

Уроки

Урок 1 - Лёгкий старт

Автор **cliva**,

10 декабря, 2022 в Уроки



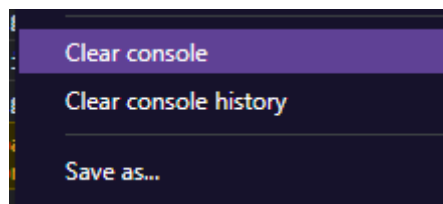
Лёгкий старт

Начинать лучше всего с основ языка, вы уже знакомы с понятием переменной, массива, функцией и т.д. Поэтому начинаем с практики, не будем пока углубляться в области видимости, классы, методы, функции и т.д. начнём с простых для понимания вещей.

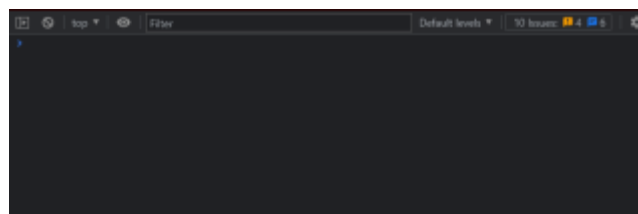
Откроем консоль вашего браузера - **Ctrl+Shift+J**

Это пожалуй одно из самых доступных средств для запуска JS кода, потому что браузер есть у каждого, я полагаю)

Отлично, консоль открыта, но вероятно там вы увидели кучу текста, ошибок или предупреждений, верим мне на слово, ничего интересного сейчас там нет, жмём ПКМ по любой строке и в контекстном меню нажимаем **clear console**



работу легче начинать с чистой консолью



И сразу же в неё пишем нашу первую строку кода js - `console.log("Hello world!")`, нажимаем интер, и видим что в консоли помимо нашей строки кода появилось сообщение Hello World! Запоминаем, что

команда `console.log()` - выводит сообщение в консоль с текстом из скобок

```
> console.log("Hello world!")
Hello world!
< undefined
```

Теперь объявим переменную, в js переменную можно объявить тремя ключевыми словами **let**, **var**, **const**

Между `let` и `var` существенных различий нет, просто запомним, что **var - устаревшее** ключевое слово и будем вместо него использовать **let**, а `const` - как можно понять, это сокращение слова "константа" - то есть это по сути не переменная, а постоянная :)

Конструкция для объявления переменной в js очень простая **ключ.слово имя = значение**;

```
let myFirstVar = 1;
const G = 9.8;
let message = "Hello world!";
let listNum = [1,2,3,4,5];
const emtyList = [];
```

Это всё переменные (или постоянные), как можно заметить нам не нужно указывать, число это, строковое значение, или массив - JS делает всё за нас, и про это не стоит забывать, я до сих пор встречаю людей, которые удивляются что для js "2"+2="22", а "2"-2=0

Возможно эти сложения и вычитания двоек кажутся сложными и непонятными, но на самом деле все проще, знак "плюс" в js не только сложение, но и "склеивание" строковых типов то есть "абв"+"где"="абвгде", а минус это только мат. операция, поэтому JS переводит строку в число, если может, а если не может он выдаст нам "NaN" что означает, что где-то мы намудрили и пытаемся вычесть что-то из строки

Объявим переменную `let x = 3` `let x = 3`, жмём интер, теперь эта переменная "хранится где-то в консоли", как же её достать(?), пробуем её написать - консоль нам тусклым текстом покажет нам её значение `> x` `< 3`, поробуем нашу переменную умножить на 4 - для этого нам нужно присвоить ей новое

значение, то есть `x = x * 4`; `> x = x * 4` `< 12`, теперь в качестве небольшой практики отнимем 2, после

приравняем эту переменную 0

```
> x = x * 4
< 12
> x = x - 2
< 10
> x = 0
< 0
```

С числами поигрались, теперь можно поиграться со строками, задаём новую переменную, пусть будет

`myText = "какой-то текст"`

```
> let myText = "какой-то текст"
< undefined
> myText
< 'какой-то текст'
```

добавим к концу этой строки " и ещё какой-то текст"

```
> myText = myText + " и ещё какой-то текст"
< 'какой-то текст и ещё какой-то текст'
```

, а теперь в начало этой же строки добавим "Мой "

```
> myText = "Мой " + myText
< 'Мой какой-то текст и ещё какой-то текст'
```

Строка - набор символов, а значит мы можем достать любой символ из строки, для этого используем квадратные скобки:

```

> myText = "Мой " + myText
< 'Мой какой-то текст и ещё какой-то текст'
> myText[2]
< 'й'
> myText[0]
< 'М'

```

- как можно заметить отсчёт идёт от 0, убедимся -

Теперь потрогаем массив - `let myArray = [0,1,2,3,4,5]` - массив от 0 до 6, достать из него значение можно также как и из строки - квадратными скобками

```

> let myArray = [0,1,2,3,4,5]
< undefined
> myArray[1]
< 1

```

, но в отличие от строки - мы можем изменить значение элемента, что и сделаем, заменим 4 на "четыре"

```

> myArray[4]="Четыре"
< 'Четыре'
> myArray
< ▶ (6) [0, 1, 2, 3, 'Четыре', 5]

```

, добавим 88 в конец массива с помощью команды `.push()`

```

> myArray.push(88)
< 7
> myArray
< ▶ (7) [0, 1, 2, 3, 'Четыре', 5, 88]

```

С переменными более менее познакомились, теперь заденем функцию, вот так её можно объявить:

```

function Name(params){
    //код функции
    return 0 //возвращаемое значение, в данном случае - 0
}

```

Объявим функцию, с помощью которой мы будем приветствовать наших виртуальных собеседников в консоли:

```

> function sayHi(name){
  console.log(name+" , добрый день!");
}

```

(перенос текста в консоли - **Shift+Enter**)

Нажимаем интер, чтобы функция "сохранилась в консоли" и поприветствуем пару "человек":

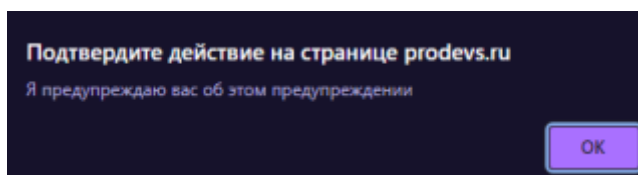
```

> sayHi("Анна")
Анна, добрый день!
< undefined
> sayHi("Вячеслав")
Вячеслав, добрый день!
< undefined
> sayHi("Игорь Олегович")
Игорь Олегович, добрый день!
< undefined

```

нажимая стрелку вверх на клавиатуре - можно быстро вводить команды из истории консоли

Ещё считаю неплохо будет упомянуть о команде `alert()` - она вызовет окно предупреждения, с заданным нами текстом, пробуем - `alert("Я предупреждаю вас об этом предупреждении")` и получаем



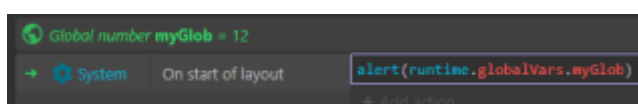
Думаю, можно закрывать консоль.

Какие-то лёгкие моменты js мы обозрели и даже потрогали, теперь можно переходить к Construct 3

Создаём пустой проект, будем тренироваться в нём, переходим сразу к списку событий, добавим условие "on start of layout", чтобы добавить скрипт в действия(action) просто жмём ПКМ по `+ Add action`

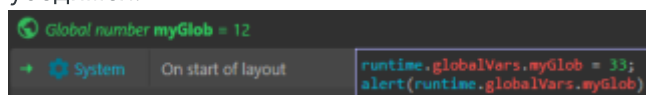
и в контекстном меню выбираем `JS Add script`, теперь напишем там `alert("Мой первый алёрт в СЗ")`, жмём `ctrl+Enter` или ЛКМ по пустому месту в списке событий и запускаем проект - получаем результат, а именно окно предупреждения с заданным текстом.

Теперь попробуем в алёрт вывести значение глобальной переменной, для этого создадим её (с любым именем), и присвоим ей какое-нибудь значение, теперь, чтобы достать её из скрипта, нам нужно прописать `runtime.globalVars.ИмяПеременной`, что и делаем:



Запускаем и любимся

С конструкцией `runtime.globalVars.ИмяПеременной` можно обращаться как с обычной переменной в js, убедимся:

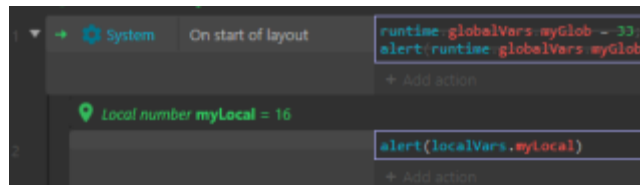


- и в правду, алёрт теперь выводит 33 вместо 12 (не

забываем про `.` точку с запятой)

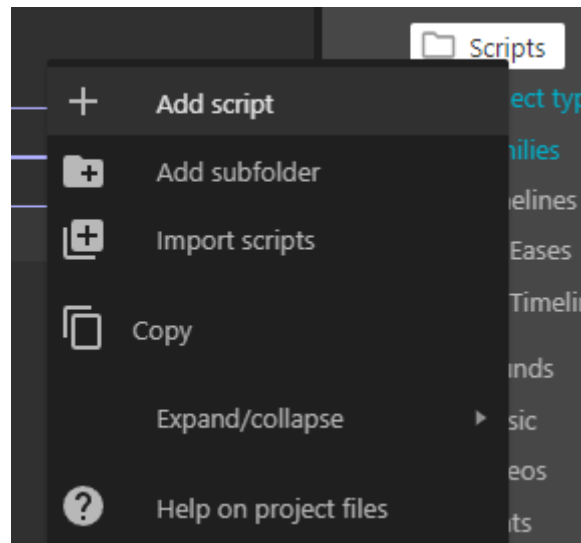
Также стоит учитывать, переменная у нас текстовая или числовая, **именно в СЗ** нельзя загнать строку в числовую переменную и наоборот, когда-нибудь мы вернёмся к этому моменту

Теперь создадим пустое под-событие (клавиша В англ.) и объявим над ним локальную переменную, до неё достать ещё проще - `localVars.ИмяПеременной`, без прелюдий запишем её в алёрт:

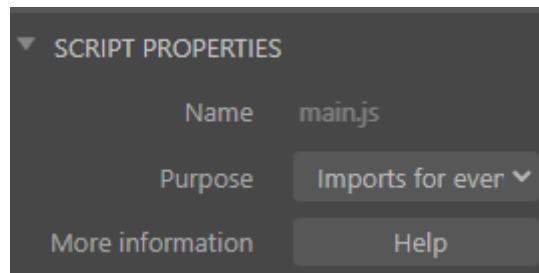


И, вы правы - с ней можно также обращаться как с обычной переменной в js, убедиться в этом вы можете сами

Теперь попробуем кое-что посложнее - объявим функцию приветствия, которую мы ранее уже объявляли в консоли. В списке событий этого не получится (точнее получится, но функция будет доступна только в текущем куске кода, что не очень интересно)
Поэтому добавляем файл скрипта в проект:



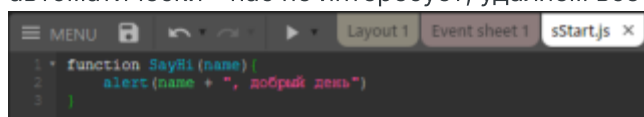
И сразу слева, в окне свойств настроим его, чтобы он импортировался в наш список событий:



(main script пока не трогаем, возможно и до него дойдём)

Имя файла можем выставить любым, я назову sStart , всё что в этом файле написано

автоматически - нас не интересует, удаляем всё и пишем там нашу функцию:



Теперь применяем все наши знания, чтобы вызвать эту функцию, параметром которой будет имя, находящееся в локальной переменной myName

 **попробуйте сначала сами (Показать контент)**

На этом, думаю, стоит сделать небольшую паузу

Домашнее задание:

- при нажатии на кнопку - сложение глобальной переменной с локальной, сумма пусть попадает в глобальную (глоб.пер. = глоб.пер. + лок.пер) и выводится в текст (в текст помещаем знач глоб. пер. ивентом)

-создать функцию на js которая выведет в алёрт квадрат числа + 1 (число пишаем в параметр функции)

-создать функцию которая выведет в алёрт сумму 4 параметров

*- создать в js файле переменную = "", на макет добавить кнопку и поле для ввода, при нажатии на кнопку пусть вызывается js функция, которая **добавит** в переменную введенный (в поле для ввода) текст и выведет переменную в алёрт

Если какое-то задание не получилось или остались вопросы по материалу - задаём вопросы в этой теме, если имеются вопросы не относящиеся именно к этой теме - переходим в клуб, во вкладке [Общее](#) есть подходящие темы.

Исходник урока - [simpleStart.c3p](#)

[Вернуться в основную тему](#)

Изменено 10 декабря, 2022 пользователем cliva

 1

[bioid.space](#)

...

@cliva Привет, спасибо за урок! Очень интересно, но пока некоторые моменты не понятны, и есть парочка вопросов:

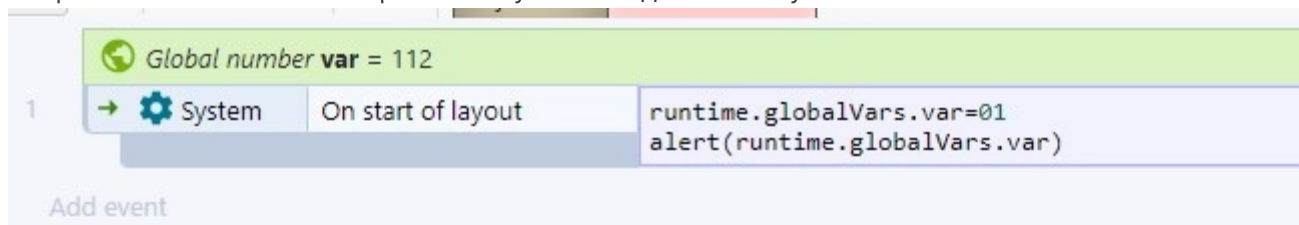
1. Забыл написать **let** и написал сразу `x = 3`, сработало, почему так?
2. Первый раз у меня функция **sayHi** почему-то не запустилась, потом перепроверил, вместо `;` поставил `:`. Значит знаки препинания критически важны, можешь чуть подробнее про пунктуацию написать. И у тебя заметил в первой функции есть `;` а во второй нет.

```
> function sayHi(name){
  console.log(name+"", "добрый день!");
}

function SayHi(name){
  alert(name + ", добрый день")
}
```

И ещё к этому же вопросу, важен ли пробел? Есть ли разница `let x = 3` и `let x=3`?

3. при замене глобальной переменной у меня выдаёт ошибку синтаксиса:



Подтвердите действие на странице preview.construct.net

Failed to load JavaScript code used in events. Check all your JavaScript code has valid syntax.

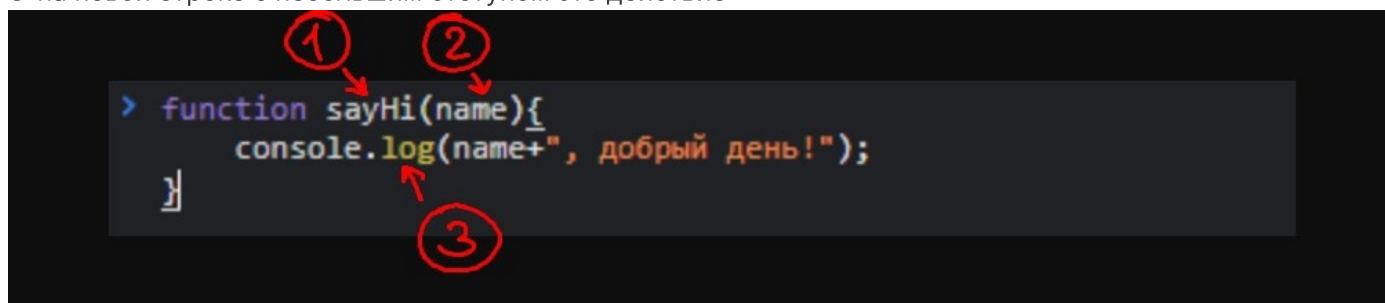
OK

4. Если я правильно понял логику написания функции, то по аналогии с конструкторскими функциями

1-это название функции

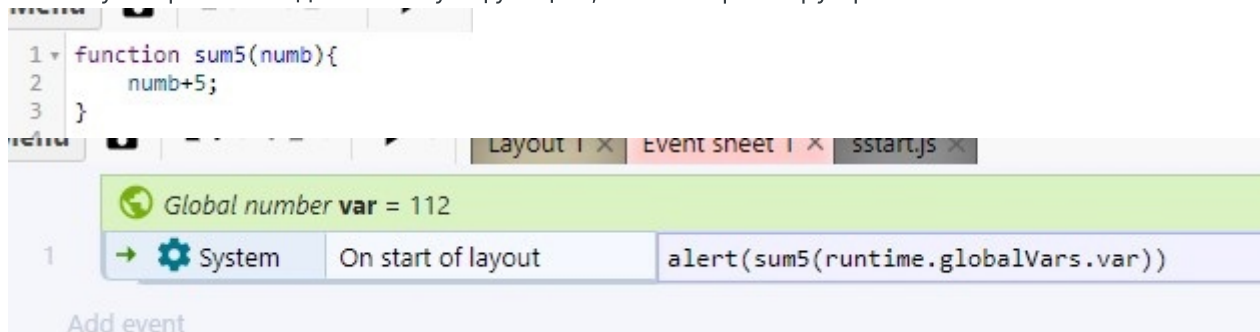
2-в скобках это параметр функции

3-на новой строке с небольшим отступом это действие



соответственно действие может быть любое,

поэтому попробовал сделать такую функцию, чтоб к параметру прибавляла 5



но в алерте вместо 117 пишет **undefined**, в чём ошибка?

Игры: dmitrygalias.itch.io

...

В 11.12.2022 в 09:50, dmitryartist сказал:

↩

1. Забыл написать **let** и написал сразу `x = 3`, сработало, почему так?

Это происходит потому, что в консоли браузера **выключен** по умолчанию **строгий режим**, его отсутствие позволяет опускать некоторые ключевые слова, но это может приводить к разным багам/ошибкам, если для куска кода активировать строгий режим, то таким образом без ключевого слова задать переменную не получится

```
> "use strict";
g = 5;
✖ ▶ Uncaught ReferenceError: g is not defined
   at <anonymous>:2:3
```

Строгий режим по дефолту включён в С3 и выключить его нельзя

В 11.12.2022 в 09:50, dmitryartist сказал:

2. Первый раз у меня функция **sayHi** почему-то не запустилась, потом перепроверил, вместо ; поставил :. Значит знаки препинания критически важны, можешь чуть подробнее про пунктуацию написать. И у тебя заметил в первой функции есть ; а во второй нет.

Также красиво как в этой [статье](#) не напишу)

Если просто - между двумя законченными строчками кода **при переносе строки** можно опустить точку с запятой:

```
> console.log(123)
console.log(456)
123
456
> console.log(123)console.log(456)
✖ Uncaught SyntaxError: Unexpected identifier 'console'
```

однако, если убрать перенос - получим ошибку

Если строка кода не закончена, js будет искать её продолжение на следующей строке:

```
> console.log(1+
  2+
  3)
6
```

Потому, лучше не опускать ; там, где в этом не уверены

В функции, цикле, классе, методе и тд, где используются фигурные скобки {...}, для отделения блока кода - последнюю точку с запятой **зачастую** опускают, я же один раз проставил её на автомате)

В твоём случае, где ты поставил двоеточие - была "незакончена" строка кода и js ожидал продолжения, ибо двоеточие в основном используется для пар **ключ: значение**, либо для троичных операторов

условие? если верно : если не верно:

```
> 5<6? console.log('математика работает'):console.log('математика НЕ работает')
математика работает
```

В 11.12.2022 в 09:50, dmitryartist сказал:

3. при замене глобальной переменной у меня выдаёт ошибку синтаксиса:

01 - это не число) Если поставить просто 1 - заработает

В 11.12.2022 в 09:50, dmitryartist сказал:

4. Если я правильно понял логику написания функции, то по аналогии с конструкторовскими функциями

- 1-это название функции
- 2-в скобках это параметр функции
- 3-на новой строке с небольшим отступом это действие

Именно, отступ можно и не делать, но лучше поддерживать читаемость кода, иначе рано или поздно выйдет что-то такое:

[illegible]

С твоей функцией проблема в том, что ты не вернул ни одного значения, попробуй перед **numb+5** добавить **return**

Изменено 11 декабря, 2022 пользователем cliva

 1

bioid.space

...

✓ В 11.12.2022 в 12:24, sliva сказал:

Также красиво как в этой [статье](#) не напишу)

Спасибо, уже читаю)

✓ В 11.12.2022 в 12:24, sliva сказал:

попробуй перед **numb+5** добавить **return**

Ура, заработало!

Изменено 11 декабря, 2022 пользователем dmitryartist

Игры: dmitrygalias.itch.io

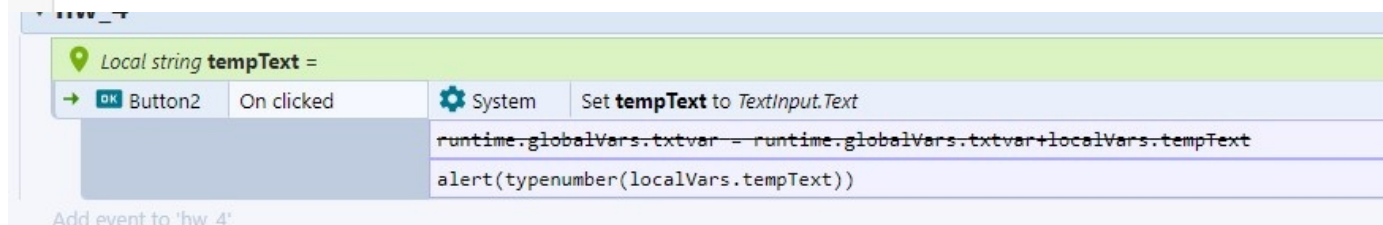
...

В 10.12.2022 в 21:01, cliva сказал:

Домашнее задание:

Сделал домашние задания самым сложным оказалось последнее, результат выдаёт но пришлось дополнительную переменную водить **tempText**, не уверен что это правильное решение:

```
1 let txtvar = "";
2 function typenumber (sometxt){
3     txtvar=txtvar+sometxt
4     return txtvar;
5 }
```



И ещё заметил что если имя параметра функции будет начинаться с цифры то выдаст ошибку, это похоже к с 01 ошибкой, но почему так нельзя так и не понял.

```
3 }
4 function f4param(1par, 2par, 3par, 4par){
5     return 1par+2par+3par+4par;
6 }
3 }
4 function f4param(par1, par2, par3, par4){
5     return par1+par2+par3+par4;
6 }
```

Игры: dmitrygalias.itch.io

...

В 15.12.2022 в 19:47, dmitryartist сказал:

но пришлось дополнительную переменную водить **tempText**, не уверен что это правильное решение:

Решение для первого урока - вполне приемлемое, во втором уроке как раз учимся работать с объектами прямее

Последние задания всегда "со звёздочкой", стараюсь придумывать ситуацию, на которую урок не наводит прямо, но как бы намекает

▼ В 15.12.2022 в 19:47, dmitryartist сказал:



И ещё заметил что если имя параметра функции будет начинаться с цифры то выдаст ошибку, это похоже к с 01 ошибкой, но почему так нельзя так и не понял.

Я и забыл, что С3 на столько лоялен к пользователю)

Почти во всех языках программирования есть запрет на использование разного рода спец-знаков и цифр в начале имени:

В js есть несколько правил по именованию переменных. Название переменной должно:

- содержать только буквы (кстати любого алфавита), цифры или символы \$ и _
- не начинаться с цифры
- не быть зарезервированным словом JS

bioid.space

< [Перейти к списку тем](#)

Последние посетители 0 пользователей онлайн

Ни одного зарегистрированного пользователя не просматривает данную страницу