

## ВОПРОСЫ

1. Раскройте суть фиксированной верстки веб-страницы?
2. Что такое резиновый макет веб-страницы?
3. Дайте определение понятия «адаптивный дизайн»
4. Что такое медиазапросы? Как подключить медиазапросы?
5. Для чего используется правило @import?
6. С какой целью применяется правило @media?
7. Для чего предназначено свойство flex?
8. Что означает @media (min-width: 560px)?
9. Расшифруйте запись @media (max-width: 960px).
10. Что означает flex: 1 2 200px;?
11. Для чего предназначено значение wrap?
12. С какой целью используется свойство flex-flow?
13. Для чего применяется свойство flex-direction?
14. С какой целью используются свойства align-items и свойство align-content?
15. Какие значения имеет свойство align-self?
16. Назовите свойства, которые имеет flex-контейнер.
17. Какими свойствами обладают flex-элементы?
18. Для чего предназначено свойство justify-content?
19. Как вставить видео? Каким образом вставить аудио?
20. Чем отличается свойство align-items от свойства align-self?
21. Как создать адаптивное меню?

## ОТВЕТЫ

1. **Фиксированная верстка веб-страницы** — это подход к созданию макетов, при котором размеры всех элементов (контейнеров, блоков, изображений и т.д.) задаются в **абсолютных единицах** измерения, чаще всего в пикселях (px). При этом макет не адаптируется под разные размеры экранов, а сохраняет строго заданные размеры и структуру.
2. **Резиновый макет (или гибкий макет)** — это способ верстки веб-страницы, при котором ширина и размеры элементов задаются в **относительных** единицах измерения, таких как проценты (%), em, rem или с использованием CSS-функций вроде calc(). Такой макет автоматически адаптируется к ширине окна браузера или контейнера, занимая доступное пространство без необходимости горизонтальной прокрутки.
3. **Адаптивный веб-дизайн** позволяет изменять структуру веб-страницы на основе ширины окна браузера на различных устройствах (планшеты, смартфоны)
4. **Медиазапросы** назначают страницам стили на основе размера окна браузера. Подключить медиазапрос можно несколькими способами:  
Внутреннее подключение:

```
@media (max-width: 480px)
{
  /* Тело медиазапроса */
}
```

CSS

Внешнее подключение через `<link>`

```
<head>
<link href="media.css" rel="stylesheet" media="(max-width: 480px)">
</head>
```

Внешнее подключение через **@import**

```
@import url(media.css) (max-width: 480px);
```

5. Правило **@import** используется для внешнего подключения медиазапроса
6. Правило **@media** используется для внутреннего подключения медиазапроса
7. **flex:** - сокращенная запись значений **flex-grow flex-shrink flex-basis**
8. **@media (min-width: 560px)** означает, что свойства, заданные в этом медиазапросе будут действительны, когда страница будет занимать ширину больше 560 пикселей.
9. **@media (max-width: 960px)** означает, что свойства внутри медиазапроса будут действительны до тех пор, пока страница занимает ширину меньше 960 пикселей.
10. **flex: 1 2 200px** означает, что элемент занимает 200 пикселей, но не может быть больше, чем 200 пикселей и меньше, чем 100 пикселей.
11. **flex-wrap** определяет поведение flex-элементов при переполнении страницы по ширине. wrap означает, что flex-элемент перенесется вниз, а nowrap означает, что переноса не будет
12. **flex-flow** - сокращенная запись flex-direction и flex-wrap
13. **flex-direction** - направление расположение flex-элементов.
14. **align-items** - выравнивание всех элементов вдоль поперечной оси (вертикаль) **align-content** - выравнивание всех строк (если их больше 1)
15. **align-self: none | flex-start | flex-end | center | baseline | stretch**
16. Свойства flex контейнера:

```
.flex-container{
  display: flex;
  flex-flow: none;
  justify-content: none;
  justify-items: none;
  align-items: none;
  align-content: none;
}
```

17. Свойства flex-элементов:

```
.flex-element{
  order: none;
  flex-grow: none;
  flex-shrink: none;
  flex-basis: none;
  align-self: none;
  justify-self: none;
  flex: 0 0 0;
}
```

18. **justify-content** отвечает за выравнивание содержимого flex-контейнера вдоль главной оси.
19. Видео можно вставить с помощью тега **<video src="">**, а аудио с помощью тега **<audio src="" controls>**

20. `align-self` используется в конкретном дочернем элементе и переопределяет значение `align-items` flex-контейнера для этого элемента.
21. Необходимо создать элемент `<input type="radio">` или `<input type="checkbox">`, затем создать для него `<label>`. После этого скрыть input и привязать появление label к медиазапросу на ширину экрана. Затем создать меню `<nav>` и скрыть его с помощью `display: none`. Добавить в код css:

CSS

```
input:checked + nav {  
  display: flex;  
}
```