

DotScript

framework

Полное руководство по DotScript
Программирование на DHub IDE

1

Начало работы

1. Об framework , Зачем он нужен
2. Логотип DotScript
3. Как взаимодействовать

Создание уведомлений

4. Знакомство с функцией noth()
5. Работа с noth()
6. Проблемы функции

Создание элементов

7. Знакомство с функцией create()
8. Знакомство с функцией stylebox()
9. Знакомство с функцией deletbox()
10. Знакомство с функцией box()
11. Работа с функциями
12. Проблемы функций

Работа с LocalStorage

13.Теория JS

14.Знакомство с функцией save()

15.Знакомство с функцией open()

16.Работа с функциями

Работа с файлами

17.Знакомство с функцией filesave()

18.Работа с функциями

19.Проблемы функции

Работа с логикой

20.Знакомство с функцией correct()

21.Работа с функциями

22.Проблемы функций

Работа с DotScript в DHub

23.Где начать программировать

24.DotScript + JavaScript

24.Работа компилятора

Работа с DotScript Native

25. Что такое DotScript Native

Подключение к проекту

Инфо.центр

26. Информационный центр контакты

DotScript

Что такое DotScript

DotScript - это framework для облегчения работы и создания элементов на JavaScript. Данный framework создан и работает на языке JavaScript.

Он создан для создания плагинов для сервиса DHub.

Не путать !

Не путайте DotScript с Microsoft **.net** и **.Script**

Новый! но тупой

Пока что DotScript еще не совершенен и немного не логичен. Но мы планируем улучшить проект и добавлять новые функции для работы

Для чайников

Изучив данное руководство вы сможете создавать плагины для DHub , даже без должных знаний JS

DotScript

Логотип DotScript



Автор: DX Desing 28.09.2024

Взаимодействие

Для работы с DotScript используются уже готовые редакторы кода



Visual Studio Code



WebStorm



DHub

Пока uPanelCode не будет поддерживать DotScript.

DotScript уже встроен:



DHub IDE

Создание уведомлений

Что такое уведомления в DHub?

Noth - это центр сообщений действий пользователя или программы за определённое время в программе (рис. 1)

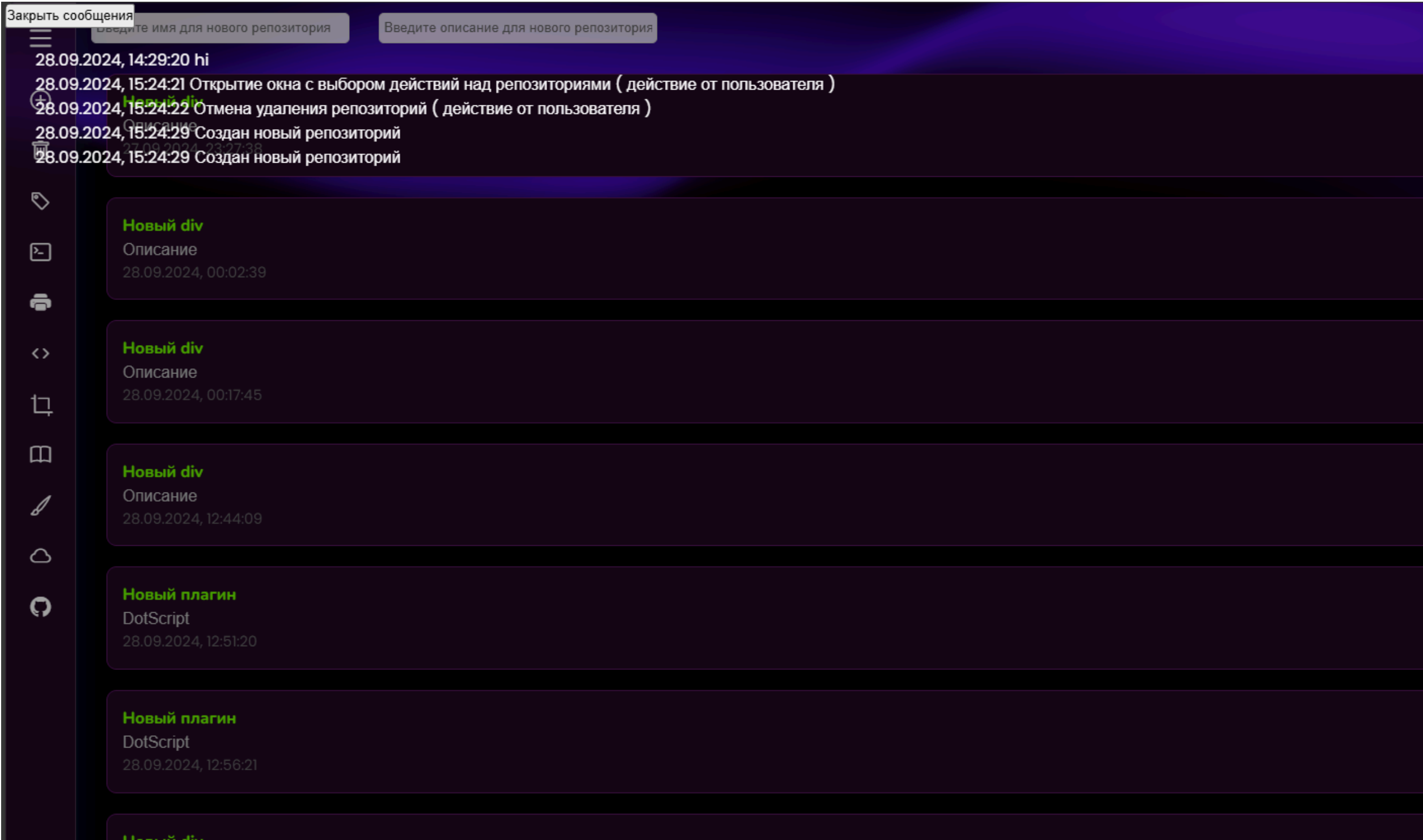


Рис.1 Центр сообщений DHub

Как работает noth?

Вот например: пользователь нажимает на кнопку “создать репозиторий” дальше создается новый репозиторий и при успешном создании этого репозитория будет отправлено сообщение в notf о том что создан новый репозиторий. (Самое простое определение)

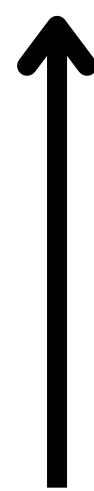
Создание уведомлений

Функция `noth()`

Что такое `noth()`

Это функция которая создает уведомление в `notf`

`noth("Hello")`



Текст сообщения

`noth("Hello" + x)`



В `noth()` можно передавать и переменные

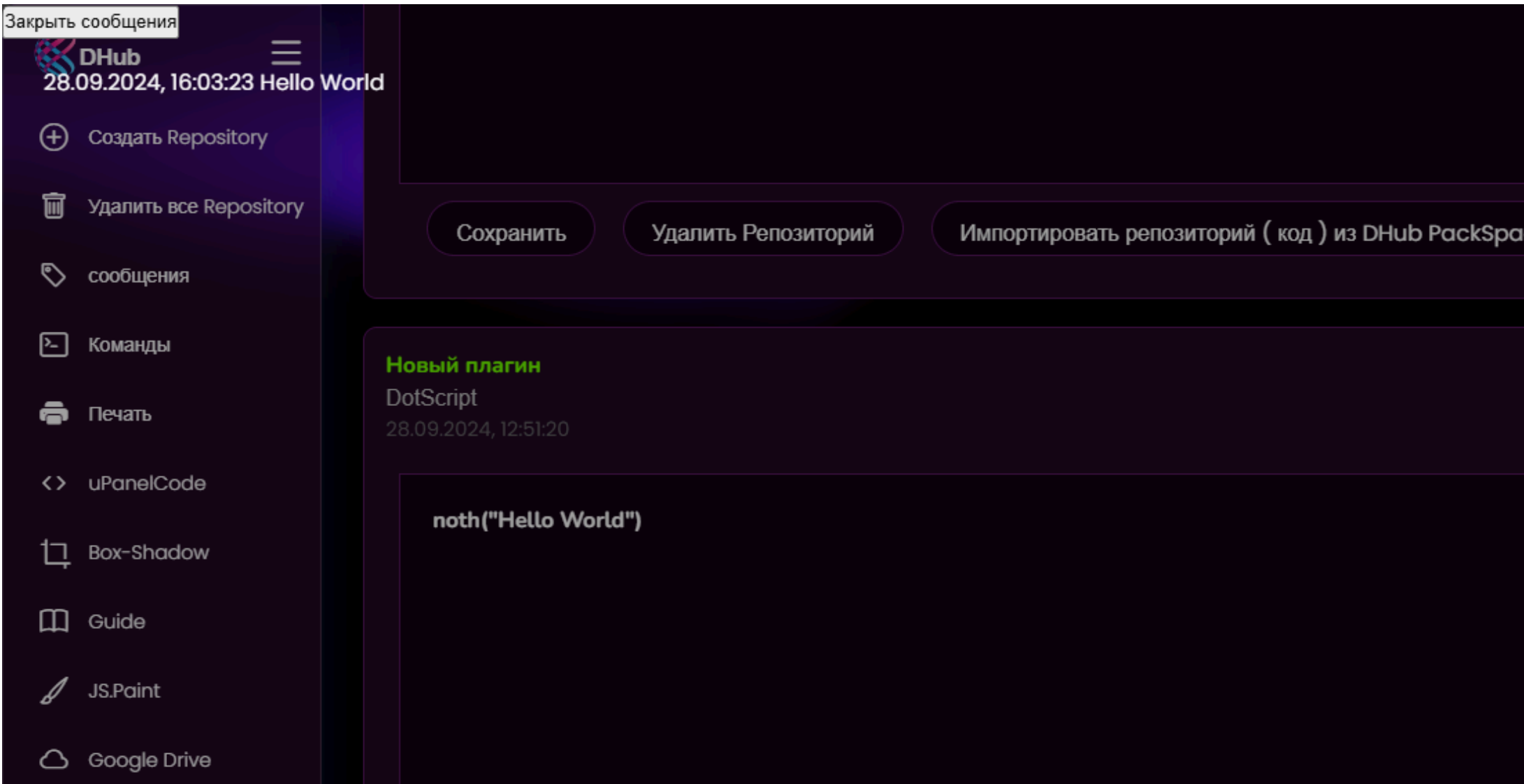
Создание уведомлений

Работа с noth()



Запуск скрипта в DHub IDE

Результат:



Создание уведомлений

Проблемы noth()

noth() он может только передать текст , передача color,fontfamily и другое не возможна. Также нельзя будет редактировать прошлое сообщение если вы уже запустили скрипт (рис.2)



Рис . 2

Нельзя изменить

Создание элементов

Функция create()

JS Теория

CreateElement - позволяет создать новый элемент, передав в параметре имя тега. После создания с элементом можно работать как с обычным элементом, а также его можно добавить на страницу методами prepend, append, appendChild, insertBefore или insertAdjacentElement.

Если записать результат работы createElement в переменную, то в этой переменной будет такой элемент, как будто бы мы получили его через querySelector или getElementById. Единственное отличие - наш элемент не будет размещен на странице. А так мы можем менять ему innerHTML, атрибуты, навешивать обработчики событий и в конце концов разместить его на странице.

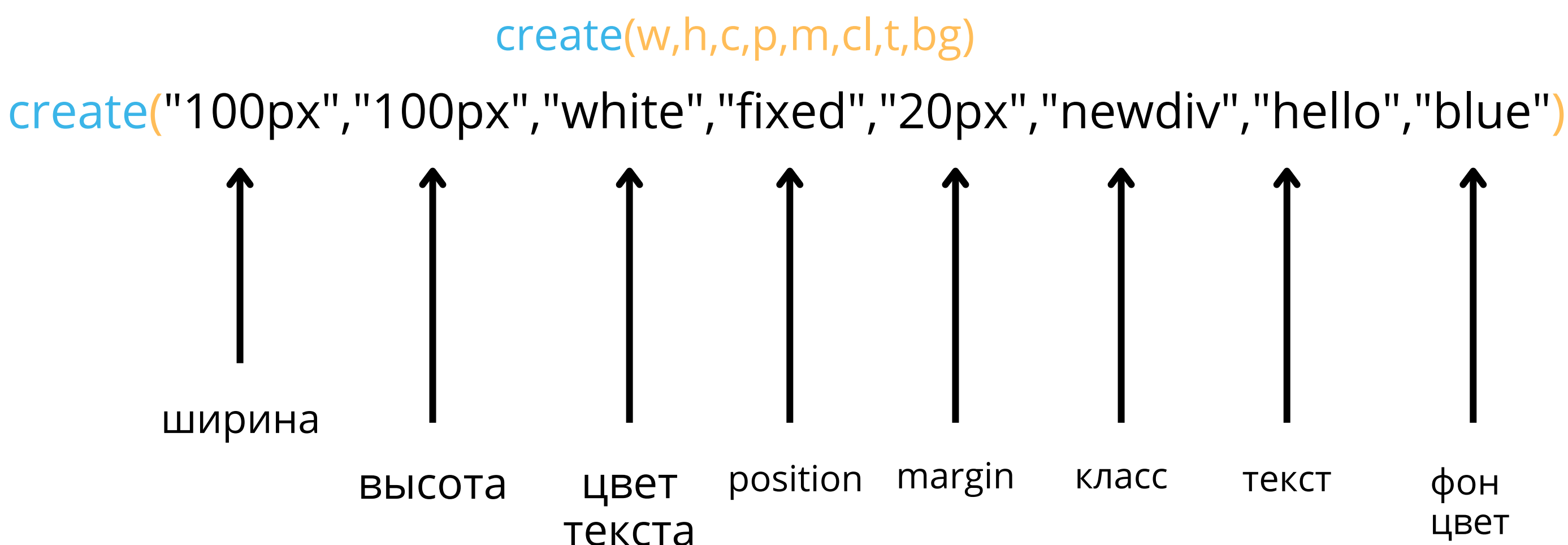
{ JavaScript }

Создание элементов

Функция create()

Что такое create() ?

Create - это функция которая создает новый элемент (div) на экране



Экономит место!

чтобы также создать элемент вам в js нужно будет писать такой код

```
let createneworc = document.createElement('div');
createneworc.style.width = w;
createneworc.style.height = h;
createneworc.style.backgroundColor = bg;
createneworc.style.margin = m;
createneworc.style.position = p;
createneworc.style.color = c;
createneworc.className = cl;
createneworc.textContent = t;
document.body.appendChild(createneworc);
```

а с DotScript

`create(w,h,c,p,m,cl,t,bg)`

Создание элементов

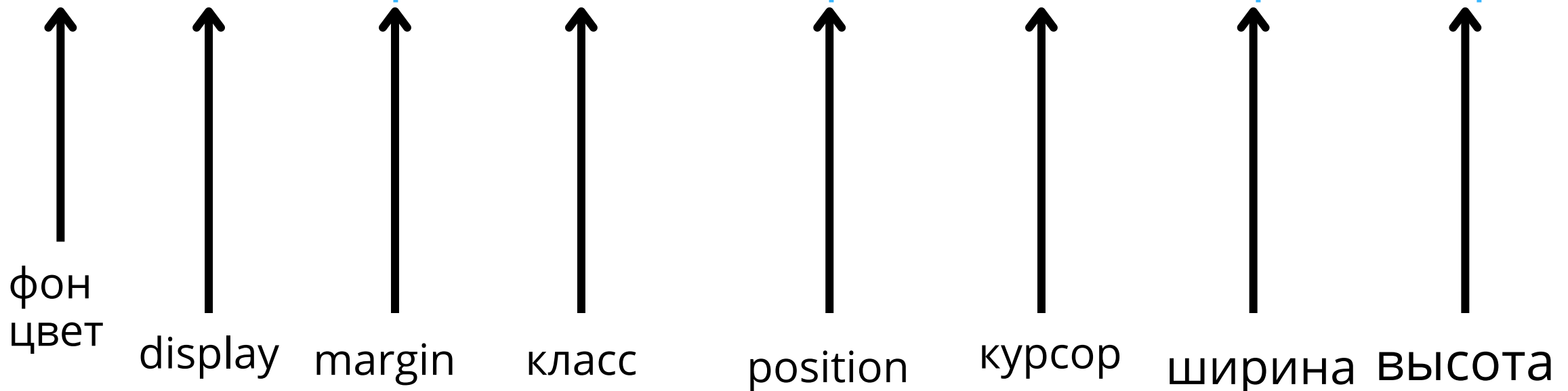
Функция `stylebox()`

Что такое `stylebox()` ?

`stylebox()` - это функция которая изменять стиль к выбранному элементу на экране

`stylebox(backw,displayw,marginw,id,pad,cursorw,widthw,heightw)`

`stylebox("red","flex","10px","newdiv","12px","move","500px","100px")`



Экономит место!

```
let stylediv = document.getElementById(id);
stylediv.style.width = widthw;
stylediv.style.height = heightw;
stylediv.style.backgroundColor = backw;
stylediv.style.margin = marginw;
stylediv.style.position = pad;
stylediv.style.display = displayw;
stylediv.cursor = cursorw;
```

а с DotScript

`stylebox(backw,displayw,marginw,id,pad,cursorw,widthw,heightw)`

Создание элементов

Функция `deletebox()`

Что такое `deletebox()` ?

`deletebox()` - это функция которая удаляет элемент на экране (не скрывает ! А удаляет)

`deletebox("newdiv")`

↑
класс

Я кубик с
классом
newdiv

+

`deletebox("newdiv")`

=

none

Создание элементов

Работа с функциями

Задача

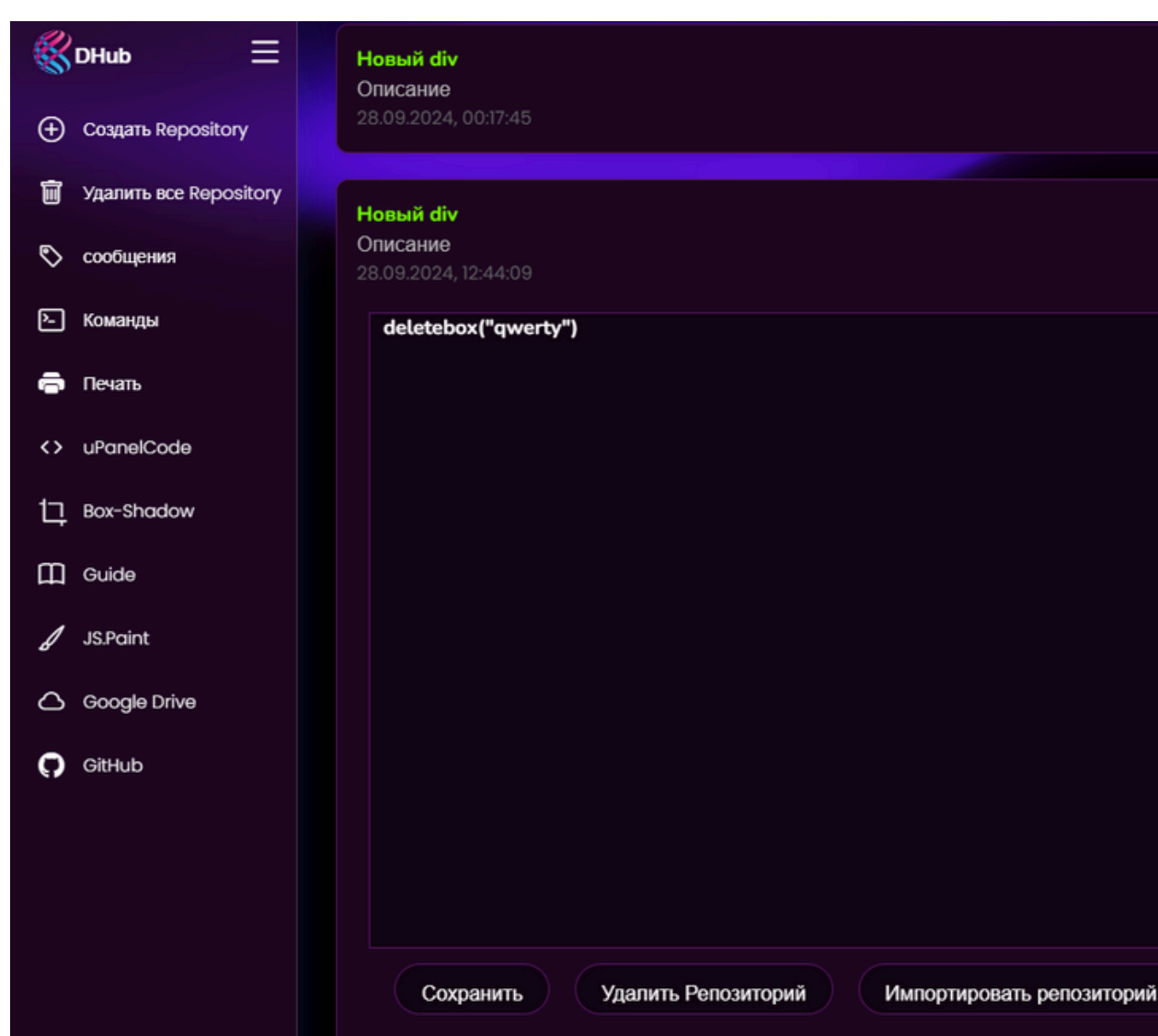
Создадим кубик 500x500 с красным фоном и с классом `qwerty` и белым текстом привет

1. Создаем куб через функцию `create()`

```
create("500px","500px","white","fixed","10px","qwerty","привет","red")
```



2. И удалим куб через функцию `deletebox("newdiv")`





Создание элементов

Проблемы

Все функции должны строго передавать свои параметры !

Пример:

 `create("500px","500px","white","fixed","10px","qwerty","привет","red")`

 `create("500px","500px","10px","qwerty","привет","red")`

Функция create должна строго передавать все свои **восемь** параметров , если меньше то будет **ошибка**.
Это правило касается всех функций DotScript

Также важна позиция параметров

`create(w,h,c,p,m,cl,t,bg)`

Если вы вместо параметра width напишите цвет или другое то будет ошибка

Это правило касается всех функций DotScript

Работа с LocalStorage

Теория JS

Это объект, хранящийся в window, который позволяет долговременно сохранять данные в браузере.

Работает как хранилище данных в формате ключ-значение — при сохранении данных мы указываем имя поля, в которое должны быть сохранены данные, и затем используем это имя для их получения.

Значения хранятся в виде строк. При попытке сохранения других типов данных, они будут приведены к строке. Например, если записать число, то при чтении нам вернётся число, записанное в строку.

Не имеет ограничений по времени хранения, может быть очищен пользователем вручную или браузером при переполнении автоматически (браузеры на основе движка WebKit, например Safari, очищают localStorage, если к нему не обращались в течение 7 дней).

Максимальный объем данных ограничен размером 5MB.

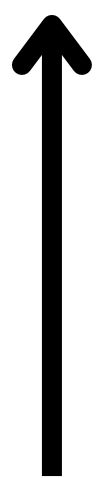
Работа с LocalStorage

функция save()

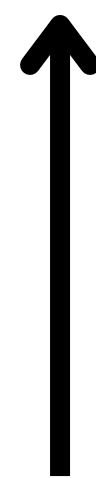
Что за save()?

save() - функция которая создает localStorage

save("newsave", "hello world")



Название
сохранения



Текст

Как она устроена ?

```
function save(name, textsave){  
    localStorage.setItem(name,  
    textsave);  
    alert("save complite!")  
}
```

Работа с LocalStorage

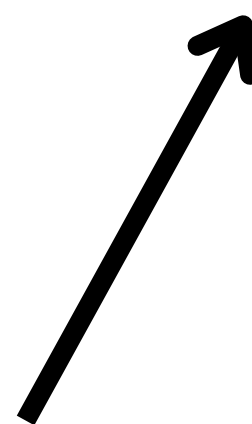
функция open()

Что за save()?

open() - функция которая показывает что хранится в LS по его ключу

`open("newsave")`

Название
сохранения



Для разработчиков плагинов

Текст из LS при вызове функции open() добавляется в переменную `localStoragevar`

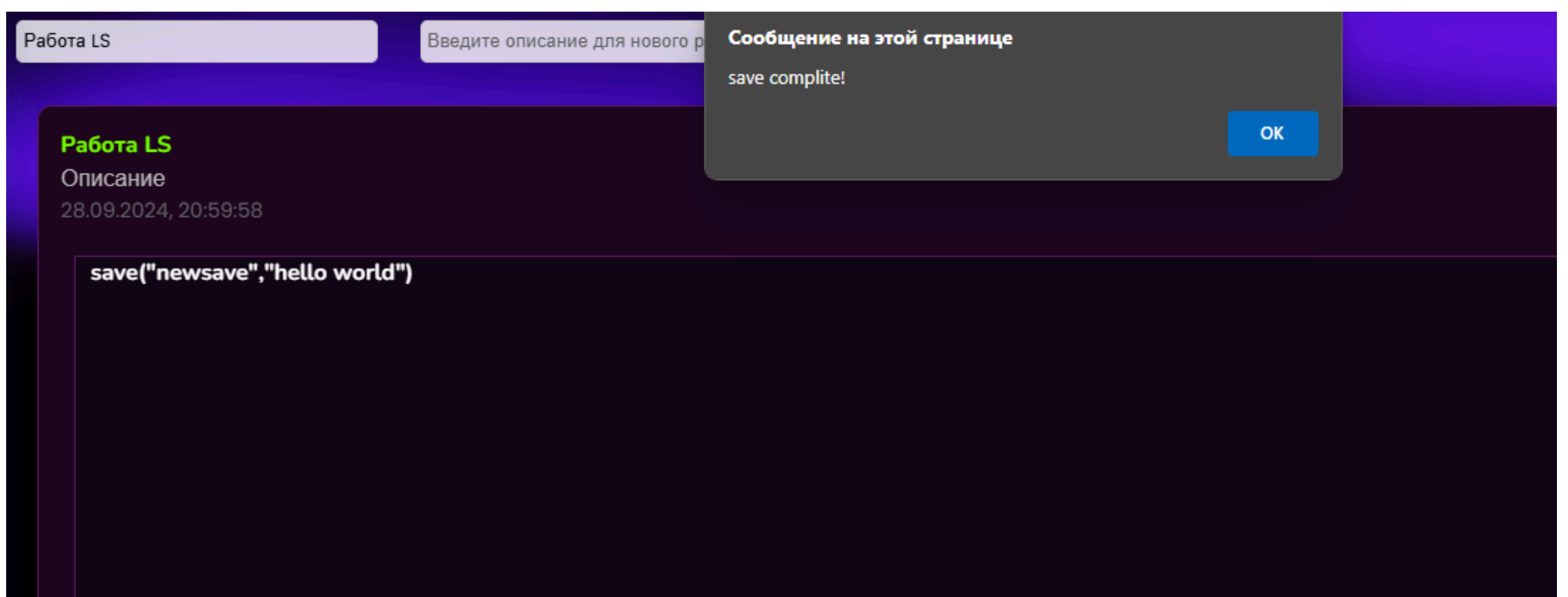
Как она устроена ?

```
function open(name){  
  let c= localStorage.getItem(name);  
  alert(c)  
}
```

Работа с LocalStorage

Работа с функциями

1. Пишем функцию `save()` и выполняем (рис.1)



Работа `save()` Рис.1

2. Пишем функцию `open()` и выполняем (рис.2)

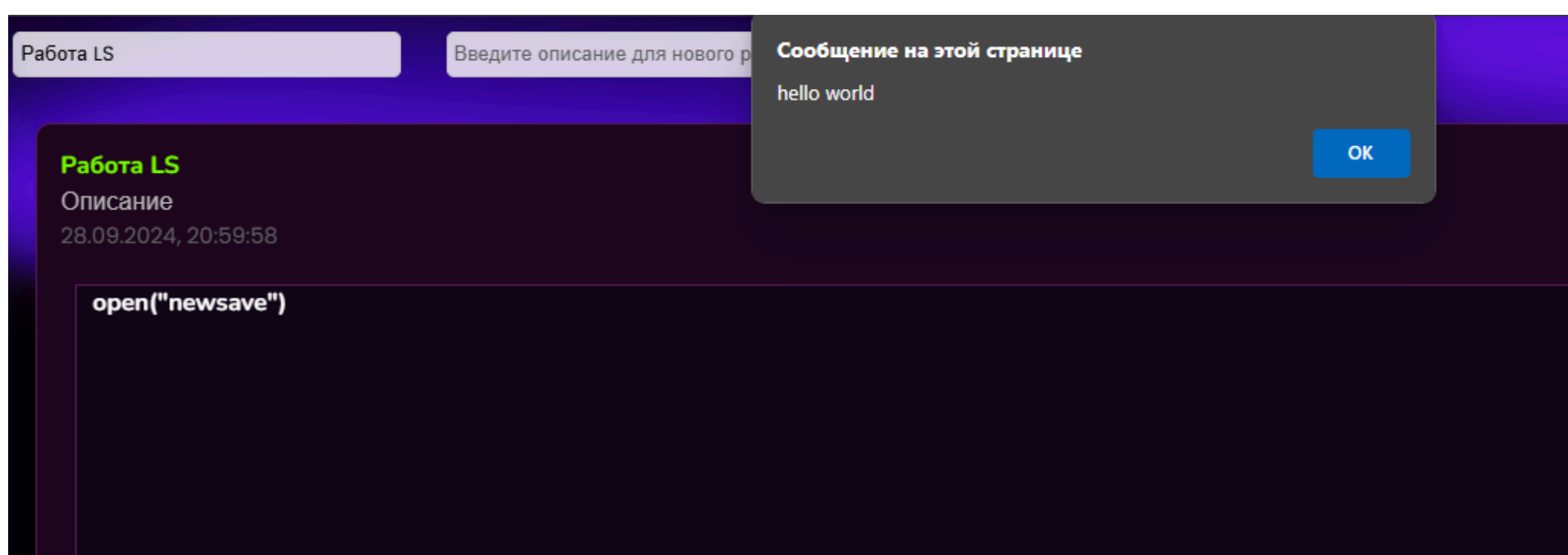


рис.2

Работа с файлами

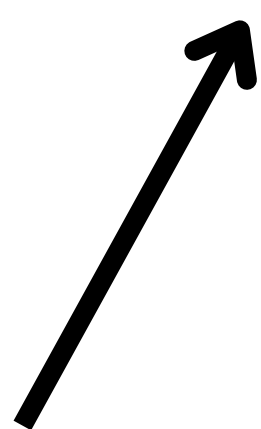
Знакомство с функцией `filesave()`

Для чего нужна функция `filesave()`?

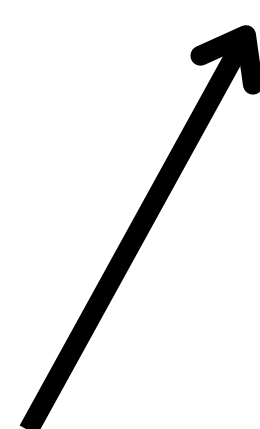
В работе с веб-страницами и приложениями на JavaScript часто возникает необходимость записи данных в файл. Например, может потребоваться сохранить результаты

В работе с веб-страницами и приложениями на JavaScript часто возникает необходимость записи данных в файл. Например, может потребоваться сохранить результаты работы скрипта или какие-то данные, введенные пользователем, в отдельный текстовый файл.

`filesave("index.html","hello world")`



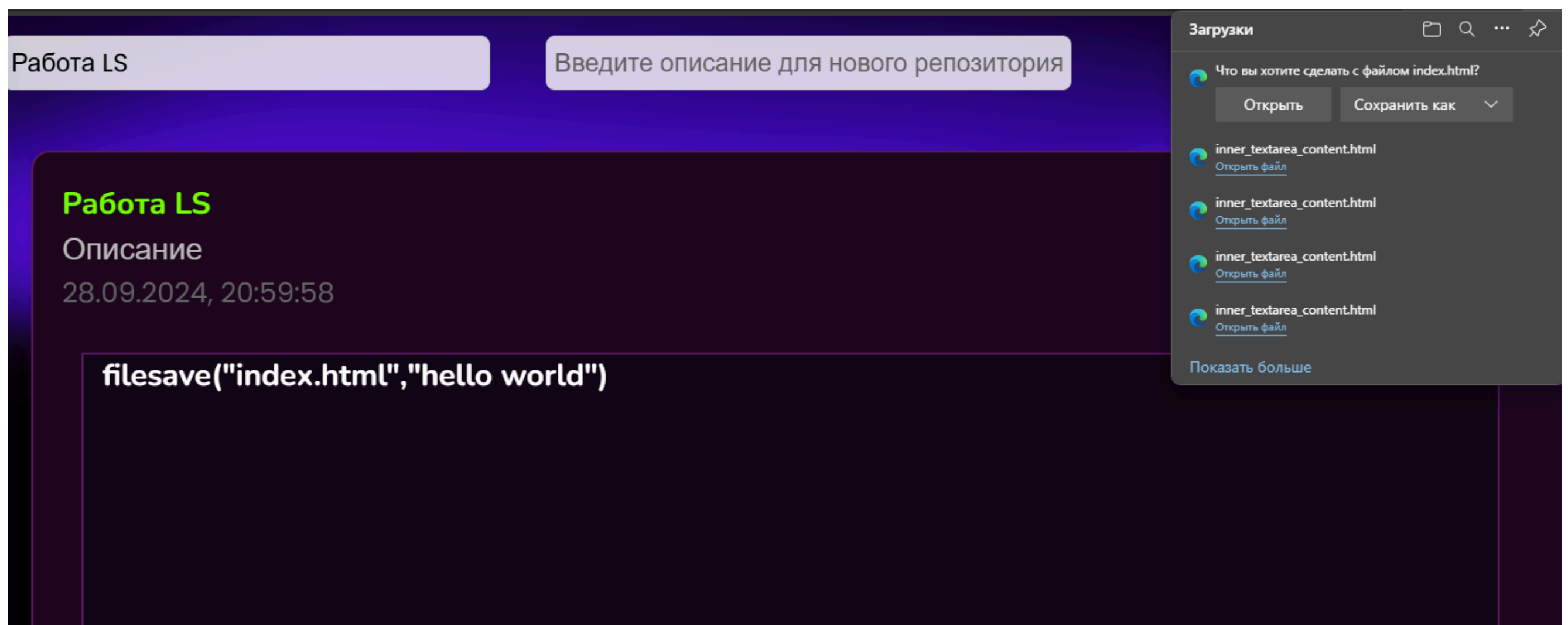
Название и формат
файла



Текст который нужно
в него поместить

Работа с файлами

Работа с `filesave()`



Поддерживаются форматы

html

js

txt

css

py

dhub

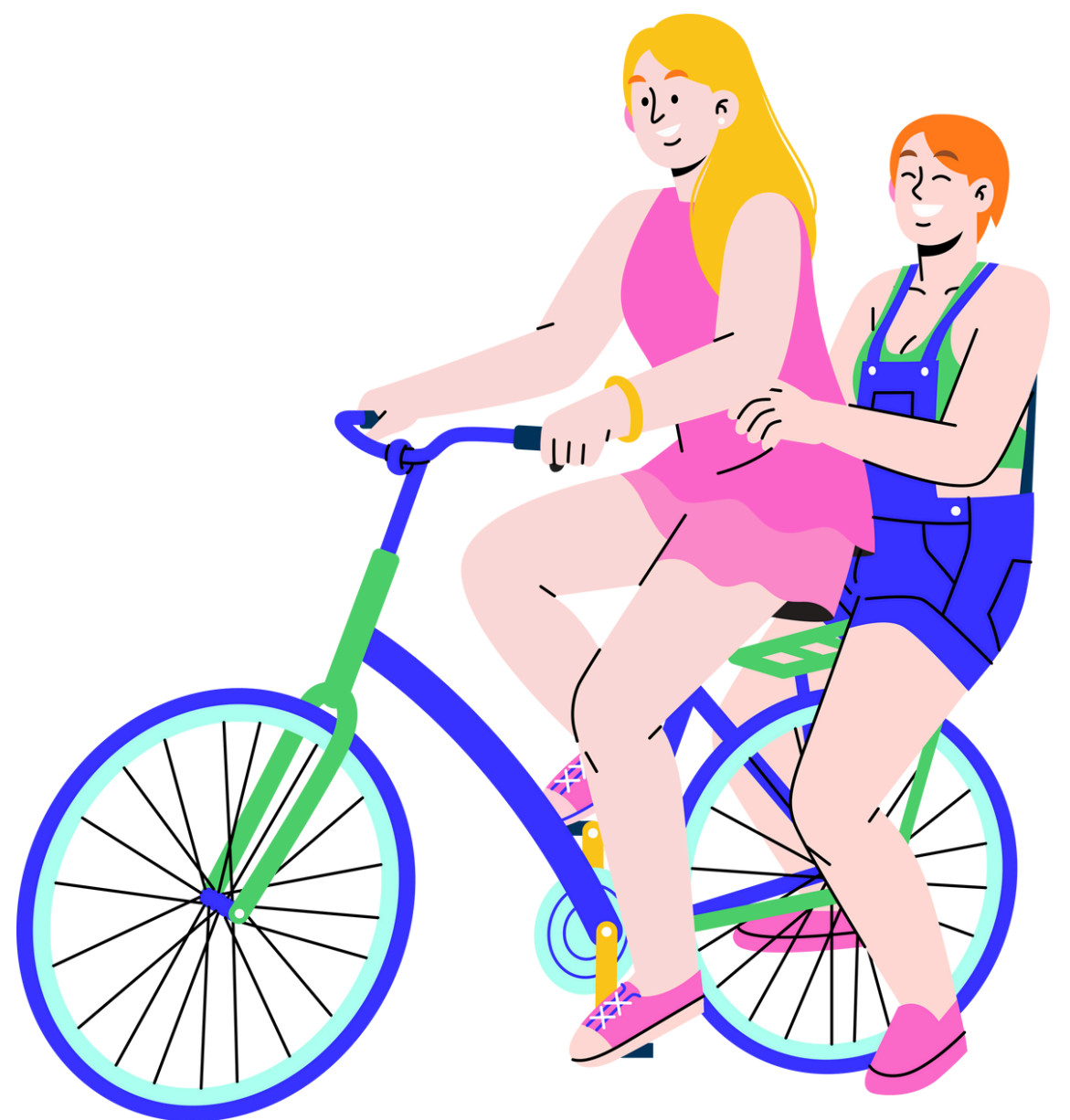
pack

vantage

Работа с файлами

Проблемы с `filsave()`

1. Нельзя сохранять в форматы `mp3` `mp4` `jpeg` `png` `svg` и другие
2. Функция пока не поддерживает чтение файлов
3. Нельзя сохранять `canvas` рисунки в функцию



Работа с логикой

Logic с correct()

JS Теория

В Javascript if() {} позволяет задать условие, при котором выполнится блок кода. Выше код внутри if будет выполнен, так как условие будет положительным (true). Ниже код внутри if не будет выполнен, так как не выполняется условие. в Javascript мы можем написать условие если-иначе.

Функция correct()

correct() - логическая функция сравнения переменных и вывода определённого значения при true или false

Если переменные будут равны то в alert покажется сообщение **text1** или **text2**

```
function correct(x,y,text1,text2)
{
  let result;
  if (x == y) {
    result = text1;
    alert(text1)
  } else {
    result =text2;
    alert(text2)
  }
  return result;
}
```

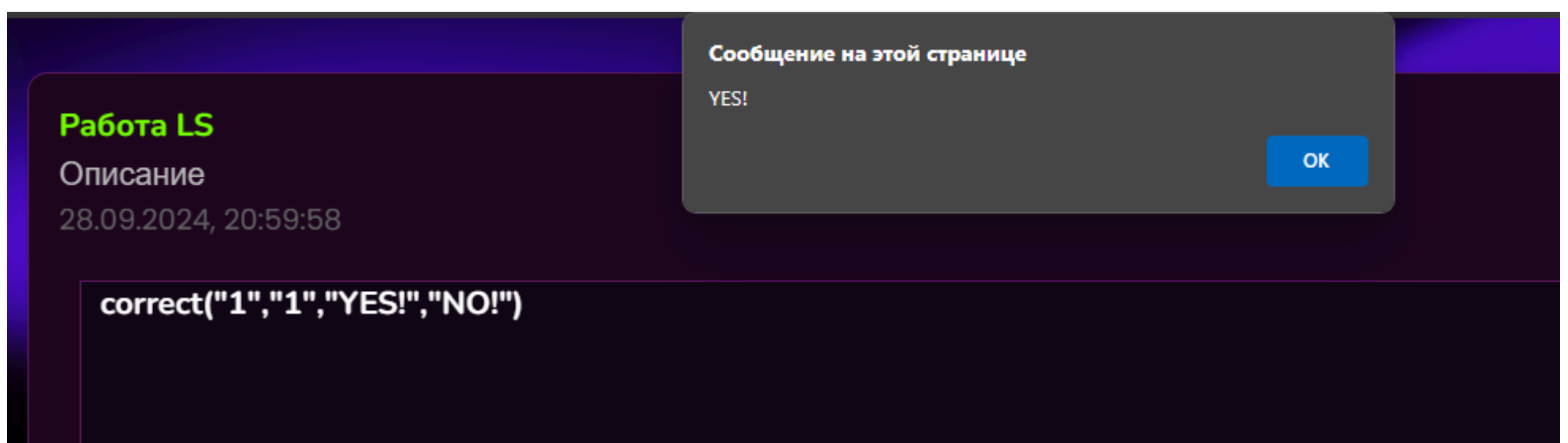
correct(1,1,"Равны!","False!")



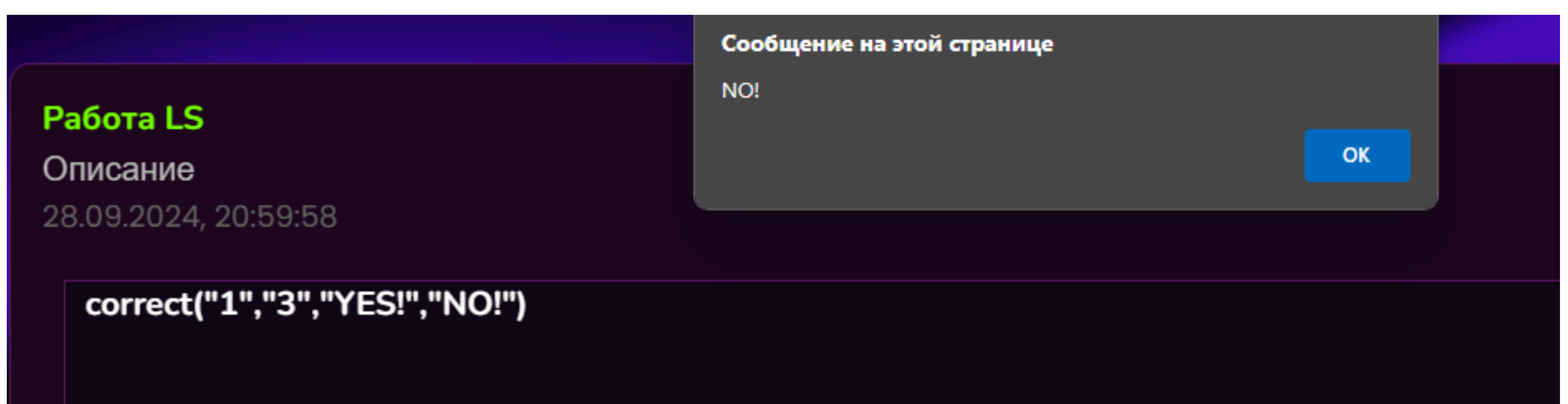
Работа с логикой

Logic работа с correct()

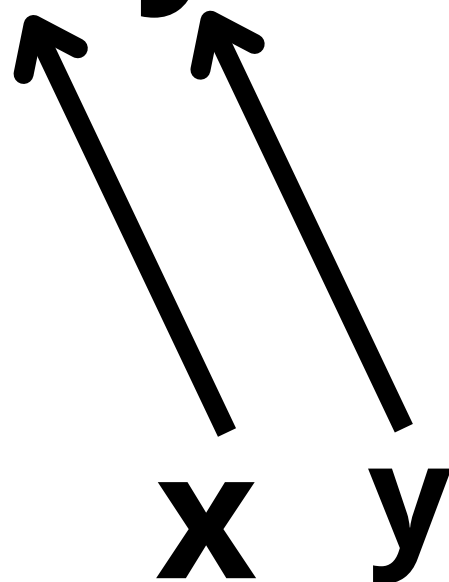
Если x равен с у



Если x не с у то



`correct(x,y,text1,text2)`



Просто да?

Работа с логикой

Проблемы correct()

Для вывода сообщения при true или false только используется alert()

Нельзя вместо сообщения вставить действие
Пример:



```
function correct(x,y,text1,text2){
  let result;
  if (x == y) {
    result = text1;
    myfunction()
  } else {
    result =text2;
    noth("no")
  }
  return result;
}
```

Только так

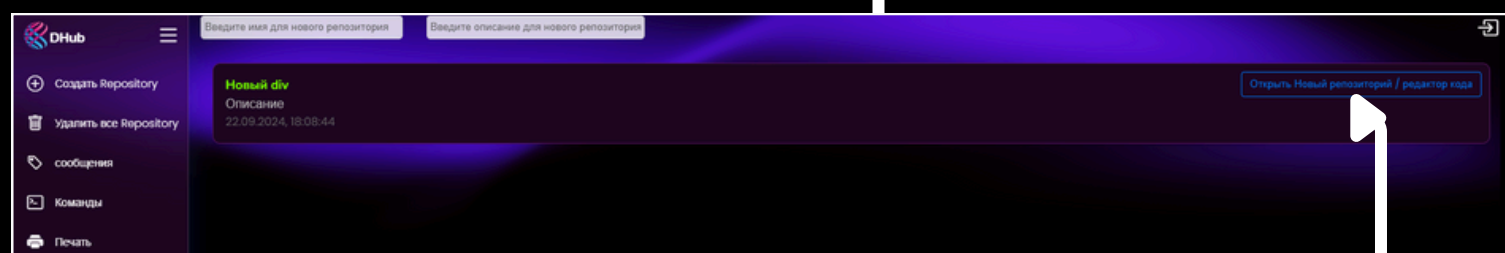
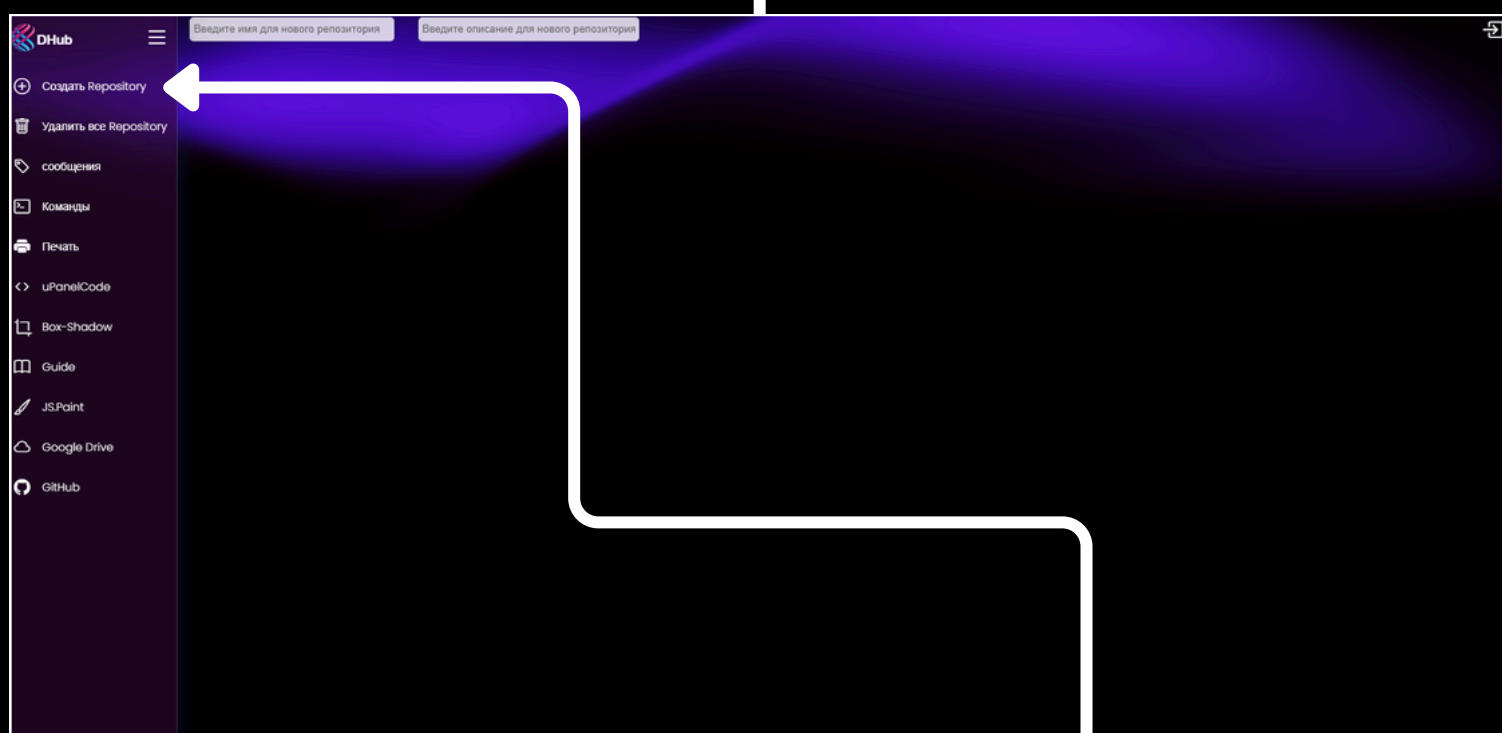
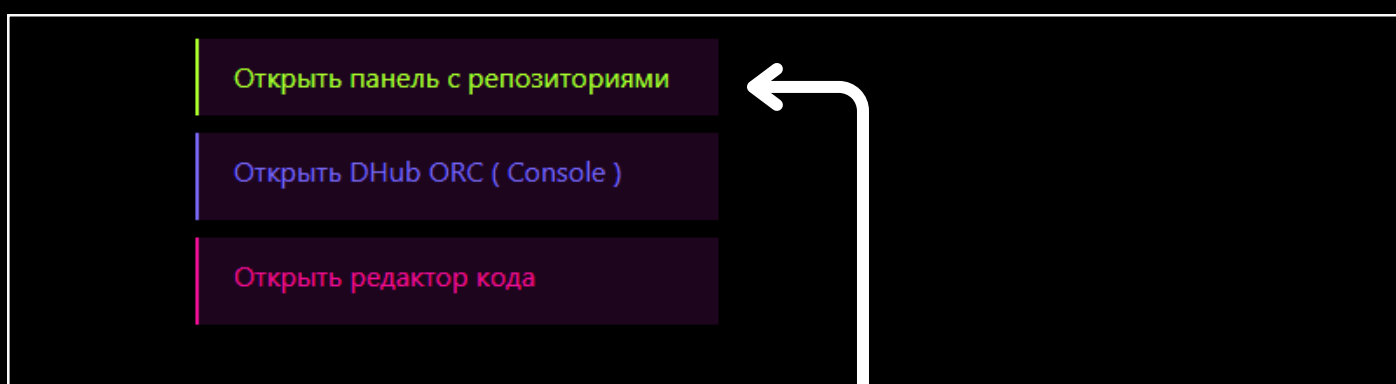
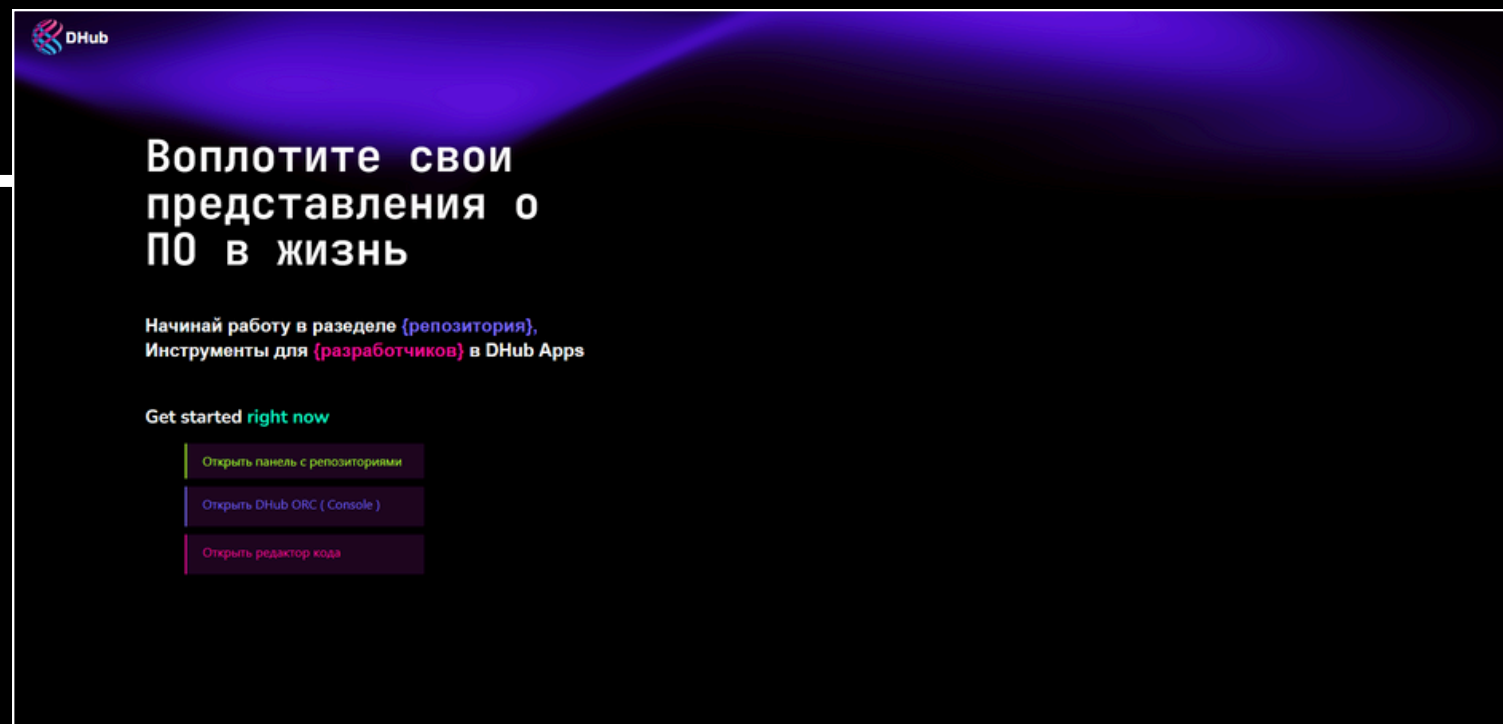
```
function correct(x,y,text1,text2){
  let result;
  if (x == y) {
    result = text1;
    alert(text1)
  } else {
    result =text2;
    alert(text2)
  }
  return result;
}
```



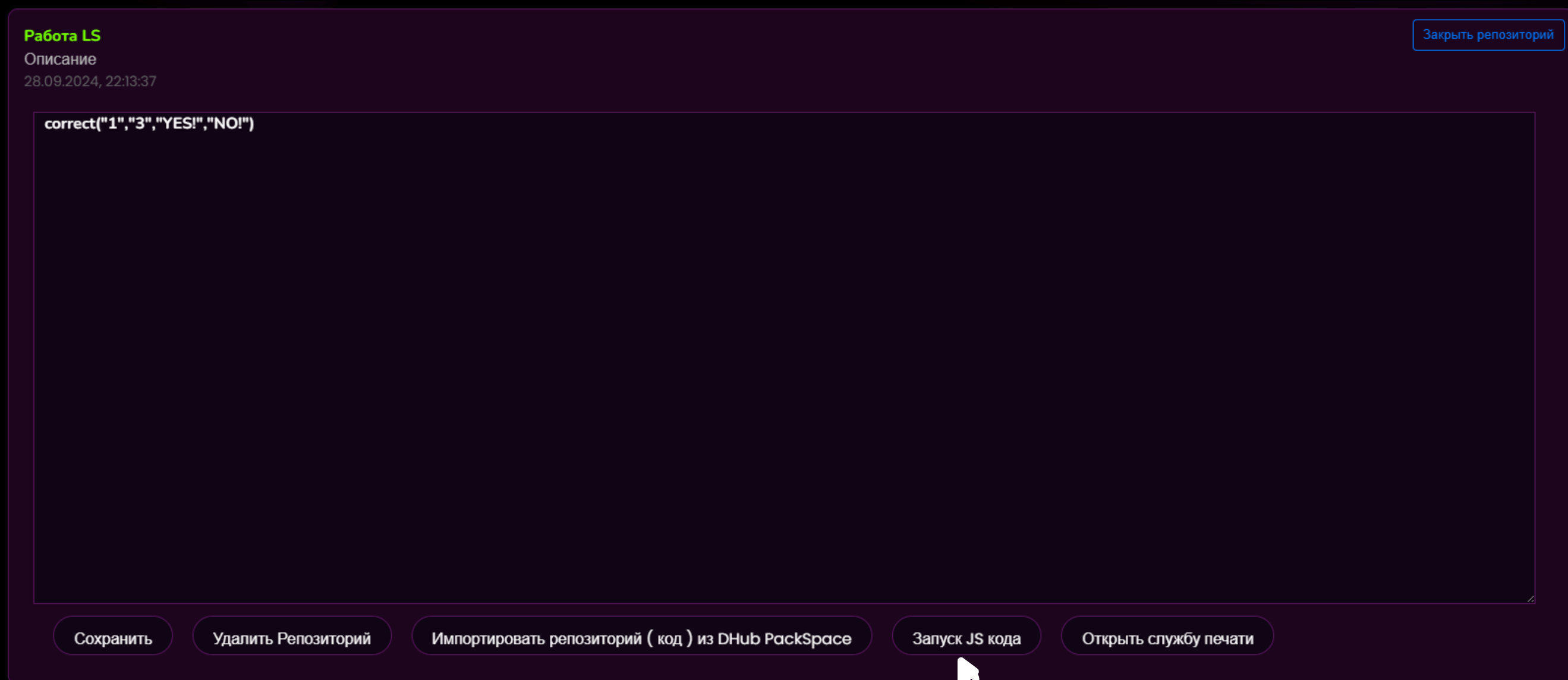
DotScript и DHub

Где начать программировать на DotScript?
в DHub

Начало работы

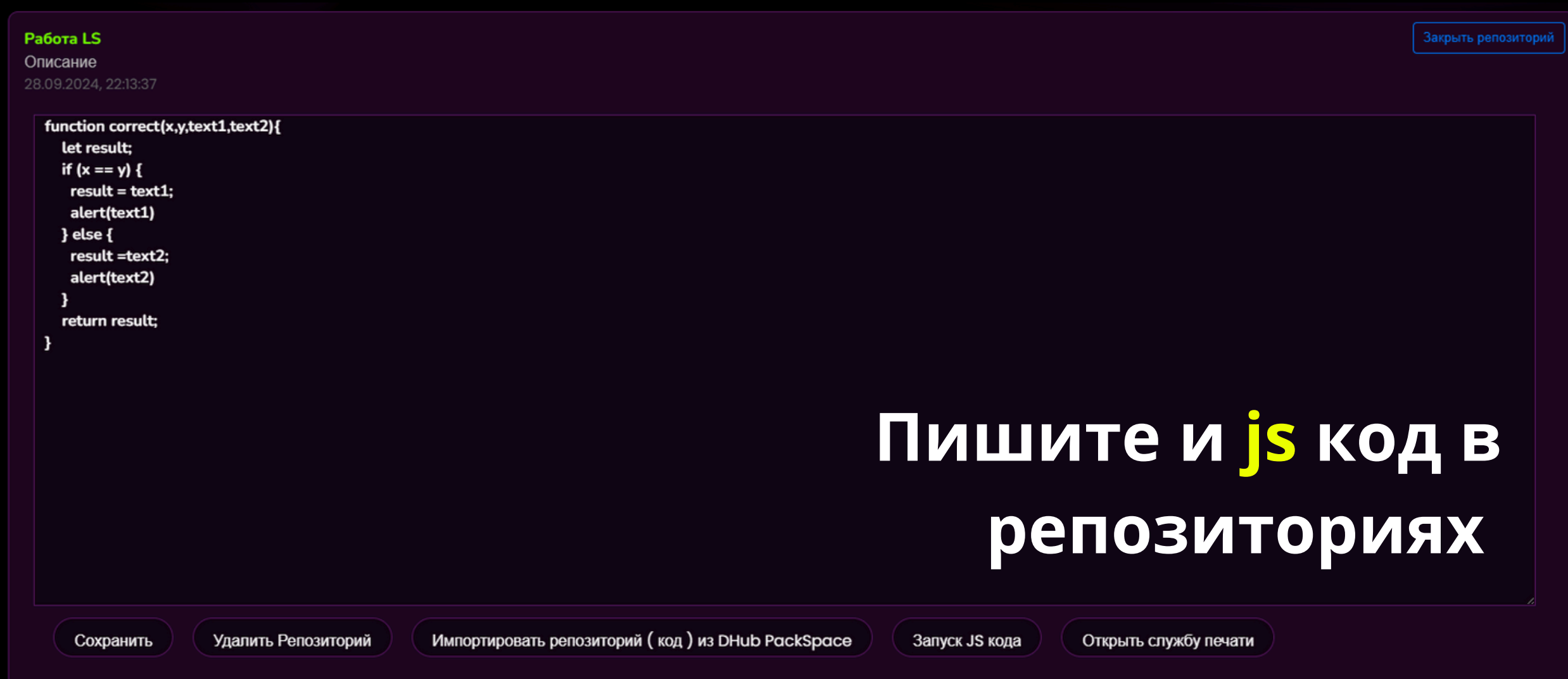


DotScript и DHub



Чтобы выполнить скрипт
нажмите на “запуск JS кода”

DotScript + JavaScript



Пишите и **js** код в
репозиториях

DotScript Native

Что такое DotScript Native

DotScript Native - это версия framework для IDE по типу VS Code и WebStorm.

Подключение к проекту

1. С GitHub скачайте DotScript Native
(ссылка на 30 стр.)

2. подключите к проекту

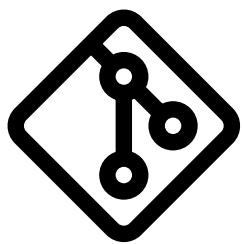
```
<script src="frame.js"></script>
```


DotScript



DHub

dhub-federation.netlify.app



Скачать **DotScript Native**

<https://github.com/DKhorov/DotScript.git>



DK Studio Community

t.me/dkdevelop



© Copyright DK Studio All rights reserved 2023-2024

Серия книг: 1 Название книг: Руководство по DotScript

Дата создания: 28.09.2024 12:00:00 Made in Belarus

Книга имеет полное право на использование в примерах сервиса DHub. Данное руководство обязательное для изучения служащих в DK Studio и DK Develop Studio. Книга создана в программе Canva, Microsoft Word 2016