

## Вёрстка многослойных элементов интерфейса

Выполнен на 51%

Введение

Углублённая теория

Методика вёрстки многослойных элементов

Вёрстка кнопки «Наверх»

Вёрстка «липкой» горизонтальной навигации

Вёрстка «липкого» sidebar

Вёрстка окна с соглашением об использовании Cookie

Вёрстка пользовательской подсказки

Вёрстка компонента Карусель/Слайдер

Вёрстка компонента Выпадающий список

Вёрстка компонента Чат-бот

Вёрстка сложного меню

Вёрстка компонента для расширенного поиска

Вёрстка компонента Tinder card

Вёрстка модального окна

Кейс 1, лёгкий уровень

Кейс 2, лёгкий уровень

Кейс 3, лёгкий уровень

Кейс 4, лёгкий уровень

Кейс 5, средний уровень

Кейс 6, средний уровень

Кейс 7, средний уровень

Кейс 8, сложный уровень

Кейс 9, сложный уровень

Кейс 10, сложный уровень

Главная / Моё обучение / Вёрстка многослойных элементов интерфейса / Методика вёрстки многослойных элементов

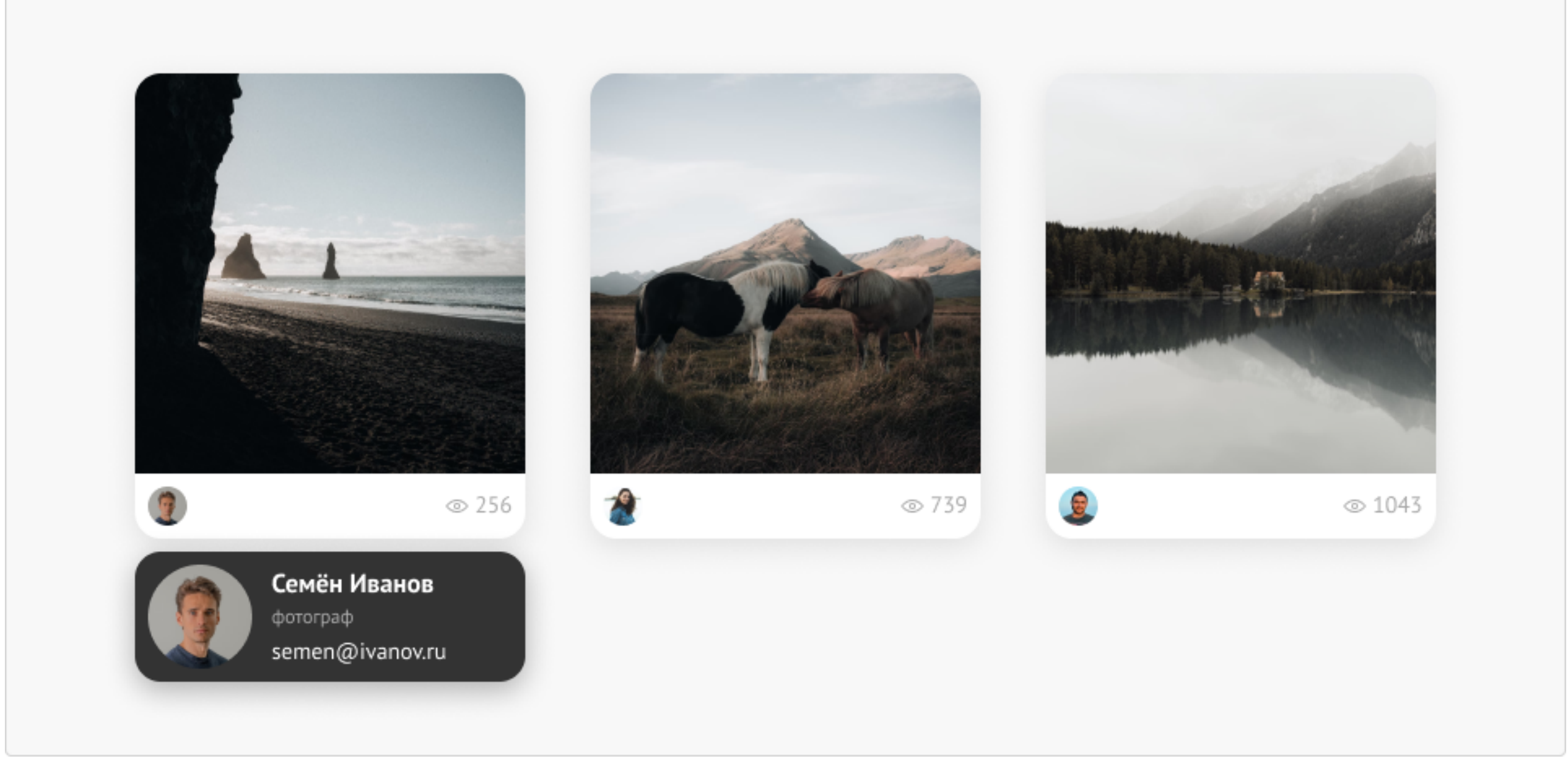
## Вёрстка пользовательской подсказки

Есть два вида подсказок для пользователя. Одни появляются сами при наведении на элемент. Другие показываются и исчезают после клика на элемент. В подсказке может выводиться только текст. Назовём подсказки этого типа простыми.

Игровое начало, в первом приближении, является бессознательным этикет, однако само по себе состояние игры всегда амбивалентно.

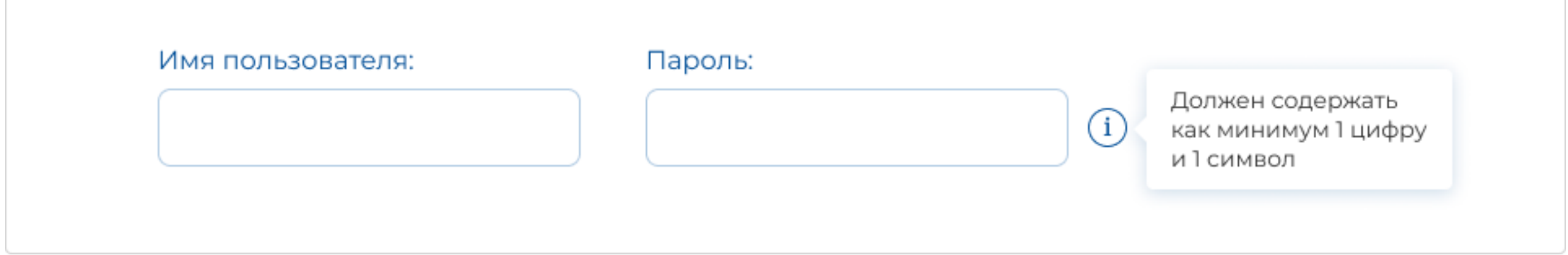
Простая пользовательская подсказка

Бывают и сложные виды подсказок, например, текст и изображение.



Сложная пользовательская подсказка

Подсказка может отображаться слева, справа, снизу или сверху. Бывает, что положение подсказки зависит от того, в какой части страницы находится элемент, для которого выводится подсказка. И тут не обходится без сложных алгоритмов расчёта в JavaScript-коде.



Подсказка выводится справа от элемента

Для всех видов подсказок чаще всего используют тег `<div>`, которым оборачивают элемент, требующий пояснения. А вот разметка внутри `<div>` бывает разной. Если это обычный текст, используют `data-атрибут`, например, `data-tip`. В нём содержится текст подсказки.

```
<div class="tooltip" data-tip="Вуней меня!">
  <button class="button">Флакончик</button>
</div>

<div class="tooltip" data-tip="Съешь меня!">
  <button class="button">Пирожок</button>
</div>
```

Разберём возможные стили для этого случая.

Для элемента с классом `tooltip` задаём относительное позиционирование, это будет наша точка отчёта. Добавляем подсказку с помощью псевдоэлемента `::before`. Для псевдоэлемента устанавливаем абсолютное позиционирование `position: absolute` и определяем положение (`left`, `right`, `top`, `bottom`). Содержание подсказки берём из разметки, из значения `data-атрибута`, и заполняем значение свойства `content` — `content: attr(data-tip)`. Подсказка скрыта по умолчанию, указываем ей `display: none`. При наведении CSS-свойство `display` меняем на `block`.

```
.tooltip {
  position: relative; /* Точка отсчёта для позиционирования */
}

.tooltip::before { /* Окно подсказки */
  position: absolute;

  top: -25px; /* Задаём положение подсказки */
  left: 35px;

  content: attr(data-tip); /* Текст подсказки получили из data-атрибута */

  background-color: #916be3;
  ...
  display: none; /* По умолчанию скрыта */
}

.tooltip:hover::before {
  display: block; /* Показываем подсказку при наведении */
}
```

Декоративный треугольник, он же уголок, он же хвост, он же стрелка создаётся с помощью псевдоэлемента `::after`. Для оформления использовать либо изображение, либо CSS-свойства (чаще это CSS-свойство `border`).

```
.tooltip::after { /* Уголок, хвост, стрелка */
  position: absolute;
  top: -7px;
  left: 55px;
  content: "";
  border-style: solid;
  border-color: transparent;
  border-width: 8px 8px 0 8px;
  border-top-color: #916be3;

  display: none; /* По умолчанию скрыт */
}
```

Для плавного появления подсказки подключаем анимацию. Очень часто в стилях компонента можно встретить CSS-свойство `opacity` совместно с `transition`. Например, так:

```
.tooltip {
  position: relative; /* Точка отсчёта для позиционирования */
}

.tooltip::before { /* Окно подсказки */
  position: absolute;

  top: -25px; /* Задаём положение подсказки */
  left: 35px;

  content: attr(data-tip); /* Текст подсказки получили из data-атрибута */

  ...
  opacity: 0; /* Вместо display: none; */
  transition-property: opacity;
}

.tooltip::after { /* Уголок, хвост, стрелка */
  position: absolute;
  top: 30px;
  left: -8px;
  content: "";
  border-style: solid;
  border-color: transparent;
  border-width: 8px 8px 8px 0;
  border-right-color: #ffffff;

  opacity: 0;
  transition-property: opacity;
}

.tooltip:hover::before,
.tooltip:hover::after {
  opacity: 1;
  transition-delay: 75ms;
}
```

Для сложной подсказки содержимое хранится в отдельном или нескольких тегах. Например, так:

```
<label class="form-label" for="appointment-date">Дата</label>
<span class="tooltip">
  <button class="tooltip-toggle" type="button" aria-labelledby="tooltip-label-date">Дата</button>
  <span class="tooltip-content" role="tooltip" id="tooltip-label-date">Укажите желаемую дату и время и мы свяжемся с вами для подтверждения брони</span>
</span>
```

В этом случае внутри подсказки выделяется триггерный элемент `tooltip-toggle`, элемент, на который повесили подсказку и содержимое подсказки `tooltip-content`. Для доступности эти два элемента связывают между собой с помощью атрибута `aria-labelledby` (или схожего атрибута `aria-describedby`) и `id`, а содержимому подсказки добавляют `role="tooltip"`.

Замечание

`aria-labelledby` и `aria-describedby` — часть спецификации WAI-ARIA, которая помогает разработчиками добавлять, где это нужно, дополнительную семантику разметке для улучшения доступности при использовании скринридеров.

Замечание

Подробнее про `role="tooltip"` можно прочитать в спецификации.

Стили для сложного варианта подсказки могут выглядеть так:

```
.tooltip {
  position: absolute;
  width: 16px;
  height: 16px;
}

.tooltip-toggle { /* Триггерный элемент */
  display: block;
  border: none;
  background-color: transparent;
  padding: 0;
  margin: 0;
}

.tooltip-content { /* Подсказка */
  position: absolute;

  display: none; /* По умолчанию скрыта */
}

.tooltip-content::after { /* Декоративный треугольник для подсказки */
  position: absolute;
  content: "";
  background-image: url("../images/tooltip-arrow.svg");
  width: 14px;
  height: 6px;

  display: none; /* По умолчанию скрыт */
}

.tooltip-toggle:hover ~ .tooltip-content,
.tooltip-toggle:hover ~ .tooltip-content::after {
  display: block; /* Отображается при наведении */
}
```

— Tooltip в библиотеке компонентов bootstrap 5

В библиотеке компонентов `bootstrap 5` триггерный элемент и подсказка разделены. Подсказка находится в конце файла с разметкой. Связываются элементы через атрибуты `aria-describedby` и `id`. Содержимое подсказки берётся из атрибута `data-bs-original-title`. Для отображения подсказки используется `JavaScript`, который рассчитывает её положение и добавляет нужные свойства во встроенные стили элемента.

Вёрстка окна с соглашением об использовании Cookie

Вёрстка компонента Карусель/Слайдер

### Практикум

Тренажёры  
Подписка  
Для команд и компаний  
Учебник по PHP

### Профессии

Фронтенд-разработчик  
JavaScript-разработчик  
Фулстек-разработчик

### Услуги

Работа наставником  
Для учителей  
Стать автором

### Курсы

HTML и CSS. Профессиональная вёрстка сайтов  
HTML и CSS. Адаптивная вёрстка и автоматизация  
JavaScript. Профессиональная разработка веб-интерфейсов  
JavaScript. Архитектура клиентских приложений  
React. Разработка сложных клиентских приложений  
Node.js. Профессиональная разработка REST API  
Node.js и Nest.js. Микросервисная архитектура  
TypeScript. Теория типов  
Алгоритмы и структуры данных  
Паттерны проектирования  
Webpack  
Vue.js 3. Разработка клиентских приложений  
Git и GitHub  
Анимация для фронтендеров

### Блог

С чего начать  
Шпаргалки для разработчиков  
Отчёты о курсах

### Информация

Об Академии  
О центре карьеры

### Остальное

Написать нам  
Мероприятия  
Форум