

Back to Basics

Как подружить верстку с логикой?

Front-End

...

Что, если я скажу тебе ...

Front-end разработка

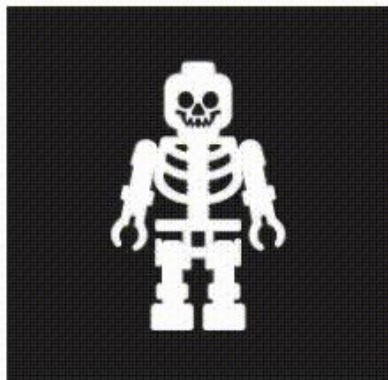
— это создание **клиентской** части сайта. Front-end разработчик занимается версткой шаблона сайта и созданием пользовательского **интерфейса**.

- *Интернеты*



Из чего состоит клиентская часть сайта?

HTML
structure



CSS
presentation/appearance



JavaScript
dynamism/action



**Почему фронты
открещиваются от верстки?**

Основные причины



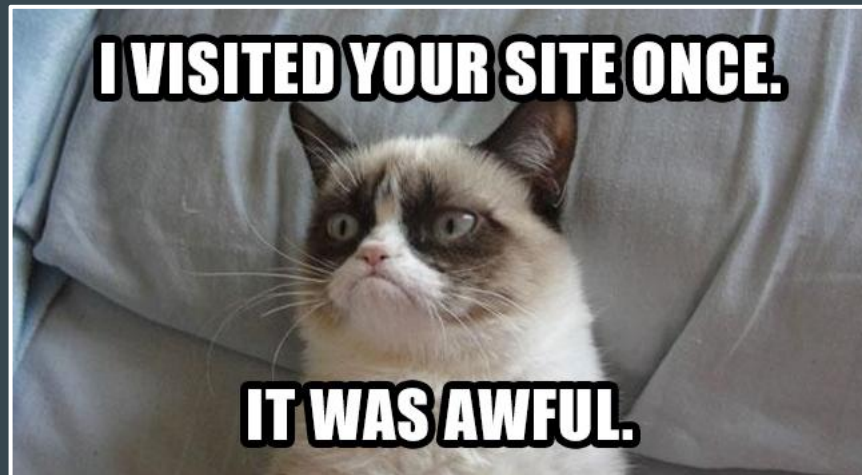
- Пришли во фронтенд из мира бэкенда\десктопа\улицы
- Считают, что Bootstrap'а достаточно
- Верстка – это вообще изи
- Ленивые ~~мудак~~и чудаки
- Редко разрабатывают сложные интерфейсы, работая преимущественно с логикой на клиенте

Уметь в верстку - это не больно

...

Чем вредно НЕ уметь в верстку

Ваша логика может быть бесконечно прекрасна, но какой в этом толк, если пользователь не может нажать на кнопку?



Чем полезно уметь в верстку

1. Повышается читаемость кода.
2. Ваши интерфейсы становятся функциональными, доступными, производительными и с плавной анимацией.
3. Дизайнеры становятся вашими друзьями.
4. SEO начинает приносить пользу.



JS - для логики

...

верстка для всего остального

!JS

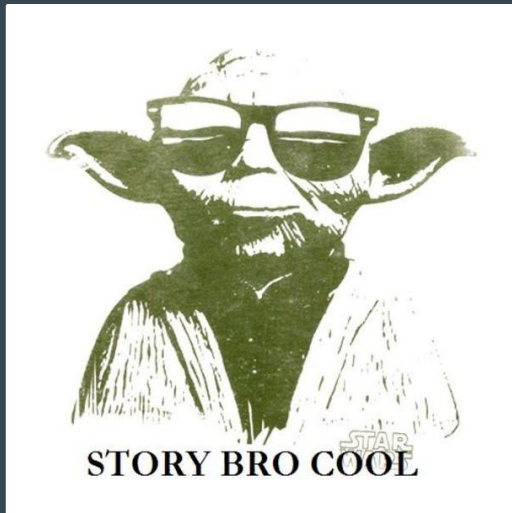
*Все, что вы можете сделать
без использования JS'а
должно быть сделано БЕЗ
использования JS'а.*



HTML и CSS сегодня

HTML5

- ★ Больше семантики
- ★ Больше функционала
- ★ Больше API
- ★ Регулярные выражения
- ★ Меньше рамок и условностей
- ★ IE8 мертв
- ★ FLASH мертв



CSS

- ★ Анимация
- ★ Плавные переходы
- ★ Градиенты
- ★ Медиа-выражения
- ★ Больше селекторов
- ★ Вычисляемые значения
- ★ Новые единицы
- ★ CSS переменные

Где пруфы, Билли?
...

НЕВНЯТНЫЙ КОНТЕНТ

```
<div class="card">
  <div class="card-header">
    <div class="card-title">{{ card.title }}</div>
    <div class="card-time">{{ card.time }}</div>
  </div>
  <div class="card-body">
    
    <span>{{ card.img.caption }}</span>
  </div>
  <div class="card-footer">
    <a :href="card.link.url" class="card-link">{{ card.link.text }}</a>
  </div>
</div>
```

ВНЯТНЫЙ КОНТЕНТ

```
<article class="card">
  <header class="card-header">
    <h3 class="card-title">{{ card.title }}</h3>
    <time class="card-time" :datetime="card.time">{{ card.time }}</time>
  </header>
  <div class="card-body">
    <figure class="card-thumb">
      
      <figcaption>{{ card.img.caption }}</figcaption>
    </figure>
    
    <p>{{ card.text }}</p>
  </div>
  <footer class="card-footer">
    <a :href="card.link.url" class="card-link">{{ card.link.text }}</a>
  </footer>
</article>
```

CSS animation VS Built-in transitions

```
<div id="list-demo">
  <button v-on:click="add">Добавить</button>
  <button v-on:click="remove">Удалить</button>
  <transition-group name="list" tag="p">
    <span v-for="item in items" v-bind:key="item" class="list-item">
      {{ item }}
    </span>
  </transition-group>
</div>
```


CSS animation VS Built-in transitions

```
.list-item {  
  display: inline-block;  
  margin-right: 10px;  
}  
.list-enter-active, .list-leave-active {  
  transition: all 1s;  
}  
.list-enter, .list-leave-to {  
  opacity: 0;  
  transform: translateY(30px);  
}
```

CSS animation VS Built-in transitions

```
var list = [...data];

list.forEach((item, index) => {
  if (index === arr.length - 1) {
    // сделать круто
  }
});
```

CSS animation VS Built-in transitions

```
.is-new-item {  
  position: relative;  
  animation: fadeIn 1s ease-out backwards;  
}  
  
@keyframes fadeIn {  
  0% {  
    opacity: 0;  
    bottom: -50px;  
  }  
  100% {  
    opacity: 1;  
    bottom: 0;  
  }  
}
```

JS vs CSS

```
<ul class="todo-list">
  <li class="todo-item done">Выпить кофе</li>
  <li class="todo-item done">Подвигать таски</li>
  <li class="todo-item done">Выпить кофе</li>
  <li class="todo-item done">Посмотреть мемасики</li>
  <li class="todo-item">Поиграть в теннис</li>
  <li class="todo-item">Пообедать</li>
  <li class="todo-item">Выпить кофе</li>
  <li class="todo-item">Написать пару строк кода</li>
  <li class="todo-item">Пойти домой</li>
  <li class="todo-item">Выпить пива</li>
  <li class="todo-item">Посмотреть мемасики до 3:00</li>
  <li class="todo-item">Очнуться в 12:00 и ползти на бровях в офис</li>
</ul>
```

JS vs CSS

```
/* JS */
document.querySelectorAll('.todo-item').forEach(item => {
  if (item.classList.contains('done')) {
    item.classList.add('hide');
  }
})

/* CSS */
.hide {
  display: none;
}
```


JS vs CSS

```
/* JS */
document.querySelector('.todo-list').classList.toggle('hide-done');

/* CSS */
.hide-done .done {
  display: none;
}
```

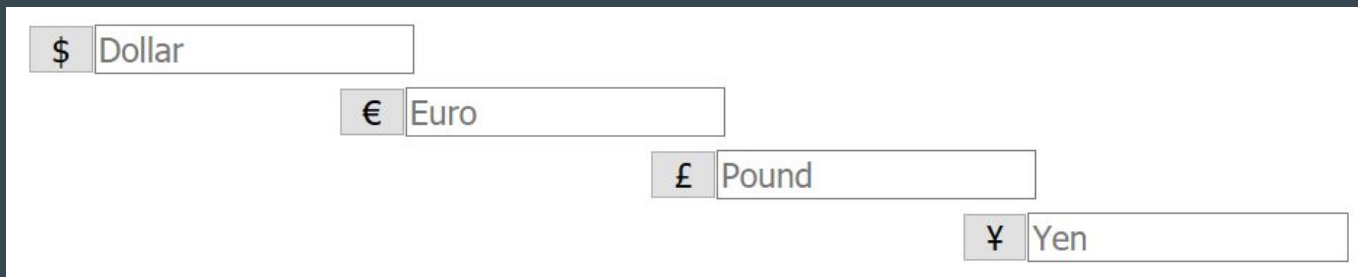
Кроссбраузерная “Утка”

Chrome



A screenshot of a web form in the Chrome browser. The form contains four currency selection controls arranged diagonally from top-left to bottom-right. Each control consists of a small square icon button followed by a text input field. The controls are: Dollar (with a '\$' icon), Euro (with a '€' icon), Pound (with a '£' icon), and Yen (with a '¥' icon). The text 'Dollar', 'Euro', 'Pound', and 'Yen' is visible in the input fields.

Firefox



A screenshot of the same web form in the Firefox browser. The layout and content are identical to the Chrome screenshot, showing four currency selection controls: Dollar (\$), Euro (€), Pound (£), and Yen (¥). The text 'Dollar', 'Euro', 'Pound', and 'Yen' is visible in the input fields.

