

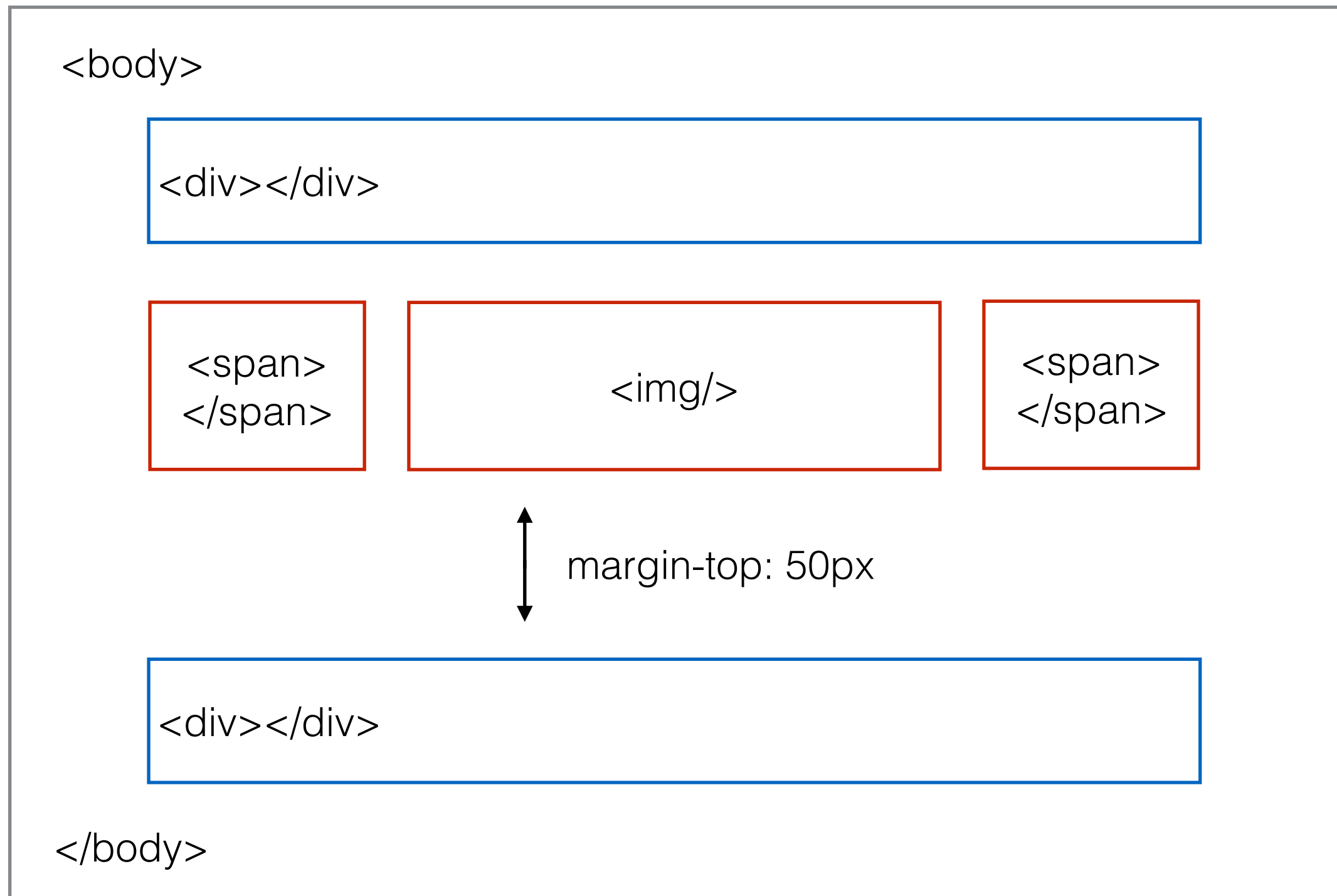
Таблицы, формы, позиционирование

План урока

1. Позиционирование
2. Для чего нужны таблицы? Где раньше использовали таблицы?
3. Структура таблицы. (Тело таблицы, заголовки таблицы, строки, столбцы)
4. Стилизация таблиц. Особенности в верстке таблиц.
5. Создадим таблицу данных.
6. Формы. Разбираем тег `<form>`
7. Какие бывают `input`'ы?
8. Атрибуты `input`'ов
9. Дополнительные элементы форм `<select>`, `<textarea>` и `<button>`

Стандартный поток элементов.

В стандартном потоке, элементы идут слева – направо и сверху – вниз.
Элементы «отталкиваются» друг от друга. Они в одном потоке.

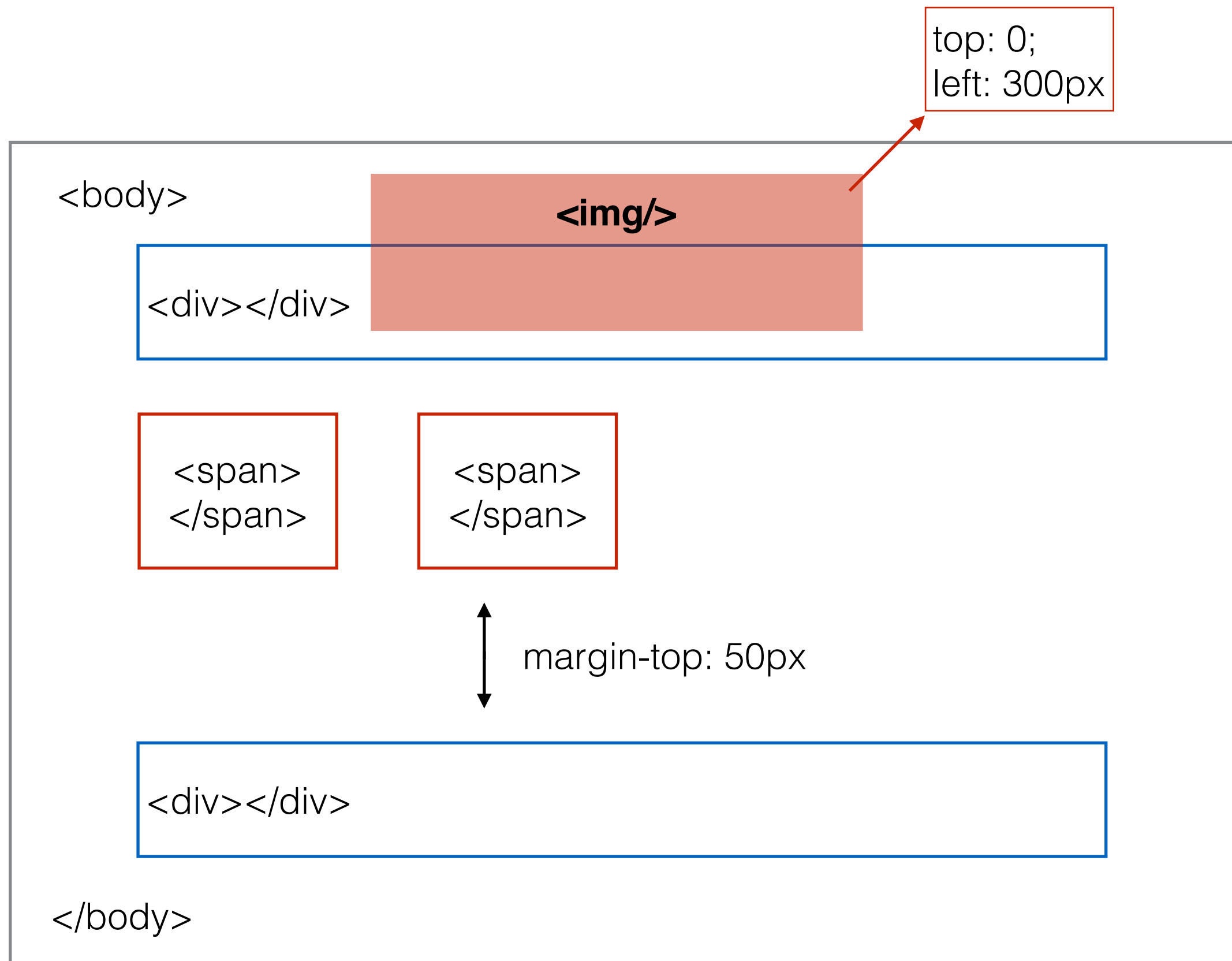


Позиционирование.

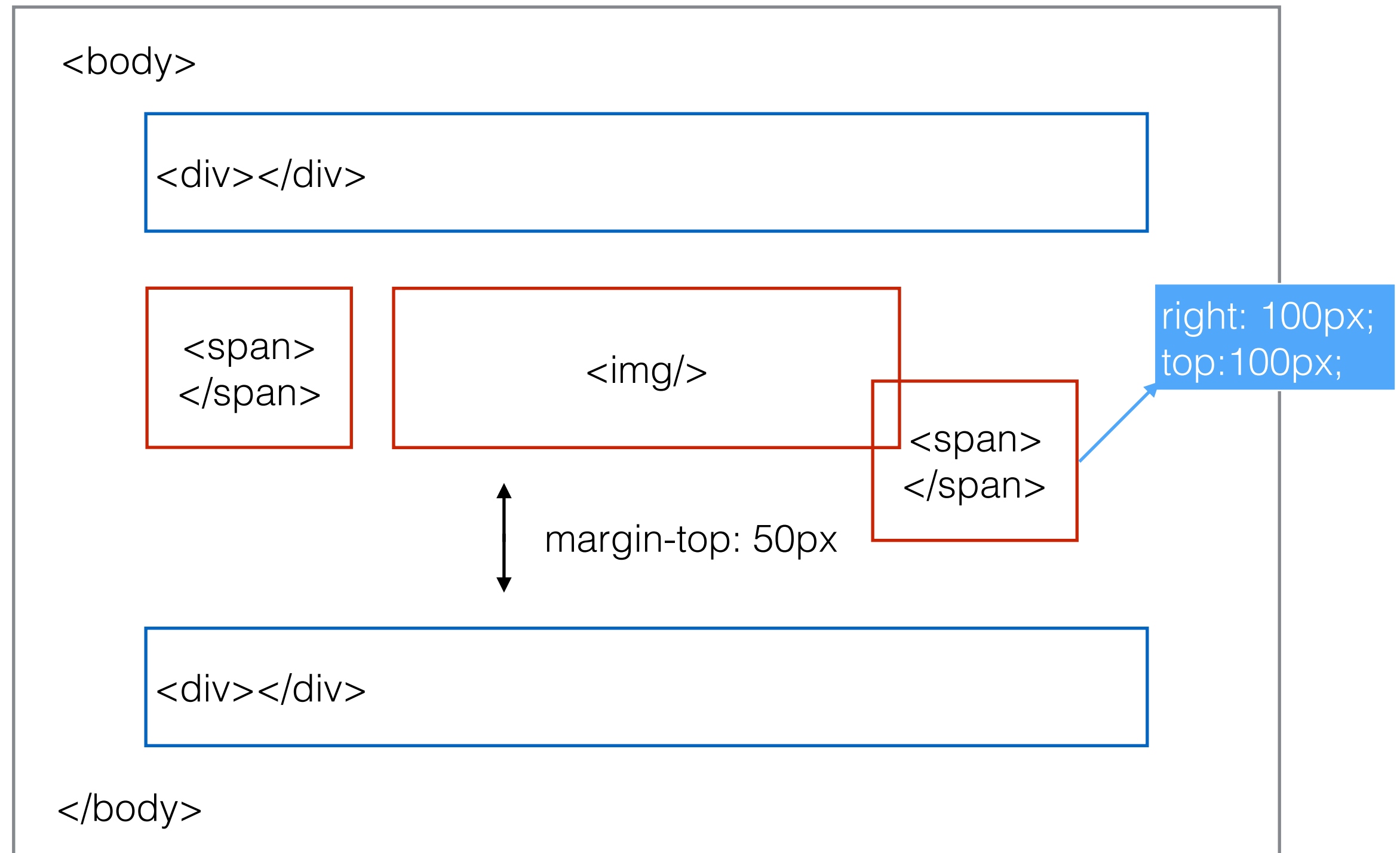
Мы можем изменить положение элемента в стандартном потоке.
Для этого у нас есть свойство:

1. **absolute** – «Вытягивает» элемент из стандартного потока, позиционируя его абсолютно, начиная отсчет от верхнего-левого угла страницы или элемента с другим position
2. **fixed** – похожее поведение на absolute, но элемент прилипает к позиции.
3. **relative** – элемент остался в потоке, но смещается относительно своего положения на странице
4. **static** – стандартное значение, свойства *top*, *bottom*, *left*, *right* – никакого эффекта не дают.

Пример абсолютного позиционирования



Пример относительного позиционирования



Свойство z-index

z-index: `auto` | `number` | `initial` | `inherit`;

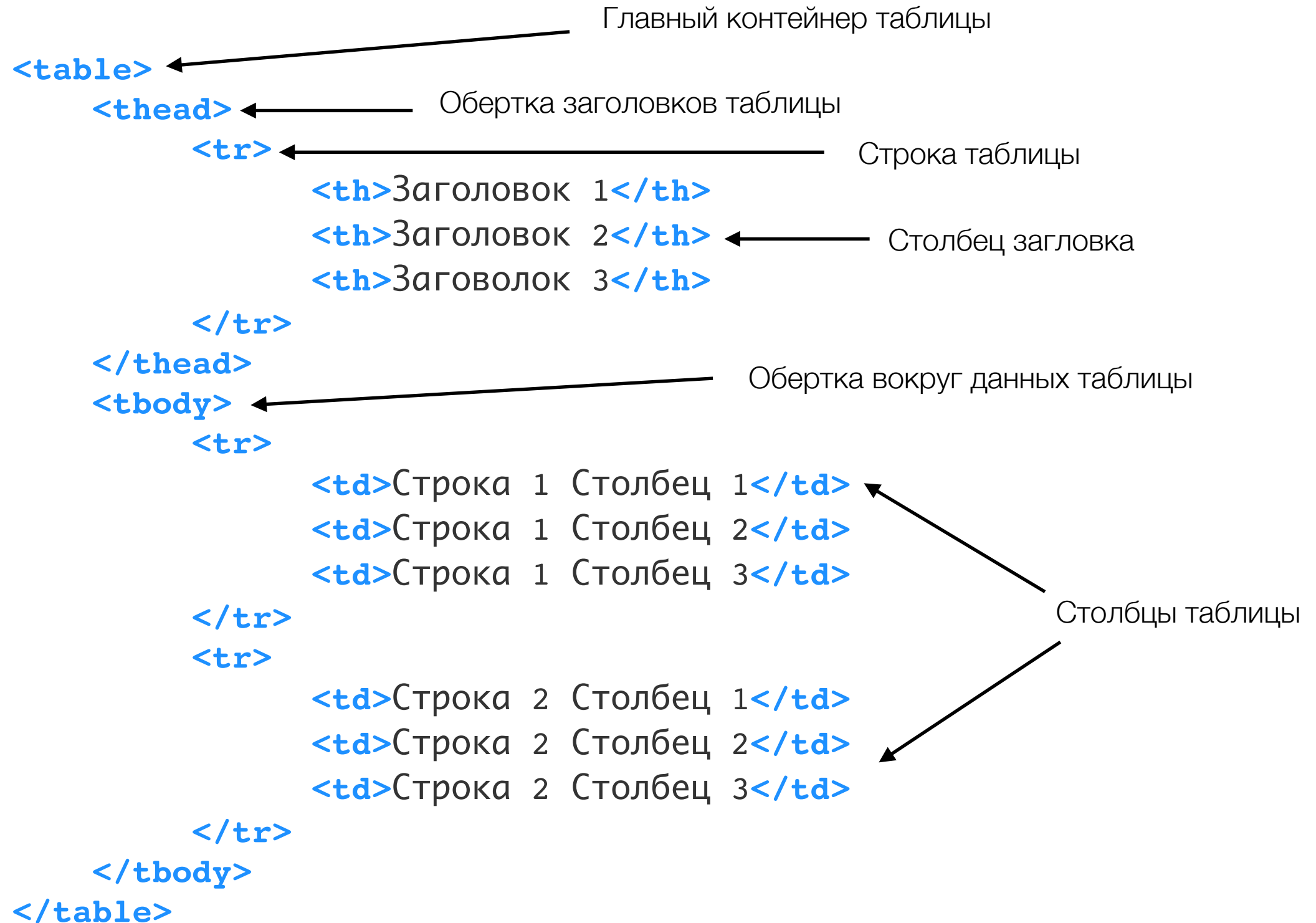
1. z-index – указывает какой элемент будет выше при позиционировании.
2. Тот у кого значение z-index выше, будет находится выше.
3. Если у родительского элемента z-index указан, то дочерний элемент никак не сможет его изменить.
4. z-index дочернего элемента будет работать только в контексте родителя

Таблицы в HTML. Для чего нужны?

Где раньше использовали таблицы?

Логотип	Меню	Контакты
Список статей	Основной контент	Правый сайдбар

Структура таблицы в HTML



Стилизация таблиц. Особенности.

1. В столбцах значения уже выровнены по центру с помощью свойства **vertical-align: middle;**
2. Стилизовать цвет, рамки, высоту, ширину, шрифт... и прочие визуальные элементы – лучше всего напрямую у тега **<td>**
3. **border-collapse: collapse** – отвечает за объединение бордеров. Его применяем ко всей таблице.
4. В таблицу без проблем можно поместить и стилизовать любой элемент: текст, картинку, инпут. Предпочтение отдается строчным элементам, однако на блочные никаких ограничений нет.
5. С помощью селекторов псевдоклассов – можно получить эффект зебры на таблице.

Практическая задача

Company	Q1	Q2	Q3	Q4
Microsoft	20.3	30.5	23.5	40.3
Google	50.2	40.63	45.23	39.3
Apple	25.4	30.2	33.3	36.7
IBM	20.4	15.6	22.3	29.3

<https://html5book.ru/krasivoe-oformlenie-tablic/>

Формы. Тег *<form>*

```
<form action="/action_page.php" method="get">  
    // Инпуты  
</form>
```

action – ссылка на страницу, которая обрабатывает данные формы. Этот атрибут обязателен, мы можем указать «#», если не знаем путь к странице

method – метод, которым передаются данные на сервер. Существует два значения post и get

enctype – метод шифрования данных. Существует 3 значения, но это на данном этапе нам неважно. В дальнейшем, нужно менять значение этого атрибута на **form-data**, иначе файл не передастся.

Так же есть другие атрибуты: **target, autocomplete, novalidate, accept-charset, name**

Типы input'ов

- **text** – тип по умолчанию, обычный текстовый инпут
- **number** – тип для чисел. Появляются справа стрелки для увеличения/уменьшения числа
- **email** – тип для задания email, очень похож на text
- **password** – прячет введенные символы за точками
- **checkbox** – позволяет поставить чекбокс(множественный выбор)
- **radio** – Выбираем одно из двух значений
- **hidden** – скрытый инпут.
- **submit** – инпут-кнопка, подтверждает отправку формы
- **reset** – при нажатии на эту кнопку, сбрасывает все значения

Атрибуты input'ов

1. **type** – указываем тип инпута. Обязательный атрибут, если мы его не укажем, валидатор выдаст ошибку. Если не указан атрибут или стоит незнакомый браузеру тип атрибута – мы получаем значение text
2. **name** – обязательный атрибут, без него сервер не сможет распознать значение инпута
3. **value** – обязательный атрибут для инпутов **checkbox, radio, submit**
4. **disabled** – делаем инпут неактивным
5. **required** – делаем инпут обязательным к заполнению
6. **autocomplete** – автозаполнение инпута. Имеет визуальный побочный эффект
7. **checked** – выбирает чекбокс или радиобаттон по умолчанию
8. **autofocus** – делает инпут фокусирующимся по умолчанию при попадании на страницу.

Дополнительные элементы в формах:

<select>, <button>, <textarea>

1. **<select>** – создает выпадающий список со значениями определенными в теге **<option>**

```
<select name="city">  
  <option value="kiev">Киев</option>  
  <option value="lvov">Львов</option>  
  <option value="kharkov">Харьков</option>  
</select>
```

2. **<button>** – создает кнопку в форме
3. **<textarea>** – создает текстовое поле для нескольких строк. В отличии от input'а с типом text – требует обязательного закрытия тега!

Практическая задача

1. Создадим форму. С обязательными полями: **имя, фамилия, номер телефона**. Необязательные поля: **пол, возраст** (можно выбрать от 18 до 99), **электронная почта, соглашение на подписку**(поставить галочку), **выбор из выпадающего списка** страны проживания(Для примера ставим любые 3 страны), **текстовый блок** с многострочным описанием.
2. Во всех текстовых полях **должны быть указаны placeholder!**
3. Добавить кнопку через тег `<input>`. При клике на кнопку мы должны попасть на страницу с надписью: Ваша форма была успешно отправлена!
4. **БОНУС:** Добавьте возможность загружать произвольный файл. Так же добавьте проверки к полям имени и электронной почты.
5. **БОНУС:** Отключите инпут где мы вводим электронную почту.

И еще немного практики...

1. Снова сгенерируем 2 абзаца, или больше, для того, чтобы получить вертикальный скролл на странице.
2. Добавим любое изображение, ограничим его по ширине в 100px.
3. Разместим изображение абсолютно с координатами top: 100px и left: 100px
4. Добавим второе изображение(можно такое же), разместим с фиксированным position и значениями top: 50px, right: 0;
5. Добавляем между абзацами еще одно изображение, применяем position: relative, значения top, left – любые, отличные от стандартных(0)
6. Что будет если изменить z-index элементу без позиционирования?
7. А что будет если указать z-index: -1?

Кто ответит на последние 2 вопроса
и выполнит задание – может быть свободен