

١

نحو اسلوبی = حاوی اسلوبی

document.body.innerHTML = 'Hello';

①

مثلاً 'Hello' عبارت مكتوبة في المتصفح  
هي لفحة في webpage مثل index.html التي نفتحها  
في المتصفح تحت علامة <body>

Type Coercion = automatic type conversion

②

template string

③

'...' = template string or backtick

template string features

④

① interpolation = let us insert a value directly into  
a string

2

`${} = insert value directly into string`

: مقدمة إلى برمجة Python

`Items(2): $2.2.24`

: Concatenation operator

`'Items(' + (1+1) + ')': $ + (2*25+72)/100`

`= 'Items(2): $2.8.24'`

: Interpolation `  
backticks

`'Items(${1+1}): $${{(2*25+72)/100}}`

② multiline String

some  
text

: Enter multiple lines using backslash

3

< 'Some\n text'

→ string as logic  
→ escape character

→ ! to escape character → \n to continue new line

? چیزی کو string کیوں کہا جائے گا ⑤

, , , " " , , , ,

single quote  
double →  
backtick

→ single quote ; اس تھا بکھر کر کوئی کامن نہیں

→ multiline strings ; اس تھا interpolation

→ multi-line template string ; backtick ;

→ head ; ⑥

→ global doble quotes ; اس تھا دوبل کوئی کامن نہیں

→ head ; visible area

٤

: HTML Onload دوبارہ جوا اس سریت Run کئے جائے ⑦

script کے ①

نیچے double quotes میں to onclick attribute ②

جوا اس سریت اجرائیں

<button onclick="alert ('good job');"

End load پھر script کی وجہ سے اس سریت کو اجراء کر دیں۔  
کوئی script میں onclick نہ رکھ دیں۔  
کوئی script کو اجراء کر دیں۔

<!-- This is a Comment --> HTML کا تنبیہ ⑧

/\* This is a Comment. \*/ CSS کا تنبیہ ⑨

<script // This is a Comment.  
  Javascript کا تنبیہ ⑩  
  </script>

5

و جادا اس سریت  
م حارہ multiline

<script>

/\*  
multi  
line  
Comment.

\*/

boolean value (11)

False  $\downarrow$   $\sim$  true صریح نہیں

Console.log(5>6) > False

: تاریخی اس سریت (12)

==== equal to

! == not equal to

== , === تاریخی (13)

not Value جو لیٹرال مکانیکی تاریخی (14)

6

برهان دو دلایل  
کوئی ایسے تو دریغ == ہے

اگر a بھر کر b' type کی ! Value ہے تو ==  
Console.log ( $5 == '5.00'$ ) ;

Result = True

لیکن اس کا بھر کر دیا جائے تو  
! Value ہے اسی == کو پڑھ لے  
جس سارے تین دلائل میں سے ایسے  
True نہیں کیونکہ Number 5 ! Value ہے اور

وہ سارے طلاق سے مادہ کرنے کے لئے  
operator == ہے (پل Triple equal )

Console.log ( $5 == '5.00'$ ) ;

Result = False

7

order of operations 14

- 1) (...)
- 2) \* /
- 3) + -
- 4) Comparison operators
- 5) Logical operators

Console.log ( $3 > 5 - 5$ );  $\xrightarrow{\text{ans}}$  True

Math.random()

(15)

Generates a random number between 0, 1

Falsy , Truthy

(16)

The number 5 behaves just like true. so we call  
This value truthy value

Falsy values = false, 0, "", NaN, Undefined, null

8

any value not on this list is truthy

else

Cart Quantity = 5

CartQuantity > 0  $\Rightarrow$  True

CartQuantity  $\Rightarrow$  truthy

: if true if (CartQuantity > 0) else false  
if (CartQuantity) { }

instead of doing comparison we can use truthy and  
false values as shortcuts

NaN = Not a Number

'text' / 5  $\Rightarrow$  we get NaN

?

Undefined = Something doesn't have a value.

let variable1;  $\leftarrow$  ~~it's~~

~~so if you try to do something with undefined, it will give you an error~~

Const variable1 = ~~let~~  $\leftarrow$  ~~it's~~ value

~~so you can't do anything with undefined~~

Const Variable1 = undefined;

Guard operator  $(!)$

False && Value2

it is impossible for both sides to be truthy. so we don't need to check right side. so The and operator stops early. This is called: short -

circuit Evaluation

It's False && console.log('hello');  $\leftarrow$  falsy

we can use the value on the left to  $\text{block}$  or  $\text{guard}$  the code on the right.

10

## (11) default operator (18)

Value1 || Value2

checks if at least 1 side is truthy

in This Situation we already know the result will be True.

true || ~~Value2~~

stops early (short circuits)

Const currency = 'EUR' || 'USD';  $\xrightarrow{\text{result}}$  EUR

Const currency = undefined || 'USD';  $\xrightarrow{\text{?}}$  USD

as you can see we can use OR operator to set a default value.

II

```
function1 {  
    // a  
    return;  
}
```

This will return Undefined.

when we use a return statement, it ends the function immediately.

```
return;  
console.log('after');
```

:d 16

The code after return doesn't run because once we return, the code returns back to where we call the function.

١٢

و سُمِّيَ Function کی جگہ کو مُوثر کیوں کریں؟  
و اس بُعدِ سُمِّی کی وجہ از Function let ComputerMove = "j".  
کے سُمِّی کی وجہ از Global نردر دیتوانیم از آنچھے جاہے، پر تم استفادہ نہیں جوں  
اُمر داخل دلیل scope ایچ راتم بینے پڑوں داخل Function کرائیں  
و خارج از scope ایچ سُمِّی وجوہ نہیں۔ حرجاً اس درستی کے سبب  
و سُمِّی کو نہیں عمل پڑوں  
button  
<button onclick = "  
    pickComputerMove()  
    ≡ دیتے مواد  
        → دلیل

و سُمِّی کی وجہ از اس سُمِّی کی وجہ از ComputerMove کی وجہ از مدار رفیعی مانند وجوہ  
و اس سُمِّی کی وجہ از onclick کی وجہ از global.  
بعد از کارکردگی : return

13

توسط return میں کوئی Value رکھنے کا معنی نہیں  
Function کا Return Value with ComputerMove  
: return ComputerMove;

حالات لئے یعنی if دادی ComputerMove with کراچی، ریاستہائے  
متحدہ ایسا کام کیا جائے کہ undefined میں onclick کا سود حاصل کرے  
ایسے with وجود ندارد.

وہی onclick  $\rightarrow$  ComputerMove with کیا کام کرے  
؛ پوچھیں Function کی مدد و مبارکہ.

const ComputerMove = pickComputerMove();

یعنی with کا معنی if دادی ComputerMove ! ComputerMove میں  
وہیں طرف میں کام کرے.

We can save the return value in the variable.

14

## Best Practice:

Keep variables inside a scope (if we can)

بارايت:

Return = gets a value out of a function.

Parameter = puts a value into a function.

بارايت فقط مع دو رطبه، و بتاريخ ما بارايت،  
پارامېتر مع مجموعه اسکوپ، ایندیکت، ایندیکت،  
وچونکي مع curly bracket دخل،  
اینداشته باش.

Default value for a parameter:

Function calculateTax (cost, taxPercent = 0.1) { }

: object

An object is another type of value.

15

another way to accessing value inside objects:

Product['name']

← bracket notation

dot notation  $\rightarrow$  مثلاً bracket notation وفقاً لـ

که وقت وقوع کی از درستگی است

آخرین وقت خطا خواهد بود dot notation

و همچنان است از دو نوع است

Console.log (Product['delivery-time'])

حروف برازشی  $\rightarrow$  فرمانات ایجاد

حروف سیوی (MATH)  $= \text{the result}$  انجام بدند

برای ایجاد و نمایش مقدار داخل bracket  $\rightarrow$  مقدار و نتیجه

negative log for math calculation  $\rightarrow$  string

Product['delivery' + 'time']

نتیجه

16

هذا يعاد أحياناً string ماسح برأسه ما تنتبه باختصار  
أو بمعنى optional حفظ دبور.

```
Const Product = {  
    name: 'shirt',  
    'delivery-time': '1 day'  
};
```

: Optional Feature

Inside an object we can save any type of value.

```
Const Product = {  
    name: 'shirt',  
    'delivery-time': '1 day',  
    rating: {  
        stars: 4.5,  
        count: 87  
    };
```

→ Nested objects

## IF

How to access nested object property:

console.log ( product.rating.count ),

87 ← result

لـ func Function دلـ count دلـ rating دلـ product دلـ product Type  
a Function is another Type of Value.

const product = {

name: 'shirt',

'delivery-time': '1 day',

rating = {

stars: 4.5,

Count: 87

},

fun: function Function() {

    console.log ('Function inside object');

}

},

18

یا dot notation: کو اسی function کے ساتھ اسے دیکھ سکتے ہیں۔ پر bracket notation

Product. Fun();

Result: function inside object

وہی میں اسی right bracket کو save کر دیا چکی ہے، Function کو

Console.log(); کو save کر دیا چکا ہے

Console → کوئی log کی وجہ سے کوئی دلیل نہیں کہ یہ کیونکہ Console

Console.log (type of console); result: object

Console.log (type of console.log); result: function

میں دیکھ سکتے ہیں Math.log(). کوئی دلیل نہیں کہ یہ کیونکہ اسی

: More Built-in objects

. وہ کوئی built-in ہے Console, Math

19

JSON

introduction / built-in

localStorage

JSON = help us work with JSON

JSON = Javascript object notation.

```
{ "name": "shirt",  
  "delivery-time": "1 day",  
  "rating": {  
    "stars": 4.5,  
    "count": 87  
  } }
```

JSON object - جملة JSON

اسن

ما و اربع بآتیت طا

جاوا اسکرپت = ابراهام

کوئی JSON نہ رہے

: لاسترینگ طا

۔ ملائیں double quotes

کوئی دو سیم اسکرپت خیلے و کوئی JSON با طبوا اسکرپت آتیت

. اس اسکرپت فونکشن ہے

20

جوابات اسکارف ایجاد آبیت جاوا اسکریپت:  
 این آبیت جاوا اسکریپت مخفف در جاوا اسکریپت اسکارف هر دو  
 این آبیت جاوا اسکریپت کو سطح کم را با قابل درست نهاده باز این حلقه

We use JSON:

- when we send data between computers.
- when we store data.

جوابات اسکریپت تبدیل کردن به JSON object

Convert:

JavaScript object → JSON

برای **stringify** آبیت در جاوا اسکریپت است که سطح کم **JSON** را تبدیل می کند.  
 این آبیت جاوا اسکریپت را به JSON تبدیل می کند.

console.log ( JSON.stringify ( product ) );

جوابات

{ "name": "shirt",

21

```
"delivery-time": "1 day",  
"rating": {  
    "Stars": 4.5,  
    "Count": 87}  
}
```

توبه نماین ~ مادر این دست حاجی اسپریت که تابع فون زیرا در جوری  
برای سایر سایر جیز JSON بگردانید

console.log(typeof JSON.stringify(product))

Result - String

Now This JSON string is ready ~~for~~ to be send  
another computer or to be saved in storage.

? put this Javascript object in JSON ~~in~~ ~~in~~

22

Convert:

JSON → Javascript object  
JSON.parse()

localStorage → Meant built-in object

Save values more permanently.

ls Variable ~ لکیوڈل سینٹر ذخیرہ درجہ . نسلیں اسے میں ادا کر سکتے ہیں۔ اسے صفت، جاری و حور طازہ، اگر ساصھا، افزیں کرنے والے خواہیں۔  
کیم و ساصھا، اب بیسے یعنی delete ls Variable پر میں اسے خواہیں۔ اسے صفت پا کر نہ کرو  
رسانے صفت پا کر نہ کرو

localStorage.setItem()

: prox String تواند ہے اسے دو کر سکتے ہیں  
lsValue ; یا ہے نہیں۔ اسے name دوString ①  
(name کوئی نہیں)، توبہ کوئی نہیں سو کر سکتے ہیں  
66/lsValue اسی Localstorage پر میں اسے ذخیرہ کر سکتے ہیں ②  
از لفی جدا ہے سو کر۔

23

LocalStorage only supports strings.

localStorage.setItem('message', 'hello');

console.log(localStorage.getItem('message'));

Result = hello

أبراجوا ملحوظاً دخراً  
localStorage

localStorage.setItem("score", JSON.stringify(score));

حوظ سعياً دخراً آتىت score دخراً دخراً بورج موسط

localStorage

: script در score لغرسن عنده بـ 0

const score = JSON.parse(localStorage.getItem('score'));

<script> const score = {  
 wins: 0,  
 losses: 0,  
 ties: ... } ;

24

localStorage; ! score, page عنوان load پیشوا  
. فی از عنوان page عنوان گواهند فک  
-

: Reset Score کتب

<button onclick = "

score.wins = 0;

score.losses = 0;

score.ties = 0;

localStorage.removeItem('score');

"> Reset score </button>

پیشوا، بازی، فریز، انتصار، w, Reset score لود خواهد

TypeError خطای. ترب خواهد

Can not Read properties of Null.

<div> localStorage; ! score لود میتواند

remove & Reset score

وچورنیا localstorage عنوان بازی وچورنیا localstorage

25

? Default value will do, now null gives us

? to local storage

<script>

let score = JSON.parse(localStorage.getItem('score'));

→ Const →  
→ let →  
if (score == null) {  
 score = {  
 wins: 0,  
 losses: 0,  
 ties: 0,  
 } } ;

→ if (!score) { → key shortcut : =)

instead of checking if a value is equal to null we  
can just use the **not** operator as a shortcut.  
Now also from the Boolean lesson we have an  
even shorter for doing this:

26

notice here that this code basically sets a default value if the score doesn't exist. In the boolean lesson we learned a technique called The default operator which is exactly what we need here:

<script>

```
let score = JSON.parse(localStorage.getItem('score')) || {  
    wins: 0,  
    losses: 0,  
    ties: 0  
};
```

If the left side is Falsy or in this case The score is null Then we will use the right side as a default value.

: again Falsy value

<= null, undefined

They both represent that something doesn't have a value

٢٧

## تَعْلِمَةٌ بِسَعَى الْعَوْنَى وَالْعَوْنَى

- null اشارةٌ فارغةٌ لا يُوجَدُ مُحْكَمٌ فِي مَفْهُومِ empty.
  - undefined مُوْجَدٌ خَارِجًا.
- ما يَحْتَاجُ إِلَيْهِ Value وَmethods, properties لِيُؤْخَذَ مِنْهُ.
- other values also have properties and methods.

console.log ('hello'.length) → 5

⇒ ('hello'.toUppercase()) → HELLO

سُرْطَانِيَّةٌ مُحْكَمٌ string فِي سَيِّفِيَّةٍ Value مُحْكَمٌ فِي مَفْهُومِ string.

وَهُوَ مُحْكَمٌ فِي string (log). اِسْعَادُكَ مُحْكَمٌ فِي string (toUppercase).

مُعْصَمُكَ مُحْكَمٌ فِي string (length).

جَاءَ الْأَسْرِيَّةُ Auto Boxing فِي Feature حَارِزٍ.

جَاءَ الْأَسْرِيَّةُ بِطُورِ الْأَتُومَاتِيَّةِ اِسْعَادُكَ String رَادِيَّةً آبَاتِ دَرِيَّةً.

وَاسِعُكَ آبَاتِ دَرِيَّةً Wrap.

28

wrapped around a special object

→ AutoBoxing: if "hello", this string type  
Undefined, null ! but int, boolean, Number etc to  
 wrap

∴ Objects are Reference

```
const object1 = {
    message: 'hello'
};
```

Computer's Memory

const object1 = reference →

```
{ message: 'hello'
}
```

значит объекты на языке JavaScript  
 value в языке Java иначе не называют  
 value в языке JavaScript

значит

29

وقتی طبق آبیجت را بگاره می‌نامیم Value واقعی یک جای در حافظه کامپیوچر  
 ابزاره سود داریم Variable حافظه یک رفرانس دارد به اسناده مالت  
 یعنی مطابع (location) در حافظه کامپیوچر. مدل یک shortCut در کامپیوچر  
 می‌باشد. اسناده مالت به جای نام فایل وجود ندارد. راهی  
 وجود ندارد به این رفرانس را مستعاره می‌نامند. فقط بداین‌دست به این رفرانس  
 است را بگذشت یک جای تغییر در حافظه کامپیوچر که برقرار می‌گردد.

Const object2 = object1

این مقدار واقع یکی از آبیجت ابزاره مالتند. فقط یکی از رفرانس ابزاره مالتند  
 که با آن object0 اسناده مالتند این مطابع را صورت می‌گیرد و فن نیست  
 همان‌یک آبیجت هنوز بزرگ داریده و این خوب نیست لکن آبیجت را بخوبی  
 نمی‌گذرد به جای آن جای اسناده مالت را کمی عالی‌تر نماید به سیار سریع تر  
 می‌باشد. به این‌گونه Copy by reference می‌گویند.

۳۰

## رَسَّاَرَهَايِّ حَالِبِ رَهْرَشْ :

Const object 1 = {  
 message : 'hello'.  
 };

(۱)

Const object 2 = object 1;

وَبَلَغَ → object1.message = 'Good job'.

ما پُرِرَى message آبَيَت ۱ رَتَهْرَشْ حَارِمْ . اولَ تَوْجَهْ لَنْدَهْ مَاهَوْنَتْمَهْ  
 كَوْرَهْ message مَرْعَطْ بَهْ آبَيَت ۱ رَتَهْرَشْ دَهْمْ بَارِعْ بَهْ آبَيَت ۱  
 كَوْرَهْ اَسْتْ . بَهْ اَسْ دَلَلْ تَوْانَتْمَهْ بَهْ object1 فَقَطْ مَهْمَالْ فَرْتَسْ  
 (آمِسْ) مَهْمَابَهْ بَهْ آبَيَت اَسْ دَهْنَتْ . بَسْ مَاهَانْ باَشْ هَفَاهَهْ  
 باَآبَيَت هَهْ طَارَهْ لَنْدَهْ وَصِنْدَهْ مَاهْ Const اَسْهَارَهْ بَلَنْمَهْ هَفَنْدَهْ مَاهَوْنَتْمَهْ  
 دَاهْلَهْ آبَيَت رَاتَهْرَشْ بَهْهَمْ .

Const object2 = object1 رَهْرَشْ آبَيَت رَاهْ دَاهْعَهْ تَقْيِمْ (۲)

بَهْ هَنَرَهْ ما رِيزْنَسْ آبَيَت رَاهْ دَاهْ تَقْيِمْ رَاهْ دَاهْ . بَهْ بَارِعْ object2  
 هَاهْ رِيزْنَسْيَهْ رَاهْ دَاهْ object1 حَارِمْ . دَهْ دَوْهَهْ آبَيَت اَسْ دَهْهَهْ دَاهْهَهْ

31

console.log (object2) :

بنا بر این امر

⇒ { message : 'Good job!' }

توجه کنید برای آبیت! در چرخه message تغیر در دو وحدت ما در چرخه  
آبیت! را تغیر خارج بودم.

we can't compare objects directly ③

const object3 = {

message: 'Good job'

console.log (object3 == object1) → False

خوب این نه رفته ای باهم نیست. و فرسته آبیت! را آبیت!

باهم تفاوت است وحده همچنانست این آبیت! می توانم هم باشد.

سپرینگ نه رفته ای باهم نیست و نیز همداخل آبیت!

console.log (object2 == object1)

True نتیجه (چون فرسته مساوی کارند)

سے یاد کرنا ہے کہ وہی ایک Variable پر کام کرنے والے  
value کو تکمیل کرنے کا لفظ رکارڈ کیا گی اور  
کام کرنے کا لفظ انواع کیا گی اور اس کا  
راحت آبیدت ہے ।

### shortcut for objects (4)

: کام کرنے والے Variable پر اگر

```
const object1 = {  
    message: 'Good job!',  
}
```

```
const message = object1.message;
```

; shortcut

```
const { message } = object1;
```

short  
destructuring

این destructuring کی وجہ سے property message کو  
object ہے تو destructuring کی وجہ سے message

33

## Shorthand Property ⑤

```
const object1 = {  
    message: message  
};
```

shortcut → const object1 = {  
 message  
};

## shorthand method ⑥

```
const object1 = {  
    message,  
    method: function function1() {  
        console.log('method');  
    }  
};
```

console.log(object1); → {message: 'Good job!', method:  
f}

34 object1.method();  
~~console.log(method);~~ → method

Short hand:

```
const object1 = {  
    message,  
    method() {  
        console.log('method');  
    }  
};
```

new function style

8 Curly

35

Document object model = DOM

ایجاد کردن ابیانت های جاوا اسکرپت می باشد  
ایجاد کردن DOM

document.body.innerHTML = 'hello';

ایجاد

گیری

ایجاد خاص می باشد . وقت طارع کار را تغییر  
و document object را تغییر می دهد و نه این طبع  
برای مرورگر لینک شود .

document.title = 'Good job';

ایجاد نام شما را web page و document object می کنید  
می توان صفحه را تغییر می داد

web page براي document object نمی شود  
لما می تواند و می تواند اینجا براي web page براي می تواند  
آن DOM می کند .

بنابراین document یک DOM می باشد  
ایجاد کردن

36

لیکن `title`, `body` میں اور `document` object پر `console.log()` dot notation کا `method` ہے

### Properties and Methods:

① `document.title`

② `document.body`

`console.log(document.body);`

`<body> .. </body>`

inside the console it will tell us that  
`document.body` is the `body` HTML Element.

We can have HTML Element inside  
JavaScript.

37

DOM = The DOM combines Javascript and HTML together.

ادخل جاواسکریپت داخل HTML Element باعث این می‌شود که کدام ایتمی این ایتم است؟  
برای این کار HTML Element را داخل جاواسکریپت داخل داشت و Convert شود.

Console.log(typeof document.body)

result: object

این پاسخ به این است که body HTML نباید ایتم است و فقط داخل جاواسکریپت قرار نماید. سریع اینکه این دلیل که در کوچک رسیده می‌باشد.  
body داخل HTML نباید innerHTML بگیرد بلکه این ایتم او را باشد و فقط می‌تواند body داخل HTML را تغییر بخورد.

HTML خالی نباید ایتم innerHTML یکی Feature text یکی new HTML باشد body داخل

38

```
document.body.innerHTML = '<button>Good job
</button>;'
```

result = Good Job

جواب مطابق با Function Full Control نیز باشد.  
ما در داخل دستور Cool Function نیز میتوانیم return کرد و این را با دستور break خارج کنیم.  
return نمیتوانیم.

document.querySelector()

برای اینجا اگر هر چیزی را در صفحه ما فرمودیم  
با document و داخل جوا اسکریپت خواهد بود.

lets us get any element from the Page and  
put it inside Javascript.

جیسا کہ document.querySelector('button');

? String بخواهد ما بخواهیم داشت String کاربردیم ایکو  
جوا اسکریپت چنانچه کام ایکو را از صفحه تخلیر. وقتی ما

٣٩

=  
 نہیں کہ اس فونکشن کو button کے لئے اپنے ایسا مخفف کرنے کا انتہا میں اور اس کے لئے اس کا اس سریکے کو رکھنا پڑتا ہے۔

```
<body>
    <button> hello </button>
```

```
<script>
    console.log(document.querySelector('button').
        innerHTML);
```

</script>

نتیجہ → hello

وہی اسے HTML داخل کا اس سریکے کو رکھنے کے لئے آجیت کر جائے۔ اس سریکے کے ساتھ button کو ہم کی آجیت کر کر کے methods و Property کو اسے دار کر کے کوئی ایجاد کر سکتے ہیں۔ innerHTML یعنی HTML کے Property کے لئے واقعی کام ہے۔ اس سے ماں ایجاد کردہ HTML کو اسے دار کر کے ایجاد کر سکتے ہیں۔

fr

document.querySelector('button').innerHTML = "Changed";

Changed

این کد نتیجه حاصل از button تغیر می‌کند

پس از این class پس از attribute ویژگی button برای این

class attribute = adds a label

برای button و جایی که اسکرپت جای اسکرپت را داشته باشد button دارد

و متغیر Variable دارای مقدار objects are Values

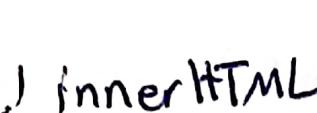
const buttonElement = document.querySelector('button');

'js-button');

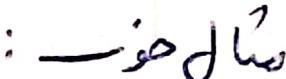
console.log(buttonElement);

نتیجه `<button class="js-button"> second button </button>`

q1

✓ `innerHTML` or `innerText`; `innerHTML`  `block-level = block`  
`innerText` will give us the text inside this Element but without all these spaces around.

`<P>` is a block element, takes up an entire line by itself.

: 

Amazon shipping calculator

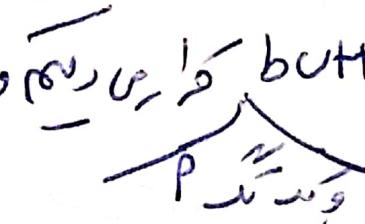
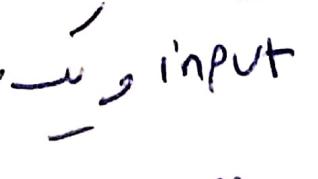
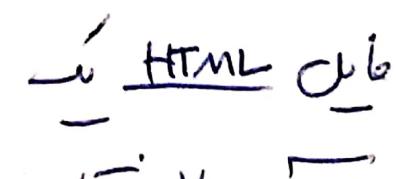
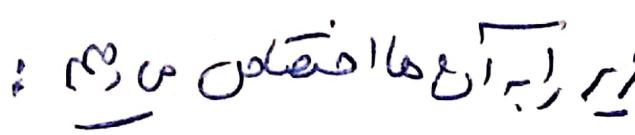
orders under \$40 = +\$10 shipping

orders over \$40 = FREE shipping

25

calculate

\$35

Showing  button,  input  `HTML` `object` `form`  
: 

42

```
<input class="js-input" placeholder="Cost of order">  
<button> calculate </button>  
<p class="js-result"></p>
```

: ممکن است `<script>` بین

و `button` بین `input` و `p` باشد  
نحو: وقتی `button` بین `input` و `p` باشد  
هر دو `input` و `p` هر دو `$40` بوده اند. `input` و `p` همچنانه `button` بین

میتوانند توسط پردازش `innerHTML` داخل `input`

را پردازند. حالا آسیب دیده باشیم. و گذشت `input` بین

`input` بین `button` و `p` باشد. `input` داخل `input` باشد

: `input` با `value` میتواند در `HTML` داشت

`<script>`

```
function calculateTotal() {
```

43

```
Gst = inputElement = document.querySelector('.js-input');
let cost = inputElement.value;
if (cost < 40) {
    cost = cost + 10;
}
document.querySelector('.js-result').innerHTML =
`$ ${cost}`;
```

پس لی! Calculate اواردیم و 20 جو Textbox میں پڑھا  
پر جو اگر وہ اسی کو 20 کے طور پر ابھر جائے تو \$20.00 ملے  
console.log(typeof Gst); سے → String  
کو String to Cost بنا جائے  
whenever we get a value from The DOM,

The value will be String.

پر جو Number یا String ہے اس کو Cost میں

٤٤

js built-in Function / حجا اس سریف جر اس جار کرد

: پرسکاٹ است گھیر زیر اکون صفحہ کل بدهیں : جو Number()

let cost = Number (InputElement.value);

وہی کار خواهد کرد

عند دفعہ جاگے قارئ میں بعد از پر درج Textbox : میں حفاظت میں سے اسے Enter کا دار درج کرنے کے نتیجے انعام پڑے : Calculate کر کر

clicks, keydowns  $\Rightarrow$  event

onclick, onkeydown  $\Rightarrow$  event Listener

ایسے event کو ایسا کہا جائے کہ اسے event

کا اجراء کرنے کے لئے کہا جائے کہ اسے event

پس اک توڑتے ایسے لونے است : وہی مالک کہ اسے event

Calculate button کا ایسا کہا جائے کہ اسے event

اکام رکھو

٤٥

خُلُونَةِ مُتَوَّبَةِ سُلْطُونَةِ نَكَارِيَرَ لَامَ سَعِيَ رَافِضَارَ رَادَهَا سَتَ?

مُنْجِمِيَّةِ **event** مُنْجِيَّةِ **keydown** دَاخِلَ آَبِيجَتِ وَرِجَوْنِيْمَ آَبِيجَتِ حَوْسَطِ جَهَادِ اسْتِرِيُّوتِ بِرِجَاتِ اِمَاهَهَا سَرِهَا سَتَ وَادِعَنِيْلِيْمَ اِمَالِيَّةِ دِرِمُورِ

Every event listener gets an even object

<input class="js-input" onkeydown="console.log(event);"

حَانَّا سَرِيَّدَرَ كَرَمَهَ اِرْصَنَهَ سَعِيَ دَاخِلَ اِنْتِرِنَتِرِنَتِرِنَل  
كَلِهَ مَاهِهَا سَتَ . دَلِيْلَهَ Key : "2" (صَعِيَّهَ 2) اِفْقَارِ خَادِمَه  
وَ حَانَّا سَرِيَّدَرَزِيَّهَ قَرِيبَهِمْ دَهَرَ...  
وَ اِنْتِرِنَتِرِنَتِرِنَل Enter اِلْهَادِرِنَتِرِنَل سَهَّلَ ٢٥ اِفْلَادِرِنَتِرِنَل سَهَّلَ

2  
5

Enter

: ۱۴۷

: ۲۶۳, ۶۷۸

<input onkeydown="  
if (event.key == 'Enter') {  
 calculateTotal();  
}"

46

Web Events

## Event Listeners

on click = click

on keydown = key press

on scroll = scrolling

on mouseenter = hovering over

on mouseleave = stop hovering over

— and many more

Event Listener Job Description  
: many ways

```
function handleCostKeydown(event){  
    if (event.key === 'Enter') {  
        calculateTotal();  
    } } }
```

٩٧

onkeydown = "handleCastKeydown(event)"

فیکل نیز بارگیری کردن و داخل event میتوانیم اینجا اینجا

variable نام داده و اینجا

باید در اینجا عدد را معرفی کرد

if a string only contains a number, and we - \* / (1)

⇒ it will be converted to number.

console.log ('25' - 5);  $\xrightarrow{\text{تبدیل}} 20$

± تبدیل

console.log ('25' + 5);  $\xrightarrow{\text{تبدیل}} 255$

so don't do MATH using strings.

window یک built-in object است (2)

(represent) کردن یک web page یعنی document را میتواند

و یعنی represent یک Browser یعنی window است لذا

48

عمر میزی داخل browser داں جو درود بردار در حقیقت داخل browser میزی داخل browser ہے۔ اما جو میزی داخل browser کی بیانات

① web page

② Console

③ pop-ups

خوب حکم document داخل browser بنایا رفع آبیت web page خوب حکم document قرار دارد window.document آبیت window

This will give us the same document object.

کسکو میں داخل browser میں آبیت window.console کسکو میں داخل browser میں آبیت window.alert

وکارہ هیچ یہ اندر رنگا ہے کیونکہ نہیں ہمارے window.pop-ups میں آبیت window.alert

وکارہ هیچ یہ اندر رنگا ہے کیونکہ نہیں ہمارے window.pop-ups میں آبیت window.alert

لیٹنگ درس ۶

q9

①

.classList = gives us control of the class attribute

بالنسبة لـ button يمكننا إضافة وremoval

one solution

when we click the button,

1- Add a class

2- This class will have different styles

classList is an object.  $\Rightarrow$  جسم

.add()  $\rightarrow$  add a class to an Element

<script>

```
const buttonElement = document.querySelector('.js-
```

```
-subscribe-button');
```

```
buttonElement.classList.add('is_subscribed');
```

إذاً is\_subscribed will be added to button once it's possible  
to add CSS rule to it in console

5p

:js.remove! button element is not subscribed ②  
buttonElement.classList.remove('is-subscribed');

body uses HTML new text for style use ③  
body { font-family: Arial; }

This will style all the text inside the body and change the font to Arial. and a good practice is to write any styles for the body at the top.

so if we need to, we can override it later.

Space on the inside = padding ④

Space on the outside = margin

padding: 10px 5px;

⑤

## 51

The first value is the vertical padding so the top and bottom, the second value is the horizontal padding or left and right.

`` ⑥  
doesn't require a closing Tag. (void Element)  
`src = source`

Tells the computer which file to load

`<link rel="stylesheet" href="style.css">` ⑦  
in header

rel tells a computer what type of file we are loading, and we set this attribute to **stylesheet**

**href** = which file to load

`<link rel="stylesheet" href="style.css">`  
in header

52

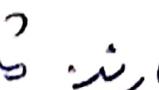
Inside an array we can put any Type of value

[1, 'hello', true, {name: 'socks'}, [1, 2]]

An array is also a value.

Console.log(typeof [1, 2]);  object

an array is an object (it's a special type of object)

myarray.length  5

.push() Adds a value to the end of the array

.splice() Removes a value from an array.

myarray.splice(0, 1);

1. index we want to remove

2. number of values to remove

53  
Loops create a scope. any variable that we create between this curly brackets will only exist inside The curly brackets :

```
while (i<=5) {  
    console.log(i);  
    i++;  
}
```

<div>  $\Rightarrow$  doesn't have any default styling.

generating HTML:

```
let todoListHTML = "";  
for (let i=0; i< myArray.length; i++) {  
    const todo = myArray[i];  
    const html = `<p> ${todo} </p>`;  
    todoListHTML += html;  
}
```

document.querySelector('.js-array').innerHTML = todoListHTML;

Cool! <div>  $\rightarrow$  your own list!

54

: قم بـ object و اخراج محتوى المثلث

const myArray = [

```

    {name: 'make dinner',
     dueDate: '2022-12-22'
    },
    {name: 'Wash dishes',
     dueDate: '2022-12-23'
    }
]
```

: قم بـ loop over myArray

let todoListHTML = '';

for (let i = 0; i < myArray.length; i++) {

const todoObject = myArray[i];

const name = todoObject.name;

const dueDate = todoObject.dueDate;

then generate HTML like this

Shortcut

```

const {name, dueDate} = todoObject;

```

55

```
const html = `<p>${name} ${dueDate}</button>
onclick="myArray.splice(${i},1);">Delete
</button><p>`;
```

الخطوة الثالثة: دобавة خاتمة call لـ onclick لـ delete الشفاعة  
الخطوة الرابعة: دобавة قرار زر اسفل مدون  
todolist.innerHTML += html;

```
}
```

```
document.querySelector('.js-array').innerHTML=
```

```
todoList.innerHTML;
```

الخطوة الخامسة: دобавة push لـ todoList

```
myArray.push(
```

```
{ name: name,
  dueDate: dueDate }
```

```
)
```

shortcut

```
myArray.push(
  {name, dueDate}
);
```

56

```
todo-grid {  
    display: grid;  
    grid-template-columns: 1fr 1fr 1fr;  
}  
How many columns are in our grid
```

Arrays are references. we learned before objects are references and arrays work the same way

```
const array1 = [1, 2, 3];
```

Variable array1 doesn't actually contain the value 1, 2, 3. This array is saved somewhere in the computer memory and variable array1 only contains a reference that points to where this array is in memory.

5f

If we create a copy of this array:

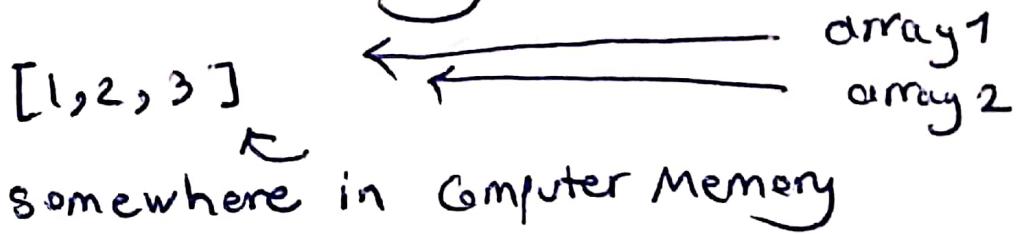
```
const array2 = array1;
```

array2 doesn't contain a copy of the values.

it's actually just contains a copy of the reference.

So, now both of these variables are pointing

To the same array.



```
array2.push(4);
```

```
console.log(array1);
```

```
console.log(array2);
```

- ▶ [1, 2, 3, 4]      ← Result
- ▶ [1, 2, 3, 4]

58

To avoid this behavior we can create a copy of the values in the array using a method called slice.

```
const array1 = [1, 2, 3];
```

```
const array2 = array1.slice();
```

```
array2.push(4);
```

```
console.log(array1);
```

```
console.log(array2);
```

Output

▶ [1, 2, 3]

▶ [1, 2, 3, 4],

Destructing

← Shortcut

```
const array1 = [1, 2, 3];
```

Get the first and second value of this array

59

Const [FirstValue, SecondValue] = [1, 2, 3];

: Cum laude shortcut

Const FirstValue = array<sup>1</sup>[0];

Const SecondValue = array<sup>1</sup>[1];

Continue → skip 1 iteration

return → is another way of breaking out of  
a loop when it's inside a function.

11 ways

functions is also values. we can save a  
function inside a variable.

Also : const Function1 = function greeting() {  
 console.log('Hello');  
};

console.log(type of function1) → Function

60

we can also run the function inside this variable.

function1();

new → hello

Anonymous Function (Function without a name)

```
const function1 = function() {  
    console.log('hello');  
};
```

Advantages → ① easier to read

② Hoisting → we can call this function before we create it in the code.

Hoisting doesn't work when we create our function and save it in a variable.

∴ Functions are value in Javascript.

61

so, what else can we do with a value?

Save Value in an object

```
const object1 = {  
    num: 2,  
    fun: function greeting() {  
        console.log('hello');  
    }  
};
```

```
object1.fun();
```

نحو hello

greeting نحن object1.fun() با مقدار 2 مطابق مسماً num  
هذا Function بالرغم من أننا

```
fun: function() { console.log('hello'); }
```

62

What else can we do with a variable:

Pass a value into a Function.

```
function run (param){
```

```
    param();
```

```
}
```

```
run(function (){
```

```
    console.log('hello');
```

```
});
```

new → hello

Because Param Contains a Function so we can call it `param()`,

passing function → returning function  
as param' callback

62

→ Callback

```
run(function() {  
    console.log('Hello');  
});
```

so anything we can do with a value we can do with a function.

Practical Example (using functions as a value)

① setTimeout() allows us to run a function in the future

setTimeout Function ex:

① A function we want to run in the future

② How long to wait before running this function

64

```
setTimeout(function () {  
    console.log('timeout')  
}, 3000);
```

عندما يُ执ّل القيمة في المقدمة، function يُ执ّل جانباً (0)  
• عندما يُ执ّل القيمة في المقدمة، function يُ执ّل

After 3 seconds → timeout

## Asynchronous Code

```
setTimeout(function () {  
    console.log('timeout')  
}, 3000);
```

```
console.log('next line');
```

65

First display → Next line

After 3 seconds → Timeout

So what's happening here?

جی خوبی ہے پر میرے سے یہ  
کہ setTimeout یعنی 3 سویں لمحے کے بعد اسی میں کام کرے گا  
لیکن اس کے بعد اسی میں کام کرنے کے لئے clearTimeout کا دفعہ  
کروں گا جو اس کو بند کر دے گا اور اس کو اسی میں  
، asynchronous code کو یعنی سونپا  
asynchronous code won't wait for a line  
to finish before going to the next line.

Synchronous code will wait for one line

to finish before going to the next line.

66

مهم در راهای خوب است این در واقعیت نیز بودن. در واقع Synchronous

کار می‌کند که Synchronous Function چنین کار می‌کند

SetTimeout ( function () {

    Console.log('time out')

}, 3000 )

اگر می‌خواهیم این اتفاق بینم شود،

که داخل function چنین کار نماید. که داخل function منتظر گم شوند و بعد از 3 ثانیه اینجا

timeout2 به اینجا آمد و بعد از timeout2 اجرای کرد.

برای اینکه asynchroneus یعنی اینجا صورت اینجا است setTimeout()

Advantage:

- it doesn't block our code for 3 seconds.

و این بسیار اهم است زیرا دیگر رایج نیست و وقتی این کامپیوتر را بکسری اجرا نماید. سل سه ثانیه کامپیوتر را بکسری و اینکی.

67

## ② Set Interval

: lesson. 2nd میں کیا ہے

It will keep running the function every 3 seconds.

→ by asynchronous function

: Rock-Paper-Scissors; Auto play کیا

: HTML file html file کیا? <button>

<button class="js-auto-play" onclick="autoPlay()>

> Auto Play </button>

: جس کے سوامی js کیا کرے

function autoPlay() {

    setInterval(function () {

        const playMove = pickComputerMove();

        playGame(playMove)

    }, 1000);

68

اگر بار ایجاد شود تر باخون یعنی ۱ و ۰ کو  
باز می آید این بار ایجاد شد، Auto play نهادم آورده است

`SetInterval()` → returns a number That we  
can use it to stop the interval.

id کو باز ایجاد نمایند و ID کو number باز  
باز نمایند اگرچه لذا سفارت بسیار ممکن است  
که `autoPlay()` باز ایجاد شود intervalId

let isAutoPlaying = false;

let intervalId;

function autoPlay() {

if (!isAutoPlaying) {

intervalId = setInterval(function () {

const playerMove = pickComputerMove();

playGame(playerMove); }, 1000);

isAutoPlaying = true;

} else {

① ۵۰

69

clearInterval (intervalId),

نقطه ②

```

    } isAutoPlaying = false;
    document.querySelector('.js-auto-play').innerHTML
        = "Auto Play";
}

```

حالاً افتراء / رالس سينم طبیعت باخونیک بازی می‌نماید  
 و اگر بعد از مان رکاراف ریدم حالتی که در آن حبیت غولن سُرخ تملخ  
 و بر علص این عقده نهاده شد از Stop و بر علص این عقده نهاده شد از Auto Play از button  
 document.querySelector('.js-auto-play').innerHTML  
 = "STOP" و افتاده شود ① نقطه / در نقطه ② افتاده صوتی  
 و همچنان دنباله ای از Auto Play = بار نقطه ② افتاده صوتی .

نحوه صنیع :

وقتی داشتی isAutoPlaying=false می‌شود متغیر Auto Play را که  
 تابع SetInterval برای اجرای شدن و تابع autoPlay() برای اجرای  
 سمتی شدن و می‌داند به فقط مکانیزم رآیع را اجرای شد چون صورتی داشت  
 استوچه SetInterval تابع **Async** است و تابع کامپونتی می‌باشد آنچه باسیم  
 است و لذا هر اتفاق بخط اجرای شد Synce

٧٤

موديل `queryselector` `setInterval` بـ `براجمـ`

`document.querySelector(...)` `isAutoPlaying=true`, `bio`  
الـ `play`. `setInterval()` `stop` `clearInterval()`  
الـ `stop` `(0, AutoPlay)` `else` `isAutoPlaying=False`  
`innerHTML` `SetInterval()` `stop` `clearInterval()`  
الـ `Auto Play` `:stop` `stop`

• `array` `for` `iterate` `loop` `forEach()`

[ 'make dinner', 'wash dishes', 'watch youtube' ]

- `forEach (function (value) {`

`console.log (value);`

});

make dinner  
wash dishes  
watch youtube

Q2

II

Function را کیہے کہ Function کو سر پاس دار کو  
وہ نہ اسی Function کو سفارت است. مابکر وہ بار اس کو دار کر دیں.  
بلاستر ای Value For Each کو فراہم کر دیں. اسی ایسے  
Value کو کوہاں دیں اور اسی کو کہاں دیں اگر کہ اسی کو دریا اسی  
Value کی قیمتی میں دیں. یعنی اب تک اسی کو اسی کو دیں اسی کو دیں  
وہ Function کا اجرام نہ کرنے والا ہے بلکہ اسی کو دیں  
کہ اسی کو اسی کو دیں اسی کو دیں اسی کو دیں اسی کو دیں  
وہ Wash dishes کو دیں اسی کو دیں اسی کو دیں اسی کو دیں  
وہ وہ کوچھ کا بھائی اسی کو دیں اسی کو دیں اسی کو دیں اسی کو دیں

طے ...

اپنے Function کو کوہاں دیں اسی کو Feature کے ForEach()  
Position کے دیا جائے کہ index کا باہمیہ کیا  
کہ دیا جائے کہ value کو کوہاں دیں اسی کو دیں اسی کو دیں  
وہ دیا جائے کہ value کو کوہاں دیں اسی کو دیں اسی کو دیں

٤٢

• return و break و continue و for each

• break و continue و for each و return

• break و continue و for each و return

: Arrow Function بـ فونکشن کی، Arrow Functions

Const arrowFunction = () => {

    console.log('hello');

};

• برگشتی

• arrowFunction(); برگشتی کی حیثیت

arrowFunction() کی حیثیت

Const regularFunction = function () {

    console.log('hello');

};

Const arrowFunction = () => {

    console.log('hello');

};

arrowFunction();

73

شادت این سی نوچهای نوچهای مبارکه در Function  $\Rightarrow$  اسماهای مبارکه  $\Rightarrow$  arrow func و صنیع arrow function دارد. ( ) ما اختیاری می‌باشد هم‌چنین وقتی ساختاری خاص را بدهیم می‌توانیم این ساختار را در arrow func که فقط کد را در خود دارد  $\Rightarrow$  قرایبیم و حجوم از این می‌خورد که خود را در این Return Statement می‌توانیم باشیم. هم‌چنین می‌توانیم برای این می‌توانیم این می‌توانیم باشیم.

مثال :

```
const oneParam = param => {
    console.log(param + 1);
};

oneParam(2);
```

نتیجه  $\rightarrow$  3

```
const oneLine = () => 2 + 3;
```

```
console.log(oneLine());
```

نتیجه  $\rightarrow$  5

74

using arrow function l, but not like this

```
[ 'make dinner',
  'wash dishes',
  'watch youtube'
]. forEach (function (value, index) {
  if (value === 'wash dishes') {
    return;
  }
  console.log(index);
  console.log(value);
});
```

});

.forEach ( (value, index) => {

}); =  
}); =

using loop

75

Passing a function into another Function ,  $\Rightarrow$

Now use arrow functions

لما تريد جعل بروتوكول arrow function ; اكتب جهازك

```
const object1 = {  
    method: () => {  
        } ;  
    } ;
```

: لما تريد أبرم shortcuts لـ methods

```
const object1 = {  
    method () {  
        } ;  
    } ;
```

76

`addEventListener()`

ما قبله هي امریقى نطاچی حابا اسبریت باکسیت ریت .  
`addEventListener` ایکریو کیت روپس کیت `onClick()`  
ما بارہ کے دھنار ایکریو کیت  
ما فوام اجراسور . ھبے کریا . ایم نفع ایکن بانے ایکت رایا .  
• `document.querySelector()`

هر ایک `addEventListener` پیش نام سے HTML کے ایک event کے ماتحت  
اجرا مکاری کر رائوند کیا ایکت باکسیت ریت .  
for what type of interaction is the event برائی کیا  
'click' button برائی کیا - we want to listen ?

Function کیا کیا . چون Function سے پڑھوں کیا کیا  
• `const arrowFunction;` ایک arrow function کیا کیا  
`const buttonElement = document.querySelector('.js-button');`

五

①

button Element . addEventListener ('click', () => {  
    console.log('click');

})

عند إضافة مصطلح click ، يتم إنشاء مثلك  
oneclick attribute [on~~click~~="click"] ، مما يجعله ينفي  
أي تأثير آخر على المثلث.

① multiple eventListeners for an event

'click2' ، 'click3' ، 'click4' ، 'click5' ، 'click6' ،  
لذلك click ، click ، click ، click ، click ، click ،  
قابلة لـ . يعني بـ addEventListener ()

② we can remove an Event Listener

removeEventListener ()

Best Practice: use .addEventListener() instead of  
onclick = " ... "

FB

• add Event Listener with keydown

type

'r' → Play rock

type 'p' → Play Paper

type 's' → Scissors

سیچھوں میں راہیں پورے رک، پیپر اسیکس

document.body.addEventListerner('keydown', () => {  
 console.log('keydown');  
});

حالا ہر جا کی سیچھوں کے راہیں پورے رک، پیپر اسیکس

کے لئے ہر کوئی دھنیوالی کیسے کر سکتے ہیں؟

هم دیکھنے کے لئے addEventListerner کا افرادی ایجاد کرنے کا کام

کرنے کا کام کیا ہے؟

٧٩

document.body.addEventListener('keydown', event)

⇒ { console.log(event.key) ; }

\* event دا اول addEventListener کافی  
\* event کردہ و بعد Function  
اطلاعات در صورت کیفیت این event کا رکار  
: پرینٹ کرنا جس کے لئے js میں

document.body.addEventListener('keydown', event) ⇒ {

if (event.key === 'r') {

    PlayGame('rock');

} else if (event.key === 'p') {

    PlayGame('paper');

} else if (event.key === 's') {

    PlayGame('scissors');

} ;

go

: ایجاد کردن HTML ; یا لایو سیدی ایجاد Select  
لیست پس از generate HTML و TodoList را  
renderTodoList();

```
function renderTodoList() {
    let todoListHTML = '';
    todoList.forEach(function(todoObject, index) {
        const {name, dueDate} = todoObject;
        const html = `<div> ${name} </div>
                    <div> ${dueDate} </div>
                    <button class="delete-todo-button"
                           onClick="todoList.splice(${index}, 1)
                           renderTodoList();"
                    > Delete </button>
                `;
        todoListHTML += html;
    });
}
```

٨١

document.querySelector('.js-todo-list').innerHTML =  
todoListHTML

ما هي فوائد إضافة Event Listener إلى onClick()؟

بما يرجع أبعد بالثلثة درجة حاداً سيرسيت باربع الخط درجاتي دارجاتي.

هذا أربع الخط درجاتي درجاتي درجاتي ايجار نسبيه ودر حجا اسر

ايجار عاليه ونعتاد آنحه هنار خاص راجع

delete todo button لـ web page html Delete call on click

① document.querySelectorAll('.js-delete-button');

will give us all the elements

: for loop console.log() ①

o !button.delete

1: < - > ~

دراین ادای todoList وکھر نار

82

ابعد عن button او بعد بـ .inster() ثم عن

81 Add: قم ... و عن button اخراجه ، ثم loop عن كل عن button اخراجه EventListener  
• loop forEach(); اما من

```
document.querySelector('.js-todo-list').innerHTML =  
    todoListHtml; كذلك لأن الصيغة الصحيحة هي صيغة الصيغة الصحيحة  
document.querySelectorAll('.delete-todo-button').forEach(  
    deleteButton, index) => {  
  
    deleteButton.addEventListener('click', () =>  
        { todoList.splice(index, 1)  
            renderTodoList(); و هذا مهم لأن نريد نرجع الـ array  
        } ); و هذا مهم لأن نريد نرجع الـ array  
    } ); و هذا مهم لأن نريد نرجع الـ array  
    • inster delete عن كل تابع char one, one,
```

63

for Each no  $\rightarrow$  iterate Current row to  $\rightarrow$  delete  
 بار اسکرپت می خواهد اولی اسٹ کو فل لے سکتے ہیں value گے  
 ویراپریم  $\rightarrow$  deleteButton یا  $\rightarrow$  remove button  
 ، Current index پر

More Features That Use  
 Functions as Value

- filter()

وہ دن سئے میں اکس ایجاد کرنا چاہیے کہ ایسے کوئی می خواہیم کیا تھا ایسا کوئی  
 می خواہیم کیا تو اس کو بنائیں۔

$[1, -3, 5] \rightarrow [5]$

Function کے لئے  $\rightarrow$  use for each کو  $\rightarrow$  filter()  
 index, value بار اسکرپت گے  $\rightarrow$  Function کے لیے دیا جائے  
 (filter) value کو لیتے رہتے تو value اولیع Filter پر ہے

84

د خریه مکمله د ریب فونکشن (اچام) لئے وسیع ھلکو، ارام  
مارفہ، سماں شاوٹ دراسع جا اسح فونکشن نے تے اسح جا دیسح  
بریکارڈنے۔ Value کا

```
[1,3,5].filter((value, index)=>{  
    return true;  
});
```

### filter

- ① create a new array []
- ② return true => put value in array
- ③ return False => not put value in array

جیسے → [1,-3,5]

البتہ اس تو پیر

د اس کا جمیلہ حاصلی فیلڈ true میں اسی بنکار اسح جا دیسح  
کرائ جیسے خواہندیبور، ای تو false ای تو true جیسے دیسح دیکھ دیجیں  
خواہندیبور

85

```
console.log([1, -3, 5].filter((value, index) => {  
    if (value >= 0) {  
        return true;  
    } else {  
        return false;  
    }  
}));
```

$\xrightarrow{\text{3rd}} \rightarrow [1, 5]$

← shorter way

```
console.log([1, -3, 5].filter((value, index) =>  
    return value >= 0;  
}));
```

$\xrightarrow{\text{3rd}} [1, 5]$

So we can return a truthy or falsy

• map()

$[1, 1, 3] \xrightarrow{\text{map}} [2, 2, 6]$

• map, filter, forEach  $\xrightarrow{\text{map}}$

36

• map()

- ① creates a new array [ ]
- ② whatever we return  $\Rightarrow$  added to new array

console.log([1, 1, 3].map((value, index) => {  
 return value \* 10;  
}));

Q: [1, 1, 1]  $\rightarrow$  [10, 10, 10]  
[1, 1, 1]  $\leftarrow$  Q:  $\Rightarrow$  return value + 10;

• map()  $\rightarrow$  transform an array into another array based on the return value.

console.log([1, 1, 3].map((value, index) => {  
 return value \* 2;  
}));

Q:  $\rightarrow$  [2, 2, 6]

; Shorter way

BF

console.log([1, 1, 3].map(value => value \* 2));  
قرآن [2, 2, 6]

چون مدل کاری بسیار طبیعی از این سه ایندیکس (index) و مقدار این ایتم را در یک آرایه می‌دانیم، می‌توانیم برای این آرایه یک فکر کنیم که برای هر یک ایتم این آرایه یک خودکاراً تابع باشد که مقدار این ایتم را در آن خودکاراً تابع بگرداند. این تابع را می‌توانیم با استفاده از دستور `map` ایجاد کنیم. این تابع می‌تواند مقدار ایتم را در آن خودکاراً تابع بگرداند و آنرا در آرایه نهاده کند. این تابع را می‌توانیم با استفاده از دستور `return` ایجاد کنیم.

closure

if a function has access to a value  
it will always have access to that value.

12 در 8:46