## Практическая работа 3. Медиа-запросы

*Цель работы:* осуществитьадаптивную верстку сайта.

*Задачи работы:*

1. Изучить синтаксис написания медиа запросов.
2. Научиться писать медиа запросы.
3. Научиться использовать подход desktop first к адаптивной верстке.
4. Сделать одностраничный сайт адаптивным.

*Инструменты:*Visual Studio Code, браузер Google Chrome.

В данной работе воспользуемся сверстанным сайтом из шестой практической работы. Так как у нас сверстана версия для desktop, то будем верстать по принципу desktop first и пойдем от больших контрольных точек к меньшим используя функцию width-max.

Так как в графическом макете не представлены контрольные точки для мобильных устройств, продумаем изменение макета при уменьшении ширины экрана самостоятельно.

На протяжении всей работы необходимо просматривать внешний вид сайта на разных устройствах, для это воспользуемся инструментами разработчика, а именно Toggle device toolbar (рис. 121). С помощью кнопки  на панели разработчика можно запустить режим просмотра страницы на устройствах разной ширины. С помощью маркеров и панели управления можно менять ширину области просмотра страницы. Следует отметить, что данный инструмент отражает лишь особенность отображения элементов страницы при разной ширине экрана и не отражает особенности их отображения в зависимости от настроек разных операционных систем и устройств. Таким образом, данный инструмент не должен быть единственным инструментом тестирования адаптивной верстки.

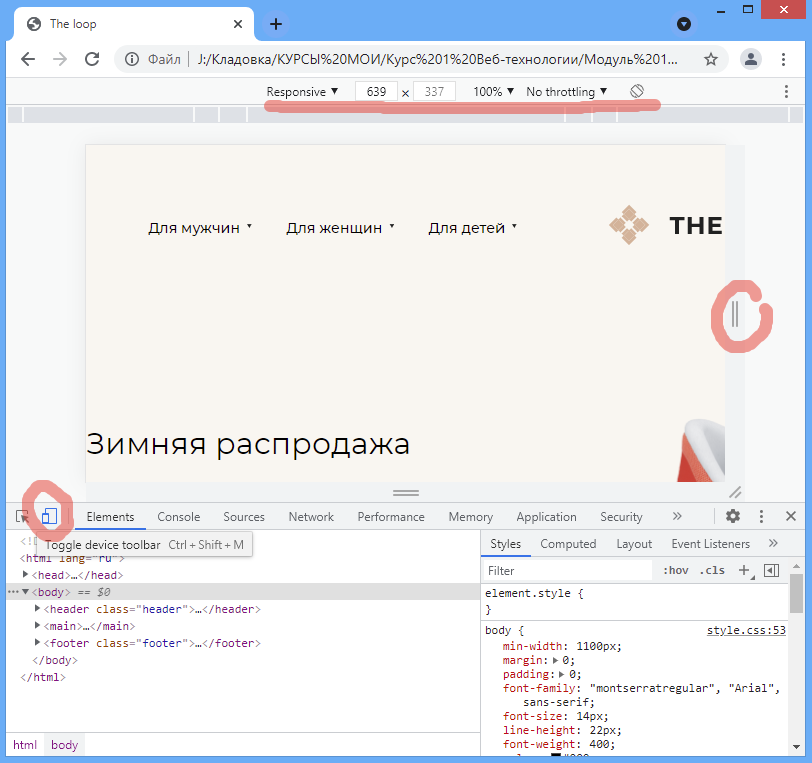
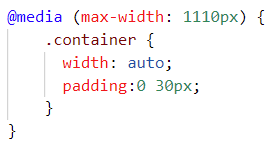


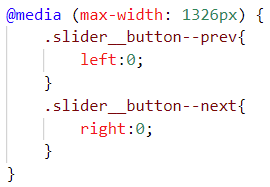
Рисунок 121 – Панель переключения устройств

**Задание 1. Адаптивная верстка шапки сайта**

Сейчас видно, что при ширине экрана меньше 1326px у сайта появляется горизонтальная полоса прокрутки. Это связано с тем что у блоков с классом container задана минимальная ширина 1110px плюс управляющие кнопки слайдера, выходящие за пределы контейнера.

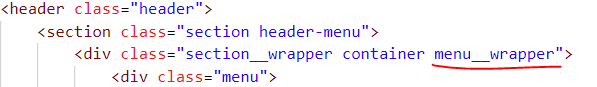
1. Для стиля body измените значение минимальной ширины на 320px.
2. Задайте медиа запрос для класса container:



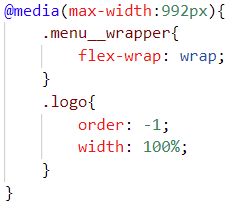
1. А для кнопок управления слайдером для той же контрольной точки задайте стили, которые сместят эти кнопки внутрь слайдера:
2. Просмотрите результат работы.
3. На контрольной точке 768px вообще скройте баннер:



1. Адаптируйте главное меню, для этого задайте дополнительный класс для секции меню:



1. Для контрольной точки 992px для главного меню создайте медиа запрос, который разрешит переносить элементы меню на разные строки и изменит порядок отображения элементов меню:



1. Просмотрите результат (рис. 122).

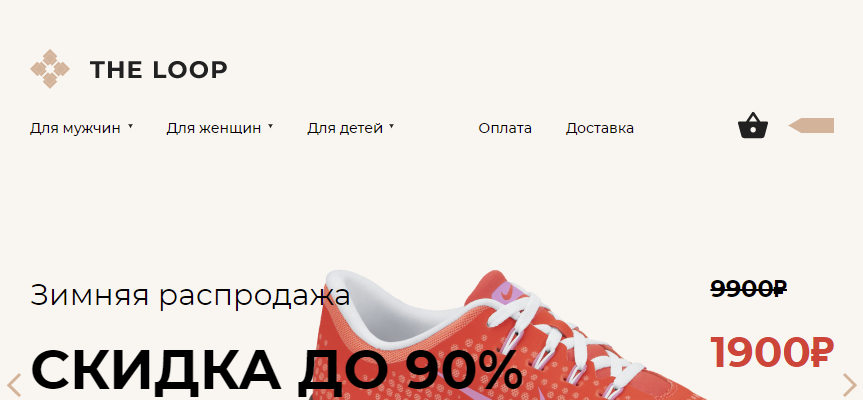
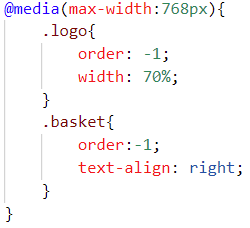


Рисунок 122 – Вид шапки сайта на контрольной точке 992px

1. Для контрольной точки 768px поднимите вверх изображения корзины (справа). Важно задать эти стили после стилей из пункта 8.



1. Просмотрите результат (рис. 123).

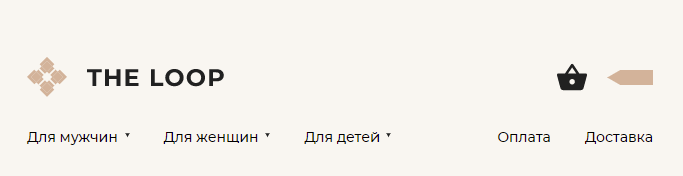


Рисунок 123 – Вид шапки сайта на контрольной точке 768px

1. Чтобы пункты меню были удобны для клика на мобильных устройствах, добавьте вертикальные padding-и к элементам-ссылкам (рис. 124). Величина padding-ов рассчитывается так: ширины и высота элементов должна быть минимум 44px минус размер шрифта (14px), полученное число (30px) поделить пополам, итого по 15px отступы сверху и снизу.

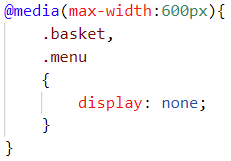
 

а б

Рисунок 124 – Область клика изначально (а) и

после добавления padding-ов (б)

1. Для контрольной точки 600px скройте все элементы шапки, кроме логотипа. Важно задать эти стили после стилей из пункта 9.



1. Просмотрите результат (рис. 125).

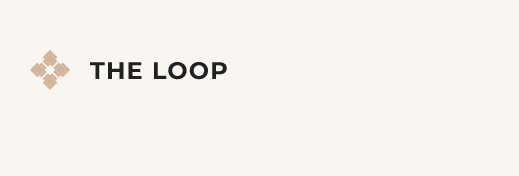
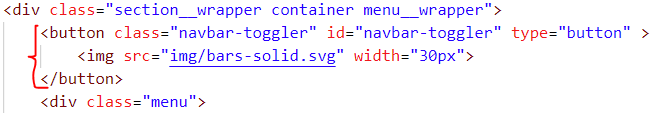


Рисунок 125 – Вид шапки сайта на контрольной точке 600px

**Задание 2. Создание меню-«гамбургер»**

1. Найдите на сайте fontawesome.com иконку bars .
2. Добавьте иконку в верстку следующим образом:



1. Просмотрите результат (рис. 126), должна появится кнопка меню.

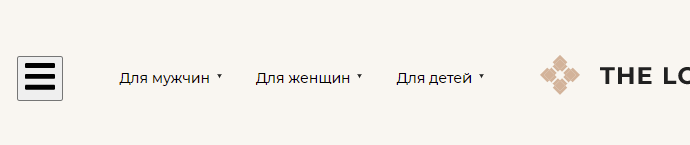
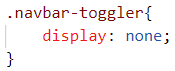


Рисунок 126 – Кнопка-«гамбургер»

1. Скройте кнопку с помощью стилей:



1. В медиа-запросе (пункт 12 из задания номер 1) сделайте кнопку видимой:



1. Просмотрите результат (рис. 127).

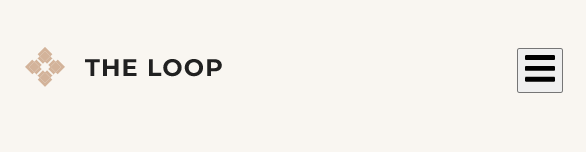
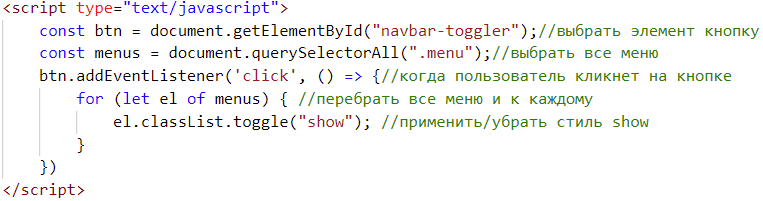
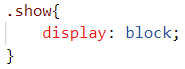


Рисунок 127 – Появление кнопки-«гамбургер» на узких экранах

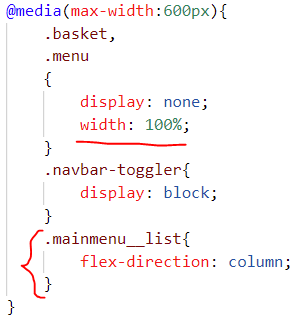
1. Для того, чтобы меню работало (выпадало) необходимо написать скрипт в конце HTML-страницы перед закрывающимся тегом body:



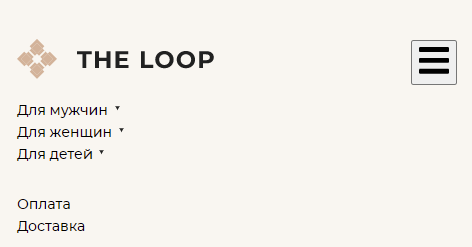
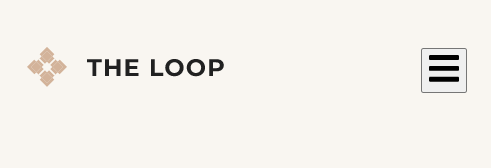
1. Добавьте стиль show обязательно после медиа запросов, связанных с главным меню:



1. Добавьте стили в медиа запрос из пункта 5 этого задания:



1. Просмотрите результат работы меню (рис. 128).

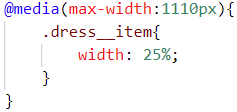


а б

Рисунок 128 – Свернутое (а) и развернутое (б) меню

**Задание 3. Адаптивная верстка секции dress**

1. Просмотрите как меняется расположение элементов в секции dress. Даже без медиа запросов раздел адаптируется к ширине окна благодаря использованию flexbox и правила flexwrap:wrap.
2. Немного изменим поведение блоков. Для контрольной точки 1110px измените значение ширины блоков с абсолютных единиц на относительные:

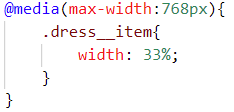


1. Просмотрите результат (рис. 129) – блоки больше не перестраиваются (не уменьшается количество блоков в ряду), а изменяется ширина блоков.



Рисунок 129 – Относительное задание размеров

1. Для контрольной точки 768px измените значение ширины блока, чтобы в ряд вставало по три блока:



1. По аналогии задайте для контрольной точки 576px – два блока в ряд, а для контрольной точки 320px – один блок в ряд.
2. Просмотрите результат работы.

**Задания для самостоятельного выполнения:**

1. Адаптируйте все остальные секции сайта из практической работы №6.
2. Проверьте, что у всех кликабельных элементов задан псевдо класс active. Если нет, то добавьте его.
3. Проверьте, что размеры всех кликабельных элементов не меньше 44 на 44px. Если есть элементы меньших размеров, то увеличьте кликабельную область за счет padding-ов у ссылок и кнопок.
4. Для узких экранов (мобильных устройств) увеличьте размеры (особенно высоту) элементов форм.