

Стать IT-шником может каждый



# HTML, CSS, JavaScript Summary Session



Преподаватель



# ИМЯ ФАМИЛИЯ

## ***Текущая должность***

Количество лет опыта

Какой у Вас опыт - ключевые кейсы

Самые яркие проекты

Дополнительная информация по вашему усмотрению

[Корпоративный e-mail](#)

[Социальные сети \(по желанию\)](#)

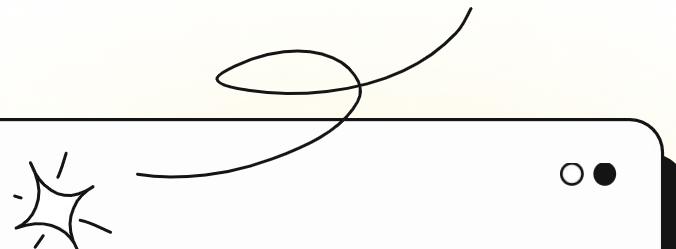


# ВАЖНО!

- Камера должна быть включена на протяжении всего занятия.
- Если у Вас возник вопрос в процессе занятия, пожалуйста, поднимите руку и дождитесь, пока преподаватель закончит мысль и спросит Вас, также можно задать вопрос в чате или когда преподаватель скажет, что начался блок вопросов.
- Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях.
- Вести себя уважительно и этично по отношению к остальным участникам занятия.
- Во время занятия будут интерактивные задания, будьте готовы включить микрофон или демонстрацию экрана по просьбе преподавателя.

# ПЛАН ЗАНЯТИЯ

- Повторение изученного за неделю
- Вопросы по повторению
- Разбор домашних заданий
- Вопросы по разбору заданий
- Задание для закрепления
- Оставшиеся вопросы



Стать IT-шником может каждый



# ПОВТОРЕНИЕ ИЗУЧЕННОГО



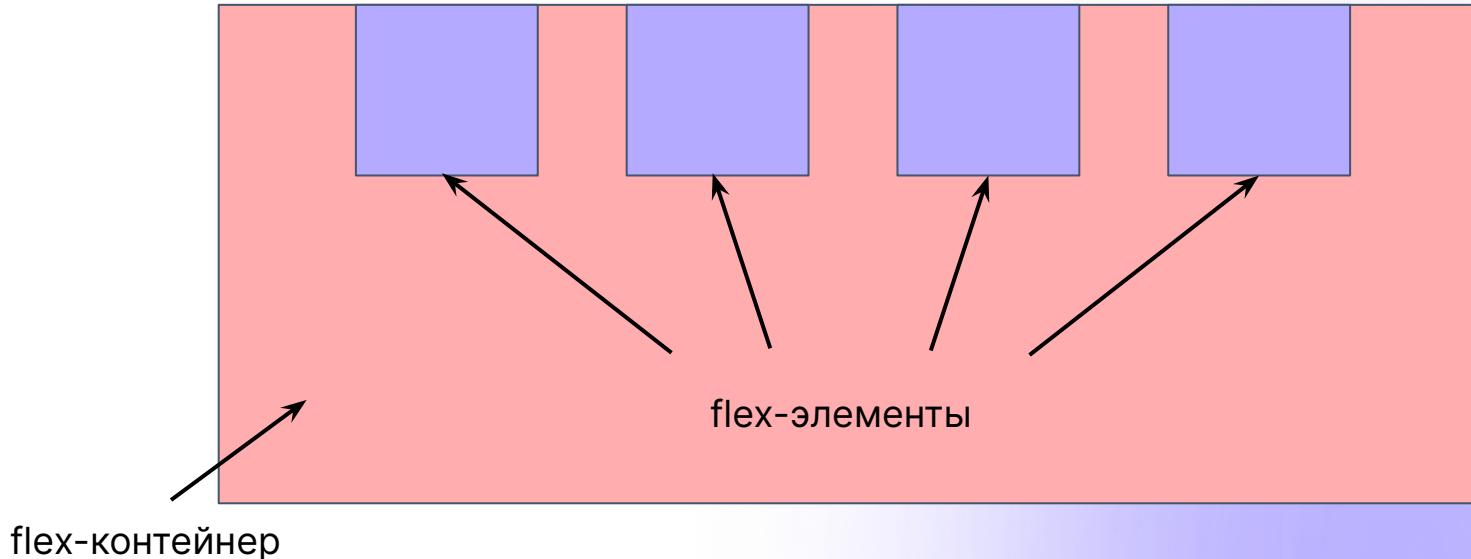


## Flexbox —

это CSS модуль раскладки **Flexible Box**, который был разработан как модель одномерного-направленного макета и как один из методов распределения пространства между элементами в интерфейсе, с мощными возможностями выравнивания.

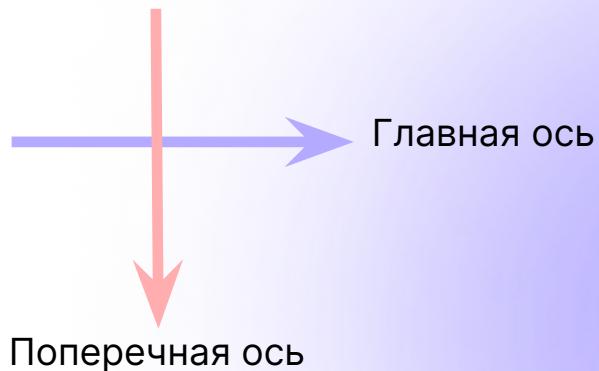


# Flexbox



# Flexbox

- Flexbox- способ работы с элементами, расположенными по одной оси
- Использует две оси - главную (main axis) и поперечную (cross axis)
- Все элементы, с которыми мы работаем располагаются внутри одного контейнера
- Чтобы сделать контейнер flex-контейнером, ему нужно задать в стилях display: flex

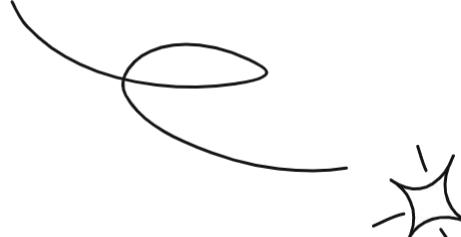


Полезно знать



## Главная ось —

это **горизонтальная ось**, определяется свойством flex-direction. Вдоль неё выстраиваются дочерние блоки элемента-контейнера.



# Главная ось

**row**

**row-reverse**

**column**

**column-reverse**

Главная ось является горизонтальной, объекты выстраиваются слева направо

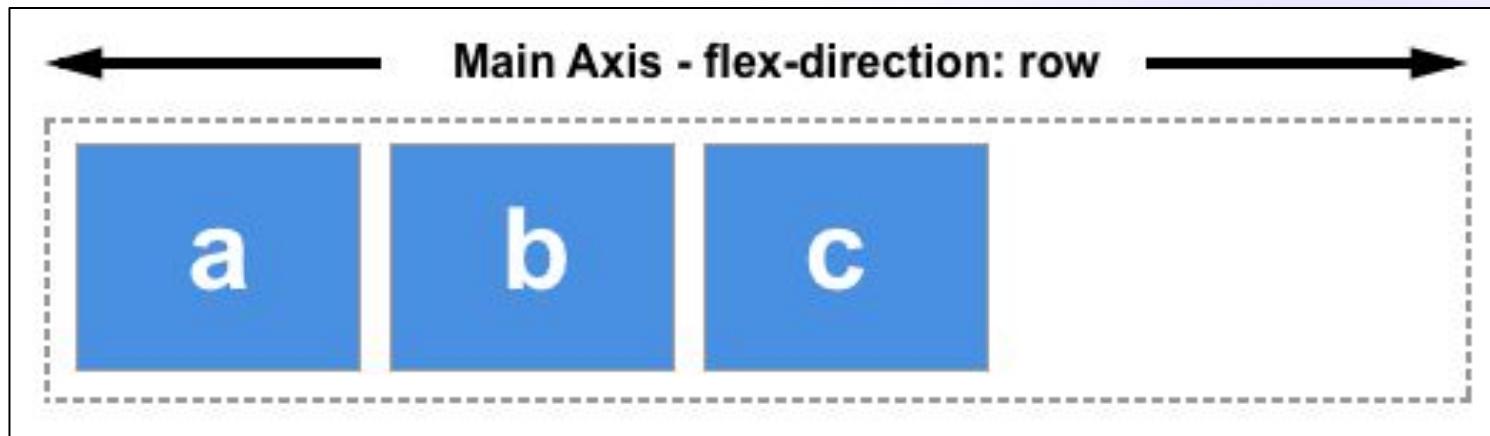
Главная ось является горизонтальной, объекты выстраиваются справа налево

Главная ось является вертикальной, объекты выстраиваются сверху вниз

Главная ось является вертикальной, объекты выстраиваются снизу вверх

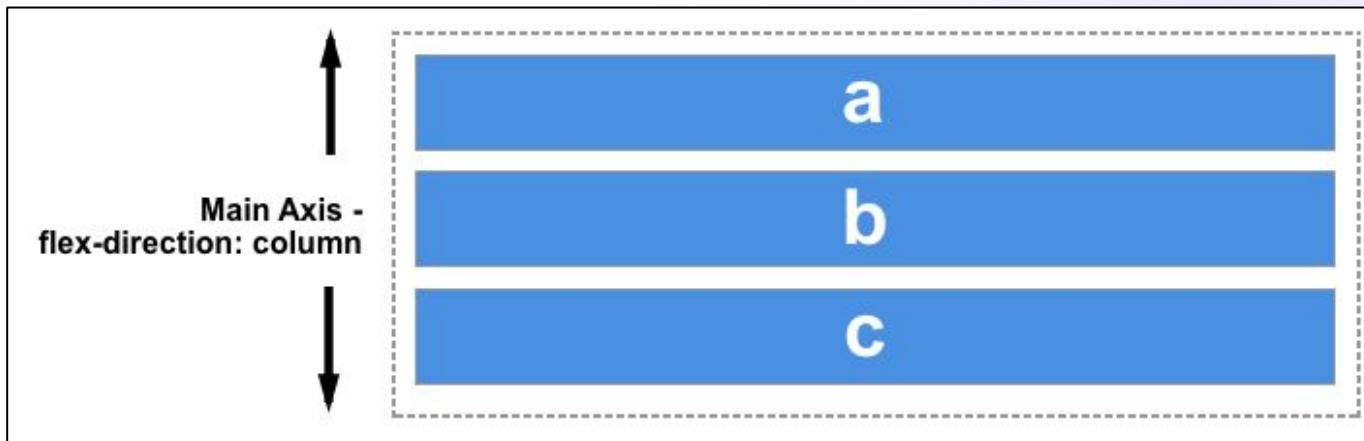
## row и row-reverse

Ось будет проходить в горизонтальном направлении (inline direction).



# column и column-reverse

Ось будет проходить в вертикальном направлении (block direction).





## Побочная ось —

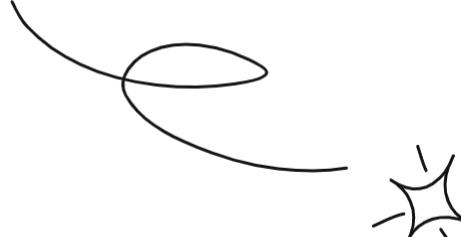
**это ось**, которая проходит перпендикулярно главной оси: если свойство flex-direction (главная ось) задано как row или row-reverse, побочная ось будет проходить в вертикальном направлении.





## justify-content —

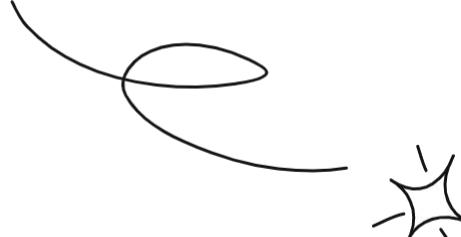
это CSS [свойство](#), которое определяет, как браузер распределяет пространство между и вокруг элементов контента вдоль главной оси flex-контейнера.





## align-items —

это CSS [свойство](#), которое выравнивает flex-элементы текущей flex-линии таким же образом, как и justify-content, но в перпендикулярном направлении.



## Значения свойства justify-content

**start**

**end**

**flex-start**

**flex-end**

Элементы располагаются друг за другом и прижимаются к началу контейнера по главной оси

Элементы располагаются друг за другом и прижимаются к концу контейнера по главной оси

Элементы располагаются друг за другом и прижимаются к началу контейнера по главной оси, в зависимости от её направления (flex-direction)

Элементы располагаются друг за другом и прижимаются к концу контейнера по главной оси, в зависимости от её направления (flex-direction)

## Значения свойства justify-content

**justify-content: start**



**justify-content: end**



## Значения свойства justify-content

**center**

**left**

**right**

**normal**

Элементы располагаются друг за другом в центре контейнера по главной оси

Элементы располагаются друг за другом и прижимаются к левому краю контейнера по главной оси

Элементы располагаются друг за другом и прижимаются к правому краю контейнера по главной оси

Элементы располагаются на своих обычных позициях, так как будто свойство justify-content не задано

## Значения свойства justify-content

**justify-content: center**

1

2

3

4

5

## Значения свойства justify-content

### **space-between**

Элементы равномерно распределены вдоль главной оси контейнера. Расстояния между каждой парой соседних элементов равны. Первый элемент прижат к началу контейнера по главной оси, а последний - к концу

### **space-around**

Элементы равномерно распределены вдоль главной оси контейнера. Расстояния между каждой парой соседних элементов равны. Пустые пространства перед первым элементом и после последнего элемента равны половине расстояния между парами соседних элементов

### **space-evenly**

Элементы равномерно распределены вдоль главной оси контейнера. Расстояния между каждой парой соседних элементов равны расстояниям от начала контейнера до первого элемента и от последнего элемента до конца контейнера

## Значения свойства justify-content

**justify-content: space-around**



**justify-content: space-between**

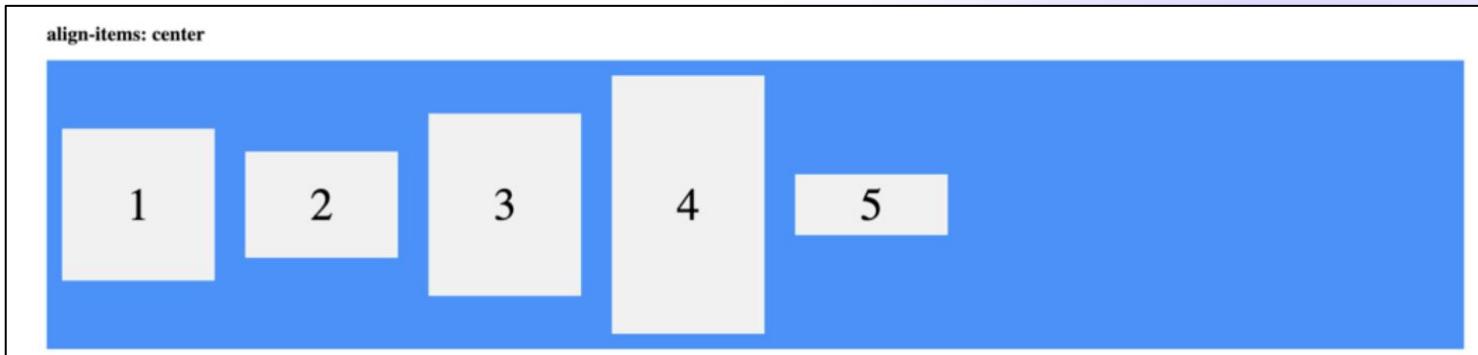


**justify-content: space-evenly**

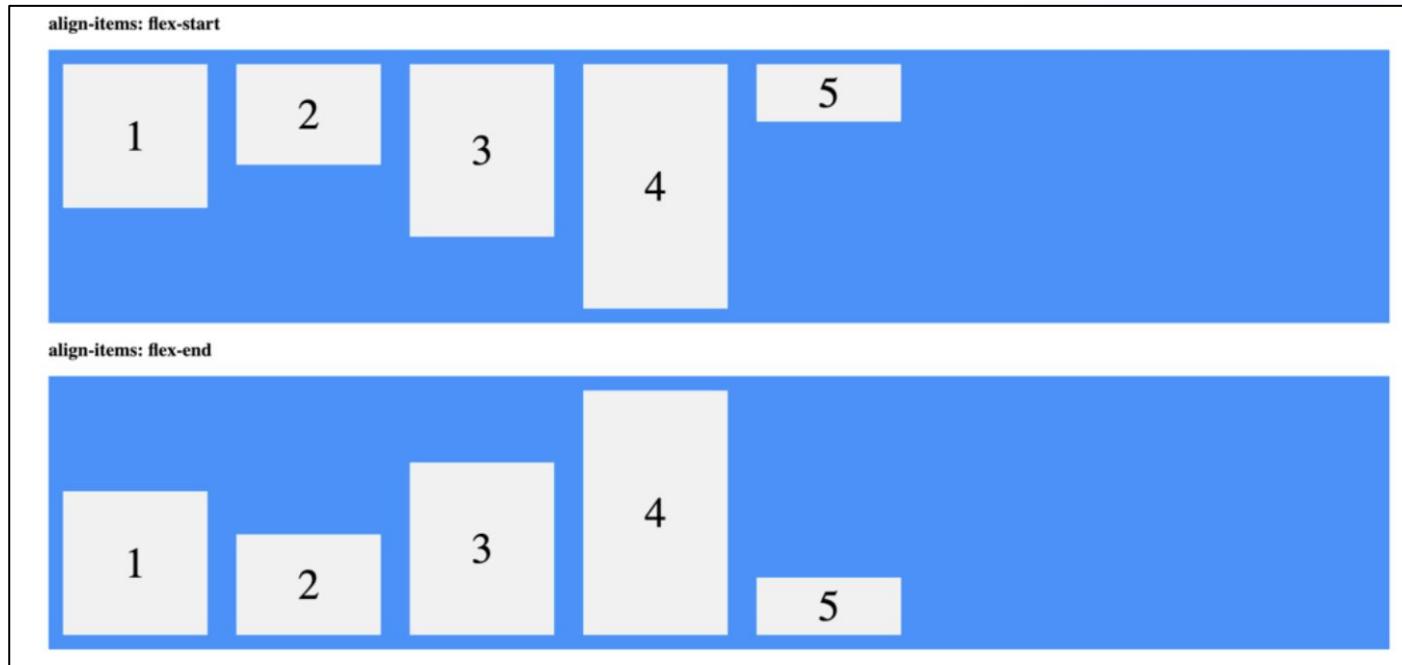


## Значения свойства align-items

Может принимать такие же значения, как justify-content (кроме space-between, space-around, space-evenly), и выравнивает элементы относительно побочной оси.



## Значения свойства align-items





## flex-wrap —

это CSS свойство, которое задает правила вывода flex-элементов — в одну строку или в несколько, с переносом блоков. Если перенос разрешён, то возможно задать направление, в котором выводятся блоки.



## Значения свойства flex-wrap

**nowrap**

Расположение в одну линию, может привести к переполнению контейнера. Свойство cross-start эквивалентно start или before в зависимости от значения flex-direction

**wrap**

Расположение в несколько линий. Свойство cross-start эквивалентно start или before в зависимости от значения flex-direction и свойство cross-end противоположно cross-start

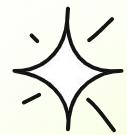
**wrap-reverse**

Ведёт себя так же, как и wrap, но cross-start и cross-end инвертированы

Стать IT-шником может каждый



# ВОПРОСЫ



Стать IT-шником может каждый



ICH  
S  
★

## Задание

Неализовать навигационно меню, применив flex-box.

[Логотип](#)



A screenshot of the Heroku website's header. It features the Heroku logo (a stylized 'H' icon) and the word 'HEROKU' in bold capital letters. To the right of the logo is a horizontal menu bar with links: 'Products', 'Marketplace', 'Pricing', 'Documentation', 'Support', and 'More'. On the far right of the header are two buttons: a white button labeled 'Log in or' and a purple button labeled 'Sign up'.



## Задание 1

Создайте HTML-страницу с тремя блоками. Все блоки должны быть расположены в одну строку.  
Используйте свойство CSS `display: flex;` в родительском контейнере, чтобы сделать его `flex` контейнером.



# Решение 1

```
<!DOCTYPE html>
<html>
<head>
    <style>
        .flex-container {
            display: flex;
        }
        .flex-item {
            border: 1px solid black;
            padding: 10px;
            margin: 10px;
        }
    </style>
</head>
<body>
    <div class="flex-container">
        <div class="flex-item">Блок 1</div>
        <div class="flex-item">Блок 2</div>
        <div class="flex-item">Блок 3</div>
    </div>
</body>
</html>
```



## Задание 2

На той же странице, что и в первом задании, используйте свойство justify-content в родительском контейнере, чтобы изменить расположение блоков внутри контейнера. Попробуйте разные значения (flex-start, flex-end, center, space-between, space-around), чтобы увидеть, как они влияют на расположение блоков.



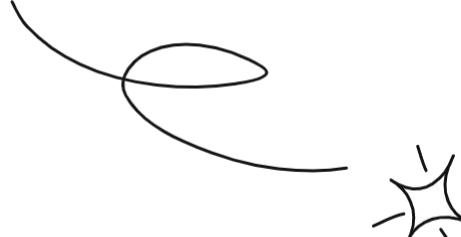
## Решение 2

```
<!DOCTYPE html>
<html>
<head>
    <style>
        .flex-container {
            display: flex;
            justify-content: space-between;
        }
        .flex-item {
            border: 1px solid black;
            padding: 10px;
            margin: 10px;
        }
    </style>
</head>
<body>
    <div class="flex-container">
        <div class="flex-item">Блок 1</div>
        <div class="flex-item">Блок 2</div>
        <div class="flex-item">Блок 3</div>
    </div>
</body>
</html>
```



## Свойство CSS flex —

это сокращённое свойство, определяющее способность гибкого элемента растягиваться или сжиматься для заполнения собой доступного свободного пространства.



# Свойство CSS flex

**flex-grow**

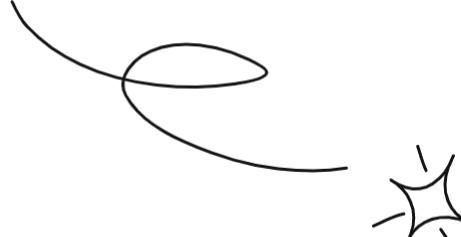
**flex-shrink**

**flex-basis**



## flex-grow —

это CSS-свойство, определяющее как много свободного пространства во flex-контейнере должно быть назначено текущему элементу.



Полезно знать



## Свободное пространство —

это разница между размером flex-контейнера и размером всех его flex-элементов вместе.



Полезно знать



## sibling-элементы —

это элементы одного уровня вложенности.



# flex-grow



# flex-grow



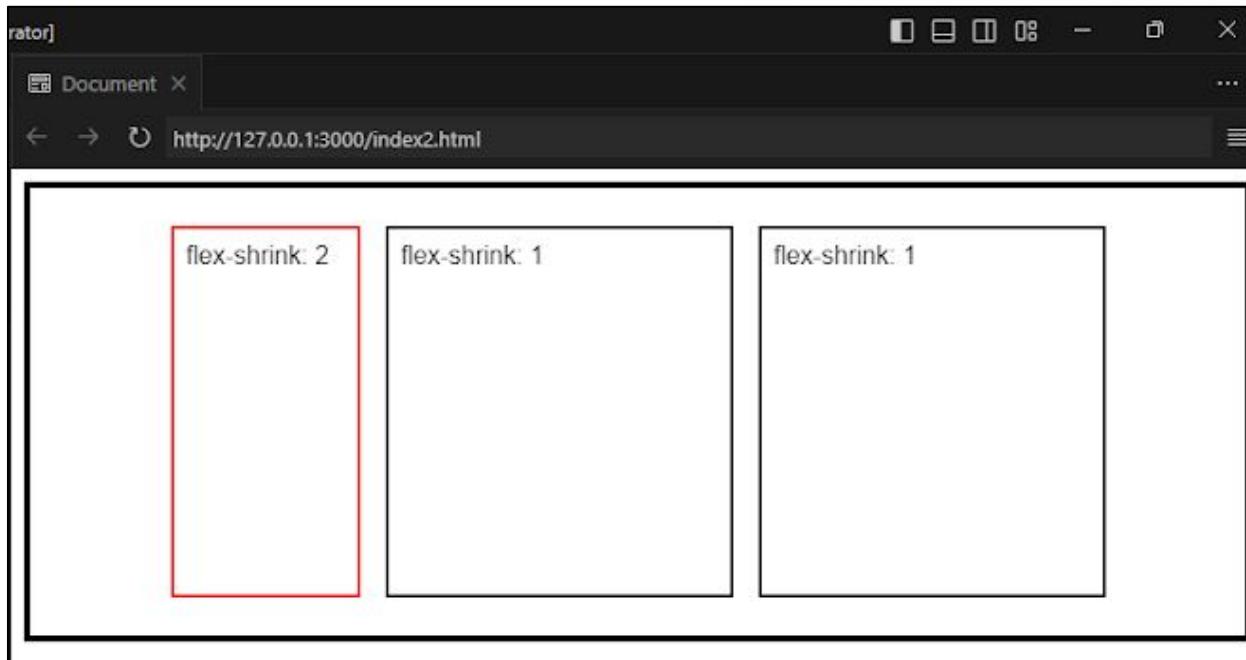


## flex-shrink —

это CSS-свойство, определяющее фактор сжатия flex-элемента.



## flex-shrink





## flex-basis —

это CSS-свойство, которое задает базовый размер flex элемента по основной оси и определяет размер контент-бокса, если не задано через box-sizing.



## Порядок элементов flex

В Flexbox есть возможность менять порядок расположения flex элементов, не влияя на исходный порядок. По умолчанию все элементы flex имеют значение order равное 0.

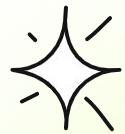
Элементы Flex с установленными на них большими значениями будут отображаться позже в порядке отображения, чем элементы с меньшими значениями порядка

Элементы Flex с одинаковым значением порядка будут отображаться в исходном порядке

Стать IT-шником может каждый



# ВОПРОСЫ

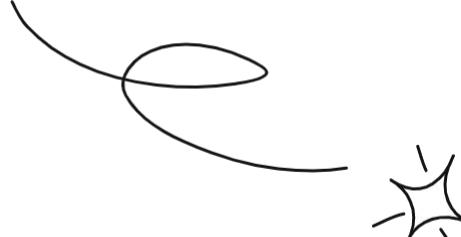


Полезно знать



## Формы —

это элемент в веб-разработке, который используются для сбора информации от пользователей на веб-страницах.



Стать IT-шником может каждый



# Важность форм

**Сбор информации от пользователей**

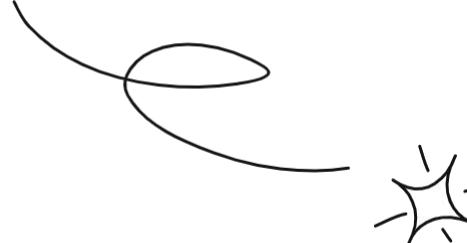
**Взаимодействие с сервером**

**Пользовательский опыт**



## <form> —

это элемент, который определяет форму и атрибуты, которые определяют поведение этой формы.



# Основные элементы формы

Элемент	Описание	Тип элемента
<b>&lt;form&gt;</b>	Создает форму, контейнер для всех элементов формы	Блочный
<b>&lt;input&gt;</b>	Однострочное поле для сбора данных	Строчный
<b>&lt;textarea&gt;</b>	Многострочное поле для сбора данных	Строчный
<b>&lt;button&gt;</b>	Кнопка	Строчный
<b>&lt;select&gt;</b>	Выпадающий список	Строчный
<b>&lt;label&gt;</b>	Контейнер для поля для сбора данных (<input>, <textarea>) и описания к нему (<p>)	Блочный

Полезно знать



## <fieldset> —

это элемент, с помощью которого поля формы можно  
разделять на логические блоки

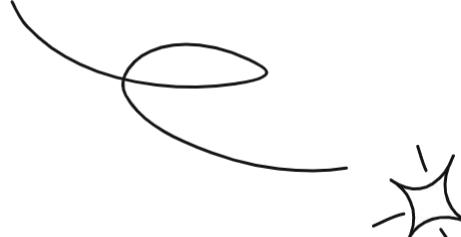


Полезно знать

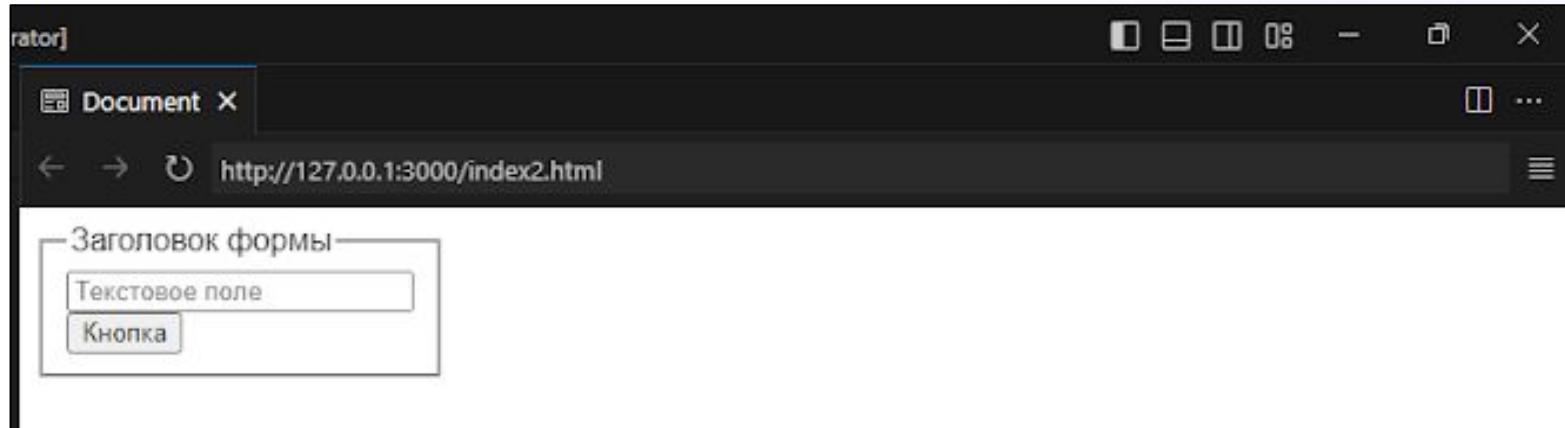


## <legend> —

это элемент, с помощью которого каждому разделу можно присвоить название.



# Результат использования fieldset и legend



## Пример использования fieldset и legend

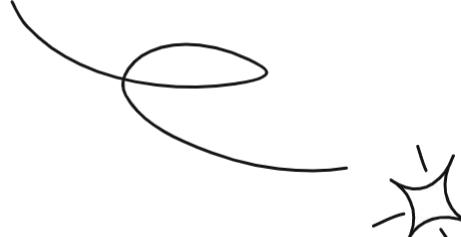
```
<form action="" style="width: 100px;">
  <fieldset>
    <legend>Заголовок формы</legend>
    <input type="text" placeholder="Текстовое поле">
    <button>Кнопка</button>
  </fieldset>
</form>
```

Полезно знать



## <label> —

это элемент, с помощью которого создаются надписи к полям  
формы.



## Пример использования label

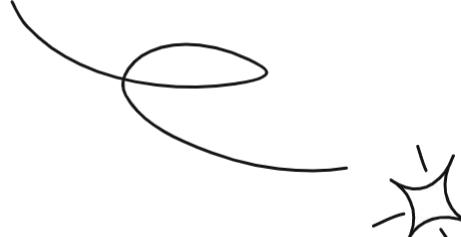
```
<form action="/action_page.php">
    <input type="radio" id="html" name="fav_language" value="HTML">
    <label for="html">HTML</label><br>
    <input type="radio" id="css" name="fav_language" value="CSS">
    <label for="css">CSS</label><br>
    <input type="radio" id="javascript" name="fav_language" value="JavaScript">
    <label for="javascript">JavaScript</label><br><br>
    <input type="submit" value="Submit">
</form>
```

Полезно знать



## Формы —

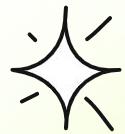
это просто HTML, можно использовать всю мощь HTML для структурирования HTML-формы.



Стать IT-шником может каждый



# ВОПРОСЫ



Полезно знать



Валидация формы —  
это проверка [данных](#), которые ввел пользователь.



# Преимущества валидации формы

Обеспечение  
корректности  
данных

Улучшение  
пользовательского  
опыта

Защита от  
нежелательных  
данных

Улучшение  
доступности



## <input> —

это тег, который используется для создания интерактивных элементов управления в веб-формах для получения данных от пользователя.





## type —

это атрибут тега `<input>`, который определяет, какой вид получают поля формы.



## Типы тега <input>

**text**

**email**

**search**

**tel**

Поле для ввода  
текста в одну  
строку

Поле для ввода  
электронной почты

Поле для поиска

Поле для ввода  
номера телефона

## Типы тега <input>

**url**

**password**

**number**

**submit**

Поле для ввода  
URL-адреса

Ввод пароля

Ввод числа

Простая кнопка  
для отправки  
формы

## Типы тега <input>

**button**

**image**

**reset**

**checkbox**

Обычная кнопка

Кнопка «Отправить»  
в виде картинки

Кнопка, которая  
сбрасывает всё, что  
пользователь ввёл в  
поля текущей  
формы ранее

Флажки, или  
чекбоксы, с  
несколько  
вариантами на  
выбор

Пиво  Чай  Кофе

## Типы тега <input>

**radio**

**range**

**color**

**file**

Круглая кнопка-переключатель для выбора 1 из нескольких вариантов

Ползунок для выбора чисел из заданного диапазона

Виджет для выбора цвета

Выбор файла для отправки

Пиво  Чай  Кофе

## Атрибуты тега <input>

**name**

**disabled**

**autocomplete**

**autofocus**

Уникальное имя  
элемента

Отключает  
элемент

Разрешает  
автозаполнение

Фокусируется на  
элементе <input>  
сразу после  
загрузки страницы

## Атрибуты тега <input>

**form**

**readonly**

**step**

**min, max**

Связывает отдельно стоящий элемент <input> с формой

Не позволяет пользователю как-либо изменять элемент, оставляя его при этом рабочим

Указывает шаг, с которым будет изменяться значение поля

Указывают минимальные и максимальные числовые значения для полей с типами number и range

## Атрибуты тега <input>

**size**

Указывает ширину поля для ввода в символах

**placeholder**

Короткая подсказка для помощи пользователю при вводе данных

**multiple**

Указывает, что пользователь может выбрать больше одного значения

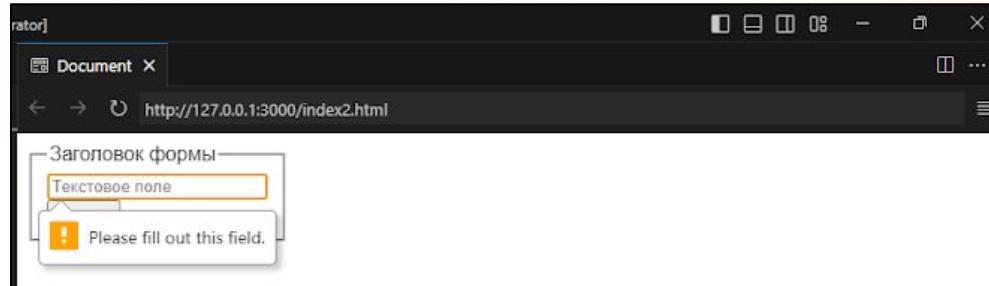
Стать IT-шником может каждый



# required



Определяет, что для отправки формы данное поле предварительно должно быть заполнено.



Стать IT-шником может каждый



## minlength и maxlenlength

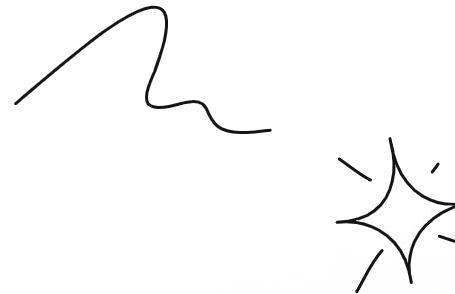


Задает минимальную и максимальную длину текстовых данных (строк).

Стать IT-шником может каждый



## min и max

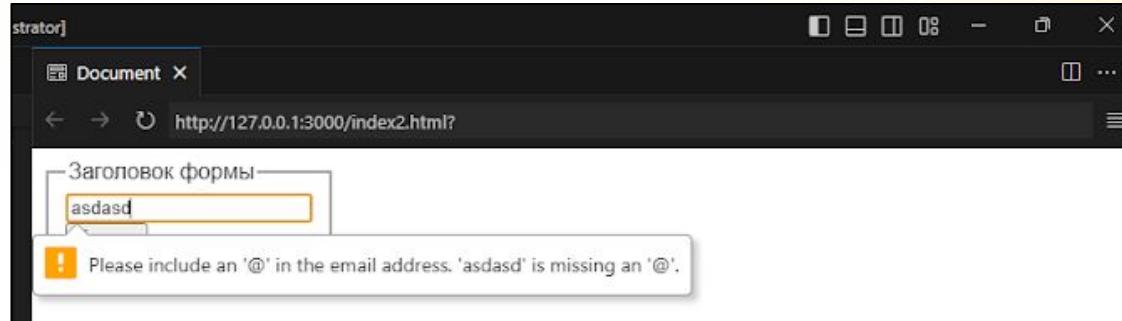


Задает минимальное и максимальное значение для поля, рассчитанного на числовой тип данных.

# type



Определяет тип данных, на который рассчитано поле: число, email-адрес или другой предустановленный тип.



Поле с type="email"

# type



Определяет тип данных, на который рассчитано поле: число, email-адрес или другой предустановленный тип.

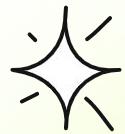
The screenshot shows a web browser window with the address bar displaying 'http://127.0.0.1:3000/index2.html?'. The main content area contains a form with a title 'Заголовок формы' and a single input field. The input field has the value '12' and a dropdown arrow icon to its right. Below the input field is a button labeled 'Кнопка'.

Поле с type=“number”

Стать IT-шником может каждый



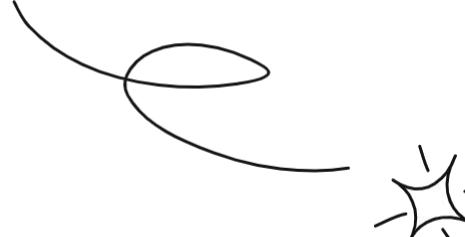
# ВОПРОСЫ





## Позиционирование —

это набор инструментов, с помощью которых можно перемещать элементы веб-страницы и определять их положение относительно браузерного окна и других объектов.



Стать IT-шником может каждый



# Важность позиционирования

Создание  
сложных  
макетов

Управление  
порядком  
наложения  
элементов

Создание  
адаптивных  
макетов

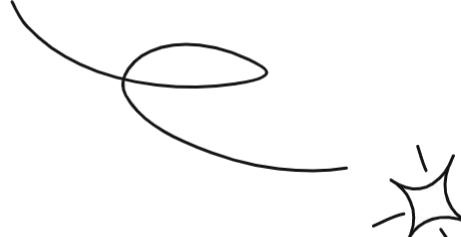
Размещение  
элементов на  
странице

Создание  
эффектов и  
анимаций



## Статическое позиционирование —

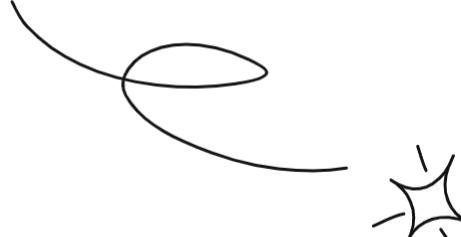
это значение по умолчанию для свойства "position". Элементы со статическим позиционированием располагаются в потоке документа и не подвержены изменению позиции с помощью свойств "top", "right", "bottom" и "left".





## Относительное позиционирование —

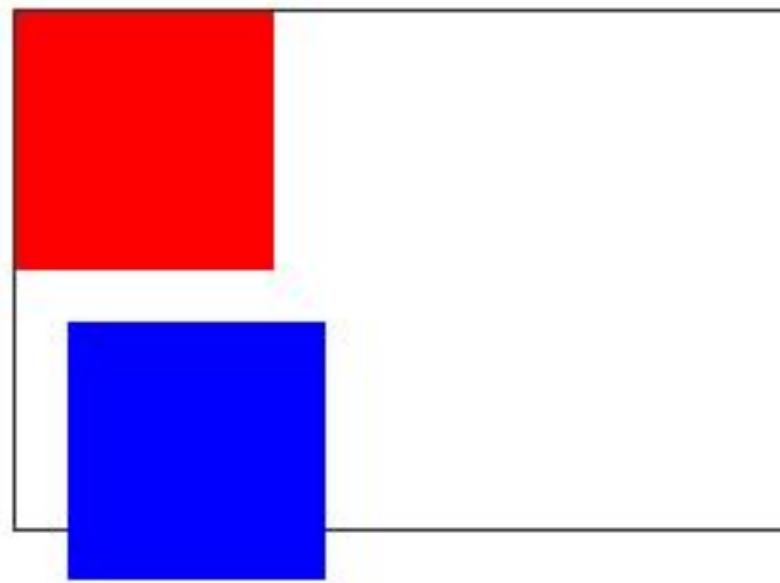
элементы могут быть сдвинуты относительно их исходного местоположения с помощью свойств "top", "right", "bottom" и "left". Они занимают свое исходное место в потоке документа, и другие элементы не изменяют свою позицию из-за них.



Стать IT-шником может каждый



## Пример относительного позиционирования



# Пример относительного позиционирования

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Title</title>
    <style>
        .container {
            position: relative;
            width: 300px;
            height: 200px;
            border: 2px solid #333;
        }

        .box {
            position: relative;
            width: 100px;
            height: 100px;
            background-color: red;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="box"></div>
        <div class="box relative-box"></div>
    </div>
</body>
</html>
```

```
.relative-box {
    position: relative;
    top: 20px;
    left: 20px;
    background-color: blue;
}
</style>
</head>
<body>
    <div class="container">
        <div class="box"></div>
        <div class="box relative-box"></div>
    </div>
</body>
</html>
```

# top, bottom, left, right

**top**

**bottom**

**left**

**right**

Определяет расстояние от верхнего края окна (или родительского элемента) до верхнего края дочернего элемента.

Определяет расстояние от нижнего края окна (или родительского элемента) до нижнего края дочернего элемента.

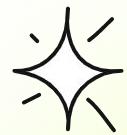
Определяет расстояние от левого края окна (или родительского элемента) до левого края дочернего элемента.

Определяет расстояние от правого края окна (или родительского элемента) до правого края дочернего элемента.

Стать IT-шником может каждый



# ВОПРОСЫ





## Абсолютное позиционирование —

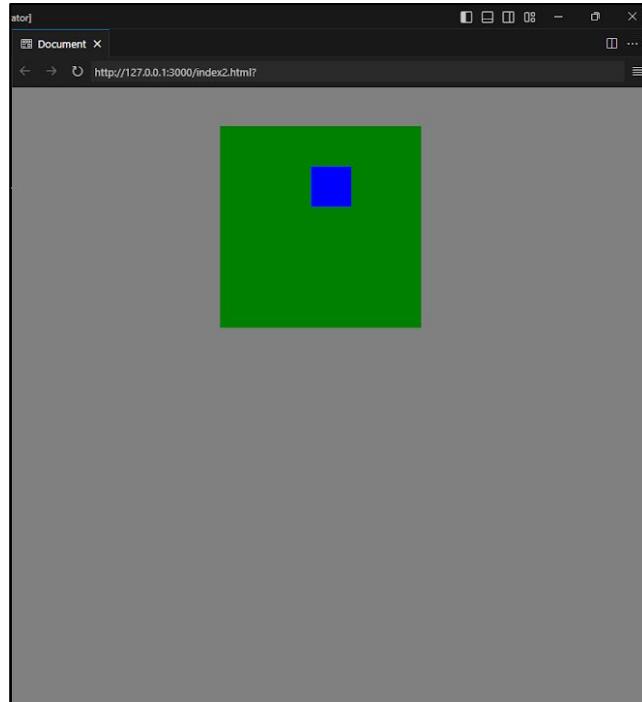
элементы полностью удаляются из потока документа и позиционируются относительно ближайшего родительского элемента с позиционированием, отличным от "static". Если такого элемента нет, то позиционирование происходит относительно окна просмотра.



Стать IT-шником может каждый



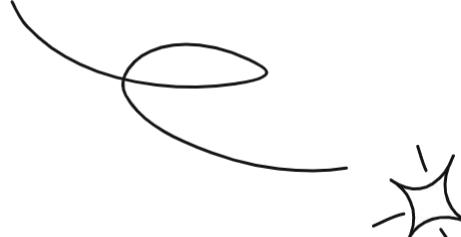
# Пример абсолютного позиционирования





## position fixed —

это свойство, которое позволяет позиционировать элемент не относительно родителя, а относительно окна браузера вне зависимости от вложенности.



# Пример position: fixed

The screenshot shows a website layout with a black header bar at the top. Below it is a white sidebar containing navigation links: [о нас](#), [о классике](#), [контакты](#), and [блог](#). The main content area has a blue background. It contains three blocks of text. The first block starts with "Да и зачем оно, это дикое и грандиозное? Море, например? Бог с ним! Оно наводит только грусть на человека: глядя на него, хочется плакать. Сердце смущается робостью перед необозримой пеленой вод, и не на чем отдохнуть взгляду, измученному однообразием бесконечной картины." The second block continues with "Рёв и бешеные раскаты валов не нежат слабого слуха: они всё твердят свою, от начала мира одну и ту же песнь мрачного и неразгаданного содержания; и все слышится в ней один и тот же стон, одни и те же жалобы будто обречённого на муку чудовища, да чьи-то пронзительные, зловещие голоса. Птицы не щебечут вокруг; только безмолвные чайки, как осуждённые, уныло носятся у прибрежья и кружатся над водой." The third block ends with "Бессилен рёв зверя перед этими воплями природы, ничтожен и голос человека, и сам человек так мал, слаб, так незаметно исчезает в мелких подробностях широкой картины! От этого, может быть, так и тяжело ему смотреть на море.

Нет, Бог с ним, с морем! Самая тишина и неподвижность его не



## position sticky —

это свойство, которое позволяет позиционировать элемент в нормальном потоке так же, как и статичный, а затем смещать при помощи свойств left, right, top, bottom или inset относительно ближайшего родителя, имеющего прокрутку.



# Пример position: fixed

The screenshot shows a mobile application interface with a dark background. At the top, there is a black header bar containing the text "Стать IT-шником может каждый". Below the header, the main content area features two chapters of a book, each enclosed in a blue rectangular box.

**Chapter IX (Left):**

**Глава IX**

Где мы? В какой благословенный уголок земли перенёс нас сон  
Обломова? Что за чудный край!

Нет, правда, там моря, нет высоких гор, скал и пропастей, ни дремучих  
лесов — нет ничего грандиозного, дикого и угрюмого.

Да и зачем оно, это дикое и грандиозное? Море, наприм.  
Оно наводит только грусть на человека: глядя на него, х  
Сердце смущается робостью перед неизбримой пелен

**Chapter X (Right):**

рождают отрадного чувства в душе: в едва заметном колебании  
водяной массы человек все видит ту же необъятную, хотя и спящую

**Глава IX**

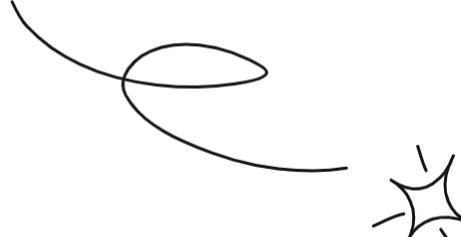
Горы и пропасти созданы тоже не для увеселения человека. Они  
грозны, страшны, как выпущенные и устремлённые на него когти и  
зубы дикого зверя; они слишком живо напоминают нам бренный  
состав наш и держат в страхе и тоске за жизнь. И небо там, над  
скалами и пропастями, кажется таким далёким и недосягаемым, как  
будто оно отступило от людей.

Не таков мирный уголок, где вдруг очутился наш герой.



## z-index —

это [свойство](#), которое используется для контроля порядка наложения элементов на веб-странице. Оно определяет, какие элементы должны находиться перед или позади других элементов при их перекрытии.





## z-index

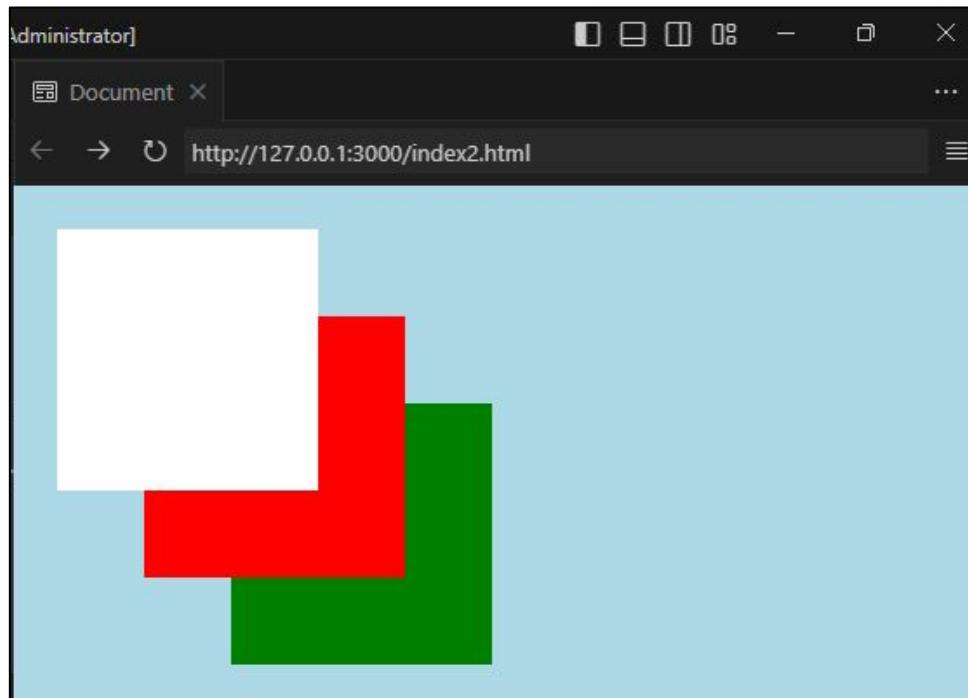


- z-index принимает целочисленные значения и определяет порядок наложения на основе этих значений.
- По умолчанию элементы имеют значение z-index равное auto.
- Свойство z-index работает только для элементов с указанной позицией (position).
- Если два перекрывающихся элемента не имеют указанного значения z-index, то элемент, который встречается позже в HTML-коде, будет отображаться поверх другого.
- Отрицательные значения z-index могут использоваться для позиционирования элементов позади других.

Стать IT-шником может каждый



## Пример z-index



Стать IT-шником может каждый



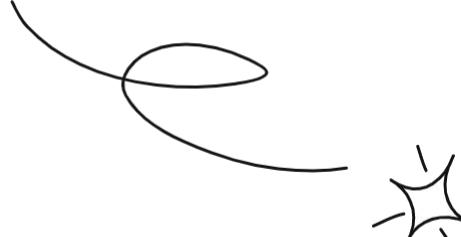
# ВОПРОСЫ





## Псевдоклассы —

это особый вид селектора, который уточняет тип или состояние. Обычно это какой-то качественный признак: реакция на наведение курсора, порядок следования и другие.



## Синтаксис псевдоклассов



```
div:hover {  
    background-color: #f89b4d;  
}
```

## Применение псевдоклассов

Контроль динамических параметров селекторов

Возможность стилизовать элемент на основе внешних факторов: истории посещений, состояния содержимого или позиций курсора мыши

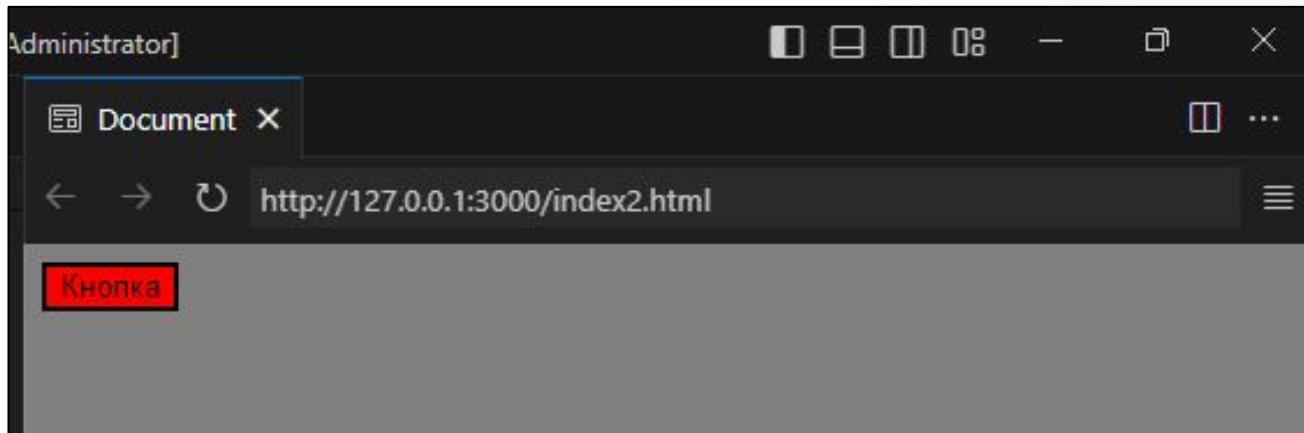


## :active —

это стандартный псевдокласс, который применяется к интерактивным элементам в момент взаимодействия.



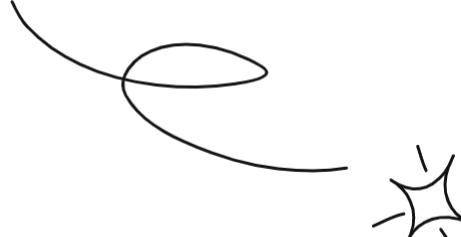
## Пример :active





## :is() —

это [стандартный псевдокласс](#), который позволяет сгруппировать схожие селекторы вместо последовательного перечисления через запятую. Будет полезно при группировке большого количества селекторов.



## Пример :is()

```
h1 a, h2 a, h3 a, h4 a, h5 a, h6 a {  
    /* ... */  
}  
/* сокращается до */  
:is(h1, h2, h3, h4, h5, h6) a {  
    /* ... */  
}
```



## :link —

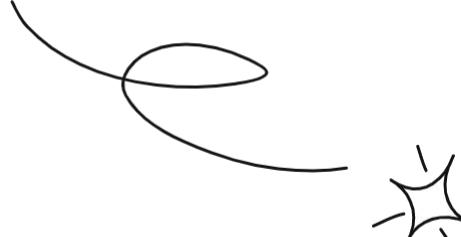
это **стандартный псевдокласс**, который характеризует еще не посещенные страницы.



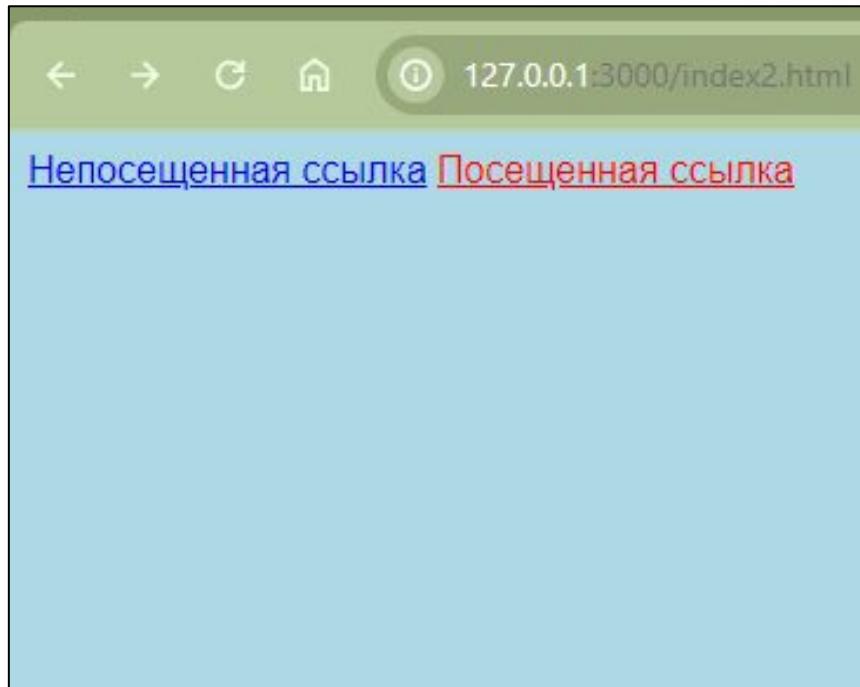


## :visited —

это стандартный псевдокласс, который характеризует посещенные страницы (в рамках одного домена).



## Пример :link





## :checked —

это стандартный псевдокласс, который применяется к элементам, состояние которых меняется с помощью атрибута checked (radio, checkbox).





## :disabled, :enabled —

это стандартный псевдокласс, который позволяют находить элементы формы по состоянию их атрибута disabled.



## Пример :disabled, :enabled

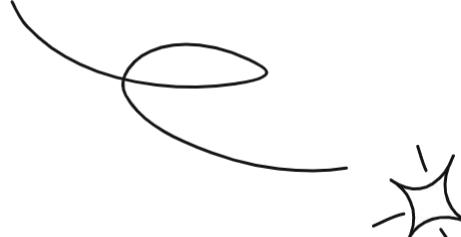
The screenshot shows a browser window with the title bar "Administrator" and the tab "Document X". The address bar displays the URL "http://127.0.0.1:3000/index2.html". The main content area contains three colored circles: red, yellow, and green. To the right of the browser window, a dark blue rounded rectangle contains the following CSS code:

```
button:disabled {  
    opacity: 50%;  
}
```

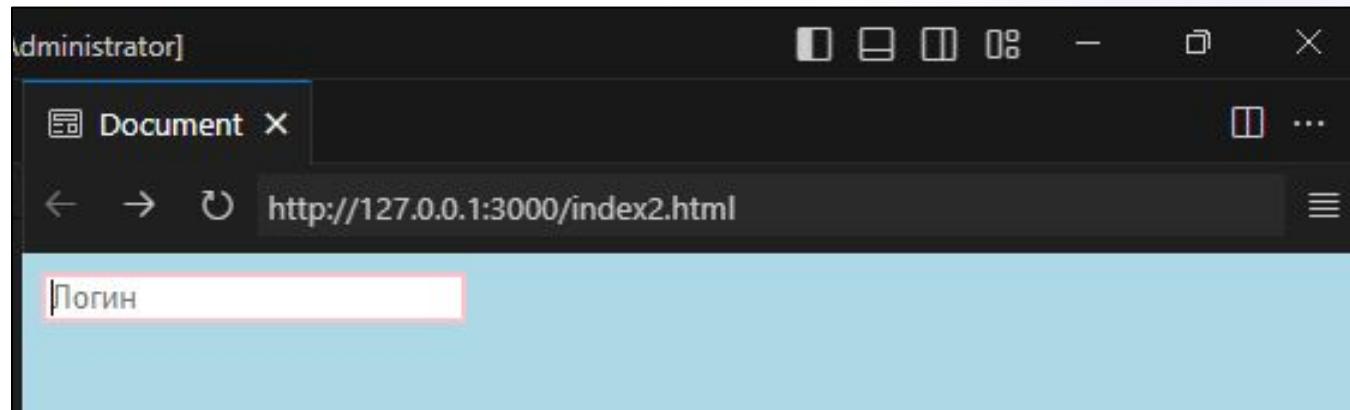


## :focus —

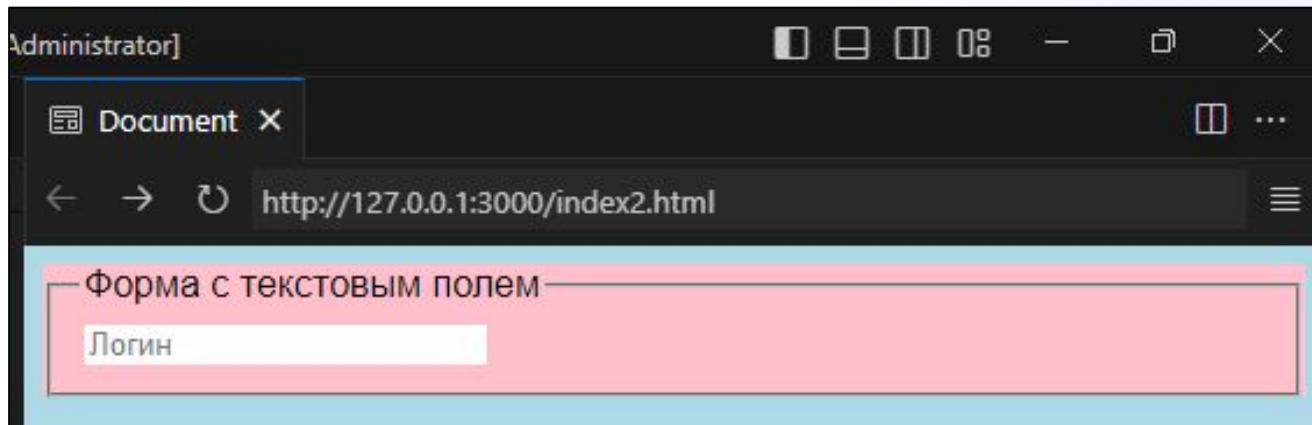
это стандартный псевдокласс, который характеризует элемент, который сейчас находится в фокусе. А :focus-within ещё обозначает элемент, внутри которого находится элемент в фокусе.



## Пример :focus



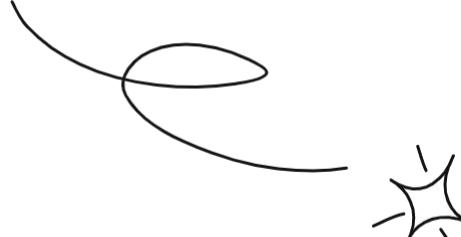
## Пример :focus-within





## :hover —

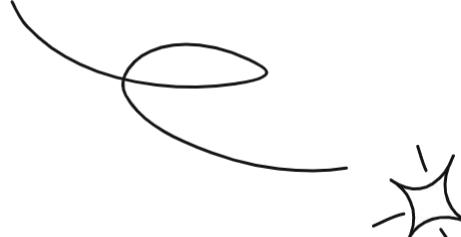
это **стандартный псевдокласс**, который характеризует состояние элемента, когда на него навели курсор.





## :fullscreen —

это [стандартный псевдокласс](#), который является признаком того, что документ развернули на всё окно (с помощью JavaScript).



## Псевдоклассы группы child

**:first-child**

**:last-child**

**:nth-child(n)**

**:nth-last-child(n)**

Первый дочерний элемент

Последний дочерний элемент

п-ный дочерний элемент, где п - номер элемента

п-ный дочерний элемент с конца, где п - номер элемента

## Псевдоклассы группы child

**:only-child**

**:nth-child(odd)**

**:nth-child(even)**

Выбирает любой элемент,  
который является единственным  
дочерним элементом своего  
родителя

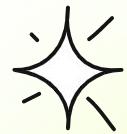
Выбирает нечётные элементы  
внутри родителя

Выбирает чётные элементы  
внутри родителя

Стать IT-шником может каждый



# ВОПРОСЫ





## Псевдоэлементы —

это элементы, которых не существует в HTML-разметке. Они создаются и позиционируются исключительно при помощи CSS. Чаще всего используются для создания различных декоративных элементов.



Полезно знать



## ::before и ::after —

это псевдоэлементы в CSS, которые позволяют добавлять контент до или после содержимого другого элемента.



## Пример ::before и ::after

```
h1::before {  
    content: "before- ";  
}  
  
h1::after {  
    content: " -after";  
}
```

before- Заголовок -after

Полезно знать

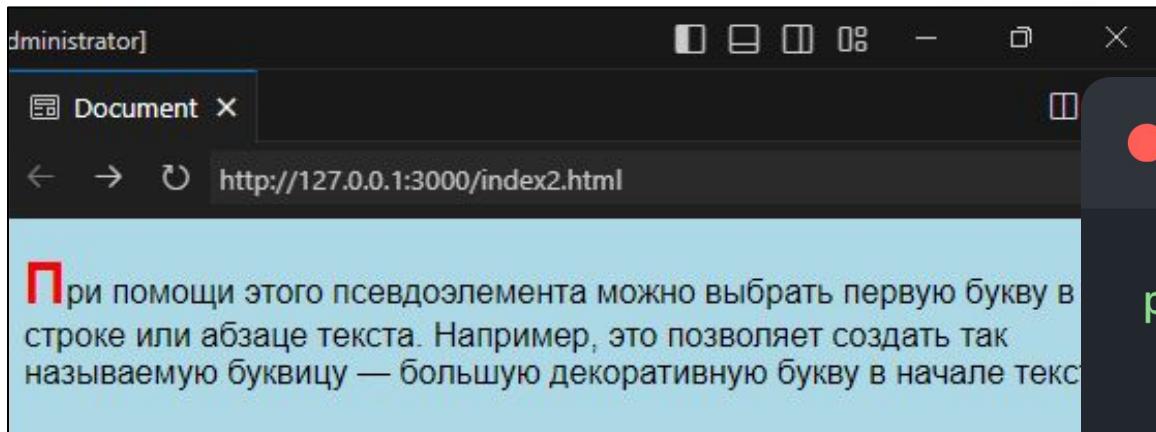


## ::first-letter —

это псевдоэлемент в CSS, который позволяет выбрать первую букву в строке или абзаце текста



## Пример ::first-letter

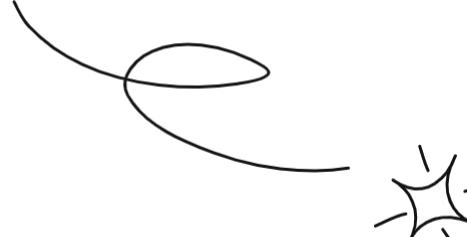


```
p::first-letter {  
    font-size: 28px;  
    font-weight: 600;  
    color: red;  
}
```

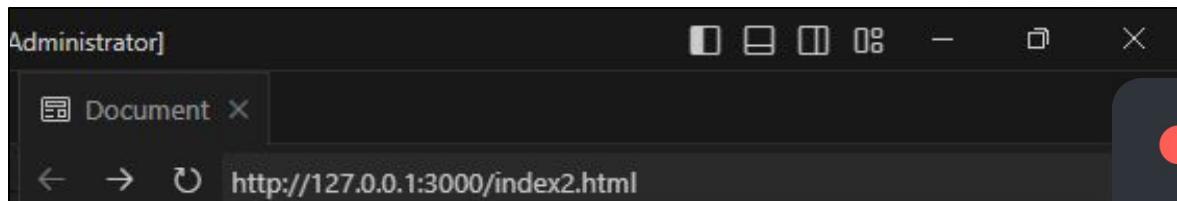


## ::first-line —

это псевдоэлемент в CSS, который позволяет выбрать первую строку текста. Работает только с блочными элементами.



## Пример ::first-line

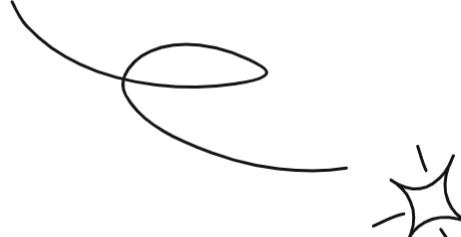


```
p::first-line {  
    font-size: 28px;  
    font-weight: 600;  
    color: red;  
}
```

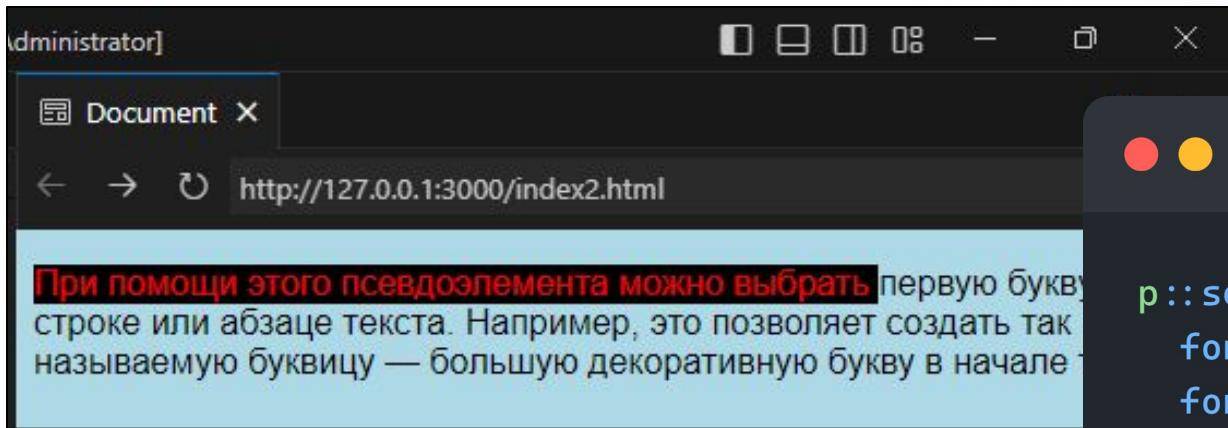


## ::selection —

это псевдоэлемент в CSS, который позволяет управлять стилем текста, который пользователь выделяет при помощи мыши.



## Пример ::selection



The screenshot shows a web browser window titled "Document" with the URL "http://127.0.0.1:3000/index2.html". The page content is a paragraph of text: "При помощи этого псевдоэлемента можно выбрать первую букву строке или абзаце текста. Например, это позволяет создать так называемую буквицу — большую декоративную букву в начале". A portion of the first sentence is highlighted with a red background and white text, demonstrating the ::selection pseudo-element.

```
p::selection {  
    font-size: 28px;  
    font-weight: 600;  
    color: red;  
    background-color: black;  
}
```

Полезно знать



## ::placeholder —

это псевдоэлемент в CSS, который позволяет стилизовать подсказку, которая выводится в поле ввода текста (<input>).



## Пример ::placeholder

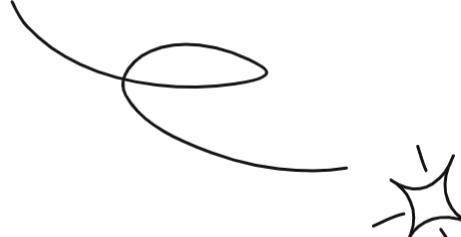
```
input::placeholder {  
    color: blue;  
    font-size: 18;  
}
```

Placeholder



## ::marker —

это псевдоэлемент в CSS, который отвечает за маркерное поле элементов списков (ul, ol).



## Пример ::marker

```
ul li::marker {  
    color: red;  
    font-size: 2em;  
}
```

- Item 1
- Item 2
- Item 3



## transition —

это **свойство**, которое используется, когда нам нужно плавно изменить CSS-свойства между двумя состояниями элемента.



## Составные свойства transition

**transition-property**

**transition-duration**

**transition-timing-function**

**transition-delay**

Указываем свойство, которое хотим плавно изменить (background-color, width, left)

Длительность перехода

Функция, описывающая скорость изменения свойства

Задержка перед началом изменения



## transition-timing-function —

это **свойство**, которое определяет функцию зависимости значения свойства от времени.



# Скорость течения анимации

**ease**

**ease-in**

**ease-out**

**ease-in-out**

Анимация начинается медленно, затем ускоряется и к концу движения опять замедляется

Анимация медленно начинается, к концу ускоряется

Анимация начинается быстро, к концу замедляется

Анимация начинается и заканчивается медленно

# Скорость течения анимации

**linear**

**step-start**

**step-end**

**steps**

Однаковая скорость от начала и до конца

Как таковой анимации нет. Стилевые свойства сразу же принимают конечное значение

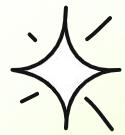
Как таковой анимации нет. Стилевые свойства находятся в начальном значении заданное время, затем сразу же принимают конечное значение

Ступенчатая функция, имеющая заданное число шагов

Стать IT-шником может каждый



# ВОПРОСЫ



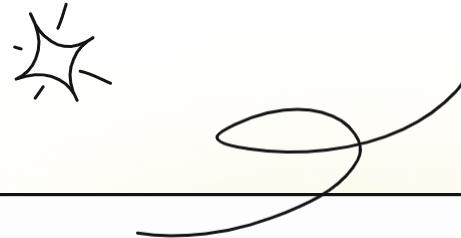
Стать IT-шником может каждый



# ЗАДАНИЕ ДЛЯ ЗАКРЕПЛЕНИЯ



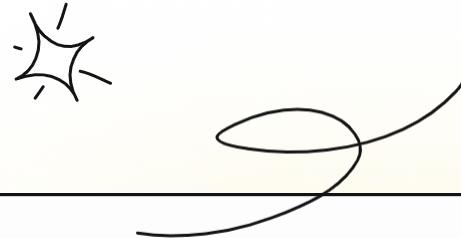
## Задание для закрепления:



Какой псевдокласс в CSS3 применяется для выбора элементов, не соответствующих заданному селектору?

1. :not-this
2. exclude
3. :not
4. except

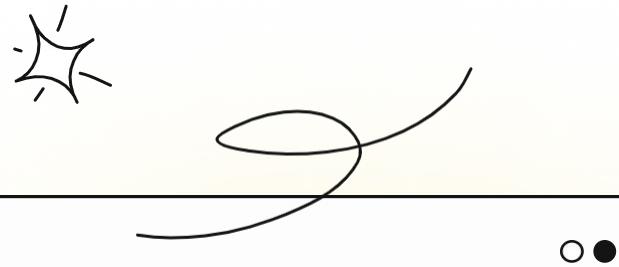
## Задание для закрепления:



Какой псевдокласс в CSS3 используется для обращения к корневому элементу документа, обычно представленному тегом <html>?

1. :root
2. main
3. global

## Задание для закрепления:



Какие два символа используются для обозначения псевдоэлементов в CSS3?

1. \*\*
2. ::
3. :::

Стать IT-шником может каждый



# ВОПРОСЫ



Стать IT-шником может каждый



# РАЗБОР ДОМАШНЕГО ЗАДАНИЯ



## Домашнее задание



Создайте HTML-страницу с контейнером(div) для списка карточек(div).

Используя CSS и свойства Flexbox, сделайте так, чтобы карточки располагались в строку и растягивались по ширине контейнера. Можно использовать любое кол-во карточек.

Каждая карточка должна содержать изображение(любое изображение) и название (любой текст).

Убедитесь, что при уменьшении размеров экрана карточки остаются читаемыми и красиво адаптируются.

# Домашнее задание



Сверстать форму по образцу. Наполнение и цвета могут быть произвольными.

Использовать flex-direction: column для формы, и flex для блока с двумя fieldset, кнопку отправки центрировать с помощью flex.

Имя—

Фамилия

Имя

Пол—

Мужской

Женский

Не хочу говорить

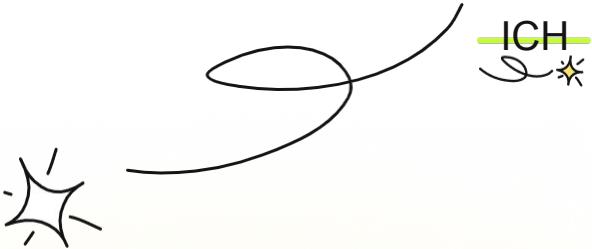
Получать рассылку по—

Email

SMS

Отправить

# Домашнее задание



Создать HTML-страницу, которая содержит 3 блока с заголовком и текстом, заголовок должен “прилипать” к верхней части страницы при прокрутке. Тегу body задать высоту в 2000px. Наполнение блоков произвольное, можно воспользоваться сайтом <https://www.lipsum.com/>.

## What is Lorem Ipsum 1

Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem ipsum.

## What is Lorem Ipsum 2

Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem ipsum.

## What is Lorem Ipsum 3

Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem ipsum.

## What is Lorem Ipsum 1

Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem ipsum.

## What is Lorem Ipsum 2

Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem ipsum.

## What is Lorem Ipsum 3

Lorem ipsum is simply dummy text of the printing and typesetting industry. Lorem ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled it to make a type specimen book. It has survived not only five centuries, but also the leap into electronic typesetting, remaining essentially unchanged. It was popularised in the 1960s with the release of Letraset sheets containing Lorem ipsum passages, and more recently with desktop publishing software like Aldus PageMaker including versions of Lorem ipsum.

## Домашнее задание

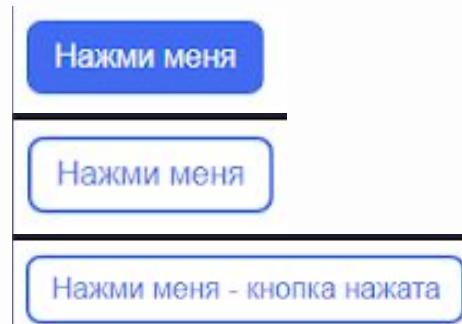


Создать HTML-страницу, которая содержит кнопку. Кнопку стилизовать на свое усмотрение, можно опираться на скриншоты.



Кнопка должна иметь 3 состояния:

- покой - основные стили
- наведение - можно изменить цвет текста, фон, границу и т.д.
- нажатие - должен появляться текст "Кнопка нажата"



Для смены состояний использовать анимированный переход.

Стать IT-шником может каждый



## ЗАКЛЮЧЕНИЕ

