

Internet- технологии

ЛЕКЦИЯ №5

КАСКАДНЫЕ ТАБЛИЦЫ СТИЛЕЙ.
ПЕРЕМЕННЫЕ.
CSS-ПРЕПРОЦЕССОРЫ

Содержание

- Свойства CSS. Background, border, display.
- CSS Flexible Box Layout Module (Flexbox).
- Проверка поддержки стилевого свойства.
- Переменные (кастомные свойства).
- CSS-препроцессоры.



Сокращенные свойства. Пример

animation: eye 3s ease-in-out infinite;

Аналогично записи:

animation-name: eye

animation-duration: 3s

animation-timing-function: ease-in-out

animation-iteration-count: infinite

Параметры анимации можно описать отдельными свойствами:

- animation-name – имя анимации, связанное с правилом @keyframes.
- animation-duration – длительность одного цикла анимации.
- animation-timing-function – функция, согласно которой должна происходить анимация каждого цикла между ключевыми кадрами.
- animation-delay – время ожидания перед запуском цикла анимации.
- animation-iteration-count: 1 Можно использовать числа меньше единицы, для примера 0.5 будет означать половину цикла анимации.



Свойства. *background*

background позволяет установить одновременно несколько характеристик фона. Ни одно значение не является обязательным. Если помимо фоновых изображений требуется задать цвет фона элемента, он указывается в последнюю очередь после перечисления.

background-image	фоновое изображение для элемента
background-position	положение фонового изображения по горизонтали и вертикали.
background-repeat	режим повторения фонового изображения (no-repeat, repeat-x (аналогично repeat no-repeat))
background-attachment	режим прокрутки фонового изображения вместе с содержимым элемента.
background-color	цвет фона элемента.



Свойства. border

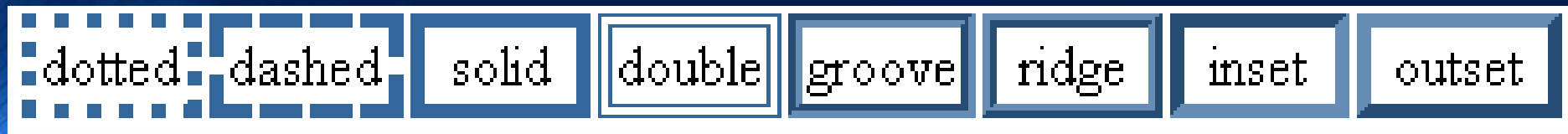
Универсальное свойство `border` позволяет одновременно установить толщину, стиль и цвет границы вокруг элемента. Для установки границы только на определённых сторонах элемента, воспользуйтесь свойствами `border-top`, `border-bottom`, `border-left`, `border-right`.

Пример:

```
border-top: 2px solid #95cd8b;
```

`border-width: 3px 7px 7px 4px;` // Разрешается использовать одно, два, три или четыре значения, разделяя их между собой пробелом. Эффект зависит от количества.

Возможные значения `border-style`: `none`, `hidden`, `dotted`, `dashed`, `solid`, `double`, `groove`, `ridge`, `inset`, `outset`.



Свойства. *display*

Свойство `display` определяет режим отображения элемента в документе. Наиболее распространенные значения этого свойства:

- `block` – элемент отображается как блочный (с новой строки).
- `inline` – элемент отображается как строковый, его содержимое выводится с того места, где окончился предыдущий элемент.
- `inline-block` – встраиваемый элемент (пример – ``). Его внутренняя часть форматируется как блочный элемент, а сам элемент как строчный.
- `list-item` – элемент выводится как блочный и добавляется маркер списка.
- `none` – временно удаляет элемент из документа.
- `flex` – элемент ведёт себя как блочный и выкладывает содержимое согласно **флекс-модели**.

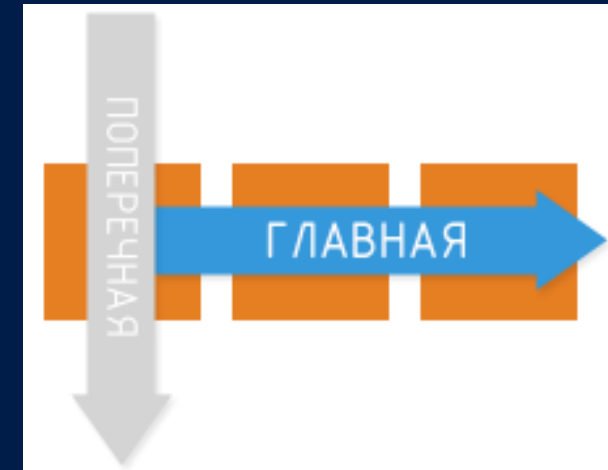


Flexbox

Flexbox позволяет контролировать размер, порядок и выравнивание элементов по нескольким осям и распределение свободного места между элементами. Flexbox определяет набор CSS свойств для контейнера (flex-контейнер, свойство `display:flex`) и его дочерних элементов (flex-блоков).

Главной осью flex-контейнера является направление, в соответствии с которым располагаются все его дочерние элементы, задается с помощью свойства `flex-direction`.

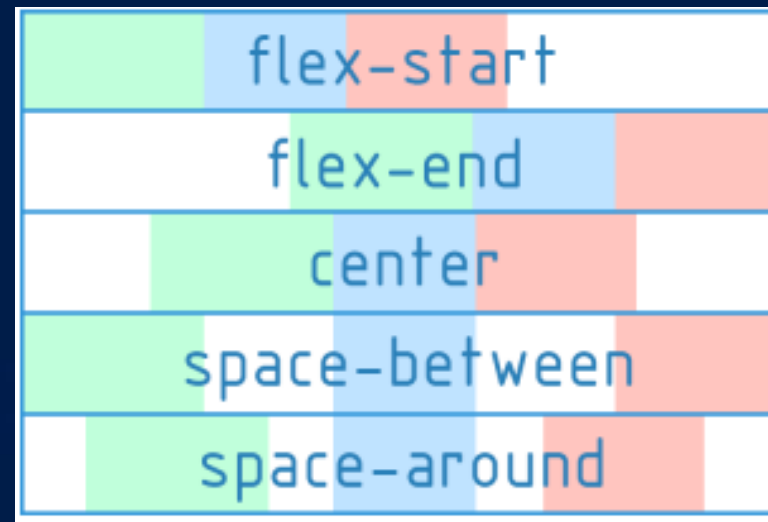
Поперечной осью называется направление, перпендикулярное главной оси. Свойство `justify-content` определяет то, как будут выровнены элементы вдоль главной оси, `align-items` – вдоль поперечной оси.



Flexbox

Основные преимущества CSS Flexible Box Layout Module (Flexbox):

- резиновая верстка всех блоков. Элементы могут сжиматься и растягиваться по заданным правилам, занимая нужное пространство.
- выравнивание по вертикали и горизонтали, базовой линии текста.
- возможность менять заданное в html расположение элементов.
- автоматическое размещение элементов занимая все свободное место на странице.
- возможность автоматического расположения элементов справа налево.



Поддержка стилевого свойства браузерами

Internet Explorer, устаревшее решение:

```
<!--[if gte IE 7]>
```

```
<style type="text/css">
```

```
P {
```

```
  color: red; /* Для браузера IE7 */
```

```
}
```

```
</style>
```

```
<![endif]-->
```

```
h1 {
```

```
  font-size: 26px;
```

```
}
```

```
@supports (font-size: calc(1rem + 12px))
```

```
{
```

```
  h1 {
```

```
    font-size: calc(1rem + 12px);
```

```
  }
```

```
}
```



CSS. Некоторые браузерные префиксы

Internet Explorer

- ms-interpolation-mode
- ms-linear-gradient
- ms-radial-gradient

Firefox

- moz-background-origin
- moz-background-size
- moz-border-left-colors
- moz-border-radius
- moz-box-shadow
- moz-box-sizing
- moz-linear-gradient
- moz-orient
- moz-radial-gradient
- moz-user-select

Opera

- o-background-size
- o-linear-gradient
- o-object-fit
- o-radial-gradient
- o-text-overflow

Safari и Chrome

- webkit-background-origin
- webkit-background-size
- webkit-border-radius
- webkit-box-shadow
- webkit-linear-gradient
- webkit-radial-gradient
- webkit-user-select



Переменные (кастомные свойства)

Переменные объявляются двумя дефисами
перед именем:
--variable-name

Объявленная
переменная доступна
всем селекторам
дочерних элементов
данного селектора.

```
.button {  
  background-color: gray;  
}  
.title {  
  color: gray;  
}  
.image-grid > .image {  
  border-color: gray;  
}  
.caption {  
  color: gray;  
}
```

```
:root {  
  --theme-color: gray;  
  --caption-text-color: gray;  
}  
.button {  
  background-color: var(--theme-color);  
}  
.title {  
  color: var(--theme-color);  
}  
.image-grid > .image {  
  border-color: var(--theme-color);  
}  
.caption {  
  color: var(--caption-text-color);  
}
```



Переменные (кастомные свойства)

Пример:

```
.test {  
  --box-shadow-color: yellow;  
  box-shadow: 0 0 30px var(--box-shadow-color);  
}  
  
.test:hover {  
  --box-shadow-color: orange;  
  /*box-shadow: 0 0 30px orange; */  
}
```

Для свойств, значения которых предполагают указание величин, они должны быть указаны в переменной, используя функцию calc.

```
:root{  
  --spacer: 10;  
}  
  
.box{  
  padding: var(--spacer)px 0; /* DOESN'T work */  
  padding: calc(var(--spacer)*1px) 0; /* WORKS */  
}
```



Значения-функции. *calc*

calc() – указание вычисляемого значения свойств, которые в качестве значений используют размер, угол, время, число. Это позволяет задавать значения, основанные на сложении или вычитании разных единиц измерений, например можно задать 100% - 20px. Если значение не может быть вычислено, оно игнорируется.

```
width: calc(20px + 20px);  
padding: calc(10% - 10px);  
height: calc(20%*2);  
width: calc(100%/3);
```



Переменные (кастомные свойства)

Значение по умолчанию указывается в функции var:

```
.shady {  
  font-size: var(--font-size, 12px);  
}
```

Переменные можно переопределять внутри медиавыражений.

```
@media (min-width: 320px) {  
  .shady {  
    --font-size: 48px;  
  }  
}
```



Переменные (кастомные свойства)

Стиль:

```
.container {
  --myBackColor: #foofo;
}

#first {
  background: var(--myBackColor);
}

#first-child {
  --myBackColor: red;
  background: var(--myBackColor);
}
```

Разметка:

```
<div class="container">
  <div id="first">
    div id="first"
    <div>&nbsp;&nbsp;&nbsp;&nbsp;</div>
    <div id="first-child">
      &nbsp;&nbsp;&nbsp;&nbsp;<div id="first-child" </div>
    </div>
    <div id="second"> div id="second"</div>
  </div>
```

```
div id="first"  
  div  
    div id="first-child"  
  div id="second"
```



Переменные (кастомные свойства). Пример

```
.btn {  
  --shadow-color: #777;  
  --gradient-from-color: #3498db;  
  --gradient-to-color: #2980b9;  
  --color: #ffffff;  
  
  background-image: linear-gradient(  
    to bottom,  
    var(--gradient-from-color),  
    var(--gradient-to-color)  
  );  
  border-radius: 28px;  
  color: var(--color);  
  padding: 10px 20px 10px 20px;  
}
```

```
body.inverted .btn{  
  --shadow-color: #888888;  
  --gradient-from-color: #CB6724;  
  --gradient-to-color: #D67F46;  
  --color: #000000;  
}
```



Переменные (кастомные свойства). Поддержка

```
@supports ( (--a: o)) {  
  /* supported */  
}  
@supports ( not (--a: o)) {  
  /* not supported */  
}
```

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
			49			10.2			
		55	60	10.1	47	10.3		4.4	
11	² 15	56	61	11	48	11	all	56	61
	16	57	62	TP	49				
		58	63		50				
		59	64						

*по данным <https://caniuse.com/#feat=css-variables>



CSS-препроцессоры. Sass

Sass (Syntactically Awesome Stylesheets) – препроцессор CSS, позволяющий создавать инструкции CSS с использованием программных конструкций вместо статических правил. Расширения SASS-файлов могут быть .sass и .scss – это зависит от выбранного синтаксиса.

SCSS:

```
$blue: #3bbfce; $margin: 16px;  
$fontSize: 14px;  
.content {  
    border: 1px solid $blue;  
}  
.border {  
    padding: $margin / 2;  
    margin: $margin / 2;  
    border-color: $blue;  
}
```

Результат в CSS:

```
.content {  
    border: 1px solid #3bbfce;  
    color: #217882;  
}  
.border {  
    padding: 8px;  
    margin: 8px;  
    border-color: #3bbfce;  
}
```



Sass. Примеры

Повторяющиеся участки кода можно сохранить в отдельной переменной и вставлять в нужных местах с помощью **@mixin**.

SCSS:

```
@mixin border-radius($radius) {  
  -webkit-border-radius: $radius;  
  -moz-border-radius: $radius;  
  border-radius: $radius;  
}  
.box-1 {  
  @include border-radius(10px);  
}  
.box-2 {  
  @include border-radius(5px);  
}
```

Результат в CSS:

```
.box-1 {  
  -webkit-border-radius: 10px;  
  -moz-border-radius: 10px;  
  border-radius: 10px;  
}  
.box-2 {  
  -webkit-border-radius: 5px;  
  -moz-border-radius: 5px;  
  border-radius: 5px;  
}
```



Sass. Компиляторы

Sass может использоваться как инструмент командной строки (необходимо установить Ruby). Для запуска Sass из командной строки используйте команду:

```
sass input.scss output.css
```

Также вы можете указать Sass следить за файлом и автоматически его компилировать в CSS при любом изменении:

```
sass --watch input.scss:output.css
```

Сторонние компиляторы:

- CodeKit (платный, Mac)
- Compass.app (платный, Win/Mac)
- Hammer (платный, Mac)
- Koala (**бесплатный**, Win/Mac)
- LiveReload (платный, Win/Mac)
- Mixture (**бесплатный**, Win/Mac)
- Scout (**бесплатный**, Win/Mac)

```
$globalVar: 10px; // global variable

.enclosing {
  $enclosingVar: 20px; // local variable

  .closure {
    $closureVar: 30px; // local variable

    font-size: $closureVar + $enclosingVar + $globalVar; // 60px
  }
}
```



Вопросы

- Зачем указывать background-color, указав background-image?
- Для чего используется свойство border-top? Возможно ли добиться того же эффекта, используя свойство border?
- Значения свойства display.
- Какие преимущества CSS Flexible Box Layout Module?
- Как проверить поддержку стилевого свойства браузерами?
- Назначение браузерных префиксов.
- Как определяются переменные в CSS?
- Для чего нужны CSS-препроцессоры?

