



Добавяне на интерактивност с HTML 5, jQuery и CSS 3

проф. г-р Асен Рахнев, докторант Мая Стоева

СЪДЪРЖАНИЕ

- I. Пояснения към домашна работа № 1
- II. Полезни инструменти за генериране на “sprites”
- III. Да рисуваме с CSS3 и HTML5
- IV. Какво представлява 960 grid-системата
- V. Как и кога да използваме мерните единици за font-size в HTML: em, rem, %, px
- VI. Трансформации с CSS3
- VII. Създаване на анимации с CSS3

Пояснения към домашна работа № 1

Да се създаде страница, написана с HTML5 и CSS3, в която:

- 1) всички графични елементи да се зареждат от един sprite файл.
- 2) Фонът ѝ да е преливка, изградена с CSS3.
- 3) Да им 2 менюта (по ваш избор дали да са хоризонтални или вертикални). Едното да има 3 точки/линкове: За мен/нас, Контакти, Медиа. Като последното да бъде връзка към страница, в която зареждате видео и аудио с HTML 5 стандартните таговете.
- 3а) Второто меню да съдържа 5 линка. Те трябва да водят до следните трансформации с css3: skew, scale, translate, rotate + 1 линк, с името "Други".
- 4) Да се използва поне един различен от стандартните шрифтове.
// да се включат префиксите за тазличните браузъри: o-, moz-,
// webkit-, ms-
// Тази домашна ще разширим по-нататък с още функционалности...

Полезни инструменти за генериране на “sprites”

<http://arnaumarch.com/en/sprites.html>

<http://spritegen.website-performance.org/>

<http://www.ms-studio.com/FontSales/felttiproman.html>

// покрай sprites и малко инфо за актуални шрифтове...

Да рисуваме с CSS3 и HTML5

<https://developer.mozilla.org/en-US/demos/detail/css3-spider-man>

<http://enrappture.com/spiderman/>

<http://enrappture.com/superman/>

<http://cssdeck.com/labs/css3-superman>

Какво представлява 960 grid-системата

<http://960.gs/>

<http://www.blueprintcss.org/>

Мрежата с 12 колони е разделена на секции по 60 pixel-а ширина. Мрежата с 16 колони се състои от части 40 pixel-а. Всяка колона има 10 пиксела margin отляво и отдясно, което ни дава 20 pixel отсъпи между колоните.

<http://960.gs/demo.html>

http://960.gs/demo_24_col.html // с 30px колони

Идеята е да ни предостави един бърз начин за създаване на прототипи за нашите сайтове с правилно позициониране и бързо подравняване.

Как и кога да използваме мерните единици за font-size в HTML: em, rem, %, px

През последните дни възникнаха доста дебати по въпроса кои точно мерни единици да използваме за оразмеряване на текста в уеб страниците. За съжаления за и против, относно всяка от тях има доста. Ето защо е добре да знаем малко повече за всяка една от тях. Кога и как се използват като теория, а накрая разбира се, сами може да решим кое да изберем.

Има 2 основни единици, които се използват основно:

- 1) Size с px
- 2) Size с em

Как и кога да използваме мерните единици за font-size в HTML: em, rem, %, px

Size с px

В момента за размерите на текста се използва най-често pixels (px). Надеждна и устойчива мярка. За съжаление потребителите на Internet Explorer— даже на IE9, а и 10 — нямат възможност да променят размера на текста като използват функциите на самия браузър за увеличаване и намаляване на font size. Кое то за самата използваемост на конкретния сайт не е добре. Последните версии на IE включват zooming, който уголемява всичко на страницата — това помага отчасти.

Като цяло залагайки на тази мерна единица, си даваме относителна сигурност под различните браузъри.

Как и кога да използваме мерните единици за font-size в HTML: em, rem, %, px

Size с em

Невъзможността да скалираме текста под IE се решава с избора на **em** като мерна единица за текста (Тя се появява около 2004).

Техниката променя основния размер на текста в body-то използвайки процентите. Тя променя нещата така, че 1em е равен на 10px, вместо на този по подразбиране - 16px. Например, за да зададем font-size равен на 14px, задайте го като 1.4em.

Как и кога да използваме мерните единици за font-size в HTML: em, rem, %, px

```
/* 1em е 10px, 0.8em е 8px, 1.6em е 16px */  
#main_content {font-size: 1.2em}  
H1 {font-size: 2em} /* displayed at 24px */  
H2 {font-size: 1.5em} /* displayed at 18px */  
H3 {font-size: 1.25em} /* displayed at 15px */  
H4 {font-size: 1em} /* displayed at 12px */
```

Когато оразмеряваме текста с **em** трябва да запомним едно правило: конкретният текст е релативен с неговия родител и се използва проста формула:

child pixels / parent pixels = child ems

Как и кога да използваме мерните единици за font-size в HTML: em, rem, %, px

```
body { font-size: 62.5%; } /* = от 16px става 10px */  
h1 { font-size: 2.4em; } /* =24px */  
p { font-size: 1.4em; } /* =14px */  
li { font-size: 1.4em; } /* =14px? */
```

Тук идва и проблемът, избирайки em-based font sizing нещата доста се усложняват. Списъкът в горното не е 14px, а 20px. За да избегнем това, трябва да зададем всички child елементи да използват 1em и така да опростим сметките.

Как и кога да използваме мерните единици за font-size в HTML: em, rem, %, px

```
body { font-size:62.5%; }  
h1 { font-size: 2.4em; } /* =24px */  
p { font-size: 1.4em; } /* =14px */  
li { font-size: 1.4em; } /* =14px? */  
li li, li p /* и т.н. */ { font-size: 1em; }
```

В общи линии тази сложна система действа отчайващо, така че какво може да направим?

Вече може да изберем новата мерна единица, придружаваща CSS3: **rem**

Как и кога да използваме мерните единици за font-size в HTML: em, rem, %, px

Единицата “**rem**” като име идва съкратено от “root em”. Ако това още не ви е приспало;), да обясним как работи тя.

Единицата **em** е зависима от font-size на родителския елемент, което причинява усложняването. Единицата **rem** е зависима от root—или html—елемента. Това означава, че можем да дефинираме определен font size на html елемента и после да зададем всички **rem** единици да са процент от него.

Как и кога да използваме мерните единици за font-size в HTML: em, rem, %, px

```
html { font-size: 62.5%; }
```

```
body { font-size: 1.4rem; } /* =14px */
```

```
h1 { font-size: 2.4rem; } /* =24px */
```

Във всички случаи, когато търсим fluid layout, достатъчно гъвкав единицата “**rem**” е идеалният избор.

Как и кога да използваме мерните единици за font-size в HTML: em, rem, %, px

Safari 5, Chrome, Firefox 3.6+, и даже Internet Explorer 9 поддържат **rem**. Но напр. Opera до 11.10 нямаше още имплементация за **rem.**, от 13.12.2011 вече и Opera поддържа **rem**.

За такива случаи трябва да предоставим fall-back алтернатива чрез **px**. Като първо дефинираме font-size в px и после отново, използвайки **rem**.

```
html { font-size: 62.5%; }
```

```
body { font-size: 14px; font-size: 1.4rem; } /* =14px */
```

```
h1 { font-size: 24px; font-size: 2.4rem; } /* =24px */
```

CSS3 преходи и трансформации

```
-webkit-transition: height 3s ease-in-out; /* За webkit браузъри */  
-moz-transition: height 3s ease-in-out; /* За Мозила */  
-o-transition: height 3s ease-in-out; /* За Опера */  
transition: height 3s ease-in-out; /* Уеб стандарт */
```

Параметрите са: свойство за прехода или ключовата дума all за всички, продължителност в секунди, - easing - ease, linear, ease-in, ease-out, ease-in-out

CSS3 преходи и трансформации

`-moz-transform: scale(2) rotate(2deg) translate(4px, 4px) skew(2deg, 2deg); /* За Мозила */`

`-webkit-transform: scale(2) rotate(2deg) translate(4px, 4px) skew(2deg, 2deg); /* За webkit браузъри */`

`-o-transform: scale(2) rotate(2deg) translate(4px, 4px) skew(2deg, 2deg); /* За Опера */`

`transform: scale(2) rotate(2deg) translate(4px, 4px) skew(2deg, 2deg); /* Уеб стандарт */`

CSS3 преходи и трансформации

Параметрите са:

Това свойство ни позволява да трансформираме елементите с познатите ни функции като:

scale /мащабитане на елемента/: 1 - елемента е без видима промяна, 1+ елемента е по-голям, $0 > x < 1$ - елемента е по-малък

rotate /завърта елемента на посочените градуси/

translate(x, y) /премества елемента на определена позиция/

skew (x, y) /смачкава елемента/

<http://www.css3maker.com/>

CSS Transitions, Transforms, Animations

TABLE 7.1 CSS3 Transition Properties

NAME	DESCRIPTION
transition-property	Specifies the name(s) of the transition to which the transition is to be applied (e.g., background, color, font-size, none, or all).
transition-duration	Defines the length of time in seconds that the transition will actually take. A value of 0s indicates that the transition is to take place immediately.
transition-timing-function	Specifies the function to be used in calculating the intermediate values during the transition. There are a number of predefined values for this property: ease, linear, ease-in, ease-out, ease-in-out, and cubic-bezier(), which allows you to define your own.
transition-delay	Defines when the transition will start. A value of 0s, the default setting, causes the transition to begin immediately.

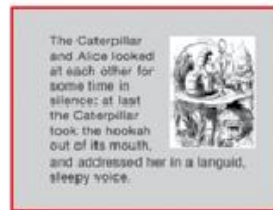
CSS 2D трансформации



original



skew



scale



translate



rotate



skewX



scaleX



translateX



rotateX



skewY



scaleY



translateY



rotateY