



Лекция 8

CSS3. Медиа запросы. Transition

Introduction to Web Development

План лекции

- Медиа запросы
- Свойство transition

CSS3. Медиа запросы

Медиа запросы позволяют использовать особые CSS-стили для конкретных устройств вывода.

IE: 9.0 (кроме вложенных медиазапросов)

Edge: 12.0

Firefox: 3.5

Chrome: 26.0

Safari: 6.1

Opera: 10.1

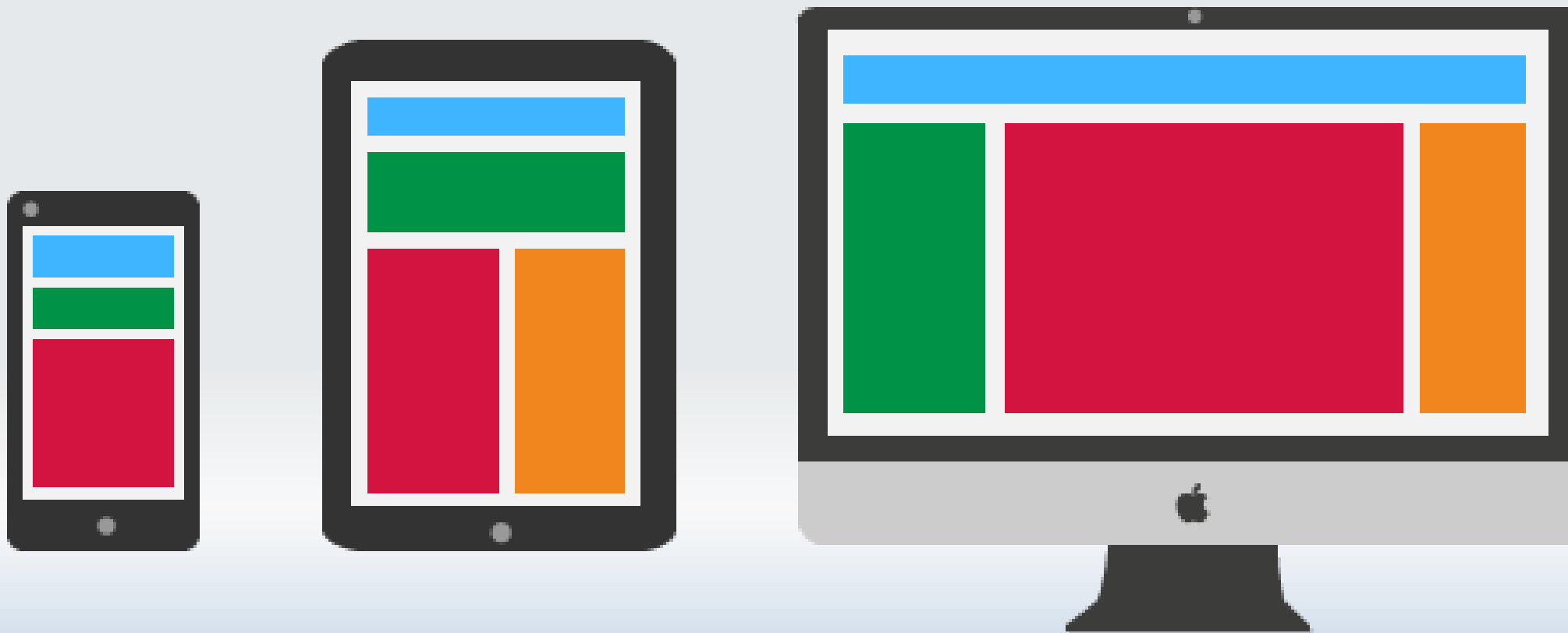
iOS Safari: 7.1

Android: 4.4

Chrome for Android: 55.0



CSS3. Медиа запросы



CSS3. Медиа запросы

Могут быть использованы для проверки многих параметров пользователя, таких как:

- **Ширина и высота области просмотра**
- **Ширина и высота устройства**
- **Ориентация (планшет / телефон в альбомном или портретном режиме)**
- **Разрешение экрана и многое другое...**

CSS3. Медиа запросы

Медиа запрос (правило **@media**) состоит из типа носителя и ноль или более выражений для проверки характеристик устройства

```
@media not|only mediatype and (media feature) {  
    CSS-Code;  
}
```

Правила **@media** всегда прописываются в конце файла CSS-стилей.

```
@media screen and (min-width: 480px) {  
    body {  
        background-color: lightgreen;  
    }  
}
```

CSS3. Медиа запросы

Значение	Описание
<code>all</code>	Подходит для всех типов устройств.
<code>print</code>	Предназначен для страничных материалов и документов, просматриваемых на экране в режиме предварительного просмотра печати.
<code>screen</code>	Предназначен в первую очередь для экранов цветных компьютерных мониторов.
<code>speech</code>	Предназначен для синтезаторов речи.

Если вы не используете операторы `not` или `only`, тип носителя не является обязательным, и подразумевается тип `all`.

CSS3. Медиа запросы. Логические операторы

Логические операторы

С помощью логических операторов можно создавать комбинированные медиазапросы, в которых будет проверяться соответствие нескольким условиям.

AND

связывает друг с другом разные условия

```
@media screen and (max-width: 600px) {  
  /* CSS-стили */;  
}
```


CSS3. Медиа запросы. Логические операторы

```
@media screen and (max-width: 699px) and (min-width: 520px) {  
    ul li a {  
        padding-left: 30px;  
        background: url(email-icon.png) left center no-repeat;  
    }  
}
```

Правило **@media all and (max-width: 600px) {...}** равнозначно правилу **@media (max-width: 600px) {...}**.

CSS3. Медиа запросы. Логические операторы

NOT

Позволяет сработать медиа запросу в противоположном случае.

- Ключевое слово `not` добавляется в начало медиа запроса и применяется ко всему запросу целиком.
- Если медиа запрос составлен с использованием оператора запятая, то отрицание будет распространяться только на ту часть, которая идет до запятой, т.е. запрос

```
@media not all and (monochrome) { ... }
```

CSS3. Медиа запросы. Логические операторы

ONLY

Применяется для старых браузеров, которые не поддерживают медиа-запросы, когда нужно скрыть правило @media от них.

```
@media only screen and (color) {  
  /* CSS-стили */;  
}
```

,

Оператор запятая работает по аналогии с логическим оператором **or**

```
@media screen, projection {  
  /* CSS-стили */;  
}
```

CSS3. Медиа запросы. Характеристики носителя

Параметр	Описание
<code>width</code>	<p>Проверяет ширину области просмотра. Значения задаются в единицах длины, <code>px</code>, <code>em</code> и т.д., например, <code>(width: 800px)</code>. Обычно для проверки используются минимальные и максимальные значения ширины.</p> <p><code>min-width</code> применяет правило если ширина области просмотра больше значения, указанного в запросе, <code>max-width</code> — ширина области просмотра меньше значения, указанного в запросе.</p>
<code>height</code>	<p>Проверяет высоту области просмотра. Значения задаются в единицах длины, <code>px</code>, <code>em</code> и т.д., например, <code>(height: 500px)</code>. Обычно для проверки используются минимальные и максимальные значения высоты.</p> <p><code>min-height</code> применяет правило если высота области просмотра больше значения, указанного в запросе, <code>max-height</code> — высота области просмотра которого меньше значения, указанного в запросе.</p>

CSS3. Медиа запросы. Характеристики носителя

<code>aspect-ratio</code>	<p>Проверяет соотношение ширины к высоте области просмотра. Широкоэкранный дисплей с соотношением сторон 16:9 может быть помечен как <code>(aspect-ratio: 16/9)</code>.</p> <p><code>min-aspect-ratio</code> проверяет минимальное соотношение, <code>max-aspect-ratio</code> — максимальное соотношение ширины к высоте области просмотра.</p>
<code>orientation</code>	<p>Проверяет ориентацию области просмотра. Принимает два значения: <code>(orientation: portrait)</code> и <code>(orientation: landscape)</code>.</p>
<code>resolution</code>	<p>Проверяет разрешение экрана (количество пикселей). Значения также могут проверять количество точек на дюйм (dpi) или количество точек на сантиметр (dpcm), например, <code>(resolution: 300dpi)</code>.</p> <p><code>min-resolution</code> проверяет минимальное разрешение экрана, <code>max-resolution</code> — максимальное.</p>

CSS3. Медиа запросы. Характеристики носителя

<code>color</code>	<p>Проверяет количество бит на каждый из цветовых компонентов устройства вывода. Например, <code>(min-color: 4)</code> означает, что экран конкретного устройства должен иметь 4-битную глубину цвета.</p> <p><code>min-color</code> проверяет минимальное количество бит, <code>max-color</code> — максимальное количество бит.</p>
<code>color-index</code>	<p>Проверяет количество записей в таблице подстановки цветов. В качестве значения указывается положительное число, например, <code>(color-index: 256)</code>.</p> <p><code>min-color-index</code> проверяет минимальное количество записей, <code>max-color-index</code> — максимальное количество записей.</p>
<code>monochrome</code>	<p>Проверяет количество битов на пиксель монохромного устройства. Значение задается целым положительным числом, например, <code>(min-monochrome: 8)</code>.</p> <p><code>min-monochrome</code> проверяет минимальное количество битов, <code>max-monochrome</code> — максимальное количество битов.</p>

CSS3. Медиа запросы. Размеры экрана

```
/* Smartphones (вертикальная и горизонтальная ориентация) ----- */
@media only screen and (min-width : 320px) and (max-width : 480px) {
/* стили */
}

/* Smartphones (горизонтальная) ----- */
@media only screen and (min-width: 321px) {
/* стили */
}

/* Smartphones (вертикальная) ----- */
@media only screen and (max-width: 320px) {
/* стили */
}
```

CSS3. Медиа запросы. Размеры экрана

```
/* iPads (вертикальная и горизонтальная) ----- */
@media only screen and (min-width: 768px) and (max-width: 1024px) {
  /* стили */
}

/* iPads (горизонтальная) ----- */
@media only screen and (min-width: 768px) and (max-width: 1024px) and (orientation: landscape) {
  /* стили */
}

/* iPads (вертикальная) ----- */
@media only screen and (min-width: 768px) and (max-width: 1024px) and (orientation: portrait) {
  /* стили */
}
```


CSS3. Медиа запросы. Размеры экрана

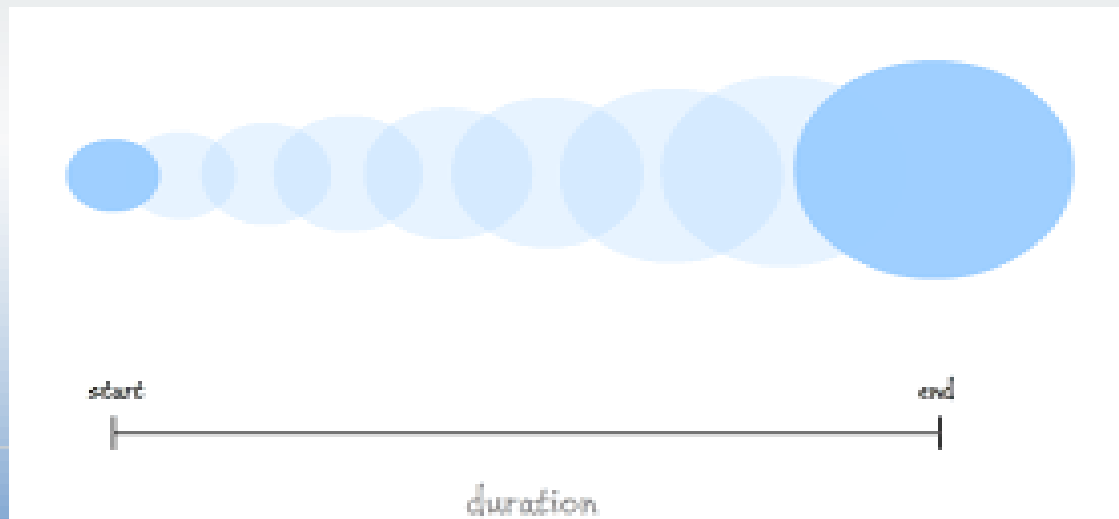
```
/* Настольные компьютеры и ноутбуки ----- */  
@media only screen and (min-width: 1224px) {  
/* стили */  
}  
  
/* Большие экраны ----- */  
@media only screen and (min-width: 1824px) {  
/* стили */  
}
```

CSS3. Transition

Transition

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

Рекомендуемое время стандартного перехода — 200ms.



IE: 10.0

Firefox: 16.0, 4.0 -moz-

Chrome: 26.0, 4.0 -webkit-

Safari: 6.1, 3.1 -webkit-

Opera: 12.1, 11.6 -o-

iOS Safari: 7.1

Opera Mini: —

Android Browser: 4.4, 4.1 -webkit-

Chrome for Android: 44

CSS3. Transition

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.

Универсальным решением, работающим в настольных и мобильных браузерах, будет обработка событий с помощью JavaScript (например, переключение классов при клике).

CSS3 переходы могут применяться **не ко всем свойствам и их значениям**.
Перечень: https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_animated_properties

CSS3. Transition

Property	Description
<u>transition</u>	A shorthand property for setting the four transition properties into a single property
<u>transition-delay</u>	Specifies a delay (in seconds) for the transition effect
<u>transition-duration</u>	Specifies how many seconds or milliseconds a transition effect takes to complete
<u>transition-property</u>	Specifies the name of the CSS property the transition effect is for
<u>transition-timing-function</u>	Specifies the speed curve of the transition effect

```
{transition: transition-property transition-duration transition-timing-function transition-delay;}
```

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

CSS3. Transition

transition-delay

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

время	Время задержки перехода указывается в секундах или миллисекундах.
<code>initial</code>	Устанавливает значение свойства в значение по умолчанию.
<code>inherit</code>	Наследует значение свойства от родительского элемента.

```
div {  
    -webkit-transition-delay: 2s; /* Safari */  
    transition-delay: 2s;  
}
```

CSS3. Transition

transition-duration

Задаёт **промежуток времени**, в течение которого должен **осуществляться переход**. Если разные свойства имеют разные значения для перехода, они указываются через запятую. Если продолжительность перехода не указана, то анимация при смене значений свойств происходить не будет.

время	Время перехода указывается в секундах или миллисекундах, например, <code>2s</code> или <code>5ms</code> .
<code>initial</code>	Устанавливает значение свойства в значение по умолчанию.
<code>inherit</code>	Наследует значение свойства от родительского элемента.

```
div {  
  -webkit-transition-duration: 5s; /* Safari */  
  transition-duration: 5s;  
}
```

CSS3. Transition

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. При создании перехода можно использовать как начальное, так и конечное состояние элемента.

<code>none</code>	Отсутствие свойства для перехода.
<code>all</code>	Значение по умолчанию. Применяет эффект перехода ко всем свойствам элемента.
<code>свойство</code>	Определяет список css-свойств, перечисленных через запятую, участвующих в переходе.
<code>initial</code>	Устанавливает значение свойства в значение по умолчанию.
<code>inherit</code>	Наследует значение свойства от родительского элемента.

CSS3. Transition

transition-property

```
div {  
    width: 100px;  
    height: 100px;  
    background: red;  
    -webkit-transition-property: width; /* Safari */  
    -webkit-transition-duration: 2s; /* Safari */  
    transition-property: width;  
    transition-duration: 2s;  
}  
  
div:hover {  
    width: 300px;  
}
```

Пример здесь: https://www.w3schools.com/cssref/tryit.asp?filename=trycss3_transition-property

CSS3. Transition

transition-timing-function

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

```
div {  
    -webkit-transition-timing-function: linear; /* Safari and Chrome */  
    transition-timing-function: linear;  
}
```

```
transition-timing-function: linear|ease|ease-in|ease-out|ease-in-out|step-start|step-end|steps(int,start|end)|cubic-bezier(n,n,n,n)|initial|inherit;
```

CSS3. Transition

```
div {
  width: 100px;
  height: 50px;
  background: red;
  color: white;
  font-weight: bold;
  -webkit-transition: width 2s; /* Safari */
  transition: width 2s;
}

/* For Safari 3.1 to 6.0 */
#div1 {-webkit-transition-timing-function: linear;}
#div2 {-webkit-transition-timing-function: ease;}
#div3 {-webkit-transition-timing-function: ease-in;}
#div4 {-webkit-transition-timing-function: ease-out;}
#div5 {-webkit-transition-timing-function: ease-in-out;}

/* Standard syntax */
#div1 {transition-timing-function: linear;}
#div2 {transition-timing-function: ease;}
#div3 {transition-timing-function: ease-in;}
#div4 {transition-timing-function: ease-out;}
#div5 {transition-timing-function: ease-in-out;}

div:hover {
  width: 300px;
}
```

transition-timing-function

https://www.w3schools.com/cssref/tryit.asp?file=name=trycss3_transition-timing-function2

CSS3. Transition

transition-timing-function

`ease`

Функция по умолчанию, переход начинается медленно, разгоняется быстро и замедляется в конце. Соответствует `cubic-bezier(0.25,0.1,0.25,1)` .

`linear`

Переход происходит равномерно на протяжении всего времени, без колебаний в скорости. Соответствует `cubic-bezier(0,0,1,1)` .

`ease-in`

Переход начинается медленно, а затем плавно ускоряется в конце. Соответствует `cubic-bezier(0.42,0,1,1)` .

`ease-out`

Переход начинается быстро и плавно замедляется в конце. Соответствует `cubic-bezier(0,0,0.58,1)` .

CSS3. Transition

transition-timing-function

`ease-in-out`

Переход медленно начинается и медленно заканчивается.
Соответствует `cubic-bezier(0.42, 0, 0.58, 1)`.

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

Позволяет вручную установить значения от 0 до 1 для кривой ускорения.

`initial`

Устанавливает значение свойства в значение по умолчанию.

`inherit`

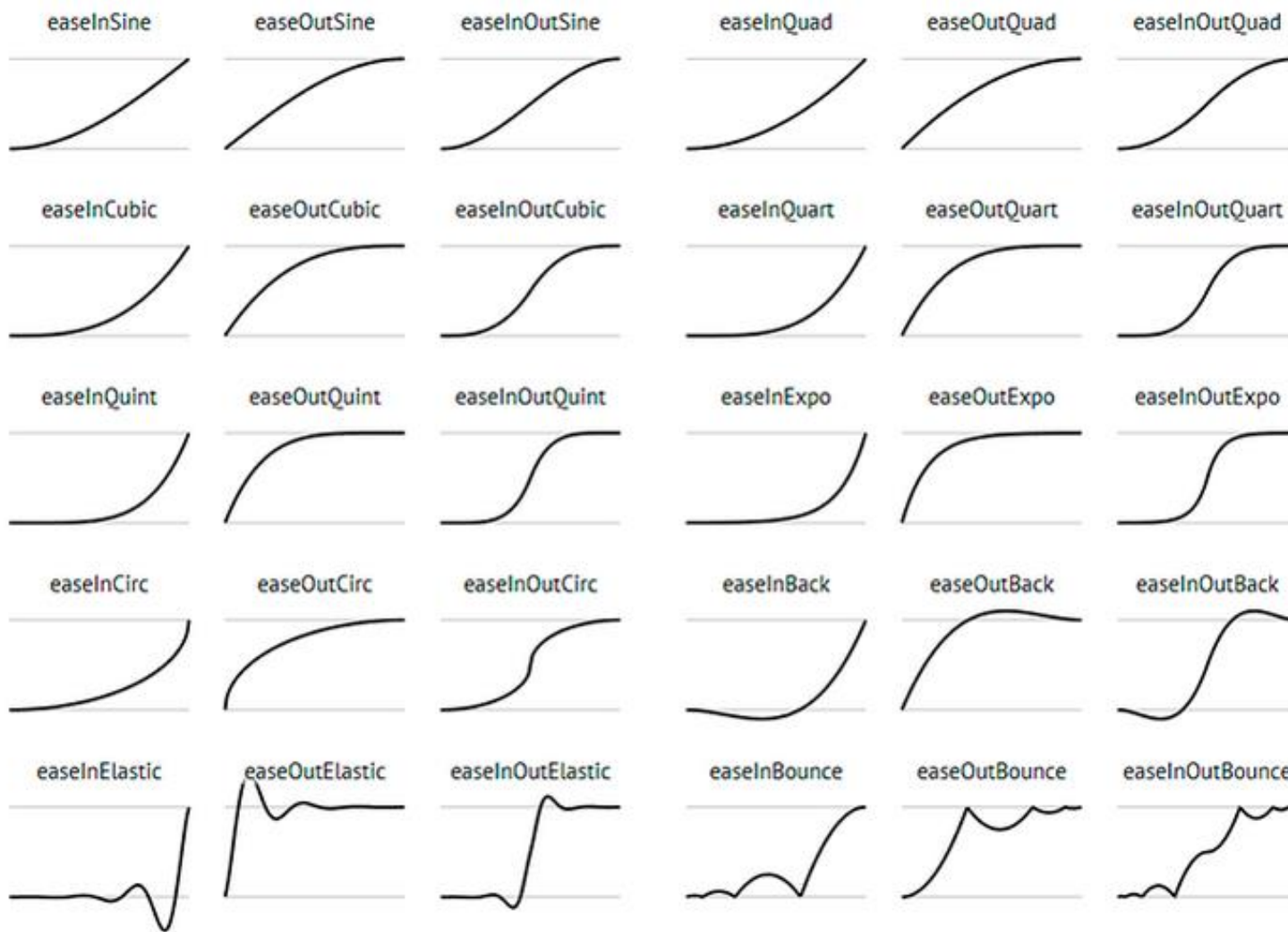
Наследует значение свойства от родительского элемента.

CSS3. Transition

transition- timing-function

Создать свою:

<http://roblaplaca.com/examples/bezierBuilder/>



CSS3. Transition

transition-timing-function

Пользовательское название	Значение функции	Пользовательское название	Значение функции
easeInSine	cubic-bezier(0.47, 0, 0.745, 0.715)	easeInQuint	cubic-bezier(0.755, 0.05, 0.855, 0.06)
easeOutSine	cubic-bezier(0.39, 0.575, 0.565, 1)	easeOutQuint	cubic-bezier(0.23, 1, 0.32, 1)
easeInOutSine	cubic-bezier(0.445, 0.05, 0.55, 0.95)	easeInOutQuint	cubic-bezier(0.86, 0, 0.07, 1)
easeInQuad	cubic-bezier(0.55, 0.085, 0.68, 0.53)	easeInExpo	cubic-bezier(0.95, 0.05, 0.795, 0.035)
easeOutQuad	cubic-bezier(0.25, 0.46, 0.45, 0.94)	easeOutExpo	cubic-bezier(0.19, 1, 0.22, 1)
easeInOutQuad	cubic-bezier(0.455, 0.03, 0.515, 0.955)	easeInOutExpo	cubic-bezier(1, 0, 0, 1)
easeInCubic	cubic-bezier(0.55, 0.055, 0.675, 0.19)	easeInCirc	cubic-bezier(0.6, 0.04, 0.98, 0.335)
easeOutCubic	cubic-bezier(0.215, 0.61, 0.355, 1)	easeOutCirc	cubic-bezier(0.075, 0.82, 0.165, 1)
easeInOutCubic	cubic-bezier(0.645, 0.045, 0.355, 1)	easeInOutCirc	cubic-bezier(0.785, 0.135, 0.15, 0.86)
easeInQuart	cubic-bezier(0.895, 0.03, 0.685, 0.22)	easeInBack	cubic-bezier(0.6, -0.28, 0.735, 0.045)
easeOutQuart	cubic-bezier(0.165, 0.84, 0.44, 1)	easeOutBack	cubic-bezier(0.175, 0.885, 0.32, 1.275)
easeInOutQuart	cubic-bezier(0.77, 0, 0.175, 1)	easeInOutBack	cubic-bezier(0.68, -0.55, 0.265, 1.55)

Рекомендованная литература:

- <https://www.w3.org/>
- <https://www.w3schools.com/css/default.asp>
- <https://webref.ru/css>
- <https://www.codecademy.com>
- <https://htmlacademy.ru/>
- <http://mattkersley.com/responsive/>
- Новая большая книга CSS

И...

<http://www.google.com/>