Поиск решений по теме Class

Сам себе задаю задачу:

1. Найти div с текстом в html и статично создать через Class: ЦВЕТ
2. Для div (в html) создать фигуру через Class: ЦВЕТ, РАЗМЕР, РАМКУ
3. По клику на кнопку: тег должен стать КРАСНЫМ или ЖЁЛТЫМ
4. По клику в checkbox: 1й тег станет КРАСНЫМ (клик в 1й чекбокс) или 2-й тег станет ЖЁЛТЫМ (клик на 2й чекбокс); если чекнуты оба чекбокса – оба тега видны со своим цветом

|  |  |
| --- | --- |
| Код HTML |  |
| < div class="boxClass" id="boxId"> ЭтоКуб </div>  КНОПКА  <script src="app.js"> </script> |  |

Содержание (пишу код через Class, extends, new):

|  |  |
| --- | --- |
| **Решение задачи №1. Найти тег и сделать в нем красный фон** |  |
| **1.1. Искал как в КЛАССе создать ссылку на тег div class="boxClass"** |  |
| **1.2. Искал как через class Two изменять цвет фона** |  |
| **1.3. Решение задания, когда в экземпляре класса догадался НЕ ИСПОЛЬЗОВАТЬ присвоение значений через объект { }** |  |
| **1.4. Решение задания№1:**  **Для каждого значения и метода - свой Class и свой экземпляр**  **( 2\_class\_oop.js )** |  |
| **1.5 Вопрос в чат: почему все срабатывает без параметра option** |  |
|  |  |
| **Решение задачи №2.** **Для div создать фигуру через Class: ЦВЕТ, РАЗМЕР, РАМКУ** **Сложности решения: extends только с родительским this; и +«px»** |  |
|  |  |
| **Решение задачи №3 ( 3\_class\_oop.js ).**  **Клик на кнопке: тег должен стать КРАСНЫМ >> ЖЁЛТЫМ**  **Сложности решения: встроить слушатель кнопки внутрь Class, использовать extends с удалением родителя** |  |
|  |  |
| **Решение задачи №4. ( 4\_class\_oop.js ).**  **По клику в checkbox: тег стать КРАСНЫМ (1й чекбокс) или ЖЁЛТЫМ (2й чекбокс); если чекнуты оба чекбокса - ОРАНЖЕВЫЙ** |  |
|  |  |
|  |  |
|  |  |

# Решение задачи №1. Найти тег и сделать в нем красный фон

**1.1. Искал как в КЛАССе создать ссылку на тег div class="boxClass":**

*(около 3часов, нашел случайно)*

|  |  |
| --- | --- |
| class One{  constructor (select ){  this.el = document.querySelector (**select.el**);  }  }  class Two extends One{  //ТоДо: задать ширина, высота, цвет  }  let showBox = new Two( {  el: **'.boxClass'**  } ) | Исходный КЛАСС – шаблон  Как я понимаю: select  В этом КЛАССЕ потом задам присвоение ширины, высоты, поля  ЭКЗЕМПЛЯР КЛАССА  Назначил el: **'.boxClass'** (чтоб находился тег через querySelector ( ) ) |

важно для кода, что значения тега задано в **showBox** внутри объекта

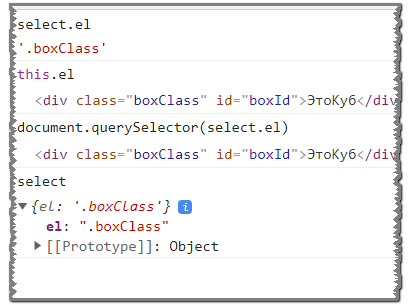
На картинке:

Что покажет **select.el**

Что покажет **this.el**

Что покажет **document.querySelector (select.el);**

Что покажет **select**

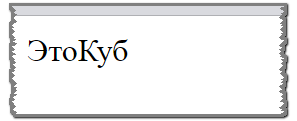


**1.2. Искал как через** class Two **изменять цвет фона**

(важно для кода, что значения тега и цвета заданы в **showBox** внутри объекта!)

|  |  |
| --- | --- |
|  |  |
| **class One{**  **constructor (select ){**  **this.el = document.querySelector (select.el);**  **}**  **}**  **class Two extends One{**    **constructor(select,** *option***){**  **super (select );**  //this.option = option; //red // не участник?  **this.el.style.background = select.color;**  **}**  **}**  **let showBox = new Two( {**  **el: '.boxClass' ,**  **color: 'red'**  **}**  NB! Это объект, поэтому option так просто не видит **)** | Исходный **class One**  ( найти div со словом «ЭтоКуб» ):  //созд.ссылку на div class="boxClass"  **class Two** расширяет **class One**  // задан ЦВЕТ для div class="boxClass"  внутри (*не понял почему*) активен только параметр **( select )**.  *option* не активен никак  (возможно из-за того, что в showBox объявления идут в объекте!).  ЭКЗЕМПЛЯР от КЛАССА **Two**  В переменной **showBox** уже указаны свойства для подстановки в классы:  **el: '.boxClass'** улетает в **class One**  **color: 'red'** улетает в **class Two** |

До отработки скрипта с классами:



После отработки скрипта с классами:

(слова выделены красным фоном)



1.3. Решение задания №1:

когда в экземпляре класса догадался НЕ ИСПОЛЬЗОВАТЬ присвоение значений через объект { }

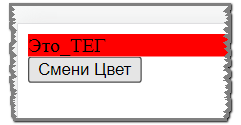
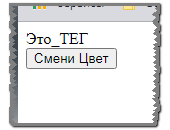
**HTML:**

<div class="boxClass" id="boxId">Это\_ТЕГ</div>

<button class="btnClass">Смени Цвет</button>

**JS:**

|  |  |
| --- | --- |
| **class BoxClass {**  **constructor (tag, color){**  **this.tag = document.querySelector (tag) ;**  **this.color = color ;**  **this.reColor ( );**  **}**  **recolor ( ) {**  **this.tag.style.background = this.color**  **}**  **}**  **let boxClass = new BoxClass (**  **'.boxClass',**  **'red'**  **)** | //для тега class="boxClass"  Найти тег  Обозначить колор  Метод смены цвета в теге  Сама ф-ция смены цвета  //Экзмепляр класса.  //Объяв.аргументы без использ-я объектов { } |



1.4. Решение задания №1:

Для каждого значения и метода - свой Class и свой экземпляр

**HTML:**

<div class="boxClass" id="boxId">Это\_ТЕГ</div>

<button class="btnClass">Смени Цвет</button>

<script src="**2\_class\_oop.js**"></script>

**JS:**

|  |  |
| --- | --- |
| **class BoxClass {**  **constructor (tag ){**  **this.tag = document.querySelector (tag) ;**  **}**  **}**  **let boxClass = new BoxClass (**  **'.boxClass'**  **)**  **class GetColor {**  **constructor (color){**  **this.color = color ;**  **this.reColor ( );**  **}**    **reColor( ){**  **boxClass.tag.style.background = this.color**  **}**  **}**  **let getcolor = new GetColor (**  **'red'**  **)** | //Класс для тега class="boxClass"  Нахожу тег с классом "boxClass"  //Экзмепляр class BoxClass.  //Объяв. аргумент без использ. объектов {}  //Класс перекрашивания фона в цвет:  //Метод: фон сделать красным  Обращение к **boxClass.tag**  //Экзмепляр class GetColor  Объявил цвет |

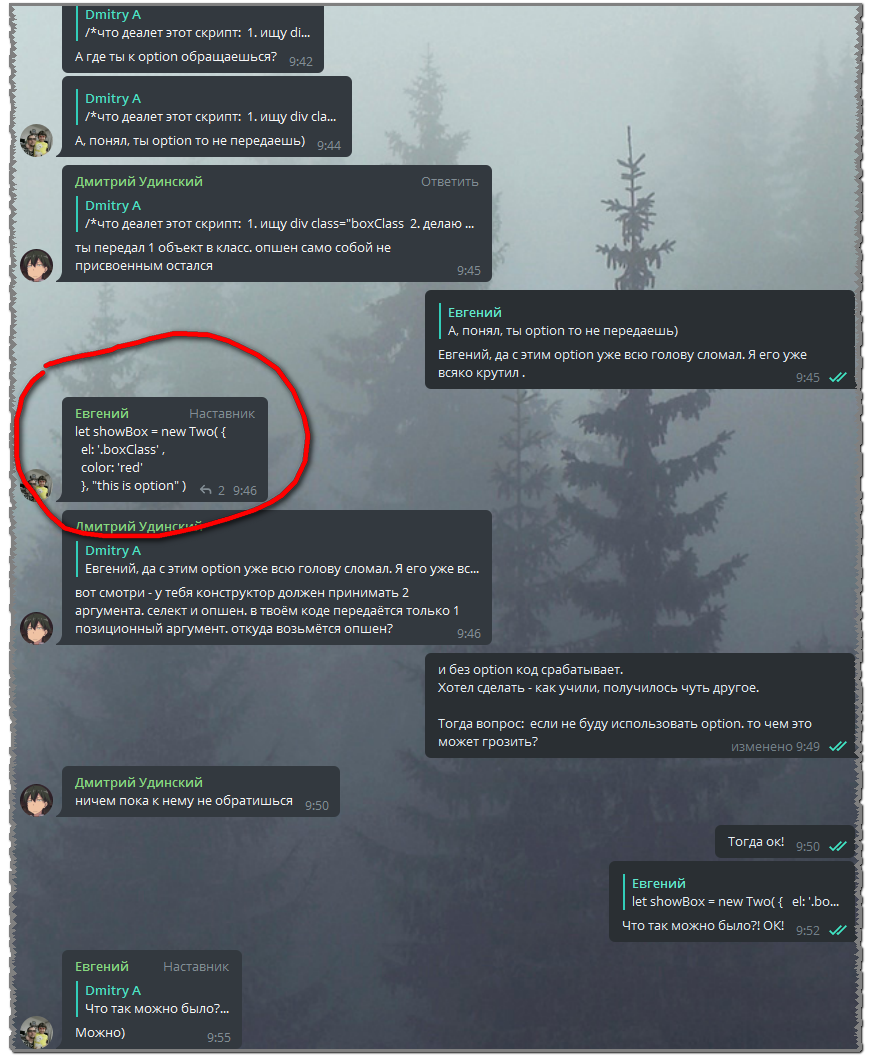
**1.5. Вопрос в чат: почему все срабатывает без параметра option**

Прошу пояснить – почему в классе Two – не видит назначенный параметр option :

``` option = undefined``` (?) То есть ```this.option``` не срабатывает нигде.

Но в то же время экземпляр этого класса showBox корректно срабатывает (делает бэкграунд красным цветом). Но делает он все это как-то удивительно – через

```select``` который был назначен в родительском class One.



Предложенный вариант:

**let showBox = new Two (**

**{ el: '.boxClass' , color: 'red' },** //объкет для тега и цвета

*"this is option…"***)** // след. Параметр

**Почему option не активен, мне пытались пояснить на примере:**

|  |  |
| --- | --- |
| НЕ указан 2-й параметр | Указан 2-й параметр |
| **let a = null;**  **let b = null;**  **function myFunc(select, option) {**  **a = select;**  **b = *option*;**  **}**  **myFunc ("this is a");** | **let a = null;**  **let b = null;**  **function myFunc(select, option) {**  **a = select;**  **b = option;**  **}**  **myFunc ("this is a", *"this is b"*);** |
| // select = "this is a"  // option = undefined  **// a**: "this is a"  **// b**:  undefined | // select = "this is a"  //option = "this is b"  **// a**: "this is a"  **// b**: "this is b" |

в конструкторе ты не передаёшь парамсы, а объявляешь каким образом эти парамсы конструируются при инициации конструктора.

если у тебя в конструкторе 2 аргумента и 2 парамсам присваиваются значения этих аргументов, то для того чтобы в значениях обоих этих парамсов что-то появилось, требуется при инициации экземпляра класса передать в конструктор ОБА аргумента.

# Решение задачи №2. Для div создать фигуру через Class:

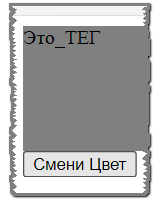
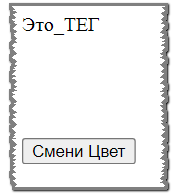
# ЦВЕТ, РАЗМЕР, РАМКУ

|  |  |
| --- | --- |
| class="boxClass"  class One{  constructor(select ){  this.el = document.querySelector ( select.el );  }  }    class Two extends One{  constructor(select, *option*){  super (select );  this.option = option; //undefined  this.el.style.background = select.color;  }  }  class Three extends One {  constructor (select){  super(select) ;  this.el.style.width = select.sizeW + 'px' ;  this.el.style.height = select.sizeH + 'px' ;  this.el.style.border = select.rendBorder;  }  }  let showBox = new Two ( {  el: '.boxClass' ,  color: 'red'  } )  let modifyBox = new Three ({  el: '.boxClass' ,  sizeW: 100,  sizeH: 100,  rendBorder: '2px solid black'    } ) | // **шаг №2.**  Class для создает ссылку на  div class="boxClass"  //**шаг №3**. Class Two для:  ЦВЕТ для div class="boxClass"  *// почему-то параметр option не активен: пояснили, что не указал его в let showBox (?) вторым параметром*  // **шаг №4.** Class Three для:  ШИРИНА ВЫСОТА ГРАНИЦА  // ширина, высота, границы  //**шаг №1.** Экзмепляр класса Two  Переменная showBox  Задаю значение для: ЦВЕТ  //**Шаг №5.** Экземпляр класса Three  Переменная modifyBox  Задаю значения для:  ШИРИНА ВЫСОТА ГРАНИЦА |

# Решение задачи №3.

# По КЛИКУ НА КНОПКУ: тег должен стать КРАСНЫМ

# От повторного клика тег становится СЕРЫМ, и так по кругу



**Без цвета красный серый**

|  |  |
| --- | --- |
|  |  |
| class Tags {  constructor (tag ){  this.tag = document.querySelector (tag) ;  }  }  let tags = new Tags (  '.boxClass' , //тег (tag)  ) | №2. для тега class="boxClass" :  №1. экз-р от class Tags: значение для tag |

Далее: расширяю class Tags

Но: смог избежать инициализации родительского class Tags

Расширяю (extends) class Tags

Но: смог избежать инициализации родительского class Tags; цель: уйти от повторного прописывания параметров родителей (уже и так указанных в исходном class Tags)

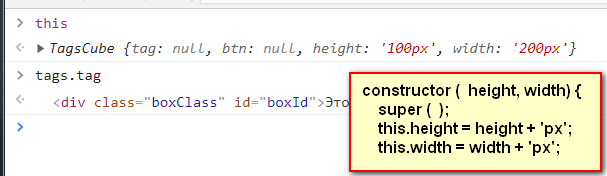
|  |  |
| --- | --- |
| class TagsCube extends Tags {  constructor ( sizeH, sizeW ) {  super ( ); // *родителям = null*  this.sizeH = sizeH + 'px';  this.sizeW = sizeW + 'px'; | №4. Проба:  Расширяю class Tags.  Для чего:  ЗАДАТЬ ФОРМУ (H x W) = 100х200рх  Убрал из constructor и super родительские: tag, btn (!)  Чтоб в let tagsCube указать только «местные» height, width  В консоли для this покажет:  TagsCube {**tag: null, btn: null**, height: '100px', width: '200px'} |

Далее – метод по заданию размеров (высота и ширина тега)

|  |  |
| --- | --- |
| this.reSize ( )  }  reSize ( ){  tags.tag.style.height = this.sizeH ;  tags.tag.style.width = this.sizeW  }  }  let tagsCube = new TagsCube (  100,  200  ) | метод по заданию размеров (высота и ширина тега)  сама функция (тег.стиль.высота = …; тег.стиль.ширина = … ;)  №3. Экзем-р класса TagsCube: указал размеры фигуры  (слово «РХ» присвоены в class’ е TagsCube) |

**Если при расширении class Tags не указать родителей, то будет null**

**Но решается - через обращение напрямую к родительскому tags.tag**



См. далее:

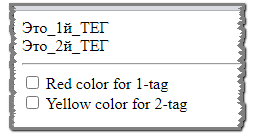
создаю **КЛАСС ДЛЯ КНОПКИ**, по клику на которую будет смена цветов тега по кругу ( красный - >серый)

|  |  |
| --- | --- |
| class BtnClick {  constructor (btn,colorRed,colorGrey){  this.btn = document.querySelector(btn) ;  this.colorRed = colorRed;  this.colorGrey = colorGrey ;  this.btn.addEventListener('click', this.listenClick ) ;  }  listenClick( ) {  *console.log (‘тест\_клика!’)*  if ( tags.tag.style.background == btnClick.colorRed) {  tags.tag.style.background = btnClick.colorGrey ;  }else {  tags.tag.style.background = btnClick.colorRed;  } } }  let btnClick = new BtnClick (  '.btnClass' , //кнопка (btn)  'red' ,  'grey'  ) | Класс для КНОПКИ '.btnClass'  Ссылка на селектор кнопки,  Инициализация двух цветов  addEventListener: **слушаю кнопку** '.btnClass'  Альтернатива для слушателя:  document.querySelector ('.btnClass'). addEventListener ('click', this.listenClick )  Метод Смены ЦВЕТА фона  // *console.log:* :Мне подсказала: что клики кнопки работают по кругу!  Условие IF для смены цветов по клику на кнопку (х2): если фон == red, тогда сделай фон = grey.  Иначе: red  Экземпляр класса BtnClick  тег кнопки  красный  серый |

**Решение задачи №4. ( 4\_class\_oop.js ).**

От клика в 1й чекбокс - 1й тег станет КРАСНЫМ ; от клика во 2й чекбокс - 2-й тег станет ЖЁЛТЫМ.

Если чекнуты оба чекбокса – оба тега видны со своим цветом.



**html**

|  |  |
| --- | --- |
| <div class="boxClass" data-box="onebox">Это\_1й\_ТЕГ</div>  <div class="boxClass" data-box="twobox">Это\_2й\_ТЕГ</div>  <hr>  <input type="**checkbox**" id="rt" name="rTag" value="red" data-color="red">  <label for="rt">Red color for 1-tag</label>  <br>  <input type="**checkbox**" id="yt" name="yTag" value="ylw" data-color="yellow">  <label for="yt">Yellow color for 2-tag</label>  <script src="4\_class\_oop.js"> </script> |  |

Не решал