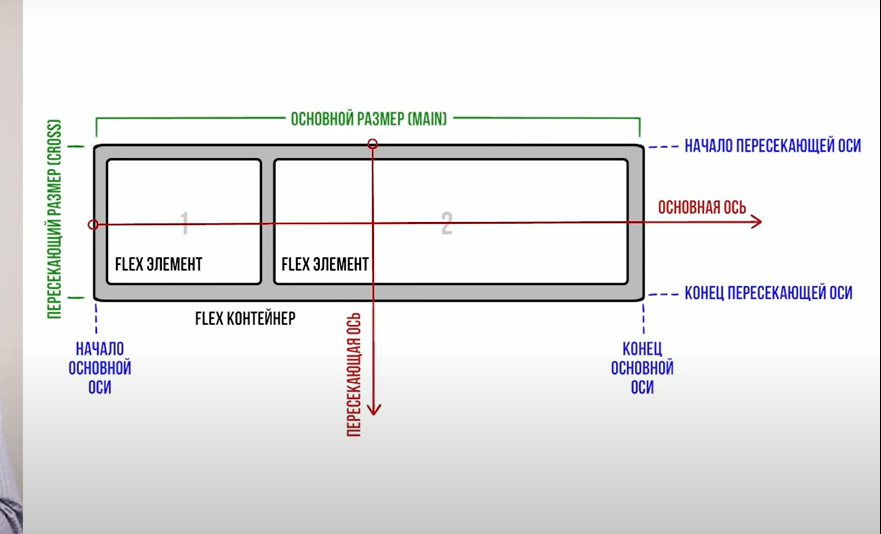
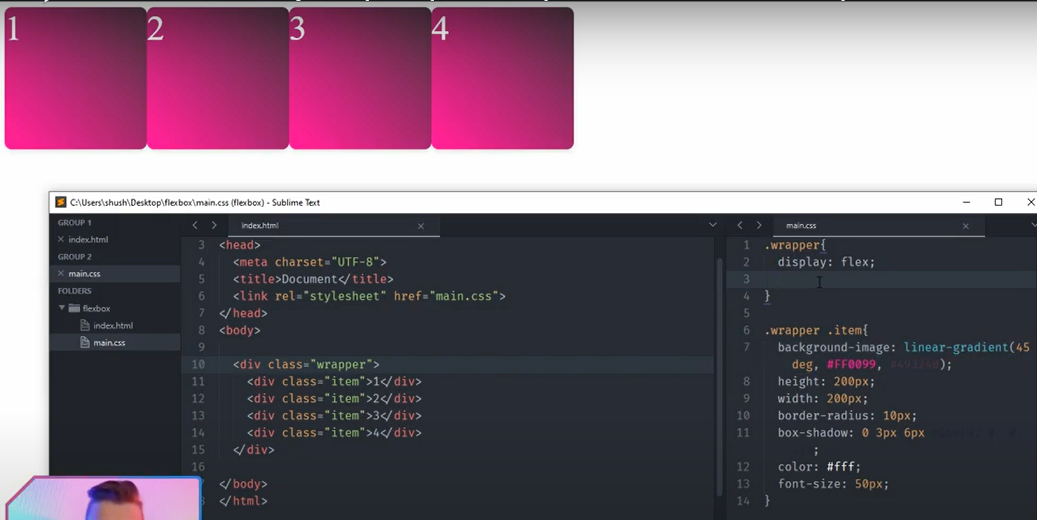
**CSS GREED**

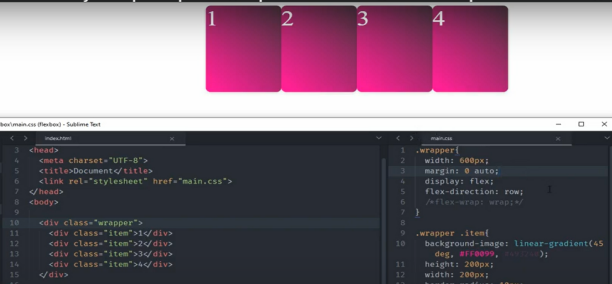


**FLEX BOX**

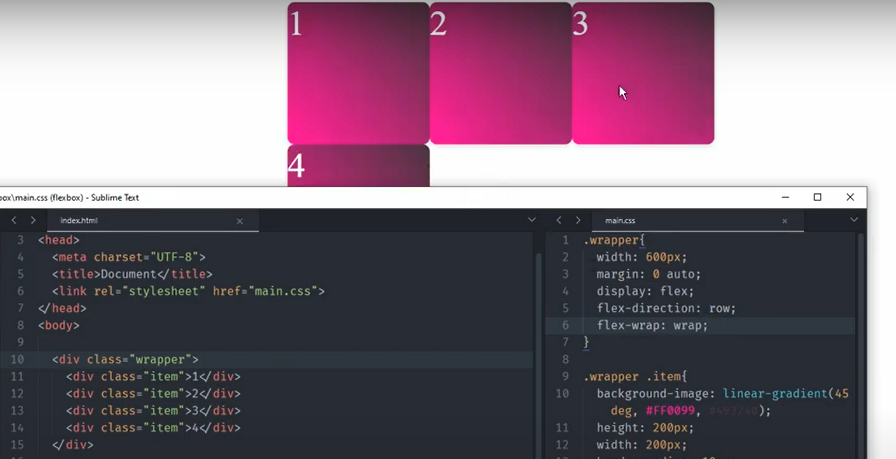




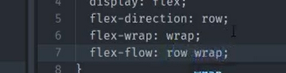
Ширины блоков 200 px, а указали общую ширину контейнера 600px, то есть блоки все не влезут (но можно применить св-во **flex-wrap: wrap** и тогда все блоки подстроятся под общую ширину):

До св-ва flex-wrap:

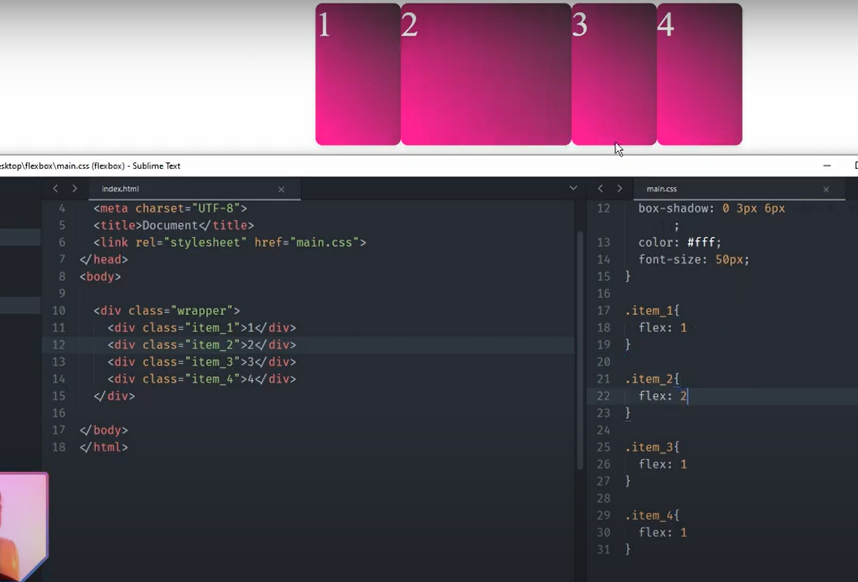
После св-ва flex-wrap:



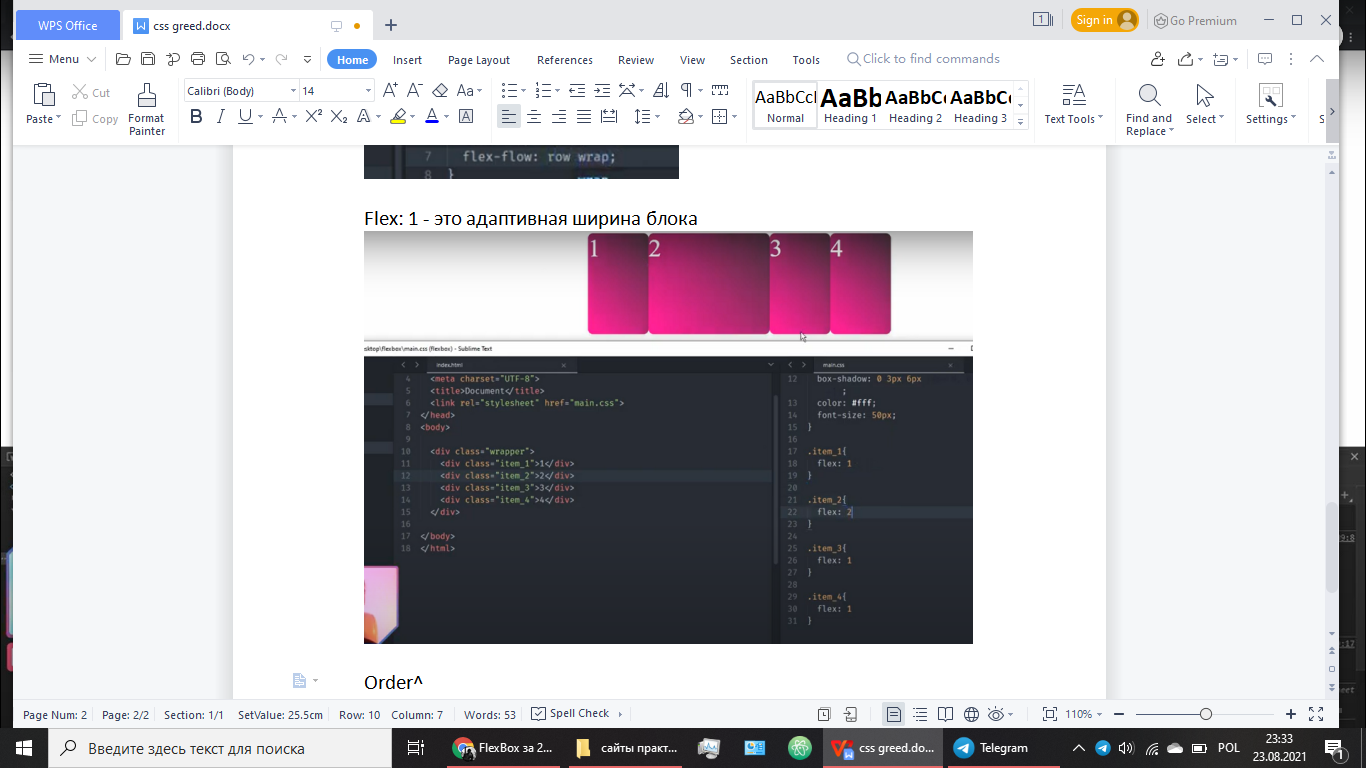
Flex-flow = flex-wrap + flex-direction



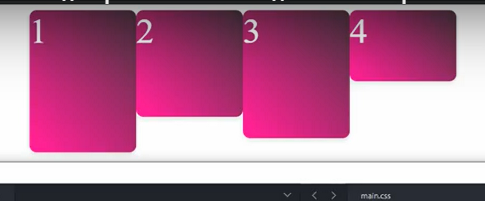
Flex: 1 - это адаптивная ширина блока

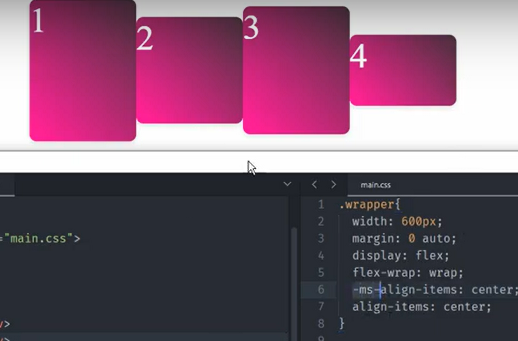


Order: 2 - меняет местами блоки, по умолчанию значение 1

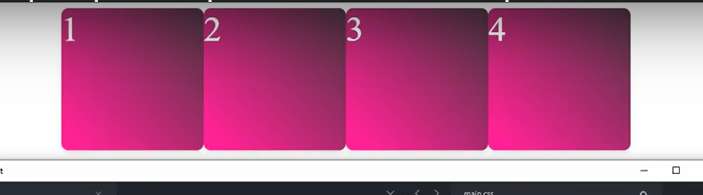


align-items: center/flex-start/flex-end/baseline… (до и после применения)

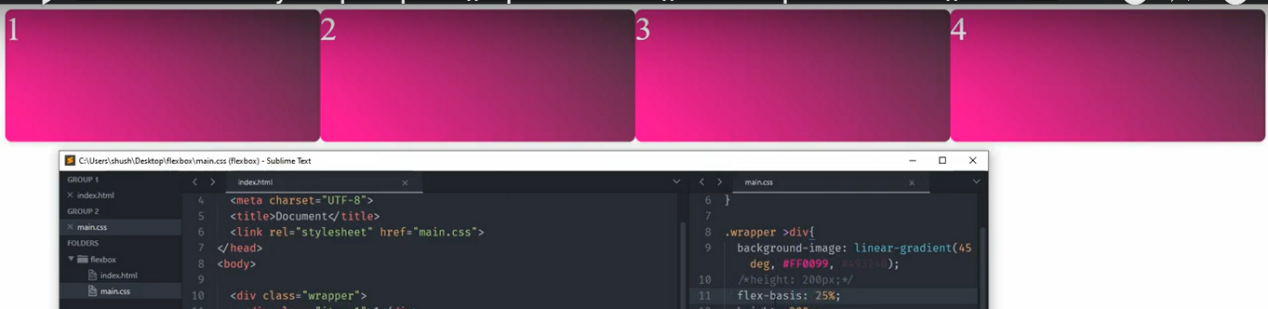




justify-content: center/flex-end/flex-start/space-between/space-around

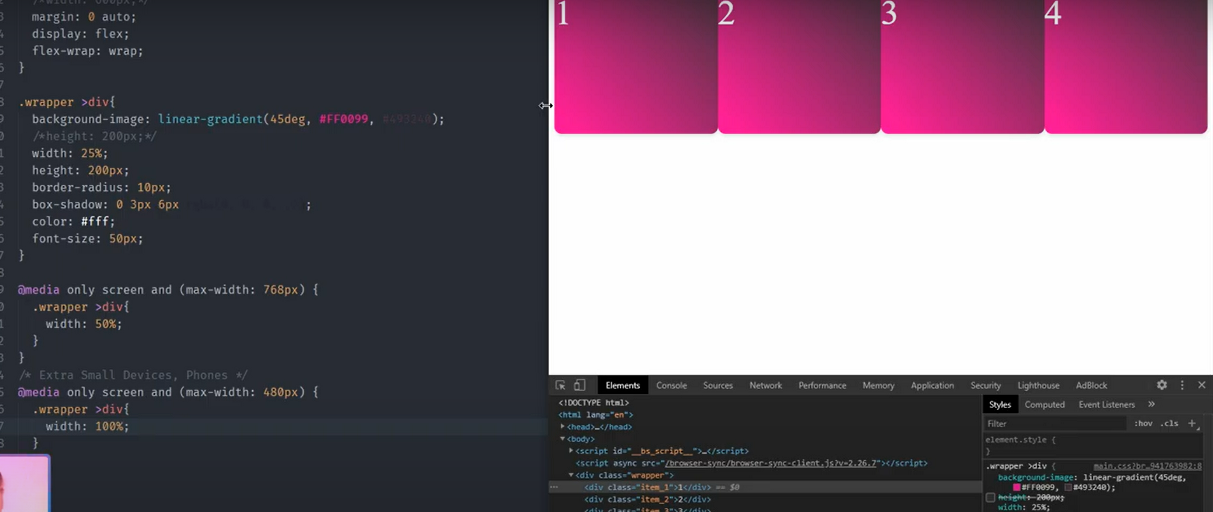
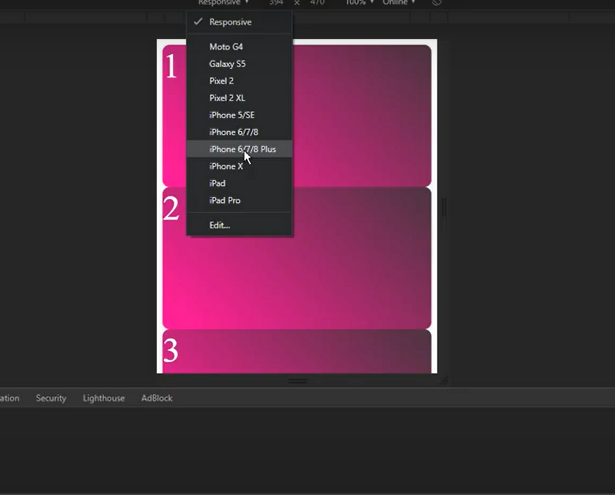


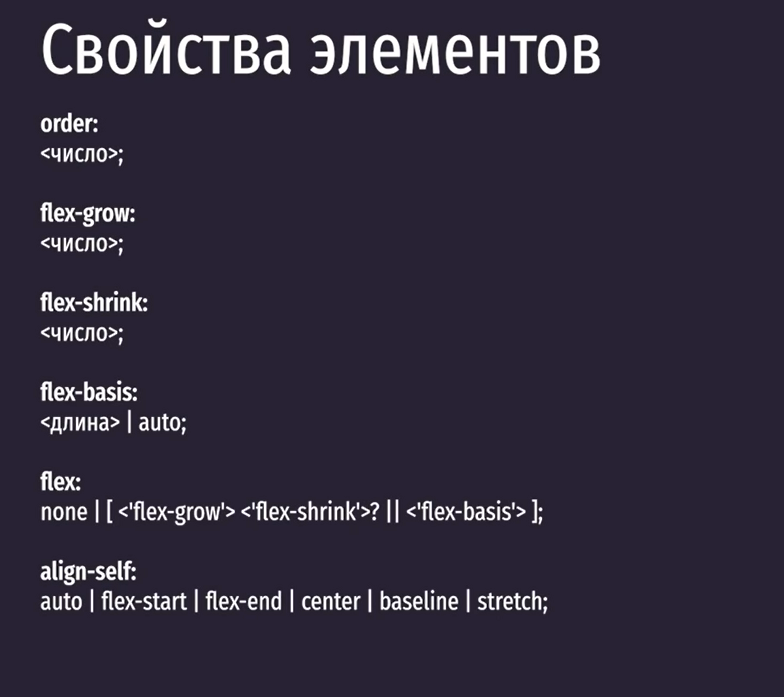
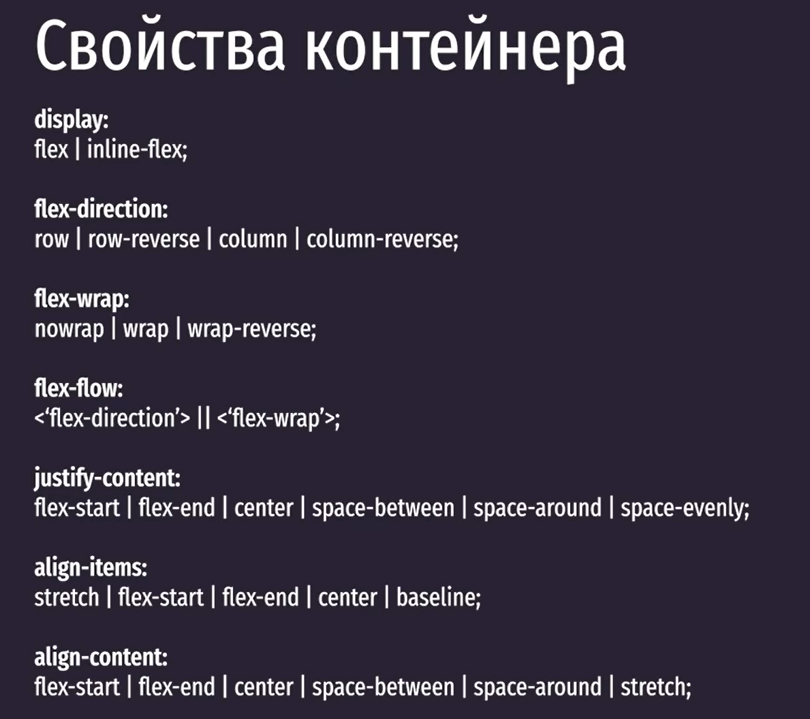
Flex-basis = width

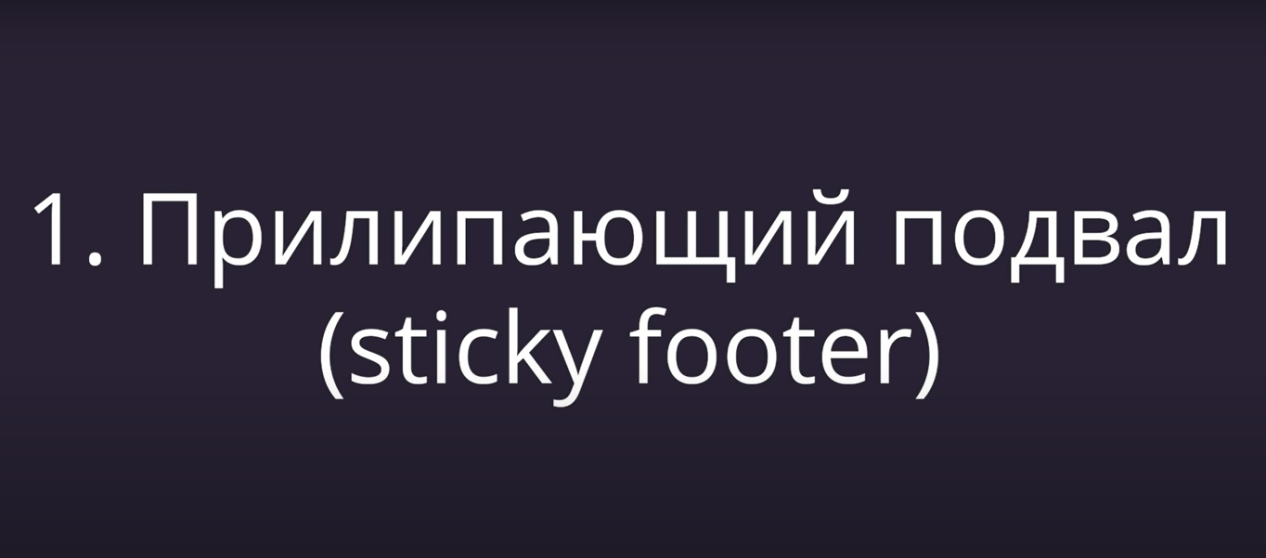


Адаптив с флексами. Прописать в начале:

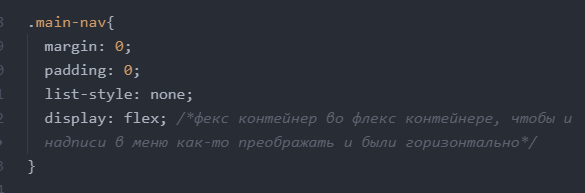








\*С ПОМОЩЬЮ FLEXBOX



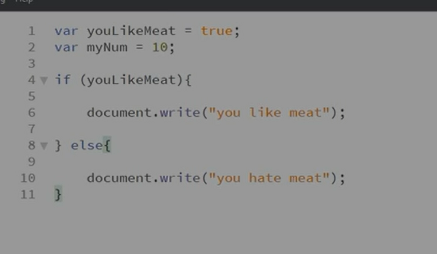
JAVA SCRIPT



console.log(“your age is less than 10”); - вывод в КОНСОЛИ фразы

document.write(“you are now over 10”); - вывод в ДОКУМЕНТЕ фразы

IF statement



ELSE IF statement   


WHILE loop



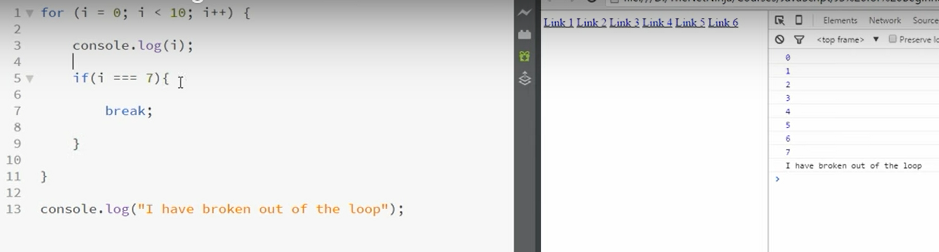
FOR loop



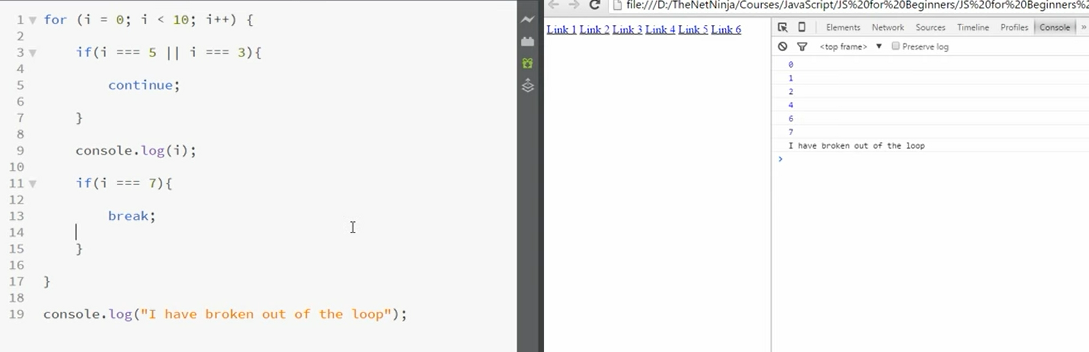


BREAK & Continue

break - при выполнении условия if выходит из цикла for



continue - пропусти выполнение дальнейшего кода и ПРОДОЛЖИ дальше выполнять цикл (не выведет 3 и 5)

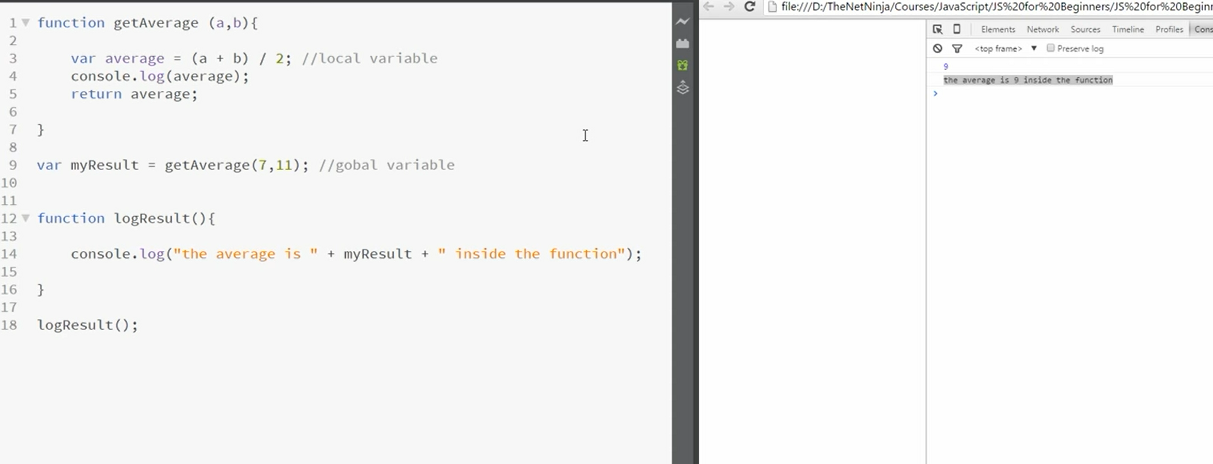


FUNCTIONS



Глобальные переменные - используются во всем файле

Локальные - используются только в функции



ОПЕРАЦИИ С ЧИСЛАМИ

Math.\*\*\*

- round(7.8) округляет значение 7.8 = 8

- floor(7.8) округляет в меньшую сторону 7.8 = 7

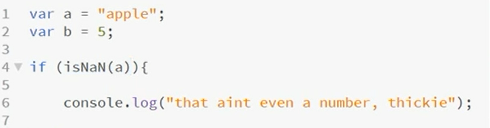
- celi(7.2) округляет в большую сторону 7.2=8

- max(\*,\*,\*) возвращает максимальное число

- PI выводит число pi

NaN - not a number

функция isNaN:

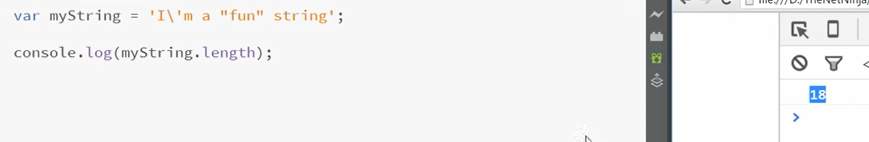
- is (a) NaN?

(в этом случае а - не число, а apple - string)

! - добавляет отрицание, например: a !== b (а не равно b)

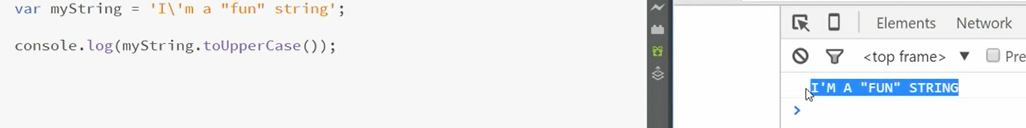


STRING



\ - игнорирует ‘ которая за ней, чтобы не закрывать скобку

.length - получает длину строки



.toUpperCase() - переделывает всю строку в заглавные буквы

.toLowerCase() - наоборот, в прописные буквы





.indexOf() - выводит позицию/номер, где встречается слово в ()

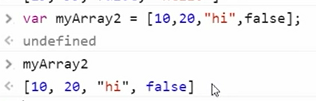
Если нет такого слова - выведет (-1)

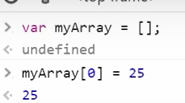


SLICE & SPLIT strings

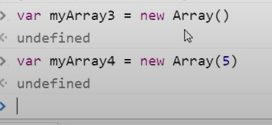
- slice(2,9) делит слово с позиции 2 до позиции 9

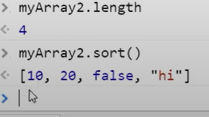
- split(“,”) делит слова когда встречает “,” и это уже как отдельный элемент массива

ARRAY - массив

- создали массив и потом внесли 0-ой элемент или так

- заполнили массив данными разного типа

- сделать рандомный массив

- length выведет длину/кол элемента массива; sort() - автоматически отсортирует массив, reverse()-наоборот

OBJECTS

- myCar - объект, drive()-его метод/функция