

Веб-технологии

План на семестр

Курсовой проект

Frontend: flex, grid, rwd

Backend: php + OpenServer + JS

4 контрольные точки



Лекция

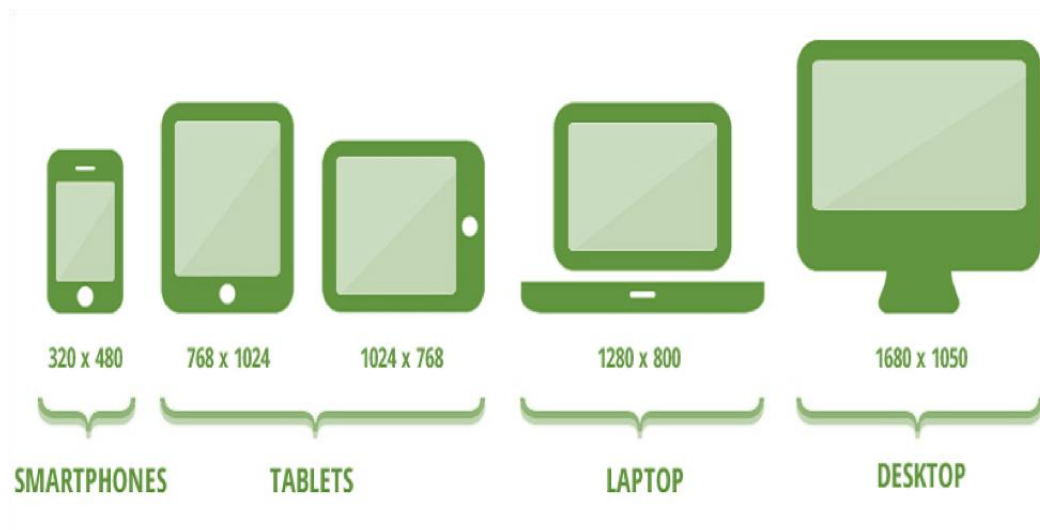
Ссылка на лекцию в Notion:

<https://attractive-veil-6fc.notion.site/Flexbox-c0f2f5ce015247eb86b84be268f2175a?pvs=4>

[перейти](#)

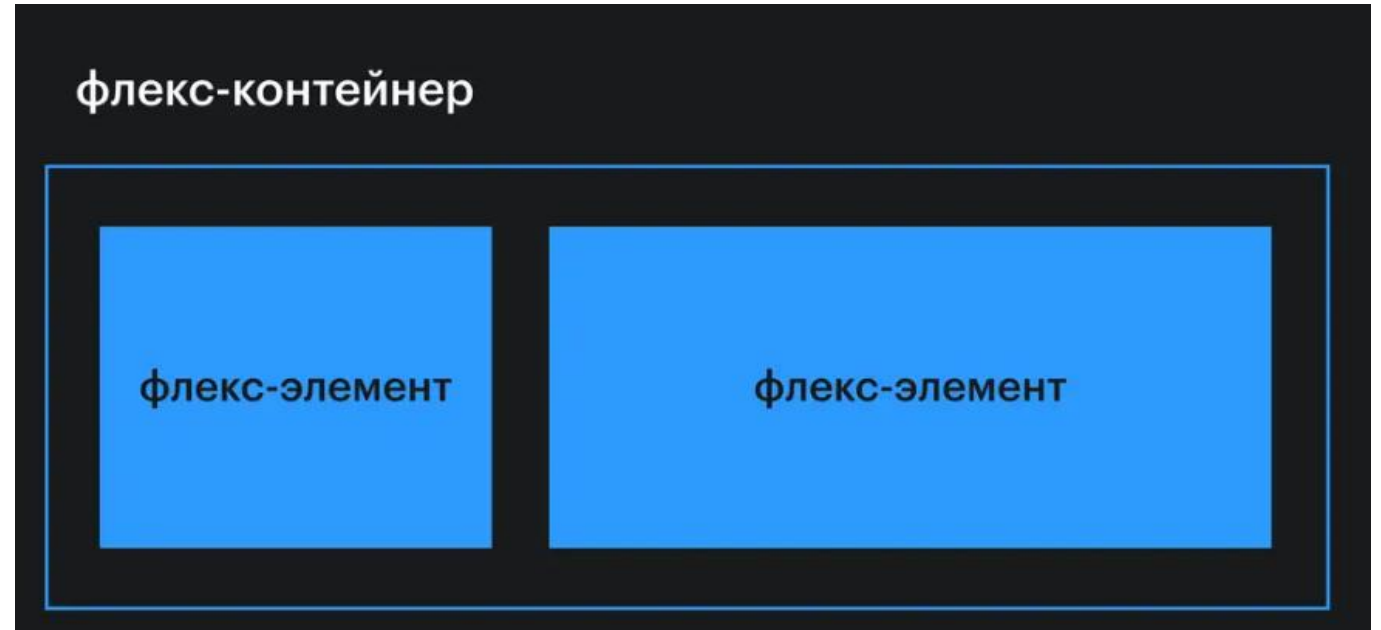
Flexbox

Идея флексбоксов появилась ещё в 2009 году, и этот стандарт до сих пор развивается и прорабатывается. Основная идея флексов – гибкое распределение места между элементами, гибкая расстановка, выравнивание, гибкое управление. Ключевое слово – *гибкое*, что и отражено в названии (flex – англ. гибко).



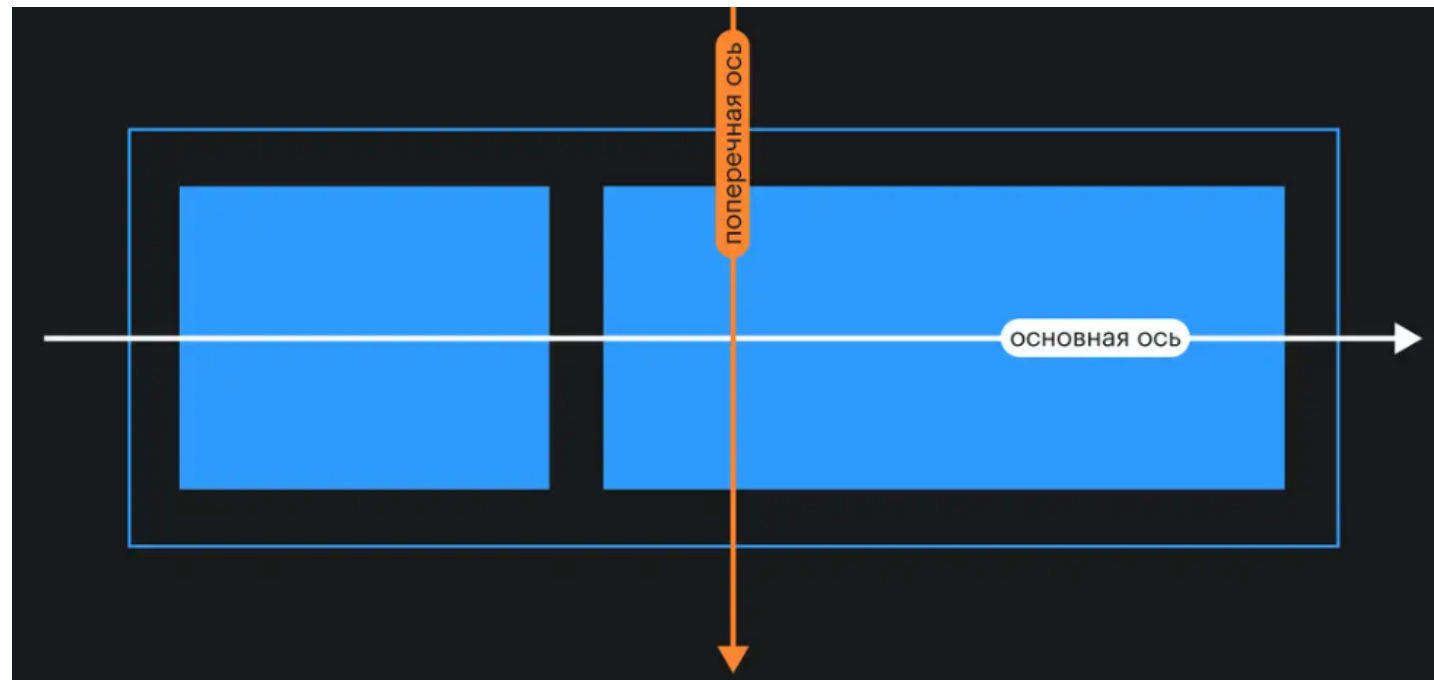
Flexbox | Термины

- ❑ **Flex-контейнер:** элемент, к которому применяется свойство `display: flex`; Вложенные в него элементы подчиняются правилам раскладки флексов.
- ❑ **Flex-элемент:** элемент, вложенный, во флекс-контейнер.



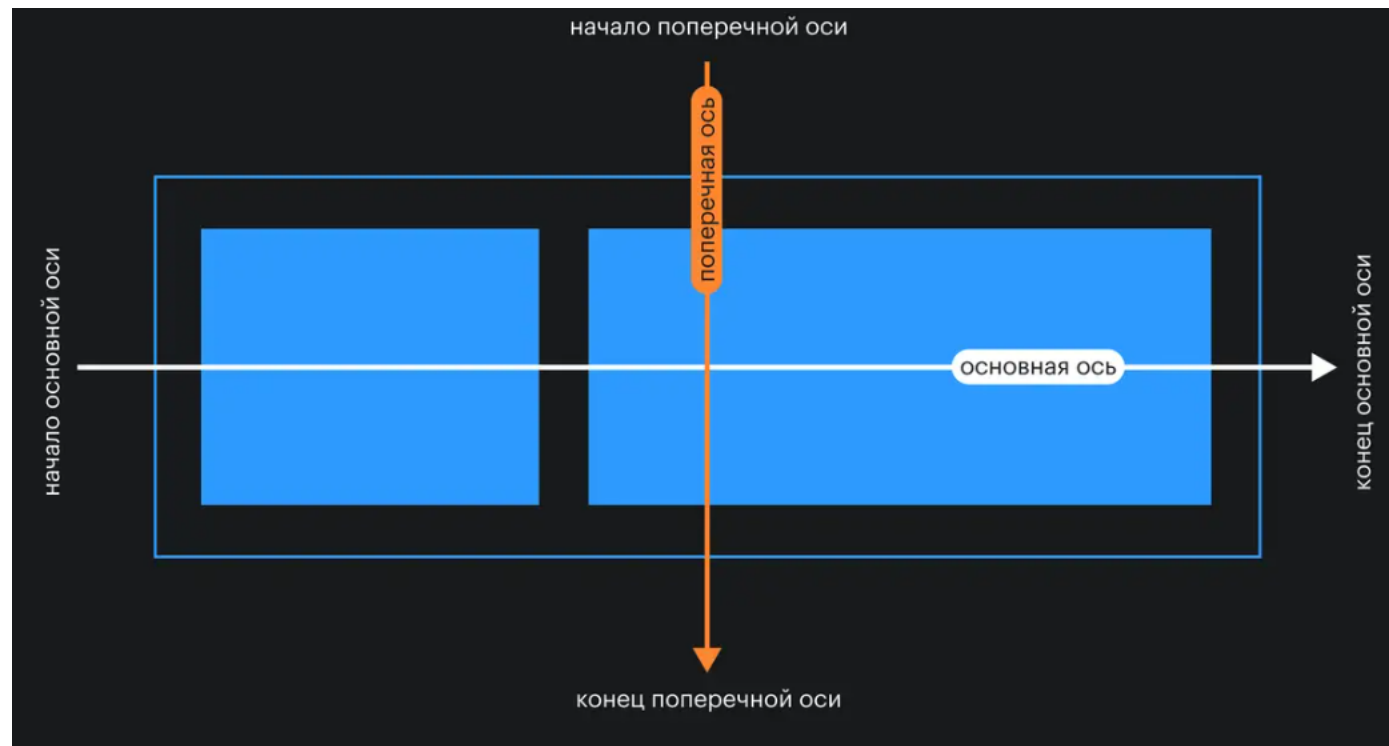
Flexbox | Термины

- ❑ **Основная ось:** основная направляющая флекс-контейнера, вдоль которой располагаются флекс-элементы.
- ❑ **Поперечная (побочная, перпендикулярная) ось:** ось, идущая перпендикулярно основной. Позже вы поймёте, для чего она нужна.



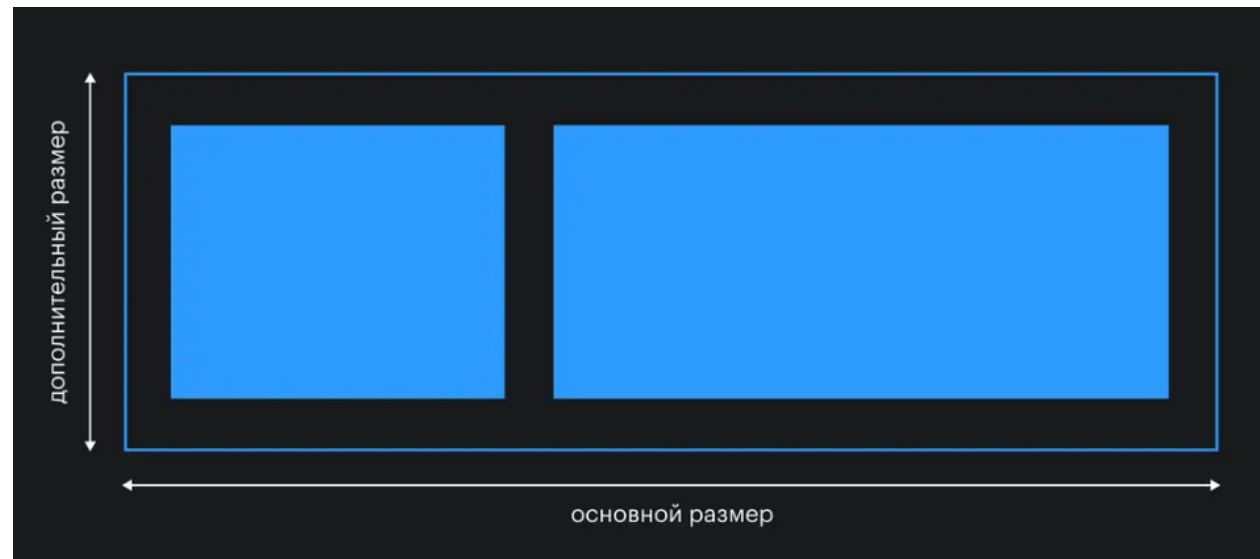
Flexbox | Термины

- ❑ **Начало / конец основной оси:**
точки в начале и в конце основной оси соответственно. Это пригодится нам для выравнивания флекс-элементов.
- ❑ **Начало / конец поперечной оси:**
точки в начале и в конце поперечной оси соответственно.



Flexbox | Термины

- ❑ **Размер по основной оси (основной размер):** размер флекс-элемента вдоль основной оси. Это может быть ширина или высота в зависимости от направления основной оси.
- ❑ **Размер по поперечной оси (поперечный размер):** размер флекс-элемента вдоль поперечной оси. Это может быть ширина или высота в зависимости от направления поперечной оси. Этот размер всегда перпендикулярен основному размеру. Если основной размер — это ширина, то поперечный размер — это высота, и наоборот.



Flexbox | Свойства флекс-контейнера

Когда мы задаём какому-то элементу значение *flex* для свойства *display*, мы превращаем этот элемент в флекс-контейнер. Внутри него начинает действовать флекс-контекст, его дочерние элементы начинают подчиняться свойствам флексбокса.

display

CSS

```
1 .container {  
2   display: flex;  
3 }
```

** снаружи флекс-контейнер выглядит как блочный элемент — занимает всю ширину родителя, следующие за ним элементы в разметке переносятся на новую строку.*

Flexbox | Display

Если контейнеру задано значение *inline-flex*, то снаружи он начинает вести себя как строчный (инлайн) элемент — размеры зависят только от внутреннего контента, встаёт в строку с другими элементами. Внутри это ровно такой же флекс-контейнер, как и при предыдущем значении.

CSS

```
1 .container {  
2   display: inline-flex;  
3 }
```

Flexbox | Flex-direction

Flex-direction – свойство управления основной и поперечной осями

CSS

```
1 .container {  
2   display: flex;  
3   flex-direction: row;  
4 }
```

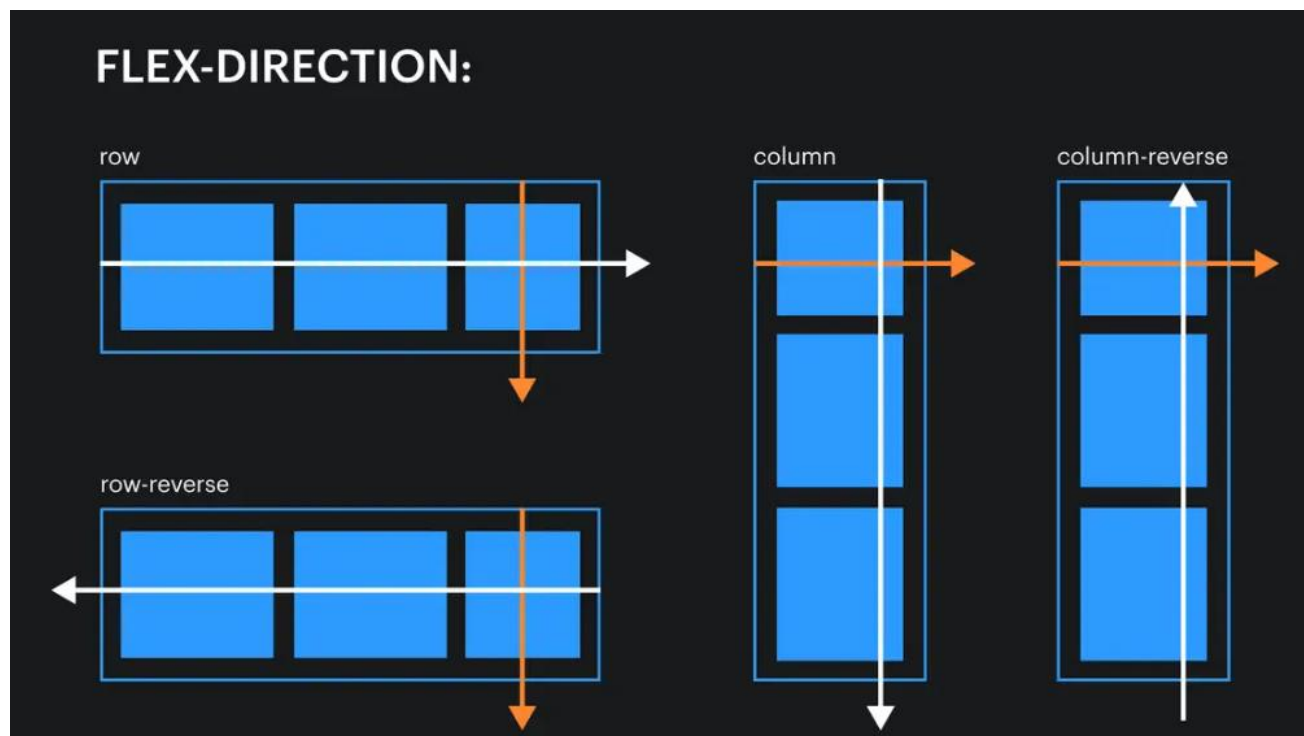
Flexbox | Flex-direction

Flex-direction – свойство управления основной и поперечной осями

Возможные значения:

- ❑ **row** (значение по умолчанию) — основная ось идёт горизонтально слева направо, поперечная ось идёт вертикально сверху вниз.
- ❑ **row-reverse** — основная ось идёт горизонтально справа налево, поперечная ось идёт вертикально сверху вниз.
- ❑ **column** — основная ось идёт вертикально сверху вниз, поперечная ось идёт горизонтально слева направо.
- ❑ **column-reverse** — основная ось идёт вертикально снизу вверх, поперечная ось идёт горизонтально слева направо.

Flexbox | Flex-direction



Важный момент: на сайтах с направлением текста справа налево, например, на сайте на арабском языке, для значений `row` и `row-reverse` основная ось будет идти в обратном направлении. Для значений `column` и `column-reverse` своё направление поменяет поперечная ось.

Flexbox | Flex-wrap

По умолчанию значение у свойства flex-wrap — nowrap. При этом флекс-элементы помещаются (или пытаются уместиться) в один ряд и не переносятся в новый ряд, даже если не влезают в размеры родителя.

Установив значение wrap, мы можем изменить это поведение, и флекс-элементы будут иметь возможность перенестись в новый ряд, если не влезают в одну линию в рамках родителя.

Ещё одно возможное значение — wrap-reverse. В этом случае элементы будут располагаться снизу вверх, заполнив собой сперва нижний ряд, а те, что не влезли, перепрыгнут в ряд выше.

Flexbox | Flex-wrap



FLEX-WRAP:

nowrap



wrap



wrap-reverse



Flexbox | Flex-flow

Flex-flow – свойство-шорткат для одновременного определения значений свойств flex-direction и flex-wrap.

CSS

```
1 .container {  
2   display: flex;  
3   flex-flow: column wrap;  
4 }
```

или

CSS

```
1 .container {  
2   display: flex;  
3   flex-flow: row nowrap;  
4 }
```

[демо](#)

Flexbox | Justify-content

Justify-content – свойство позволяет выравнивать флекс-элементы внутри флекс-контейнера по основной оси.

CSS

```
1 .container {  
2   display: flex;  
3   flex-flow: column wrap;  
4 }
```

или

CSS

```
1 .container {  
2   display: flex;  
3   flex-flow: row nowrap;  
4 }
```

демо