200, 100, 100);  
    background-color: yellow;  
    font-size: 16px;  
  }  
  100% {  
    font-weight: bold;  
    transform: translateX(50px);  
  }  
}
```




CSS Animations

После того, как описали анимацию - ее нужно добавить к целевому элементу (элементам) с помощью свойства `animation-name`

```
.sun {  
  
background: red;  
  
animation-name: awesomeAnimation;  
  
}
```



CSS Animations

Свойства анимаций:

`animation-name: awesomeAnimation;`

`animation-duration: 0.4s;`

`animation-timing-function: linear;`

`animation-delay: 2s;`

`animation-iteration-count: 4| infinite;`

`animation-direction: normal | reverse | alternate | alternate-reverse ;`