Ответы неделя 5:

1. flexbox-позиционирование позволяет создать адаптивный дизайн проще, чем с использованием других видов позиционирования. Можно расставить элементы в контейнере, распределить доступное пространство между ними, и выровнять их тем или иным способом даже если они не имеют конкретных размеров.

2. Позиционирование через `display: table подходит:

- Горизонтальное выравнивание:

element {

display: table;

margin: auto;

}

при этом ширина таблицы динамическая.

-Вертикальное выравнивание:

element {

display: table;

}

и элементу, который является дочерним в HTML разметке, задаем свойства и значения

element {

display: table-cell;

text-align: center;

vertical-align: middle;

}

ячейка table-cell, растягивается на всю ширину таблицы, а содержимое внутри ячейки просто центрируется внутри неё при помощи text-align и vertical-align.

Итог Блок выровнен по горизонтали, и текст по вертикали.

3. Горизонтальное выравнивание по центру:

1) класс center со свойством text-align: center к родительскому блоку и отобразим дочерний блок, как inline-block (используется редко)

2) margin: auto свойство мы прописываем дочернему блоку (актуальный способ). значение auto, задает автоматический отступ с левой и с правой стороны от вложенного блока, пространство с обеих сторон должно рассчитываться поровну.

3) Свойство position: absolute. (не рекомендован к использованию, противоречит принципу независимого переиспользования блоков). Когда к дочернему блоку применяется position: absolute, то оно перестает влиять на родителя. Поэтому для родителя мы задаем position: relative, чтобы позиционироваться от него. А ребенку задается position: absolute, смещаем его на 50% влево и задаем отступ на половину его ширины.

4) Свойство flexbox (популярное)

Flexbox (простой и современный способ горизонтального выравнивания по центру), для родителя добавляются два флексовых свойства.

<div class="wrap flexbox">

<div class="block"></div>

</div>

.flexbox { display: flex;

justify-content: center;

}

5) Свойство grid (современное). Добавляем родителю всего два grid свойства, а вложенному блоку - три grid свойства.

<div class="wrap grid-parent">

<div class="block grid-child"></div>

</div>

.grid-parent {

display: grid;

grid-template-areas: "a";

}

.grid-child {

grid-area: a;

justify-self: center;

}

4. Оси во флекс-верстке:

Первая ось - главная (по умолчанию она направлена слева направо).

Вторая - поперечная (по умолчанию направлена сверху вниз), она всегда перпендикулярно главной.

Главная ось задаёт основное направление flex-элементов во flex-контейнере, а поперечная ось определяет их направление при переносе на новую линию.

По умолчанию элементы во flex-контейнере располагаются вдоль направления главной оси (т.е. слева направо) на одной линии.

Направление главной оси можно изменить, осуществляется это с помощью CSS-свойства flex-direction.

row (слева направо) - по умолчанию

row-reverse (справа налево)

column (сверху вниз)

column-reverse (снизу вверх)

С помощью этого свойства можно сделать так, чтобы flex-элементы располагались не рядами (rows), а колонками (columns) (с помощью значения column или column-reverse).

Значения wrap и wrap-reverse CSS-свойства flex-wrap определяют направление поперечной оси.

Свойства flex-direction и flex-wrap (указывают с помощью flex-flow).

5. margin: auto во флекс-верстке:

-С помощью флексбокса можно отцентровать элемент по вертикали и горизонтали так, чтобы центровка сохранялась при изменении размеров элемента или контейнера.

Если задать контейнеру раскладку флексбокса, а дочернему флекс-элементу margin: auto. В этом случае флекс-элемент уменьшит свой размер под содержимое и отцентруется по вертикали и горизонтали.

-Родительский div с div’ом потомком внутри:

Если выставить margin-left и margin-top на auto, мы отодвинем flex-элемент в нижний правый угол родителя:

Выставление свойства margin flex-потомку, оттолкнет его в указанном направлении.

Выставив margin-left на auto - оттолкнем потомка вправо.

Выставив margin-top на auto - оттолкнем потомка вниз.

-Установив значение атрибута margin: auto для одного элемента в наборе с flex-элементами, выровненных с начала, можно создать разделённую навигацию. Это работает с разметкой flexbox и свойствами выравнивания. Как только для элемента со свойством margin: auto не остаётся свободного места, он станет вести себя также, как и все остальные flex элементы, и сократится в объёме, пытаясь уместиться в доступное пространство.

6. Cвойство box-sizing определяет как вычисляется общая ширина и высота элемента. Свойство box-sizing позволяет изменить этот алгоритм, чтобы свойства width и height задавали размеры не контента, а размеры блока.

7. CSS-свойство flex-grow. Cвойство определяет, может ли  начальная ширина flex-элемента увеличиваться. Увеличение ширины flex-элемента осуществляется за счёт свободного пространства линии. В качестве значения CSS-свойства `flex-grow` указывается целое число. Именно это значение и определяет (если оно больше или равно 1) какую часть свободного пространства flex-элемент заберёт себе

CSS-свойство flex-shrink. Свойство определяет, может ли ширина flex-элемента уменьшиться. Уменьшение ширины flex-элемента будет осуществляться только в том случае, если ширины линии будет не достаточно для отображения всех flex-элементов, расположенных в ней. Необходимая ширина рассчитывается на основании начальной ширины, которую имеет каждый flex-элемент в ней.

8. align-items: stretch

9. 148.52\*17.807

10. Свойство order определяет порядок, используемый для размещения flex элементов в их flex контейнере относительно остальных. Элементы располагаются в восходящем порядке по значению order. Элементы с одинаковым значением order располагаются в том порядке, в каком они находятся в исходном коде.

Если элемент не является флекс элементом (не находится внутри родительского элемента, который является блочным, или строчным флекс контейнером), то свойство order не окажет на такой элемент никакого эффекта.

11. Код в отдельном файле.

12. Значение fixed похоже на absolute. Применяется такой тип позиционирования для создания меню, вкладок, заголовков, в общем, любых элементов, которые должны быть закреплены на веб-странице и всегда видны пользователю. В отличие от абсолютного позиционирования привязывается к указанной свойствами left, top, right и bottom точке на экране и не меняет своего положения при прокрутке веб-страницы.

13. С помощью блочных элементов.

14. [CSS](https://developer.mozilla.org/ru/docs/Web/CSS) свойство flex-basis задаёт базовый размер флекс элемента по основной оси. Это свойство определяет размер контент-бокса, если не задано иначе через [box-sizing](https://developer.mozilla.org/ru/docs/Web/CSS/box-sizing). Минимальный размер [содержимого](https://www.w3.org/TR/css-sizing-3/#min-content-inline-size) — это наименьший размер, который может иметь блок, не вызывая переполнения. Максимальный размер [содержимого](https://www.w3.org/TR/css-sizing-3/#max-content-inline-size) — это максимальный размер, который может содержать содержимое коробки.