

Лабораторная работа № 10

Тема работы: «Разработка второго варианта таблиц стилей для HTML документа»

1. Цель работы

Формирование умений модификации таблиц стилей для создания альтернативного оформления страниц, без редактирования HTML-документа.

2. Задание

Сверстать HTML-страницу используя наглядный макет psd, используя средства HTML и CSS.

3. Оснащение работы

ПК, текстовый редактор, браузер.

4. Основные теоретические сведения

Блочная модель. В блочной модели элемент рассматривается как прямоугольный контейнер, имеющий область содержимого и необязательные рамки и отступы (внутренние и внешние). Свойство `display` определяет тип контейнера элемента. Для каждого элемента существует значение браузера по умолчанию. Пример визуализации представлен на рисунке 10.1.

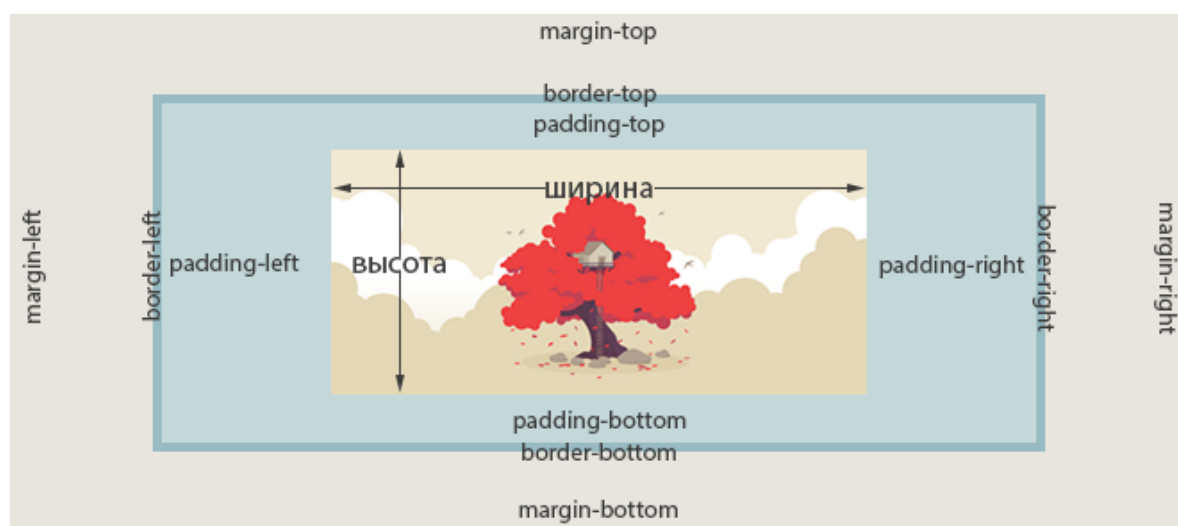


Рисунок 10.1 – Блочная модель элемента

Область содержимого — это содержимое элемента, например, текст или изображение.

Внутренний отступ задаётся свойством `padding`. Внутренний отступ — это расстояние между основным содержимым и его границей (рамкой). Если для элемента задать фон, то он распространится также и на поля элемента.

Внутренний отступ не может принимать отрицательных значений, в отличие от внешнего отступа.

Внешний отступ задаётся свойством `margin`. Он добавляет расстояние снаружи элемента от внешней границы рамки до соседних элементов, тем самым разделяя элементы на странице. Внешние отступы всегда остаются прозрачными и через них виден фон родительского элемента.

Значения `padding` и `margin` задаются в следующем порядке: верхнее, правое, нижнее и левое.

Граница, или рамка элемента, задаётся с помощью свойства `border`. Если цвет рамки не задан, она принимает цвет основного содержимого элемента, например, текста. Если рамка имеет разрывы, то сквозь них будет проступать фон элемента.

Внешние, внутренние отступы и рамка элемента не являются обязательными, по умолчанию их значение равно нулю. Тем не менее, некоторые браузеры добавляют этим свойствам положительные значения по умолчанию на основе своих таблиц стилей. Очистить стили браузеров для всех элементов можно при помощи универсального селектора:

```
* {  
  margin: 0;  
  padding: 0;  
}
```

Создание медиа-запросов.

В общем случае медиа-запрос состоит из ключевого слова, описывающего тип устройства (необязательный параметр) и выражения, проверяющего характеристики данного устройства. Из всех характеристик чаще всего проверяется ширина устройства `width`. Медиа-запрос является логическим выражением, которое возвращает истину или ложь.

Медиа-запросы могут быть добавлены следующими способами:

1) С помощью HTML:

```
<link rel="stylesheet" media="screen and (color)" href="example.css">
```

2) С помощью правила `@import` внутри элемента `<style>` или внешней таблицы стилей:

```
@import url(color.css) screen and (color);
```

3) Непосредственно в коде страницы:

```
<style>
```

```
@media (max-width: 600px) {  
  #sidebar {display: none;}
```

```
}  
</style>
```

4) Внутри таблицы стилей style.css:

```
@media (max-width: 600px) {  
  #sidebar {display: none;}}
```

Таблица стилей, прикрепленная через тег <link>, будет загружаться вместе с документом, даже если её медиа-запрос вернет ложь.

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

Оператор and связывает друг с другом разные условия:

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

Стили этого запроса будут применяться только для экранных устройств с шириной области просмотра не более 600px.

```
@media (min-width: 600px) and (max-width: 800px) {  
  /* CSS-стили */;  
}
```

Стили этого запроса будут применяться для всех устройств при ширине области просмотра от 600px до 800px включительно.

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

Оператор запятая работает по аналогии с логическим оператором or.

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

В данном случае CSS-стили, заключенные в фигурные скобки, сработают только для экранных или проекционных устройств.

Оператор not позволяет сработать медиазапросу в противоположном случае. Ключевое слово not добавляется в начало медиазапроса и применяется ко всему запросу целиком, т.е. запрос

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

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

Если медиазапрос составлен с использованием оператора запятая, то отрицание будет распространяться только на ту часть, которая идет до запятой, т.е. запрос

@media not screen and (color), print and (color)

будет эквивалентен запросу

@media (not (screen and (color))), print and (color)

Оператор only используется, чтобы скрыть стили от старых браузеров (поддерживающих синтаксис медиа-запросов CSS2).

media="only screen and (min-width: 401px) and (max-width: 600px)"

Эти браузеры ожидают список медиа-типов, разделённых запятыми. И, согласно спецификации, они должны отсекают каждое значение непосредственно перед первым неалфавитно-цифровым символом, который не является дефисом. Таким образом, старый браузер должен интерпретировать предыдущий пример как media="only". Поскольку данного типа медиа-типа не существует, то и таблицы стилей будут игнорироваться.

Тип носителя представляет собой тип устройства, например, принтеры, экраны. Более подробное описание представлено в таблице 10.1.

Таблица 10.1 – Типы носителей

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

Метатег viewport. Для управления разметкой в мобильных браузерах используется метатег viewport. Изначально данный тег был представлен разработчиками Apple для браузера Safari на iOS. Мобильные браузеры отображают страницы в виртуальном окне просмотра, которое обычно шире, чем экран устройства. С помощью метатега viewport можно контролировать размер окна просмотра и масштаб.

Страницы, адаптированные для просмотра на разных типах устройств, должны содержать в разделе <head> метатег viewport.

<meta name="viewport" content="width=device-width, initial-scale=1.0">

Свойство width определяет виртуальную ширину окна просмотра, значение device-width — физическую ширину устройства. Другими словами, width отражает значение document.documentElement.clientWidth, а device-width — screen.width. Разница между width и device-width представлена на рисунке 10.2



Рисунок 10.2 – Разница между width и device-width

При первой загрузке страницы свойство `initial-scale` управляет начальным уровнем масштабирования, `initial-scale=1` означает, что 1 пиксель окна просмотра = 1 пиксель CSS.

На какие размеры экрана нужно ориентироваться

При составлении медиазапросов нужно ориентироваться на так называемые переломные (контрольные) точки дизайна, т.е. такие значения ширины области просмотра, в которых дизайн сайта существенно меняется, например, появляется горизонтальная полоса прокрутки. Чтобы определить эти точки, нужно открыть сайт в браузере и постепенно уменьшать область просмотра.

```
/* 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 */
}
```

```

}
/* 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 */
}
/* iPad 3*****/
@media only screen and (min-width: 768px) and (max-width: 1024px) and
(orientation: landscape) and (-webkit-min-device-pixel-ratio: 2) {
/* стили */
}
@media only screen and (min-width: 768px) and (max-width: 1024px) and
(orientation: portrait) and (-webkit-min-device-pixel-ratio: 2) {
/* стили */
}
/* Настольные компьютеры и ноутбуки ----- */
@media only screen and (min-width: 1224px) {
/* стили */
}
/* Большие экраны ----- */
@media only screen and (min-width: 1824px) {
/* стили */
}
/* iPhone 4 ----- */
@media only screen and (min-width: 320px) and (max-width: 480px) and
(orientation: landscape) and (-webkit-min-device-pixel-ratio: 2) {
/* стили */
}
@media only screen and (min-width: 320px) and (max-width: 480px) and
(orientation: portrait) and (-webkit-min-device-pixel-ratio: 2) {

```

```

/* стили */
}
/* iPhone 5 ----- */
@media only screen and (min-width: 320px) and (max-height: 568px) and
(orientation: landscape) and (-webkit-device-pixel-ratio: 2){
/* стили */
}
@media only screen and (min-width: 320px) and (max-height: 568px) and
(orientation: portrait) and (-webkit-device-pixel-ratio: 2){
/* стили */
}
/* iPhone 6 ----- */
@media only screen and (min-width: 375px) and (max-height: 667px) and
(orientation: landscape) and (-webkit-device-pixel-ratio: 2){
/* стили */
}
@media only screen and (min-width: 375px) and (max-height: 667px) and
(orientation: portrait) and (-webkit-device-pixel-ratio: 2){
/* стили */
}
/* iPhone 6+ ----- */
@media only screen and (min-width: 414px) and (max-height: 736px) and
(orientation: landscape) and (-webkit-device-pixel-ratio: 2){
/* стили */
}
@media only screen and (min-width: 414px) and (max-height: 736px) and
(orientation: portrait) and (-webkit-device-pixel-ratio: 2){
/* стили */
}
/* Samsung Galaxy S3 ----- */
@media only screen and (min-width: 320px) and (max-height: 640px) and
(orientation: landscape) and (-webkit-device-pixel-ratio: 2){
/* стили */
}
@media only screen and (min-width: 320px) and (max-height: 640px) and
(orientation: portrait) and (-webkit-device-pixel-ratio: 2){
/* стили */
}
/* Samsung Galaxy S4 ----- */

```

@media only screen and (min-width: 320px) and (max-height: 640px) and (orientation: landscape) and (-webkit-device-pixel-ratio: 3){

/* стили */

}

@media only screen and (min-width: 320px) and (max-height: 640px) and (orientation: portrait) and (-webkit-device-pixel-ratio: 3){

/* стили */

}

/* Samsung Galaxy S5 ----- */

@media only screen and (min-width: 360px) and (max-height: 640px) and (orientation: landscape) and (-webkit-device-pixel-ratio: 3){

/* стили */

}

@media only screen and (min-width: 360px) and (max-height: 640px) and (orientation: portrait) and (-webkit-device-pixel-ratio: 3){

/* стили */

}

5. Порядок выполнения работы

1. В заданиях к лабораторной работе №9 написать 3 разных медиа-запроса.

6. Форма отчета о работе

Лабораторная работа № ____

Номер учебной группы _____

Фамилия, инициалы учащегося _____

Дата выполнения работы _____

Тема работы: _____

Цель работы: _____

Оснащение работы: _____

Результат выполнения работы:

7. Контрольные вопросы

1. Перечислите способы использования каскадных таблиц стилей. Укажите преимущества и недостатки каждого из них.

2. Как создать медиа-запрос?

3. Для чего нужны медиа-запросы?

4. Какие бывают виды медиа-запросов?

8. Рекомендуемая литература

1. **Макфарланд, Д.** Новая большая книга CSS / Дэвид Макфарланд. – СПб.: Питер, 2016. – 720с.
2. **Никсон, Р.** Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5 /Р. Никсон. – 4-е изд. – СПб.: ООО «ПИТЕР М», 2017. – 768 с.
3. **Прохоренок, Н.А.** HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера / Н.А.Прохоренок. – СПб.: БХВ-Петербург, 2010. – 912с.
4. **Робсон, Э.** Изучаем HTML, XHTML и CSS / Э. Робсон. – 2-е изд. – СПб.: ООО «ПИТЕР М», 2017. – 720 с.
5. **Фрейн, Б.** HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств / Бен Фрейн. – СПб.: Питер Пресс, 2017. — 272с.