

# Homework #27

## 1

### Опишите все значения свойств:

- **@charset**

Правило @charset определяет кодировку символов, используемую в таблице стилей. Правило должно быть указано самым первым в таблице стилей и ему не должен предшествовать какой-либо символ, даже пробел. Если задано несколько правил @charset, то будет использоваться только первое. Правило может быть полезно при использовании не-ASCII символов в некоторых CSS-свойствах, например, content.

Не допускается одновременное задание кодировки в таблице стилей с помощью правила @charset и на HTML-странице внутри элемента <style>, или как значение атрибута свойства style.

- **@document**

@document устанавливает стилевые правила на основе адреса документа. К примеру, если адреса мобильной и основной версии сайта различаются, то мы можем задать собственный стиль элементов для каждой версии.

Значения

- url()

Указывает точный адрес страницы, для которой применяются стилевые правила. Адрес пишется внутри скобок url().

- url-prefix()

Значение, с которого начинается адрес документа.

- domain()

Домен или поддомен сайта.

- regexp()

Регулярное выражение которому соответствует адрес.

- **@font-face**

Правило @font-face позволяет подключать разнообразные пользовательские шрифты. Браузер загружает шрифты в кэш и использует их для оформления текста на странице. Такой подход

называется встраиванием шрифтов, а встроенные шрифты — веб-шрифтами.

Правило `@font-face` нужно размещать перед всеми остальными правилами CSS, так как этот прием улучшит производительность страницы. Загружаемые шрифты можно помещать в специальную папку `fonts`, созданную на сервере.

- **`@import`**

Правило осуществляет вставку содержимого таблицы стилей в определенный файл CSS. Чтобы правило корректно отработало, оно не должно стоять после других объявлений. Исключением является `@charset` и, естественно, само правило `@import`.

Значением для этого правила выступает путь к документу. После адреса указывается разновидность носителя. Ниже представлены значения, через который указывают, какой носитель используется

	Носитель	Пояснение
<b>all</b>		Соответствует всем возможным носителям. Это значения установлено по умолчанию
<b>aural</b>		Устройство, которое считывает информацию и преобразует ее в голосовой формат. В качестве примера выступает речевой браузер.
<b>braille</b>		Носитель, который работает по принципу Брайля. Такие устройства предназначены для людей, которые не видят.
<b>handheld</b>		Соответствует КПК и похожим устройствам.
<b>print</b>		Печатающие устройства.
<b>projection</b>		Проектор.
<b>screen</b>		Монитор.
<b>tv</b>		Телевизор.

- **`@keyframes`**

Правило `@keyframes` устанавливает ключевые кадры при анимации элемента. Ключевой кадр это свойства элемента (прозрачность, цвет, положение и др.), которые должны применяться к элементу в заданный момент времени. Таким образом, анимация представляет собой плавный переход стилизованных свойств от одного ключевого кадра к другому. Вычисление промежуточных значений между такими кадрами берёт на себя браузер.

Значения

➔ `<переменная>`

Уникальная переменная, которая связывает @keyframes с animation, через это свойство настраивается время анимации и другие её параметры.

→ from

Первый ключевой кадр, аналогичен 0%.

→ to

Последний ключевой кадр, аналогичен 100%.

→ <проценты>

Устанавливает ключевой кадр в процентах от времени всей анимации.

- **@media**

Наряду с типами носителей в CSS3 включена поддержка различных технических параметров устройств, на основе которых требуется загружать те или иные стили. К примеру, можно определить смартфон с максимальным разрешением 640 пикселей и для него установить одни стилевые свойства, а для остальных устройств другие. Также можно выявить различные характеристики вроде наличия монохромного экрана, ориентации (портретная или альбомная) и др. Все характеристики легко комбинируются, поэтому допустимо задать стиль только для устройств в альбомной ориентации с заданным разрешением экрана.

Возможности медиа-запросов не ограничиваются выявлением мобильных устройств, с их помощью можно создавать адаптивный макет. Такой макет подстраивается под разрешение монитора и окна браузера, меняя при необходимости ширину макета, число колонок, размеры изображений и текста. Медиа-запросы ограничивают ширину макета и при достижении этого значения (к примеру за счёт уменьшения окна или при просмотре на устройстве с указанным размером) уже применяется другой стиль.

Все запросы начинаются с правила [@media](#), после чего следует условие, в котором используются типы носителей, логические операторы и медиа-функции. Типы носителей перечислены в табл. 1.

Табл. 1. Типы носителей и их описание

Тип	Описание
all	Все типы. Это значение используется по умолчанию.
print	Принтеры и другие печатающие устройства.
screen	Экран монитора.
speech	Речевые синтезаторы, а также программы для воспроизведения текста вслух. Сюда, например, можно отнести речевые браузеры.

- **@page**

Правило @page позволяет задать значение полей при печати документа или для страниц, у которых тип носителя задан как print.

Значения

Значения

Для установки полей на странице используется универсальное свойство margin или его производные — **margin-top**, **margin-right**, **margin-bottom** и **margin-left**, задающие соответственно отступ сверху, справа, снизу и слева. Отрицательное значение допускается, но часть страницы может оказаться «отрезанной». Свойства **border** и **padding** не применимы.

Ключевое слово **:left** позволяет задать поля для всех левых страниц, **:right** — для всех правых, а **:first** — для первой страницы.

- **@viewport**

Позволяет оптимизировать макет веб-страницы в зависимости от различных устройств и их размеров. Является аналогом <meta name="viewport">, но переносит параметры из HTML в правила CSS.

Значения

Дескриптор	Допустимые значения	Описание
min-width	Любые единицы CSS для размера.	Минимальная ширина области просмотра.
max-width	Любые единицы CSS для размера.	Максимальная ширина области просмотра.
width	Любые единицы CSS для размера, а также device-width.	Одновременно устанавливает min-width и max-width.
min-height	Любые единицы CSS для размера.	Минимальная высота области просмотра.

max-height	Любые единицы CSS для размера.	Максимальная высота области просмотра.
height	Любые единицы CSS для размера, а также device-height.	Одновременно устанавливает min-height и max-height.
zoom	<ul style="list-style-type: none"> <li>• auto — браузер автоматически устанавливает масштаб, чтобы целиком отобразить документ;</li> <li>• &lt;число&gt; — положительное число определяет масштаб, значение 1.0 соответствует 100%;</li> <li>• &lt;проценты&gt; — масштаб можно задавать в процентах, 100% указывает, что масштабирование не требуется.</li> </ul>	Устанавливает начальный масштаб документа, может задаваться как положительное дробное число или в процентах. По умолчанию установлено значение auto.
min-zoom	Аналогично.	Устанавливает минимальный масштаб документа. По умолчанию установлено значение auto.
max-zoom	Аналогично.	Устанавливает максимальный масштаб документа. По умолчанию установлено значение auto.
user-zoom	<ul style="list-style-type: none"> <li>• zoom — пользователь может масштабировать документ;</li> <li>• fixed — пользователю запрещено изменять масштаб.</li> </ul>	Определяет, может пользователь масштабировать документ или нет. По умолчанию установлено значение zoom.
orientation	<ul style="list-style-type: none"> <li>• auto — устанавливает ориентацию автоматически на основе положения устройства;</li> <li>• portrait — документ должен быть зафиксирован в портретной ориентации;</li> <li>• landscape — документ должен быть</li> </ul>	Управляет ориентацией документа. По умолчанию установлено значение auto.

зафиксирован в  
альбомной ориентации.

## 2

### Опишите правила описания селекторов:

- [attr]

Обозначает элемент с атрибутом по имени attr

- [attr='value']

Обозначает элемент с именем атрибута attr и значением в точности совпадающим с value.

- [attr^='value']

Обозначает элемент с именем атрибута attr значение которого начинается с "value"

- [attr\$='value']

Обозначает элемент с именем атрибута attr чье значение заканчивается на "value"

- [attr\*='value']

Обозначает элемент с именем атрибута attr чье значение содержит по крайней мере одно вхождение строки "value" как подстроки.

- [attr~='value']

Обозначает элемент с именем атрибута attr значением которого является набор слов разделенных пробелами, одно из которых в точности равно value

- [attr|='value']

Обозначает элемент с именем атрибута attr. Его значение при этом может быть или в точности равно "value" или может начинаться с "value" со сразу же следующим "-" (U+002D). Это может быть использовано когда язык описывается с подкодом.

**Пример:**

```

1  /* Все span с атрибутом "lang" будут жирными */
2  span[lang] {font-weight:bold;}
3
4  /* Все span в Португалии будут зелеными */
5  span[lang="pt"] {color:green;}
6
7  /* Все span с американским английским будут синими */
8  span[lang~="en-us"] {color: blue;}
9
10 /* Любые span на китайском языке будут красными, как на упрощенном китайском (zh-CN) так и на традиционном (zh-TW) */
11 span[lang|="zh"] {color: red;}
12
13 /* Все внутренние ссылки будут иметь золотой фон */
14 a[href^="#"] {background-color:gold}
15
16 /* Все ссылки с url заканчивающимся на .cn будут красными */
17 a[href$=".cn"] {color: red;}
18
19 /* Все ссылки содержащие "example" в url будут иметь серый фон */
20 a[href*="example"] {background-color: #CCCCCC;}

```

### 3

## Приведите список свойств, к которым применима анимация.

### animation-name

Значения:

none	Означает отсутствие анимации. Также используется, чтобы отменить анимацию элемента из группы элементов, для которых задана анимация. Значение по умолчанию.
имя анимации	Имя анимации, которое связывает правило @keyframes с селектором.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

### animation-duration

Значения:

время	Указывает время, которое анимация занимает для завершения одного цикла. Отрицательные значения недействительны. Если время равно 0s, ключевые кадры анимации не действуют, но сама анимация происходит мгновенно. Значение по умолчанию 0s.
-------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## animation-timing-function

Значения:

linear

Линейная функция, анимация происходит равномерно на протяжении всего времени, без колебаний в скорости.

### функции Безье

ease

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

ease-in

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

ease-out

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

ease-in-out

Анимация 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. [На этом сайте](#) вы сможете построить любую траекторию скорости изменения анимации.

### пошаговые функции

step-start

Задаёт пошаговую анимацию, разбивая анимацию на отрезки, изменения происходят в начале каждого шага. Вычисляется в `steps(1, start)`.

step-end

Пошаговая анимация, изменения происходят в конце каждого шага. Вычисляется в `steps(1, end)`.



Ступенчатая временная функция, которая принимает два параметра. Первый параметр указывает количество интервалов в функции. Это должно быть положительное целое число больше 0, если вторым параметром не является `jump - none` — в этом случае оно должно быть положительным целым числом больше 1. Второй параметр, который является необязательным, указывает позицию шага — момент, в котором начинается анимация, используя одно из следующих значений:

`steps`(количество шагов, положение шага)

- `jump - start` — первый шаг происходит при значении 0
- `jump - end` — последний шаг происходит при значении 1
- `jump - none` — все шаги происходят в пределах диапазона (0, 1)
- `jump - both` — первый шаг происходит при значении 0, последний — при значении 1
- `start` — ведет себя как `jump - start`
- `end` — ведет себя как `jump - end`

Со значением `start` анимация начинается в начале каждого шага, со значением `end` — в конце каждого шага с задержкой. Задержка вычисляется как результат деления времени анимации на количество шагов. Если второй параметр не указан, используется значение по умолчанию `end`.

`initial`  
`inherit`

Устанавливает значение свойства в значение по умолчанию.  
Наследует значение свойства от родительского элемента.

### **animation-iteration-count**

Значения:

`infinite` Анимация проигрывается бесконечно.

число Анимация будет повторяться указанное количество раз. Если число не является целым числом, анимация закончится в середине последнего цикла.  
Отрицательные числа недействительны. Значение 0 вызывает мгновенное срабатывание анимации.

### animation-direction

Значения:

normal	Все повторы анимации воспроизводятся так, как указано. Значение по умолчанию.
reverse	Все повторы анимации воспроизводятся в обратном направлении от того, как они были определены.
alternate	Каждый нечетный повтор цикла анимации воспроизводятся в нормальном направлении, каждый четный повтор воспроизводится в обратном направлении.
alternate-reverse	Каждый нечетный повтор цикла анимации воспроизводятся в обратном направлении, каждый четный повтор воспроизводится в нормальном направлении.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

### animation-play-state

Значения:

running	Анимация выполняется. Значение по умолчанию.
paused	Анимация приостанавливается. При перезапуске анимация начинается с того места, где она была остановлена, как если бы «часы», управляющие анимацией, остановились и снова запустились. Если анимация остановлена во время задержки, при повторном воспроизведении время задержки также возобновляется.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

### animation-delay

Значения:

время	Время определяет длительность задержки между началом анимации (когда анимация применяется к элементу через свойства) и когда она начинает выполняться. Отрицательные значения разрешены, такая задержка начинает анимацию с определенного момента внутри её цикла, т.е. со времени, указанного в задержке. Это позволяет применять анимацию к нескольким элементам со сдвигом фазы, изменяя лишь время задержки. Чтобы анимация началась с середины, нужно задать отрицательную задержку, равную половине времени, установленном в <code>animation-duration</code> . Значение по умолчанию 0s.
-------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## animation-fill-mode

Значения:

none	Значение по умолчанию. Состояние элемента не меняется до или после воспроизведения анимации.
forwards	После того, как анимация заканчивается (как определено значением <code>animation-iteration-count</code> ), анимация будет применять значения свойств к моменту окончания анимации. Если <code>animation-iteration-count</code> больше нуля, применяются значения для конца последней завершенной итерации анимации (а не значения для начала итерации, которое будет следующим). Если значение <code>animation-iteration-count</code> равно нулю, применяемыми значениями будут те, которые начнут первую итерацию (так же, как и в режиме <code>animation-fill-mode: backwards</code> );).
backwards	В течение периода, определенного с помощью <code>animation-delay</code> , анимация будет применять значения свойств, определенные в ключевом кадре, которые начнут первую итерацию анимации. Это либо значения ключевого кадра <code>from</code> (когда <code>animation-direction: normal</code> или <code>animation-direction: alternate</code> ), либо значения ключевого кадра <code>to</code> (когда <code>animation-direction: reverse</code> или <code>animation-direction: alternate</code> ).
both	Позволяет оставлять элемент в первом ключевом кадре до начала анимации (игнорируя положительное значение задержки) и задерживать на последнем кадре до конца последней анимации.

---

## 4

### Как и для чего применять свойство `order`?

CSS свойство **order** определяет порядок, используемый для размещения flex элементов в их flex контейнере. Элементы располагаются в восходящем порядке по значению **order**. Элементы с одинаковым значением **order** располагаются в том порядке, в каком они находятся в исходном коде.

---

## 5

### Какое свойство отвечает за правила выделения объекта или текста в объекте?

Псевдоэлемент **::selection** позволяет применить стили к части документа, который был выделен пользователем (например, с помощью мыши).

Разрешённые свойства

Только небольшое подмножество CSS-свойств может быть использовано с псевдоэлементом **::selection** :

- color
- background-color
- cursor
- caret-color
- outline и его длинные записи
- text-decoration и связанные свойства
- text-emphasis-color
- text-shadow

**Синтаксис:**

```
/* синтаксис Firefox */  
::-moz-selection
```

```
::selection
```