



**20.35**  
УНИВЕРСИТЕТ



Минцифры  
России



Fullstack-разработка

Vue

Атрибуты

## Цели урока:

1. Найдём замену `classList`
2. Научимся менять стили

# Разбор ДЗ

## Выбор языка

# Практика

## Галерея строк

# Теория

## Атрибуты



## Как сделать галерею с изображениями?

Назад



Вперёд

# img

Чтобы менять изображения, давайте разберёмся, как **менять атрибуты**.

```
<template>  
    
</template>
```



# data

У каждого компонента есть **данные**.  
 Это **объект** со свойствами для **шаблона**.

```

<script>
export default {
  data() {
    return {
      username: 'Никита'
    }
  }
}
</script>
  
```

```

<template>
  <h1>
    Меня зовут {{ username }}
  </h1>
</template>
  
```

# img

Сохранить путь до изображения в **data** и попробуем подставить в шаблон...

```

<script>
export default {
  data() {
    return {
      image: 'src/assets/plant.png'
    }
  }
}
</script>

```



```

<template>
  
</template>

```

# Директива v-bind

Для подстановки в шаблон атрибута используется директива `v-bind`.

```
<template>  
    
</template>
```



Или его более короткая версия...

```
<template>  
    
</template>
```

# Директива v-bind

Таким образом можно подставить любое **data-свойство** в любой **атрибут**.

```
data() {  
  return {  
    image: 'src/assets/plant.png',  
    imageWidth: 500,  
    imageClass: 'img img-fluid'  
  }  
}
```

```
<template>  
    
</template>
```

Дальше – дело техники!

Назад



Вперёд

# Практика

## Пароль

# Теория

## Классы



Как бы мы решали такую задачу  
на чистом JS?

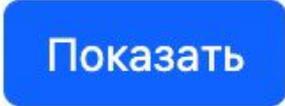
**Кнопка тогл**

Показать

## Работа с DOM

```
let button = document.querySelector('.toggle');  
let node = document.querySelector('.secret');  
  
button.addEventListener(`click`, function () {  
  node.classList.toggle('hidden');  
})
```

### Кнопка тогл



Показать



Вместо работы с DOM в Vue работает принцип **реактивности**:

Изменение **данных** приводит к перерисовке **компонента**.

Давайте решим эту задачу через **data, v-bind и @click**.

# Изменение класса

Атрибут `:class` можно задать в виде объекта со значениями `true` или `false`.

```
<div :class="{  
  'square': true,  
  'green': false,  
  'border': true  
}">  
  ...  
</div>
```

Осторожно,  
хитрый синтаксис!

```
<div class="square border">  
  ...  
</div>
```

# Изменение класса

Вернёмся к задаче...

По принципу реактивности ставим добавление класса в зависимость от data-свойства.

```
<div :class="{  
  'hidden': isHidden,  
}">  
  ...  
</div>
```

```
data() {  
  return {  
    isHidden: true  
  }  
}
```

# Изменение класса

Добавляем метод, который будет менять **data-атрибут** класса по **событию** клик.

```
methods: {  
  toggle() {  
    this.isHidden = !this.isHidden;  
  }  
}
```

```
<button @click="toggle">  
  Скрыть  
</button>
```

Готово!

*Останется лишь заменить  
текст кнопки :)*

**Кнопка тогл**

Показать

# Практика

## Редактор текста

# Теория

## Стили

Иногда классы не дают нам достаточно гибкости для изменений.

Тогда на помощь приходят **стили!**

**B**

/

T

-

+

Съешь ещё этих мягких французских  
булочек

# Изменение стиля

Атрибут `:style` можно задать в виде объекта с CSS-свойствами.

```
<div :style="{  
  'font-size': '20px',  
  'color': 'red'  
}">  
  ...  
</div>
```

# Изменение стиля

Конечно, эти **стили** можно сделать реактивными, привязав их к **data-свойствам!**

```
<div :style="{  
  'font-size': size + 'px',  
  'color': color  
}">  
  ...  
</div>
```

# Практика

## Редактор изображения

## Директива v-bind

1. Пишется через двоеточие  
`:src="imageSrc"`
2. Классы через объект  
`:class="{ 'isDark': true }"`
3. Стили через объект  
`:style="{  
 'background-color': 'black'  
}"`

## Цели урока:

- ✓ Нашли замену `classList`
- ✓ Научились менять стили

## Домашнее задание