1. **Фиксированная вёрстка** (также она называется статической) **представляет** собой такой вид дизайна **страницы**, в котором размеры указываются точно в пикселях (px) и являются статичными вне зависимости от размеров экрана. ... Такая **страница** с резиновым дизайном займёт всю ширину окна браузера
2. «Резиновым дизайном» будем называть структуру веб-страницы, которая автоматически приспосабливается под определенную ширину окна браузера пользователя.
3. **Адаптивный дизайн** – это конфигурация, в которой сервер отправляет одинаковый HTML-код на любые устройства, а размеры элементов страницы корректируются с помощью CSS.
4. Используя flexbox-вёрстку, и медиа запросы
5. **Медиа-запросы** (media queries) – это правила CSS, которые позволяют управлять стилями элементов в зависимости от значений технических параметров устройств.
6. С помощью правила **@media**
7. Правило @import позволяет импортировать содержимое CSS-файла в текущую стилевую таблицу. @import не разрешается вставлять после любых объявлений кроме @charset или другого @import.
8. –
9. Свойство [CSS](https://developer.mozilla.org/ru/docs/Web/CSS) **flex** - это [сокращённое свойство](https://developer.mozilla.org/ru/docs/Web/CSS/Shorthand_properties), определяющее способность гибкого элемента растягиваться или сжиматься для заполнения собой доступного свободного пространства. Это свойство устанавливает [flex-grow](https://developer.mozilla.org/ru/docs/Web/CSS/flex-grow), [flex-shrink](https://developer.mozilla.org/ru/docs/Web/CSS/flex-shrink) и [flex-basis](https://developer.mozilla.org/ru/docs/Web/CSS/flex-basis).
10. –
11. –
12. Что означает flex: 1 2 200px;?

[flex-grow](https://developer.mozilla.org/ru/docs/Web/CSS/flex-grow)=1

CSS-свойство **flex**-**grow** определяет как много свободного пространства во **flex**-контейнере должно быть назначено текущему элементу (**flex grow** factor — «коэффициент увеличения **flex**»). Свободное пространство — разница между размером **flex**-контейнера и размером всех его **flex**-элементов вместе

[flex-shrink](https://developer.mozilla.org/ru/docs/Web/CSS/flex-shrink)=2

**flex-shrink**- свойство  [CSS](https://developer.mozilla.org/en-US/docs/Web/CSS) , которое определяет фактор сжатия flex-элемента. Flex-элементы будут заполнять контейнер в зависимости от значений  flex-shrink, когда стандартная ширина flex-элементов шире, чем flex-контейнер.

f[lex-basis](https://developer.mozilla.org/ru/docs/Web/CSS/flex-basis).=300px

[CSS](https://developer.mozilla.org/en-US/docs/Web/CSS)  свойство **flex-basis** задаёт базовый размер флекс элемента по **основной оси** . Это свойство определяет размер контента-бокса

1. Свойство **flex**-**flow** является сокращенной формой записи свойств **flex**-direction и **flex**-wrap и позволяет за одну декларацию задать направление главной оси и возможность переноса флекс-элементов на новую строку (т. е. многострочность поперечной оси). Значение по умолчанию "**flex**-**flow**: row nowrap;"
2. **flex**-**direction** - CSS свойство , указывающее на то,как **flex**-элементы располагаются во **flex**-контейнере по главной оси и направлению (normal или reversed). Обратите внимание, что значения row и row-reverse зависят от направления **flex**-контейнера.
3. **Align-items**

Данное свойство - позволяет выравнивать элементы по вертикали. Применимо только к контейнеру. По умолчанию равно stretch. При этом свойстве элементы растягиваются по всей длине оси.

**Align-content**

align-content - позволяет выравнивать элементы если они находятся на нескольких рядах. Очень похоже на justify-content, но применяется относительно поперечной оси. Данное свойство применимо только к контейнеру.

17) **Аlign-self**

Данное свойство применяется для выравнивания одного элемента. Данное свойство позволяет изменить значение установленное свойством align-items у контейнера. Т.е. используя свойство align-items мы можем выровнять все элементы по верхнему краю а последний элемент используя свойство аlign-self - выровнять по нижнему краю.

## 18) flex-direction

Это свойство указывает, как flex-элементы располагаются во flex-контейнере, задавая направление главной оси flex-контейнера. Элементы могут располагаться по двум основным направлениям — как строки по горизонтали или как столбцы по вертикали.

## flex-wrap

Исходная концепция flexbox — это контейнер для установки своих элементов в одну строку. Если flex-контейнер располагает свои элементы в одну или несколько строк, то свойство flex-wrap управляет направлением, в котором эти новые строки располагаются.

## flex-flow

Это свойство является сокращением для установки свойств flex-direction и flex-wrap.

## justify-content

Свойство justify-content выравнивает flex-элементы вдоль главной оси текущей строки flex-контейнера. Свойство помогает распределять оставшееся свободное пространство, когда все flex-элементы в строке не растягиваются или растягиваются, но достигли максимального размера.

## align-items

flex-элементы могут быть выровнены по поперечной оси текущей строки flex-контейнера, подобно justify-content, но в перпендикулярном направлении. Это свойство устанавливает выравнивание по умолчанию для всех flex-элементов, включая анонимных.

## align-content

Свойство align-content выравнивает строки внутри flex-контейнера, когда в поперечной оси имеется дополнительное пространство, подобно тому, как justify-content выравнивает отдельные элементы по главной оси.

## 19) order

Свойство order управляет порядком, в котором дочерние элементы появляются внутри flex-контейнера. По умолчанию они располагаются в том порядке, как добавлены исходно во flex-контейнер.

## flex-grow

Это свойство задаёт коэффициент роста, который определяет, насколько flex-элемент будет увеличиваться относительно остальных flex-элементов во flex-контейнере, при распределении положительного свободного пространства.

20) свойство  **justify-content** определяет, как браузер распределяет пространство между и вокруг элементов контента вдоль [главной оси](https://developer.mozilla.org/ru/docs/Glossary/Main_Axis) гибкого контейнера, или вдоль строчной оси сетки контейнера.

21) <audio *controls* *src*="../audio/audio2.mp3"></audio>

<video *controls* *src*="../video/vid0.mp4" *class*="video"></video>

22)

Для определения, будут ли переноситься элементы, при сжатии flex- контейнера

23)

24)flexbox-вёрстки и медиа запросов