

# Выравнивание элементов форм

До появления флексбоксов выравнивание элементов формы порой вызывало множество трудностей. Но флексбоксы существенно упрощают этот процесс. Давайте начнём с простого примера.

Вот форма *без* флексбоксов.

☐ Тёмная тема

```
<!doctype html>
<title>Пример</title>
<style>
.wrapper {
  background-color: whitesmoke;
  list-style-type: none;
  padding: 0;
  border-radius: 3px;
}
.form-row {
  padding: .5em;
}
.form-row > label {
  padding: .5em 1em .5em 0;
}
.form-row > input,
.form-row > button {
  padding: .5em;
}
.form-row > button {
  background: gray;
  color: white;
  border: 0;
}
</style>
<form>
  <ul class="wrapper">
    <li class="form-row">
      <label for="name">Имя</label>
      <input type="text" id="name">
    </li>
    <li class="form-row">
      <label for="townborn">Место рождения</label>
      <input type="text" id="townborn">
    </li>
    <li class="form-row">
      <label for="email">Адрес электронной почты</label>
      <input type="email" id="email">
    </li>
    <li class="form-row">
      <button type="submit">Отправить</button>
    </li>
  </ul>
</form>
```

The screenshot shows a web browser window with the title 'Пример'. The address bar contains the text 'Найдите в Google или введите адрес'. The main content area displays a form with three input fields and a submit button. The first input field is labeled 'Имя', the second 'Место рождения', and the third 'Адрес электронной почты'. The submit button is labeled 'Отправить'.

Данная форма использует небольшое форматирование, но ширина всех полей ввода одинакова. Это приводит к тому, что правый край выглядит немного неровным, поскольку поля формы не располагаются прямо друг под другом. Если вас подобное устраивает, то хорошо, и нет необходимости что-либо делать дальше.

Но что если мы хотим растянуть поля ввода до правого края? Без флексбоксов получить такой результат было бы несколько болезненно. Особенно, если сама форма гибкая (то есть она расширяется и сжимается в зависимости от ширины родительского элемента или области просмотра). Если форма гибкая, то каждое поле ввода тоже должно расширяться и сжиматься по мере необходимости. В этом случае вы не можете использовать фиксированную ширину для полей ввода, потому что как только ширина формы изменится, ширина полей станет некорректной. Так что потребуется схитрить, чтобы всё работало правильно.

Тем не менее, флексбоксы позволяют сделать это без каких-либо магических трюков.

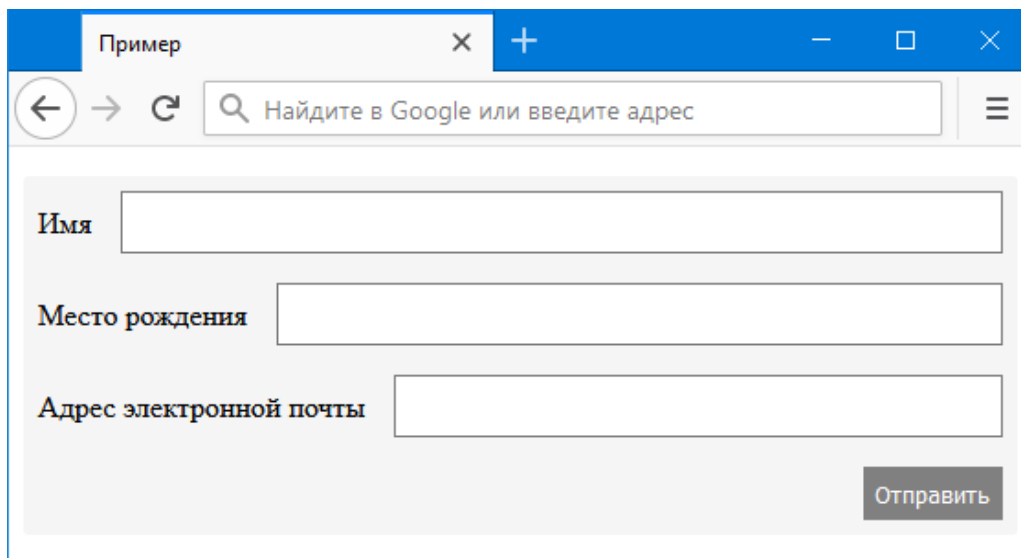
☐ Тёмная тема

```
<!doctype html>
<title>Пример</title>
<style>
.wrapper {
  background-color: whitesmoke;
  list-style-type: none;
  padding: 0;
  border-radius: 3px;
}
.form-row {
  display: flex;
  justify-content: flex-end;
  padding: .5em;
}
.form-row > label {
  padding: .5em 1em .5em 0;
}
.form-row > input {
  flex: 1;
}
.form-row > input,
.form-row > button {
  padding: .5em;
```

```

}
.form-row > button {
  background: gray;
  color: white;
  border: 0;
}
</style>
<form>
  <ul class="wrapper">
    <li class="form-row">
      <label for="name">Имя</label>
      <input type="text" id="name">
    </li>
    <li class="form-row">
      <label for="townborn">Место рождения</label>
      <input type="text" id="townborn">
    </li>
    <li class="form-row">
      <label for="email">Адрес электронной почты</label>
      <input type="email" id="email">
    </li>
    <li class="form-row">
      <button type="submit">Отправить</button>
    </li>
  </ul>
</form>

```



Сперва для каждого класса `.form-row` устанавливаем отображение `display: flex`.

```

.form-row {
  display: flex;
  justify-content: flex-end;
  padding: .5em;
}

```

Обратите внимание, что мы также использовали `justify-content: flex-end`, но это не обязательно. В этом случае кнопка просто перемещается вправо.

Теперь, когда каждая строка формы является флекс-контейнером, мы применяем следующий код к полям ввода.

```

.form-row > input {
  flex: 1;
}

```

Таким образом, применяя `flex: 1` к полям ввода, мы заставляем эти поля использовать всё доступное свободное пространство после меток `<label>`. При этом метки сохраняют свою обычную ширину с учётом `padding`. Поля ввода начинаются сразу после них и растягиваются до самого конца.

## Поля ввода одинаковой ширины

Вы можете изменить приведённый выше пример так, чтобы *левый* край полей формы также был в линию (как и правый край).

Вроде этого.

☐ Тёмная тема

```
<!doctype html>
<title>Пример</title>
<style>
.wrapper {
  background-color: whitesmoke;
  list-style-type: none;
  padding: 0;
  border-radius: 3px;
}
.form-row {
  display: flex;
  justify-content: flex-end;
  padding: .5em;
}
.form-row > label {
  padding: .5em 1em .5em 0;
  flex: 1;
}
.form-row > input {
  flex: 2;
}
.form-row > input,
.form-row > button {
  padding: .5em;
}
.form-row > button {
  background: gray;
  color: white;
  border: 0;
}
</style>
<form>
  <ul class="wrapper">
    <li class="form-row">
      <label for="name">Имя</label>
      <input type="text" id="name">
    </li>
    <li class="form-row">
      <label for="townborn">Место рождения</label>
      <input type="text" id="townborn">
    </li>
    <li class="form-row">
      <label for="email">Адрес электронной почты</label>
      <input type="email" id="email">
    </li>
    <li class="form-row">
      <button type="submit">Отправить</button>
    </li>
  </ul>
</form>
```

A screenshot of a web browser window with the title 'Пример'. The address bar shows a search prompt 'Найдите в Google или введите адрес'. The main content area contains a form with three input fields. The first field is labeled 'Имя', the second 'Место рождения', and the third 'Адрес электронной почты'. All three fields are of equal width. To the right of the third field is a button labeled 'Отправить'.

Всё, что мы здесь сделали, это добавили `flex: 1` к меткам, а для полей ввода изменили на `flex: 2`.

```
.form-row > label {
  padding: .5em 1em .5em 0;
  flex: 1;
}
.form-row > input {
  flex: 2;
}
```

Это задаёт ширину полей формы как удвоенная ширина меток. Вы можете настроить это соотношение в зависимости от ширины формы или ожидаемой ширины. К примеру, можно использовать соотношение 1 к 3 или 1 к 4 для широких форм.

Если вы используете форму с фиксированной шириной, то можете установить одно соотношение, которое лучше всего подходит для формы. Если это гибкая форма, которая меняет ширину в зависимости от ширины области просмотра, то можете использовать медиа-запросы, чтобы установить разное соотношение для каждой точки останова.

## Адаптивные поля ввода одинаковой ширины

Вот пример с медиа-запросами, которые меняют соотношение при изменении размера области просмотра.

☐ Тёмная тема

```
<!doctype html>
<title>Пример</title>
<style>
.wrapper {
  background-color: whitesmoke;
  list-style-type: none;
  padding: 0;
  border-radius: 3px;
}
.form-row {
  display: flex;
  justify-content: flex-end;
  padding: .5em;
}
.form-row > label {
  padding: .5em 1em .5em 0;
  flex: 1;
}
.form-row > input {
  flex: 2;
}
.form-row > input,
.form-row > button {
  padding: .5em;
}
.form-row > button {
```

```

background: gray;
color: white;
border: 0;
}
@media screen and (min-width: 768px) {
.form-row > input {
flex: 3;
}
}
@media screen and (min-width: 992px) {
.form-row > input {
flex: 4;
}
}
@media screen and (min-width: 1200px) {
.form-row > input {
flex: 5;
}
}
</style>
<form>
<ul class="wrapper">
<li class="form-row">
<label for="name">Имя</label>
<input type="text" id="name">
</li>
<li class="form-row">
<label for="townborn">Место рождения</label>
<input type="text" id="townborn">
</li>
<li class="form-row">
<label for="email">Адрес электронной почты</label>
<input type="email" id="email">
</li>
<li class="form-row">
<button type="submit">Отправить</button>
</li>
</ul>
</form>

```

Посмотрите эту форму на большом экране, затем уменьшите размер браузера. Вы должны увидеть как регулируется ширина полей ввода, когда вы уменьшаете окно браузера.