- HTML syntax

• HTML semantics and markup tags

**Header** (вводная часть смыслового раздела или сайта, повторяется на всех страницах), **nav**(ссылки на части страницы или другие страницы сайта), **main**(только один для неповторяемой информации на страницах сайт, не включать повторяющиеся вещи – футтеры и хэдеры), **article** (твиты, статьи, комментарии, посты – независимая, отделяемая смысловая единицы страницы). **Section** (смысловой раздел документа, неотделяемая часть страницы – блоки, для группировки элементов к одной логической секции на странице), **aside**(косвенно связанная с информацией на странице часть), **footer** (заключительная часть смыслового раздела или страницы).

• HTML Adoptive markup

**<meta name="viewport" content="width=device-width, initial-scale=1">**

Content – показывает, что ширина сайта будет равна ширине девайса (это не вычитание)

Initial-scale – масштаб 1 к 1 будет, без зума.

Медиа-запросы

Флексбоксы и гриды

• HTML Types of elements positioning: normal, fixed, absolute, relative, sticky

• CSS syntax and common attributes

• HTML/CSS layout techniques: flexbox vs grid and differences between them

Макеты Веб-дизайна. Флексбокс придуман для одномерных макетов, а Грид – для двухмерных макетов. Флексбокс делает акцент на содержимом, а грид – на макете.

Одномерный – может управлять элементами только по горизонтали или только по вертикали. Двухмерный – по вертикали и горизонтали сразу.

Грид выравнивает элементы в строках и колонках.

Флексбокс

• CSS points: REM, EM, PX, PT, CM, IN

**Абсолютные величины – такие размеры, как написано**

1cm – 38px

1mm – 3.8px

1pt – 4/3 px

1px – точное и конкретное задание размеров

1in – 96px

**Относительные – зависят от чего-то**

1rem – относительно шрифта html{font-size:15px}

Em – относительный, определяются по текущему контексту. Если перед дивом был див с размером 1.5ем, тогда див внутри со значением 1.5 ем будет в 1.5 раза больше родительского.

% - берется от значения такого же свойства у родителя, но не всегда. В line-height берется от текущего размера шрифта. В margin-left берется от ширины родительского блока. Width/height обычно берутся от ширины и высоты родителя, но если position: fixed – тогда берется от размера окна

vw,vw,vmin,vmax – относительно размера экрана

• CSS breakpoints

Медиа-запросы всегда находятся в конце CSS файла

**@media screen/print/all/speech and (max-width/min-width)**

• JavaScript mounting and types of scripts loading: default, async, defer

**Defer** предназначен только для внешних скриптов – если нет src в теге script, тогда дефер просто проигнорируется. Вообще, он предназначен, чтобы сказать браузеру, чтобы тот загружал дальше страницу, при это загружая скрипты. Он начнет выполнять скрипты после того, как весь ДОМ будет отрендерен. При этом сохранит последовательность скриптов. Единственное, это все будет сделано до события DOMContentLoaded.

**Async** – только для внешних скриптов. Он будет проигнорирован, если нет src в теге script. Скрипт абсолютно независим: страница не ждет асинхронных скриптов, а обрабатывается и отображается; событие DOMContentLoaded и асинхронные скрипты не ждут друг друга (если скрипт загрузиться раньше, если обработается раньше ДОМа); остальные скрипты не ждут async и он не ждет никого (кто первый загрузился, тот первый и выполнится).

• Interaction with user components and event listeners implementation

**Event handler** – аттрибуты, которые добавляют к хтмл тегу. Только один обработчик ивента может быть привязан к онклик одного элемента.

**Event Listeners** – много листенеров может быть добавлено для одного ивента.

Удалять ивент листенеры, когда уже не нужны, чтобы не засорять память.

What is CVS and CVS revision

CVS – система контроля версий. С ее помощью я могу записать историю изменений файлов. С помощью такой системы я могу обратиться к старым версиям, чтобы точно выяснить, что привело к ошибке. Конечно, можно хранить каждые измененные файлы, но для этого надо много места. CVS же делает все в одном файле – просто вносит изменения в некоторые места, а не в весь файл. Можно использовать, когда работаешь не один на проекте. Каждый разработчик работает в своем каталоге, а система уже объединяет полученные результаты.

CVS revisions are the version numbers that CVS assigns to files. All files in CVS have a corresponding revision, and whenever you commit a change, the revision number of the changed files is incremented. (начинается с 1.1 и идет дальше – первая 1 никогда не будет двойкой, будет такое 1.123)

• Basic Git commands: Add, Remove, Rename, Commit, Amend, Cherry-pick and Revert

git add – добавляет содержимое рабочего каталога в индекс (staging area) для последующего коммита. Гит коммит использует только этот индекс, поэтому я могу использовать гит адд для сборки слепка моего след коммита.

git commit – берет все данные из индекса, куда я их добавил через гит адд, и сохраняет их слепок во внутренней базе данных, а затем сдвигает указатель текущей ветки на этот слепок.

git rename (git mv) – git mv old.txt new.text. После чего надо сделать коммит, так как я сохранил в рабочем каталоге

git status – показывает состояние файлов в рабочем каталоге и индексе: какие файлы изменены, но не добавлены в индекс; какие ожидают коммита в индексе. Так же показываются подсказки о том, как изменить состояние файлов.

git rm – git rm deleteing.txt – добавление в индекс перед коммитом. Добавить флаг –r после rm для удаления папки.

git commit - - amend - - no-edit – изменение последнего коммита без изменения текста (no-cache). Это делается на основе того, есть ли что-то в индексе после git add. Если там есть файл, тогда он будет добавлен в измененном коммите, если нет, то можно просто изменить текст –m коммита. Коммит будет полностью заменен и старый, который мы изменили, не будет виден вообще.

git cherry-pick commitSHA(берется из git log) – помогает взять коммит из другой ветки и перекинуть на текущую. Чтобы это сработало, working tree должно быть чистым (без изменений из ветки, в которую перетягиваю коммит). Чистое рабочее дерево – это когда моя рабочая директория такие же, как и мой последний коммит в гит репозиторий. Нет untracked files, modified files or files that need to be commited.

??Git revert ??

• GIT: local repository vs remote repository and git branches

• Git commands to operate local/remote repos: Init, Fetch, Push, Remote

• Mering Git commands and git flow: Merge, Reverse-Merge, Rebase, Squash