Types of variables in javascript:

https://www.youtube.com/watch?v=7tdXmlK\_YXI

Var

Const

Let

### **The Object Datatype**

The object data type can contain:

1. An object  
2. An array  
3. A date

DataTypes   
  
String

Number

Boolean

Array

Object

Null

For weather application we well be make the api => in weather map

And copy the link off api call in buit-in api request in past it in url

Events in javascripts :

Onclick

onclick="window.print()"

onmouseover

onmouseout

onkeydown

onload

onchange

JavaScript Type Operators

Typeof return the type of variable

Instanceof return true when the object is instance of object type

Let x = 5; Let y = x\*\*2; توان یا طاقت let y = math.pow(x,2); both of this uses for tawan

JavaScript String methods:

charAt()

charCodeAt()

at() د کرکترکود ات سره یوشان ته دی توپیر یی دادی چی پدی کی منفی انډیکس هم ورکولی شو

slice(start postion index of string , end postion index)یا (start position only)ex slice(2) or slice(-12) it delete the array elements

substring()and substr()  is similar to slice().

toUpperCase()

toLowerCase()

concat() ex : text1.concat(text2);

trim()

trimStart()

trimEnd()

padStart(numberfo padded letter , “padded letter”)

padEnd();

repeat(); ex: *string*.repeat(count)

replace() ex :sting .replace(“one word in prograph” , “that word you want”);

when you want to remove the case insitive of this use /i

let text = "Please visit Microsoft!";  
let newText = text.replace(/MICROSOFT/i, "W3Schools");

replaceAll()

split()

below all is used for searching porpus

indexOf()

lastIndexOf()

search()

let text = "Please locate where 'locate' occurs!";  
text.search("locate");

match()

matchAll()

includes()=>it return the true and false

let text = "Hello world, welcome to the universe.";

document.getElementById("demo").innerHTML = text.includes("worlds" , 4 option );

 startsWith() and endsWith() => return true and false

toFixed()

number in js

NaN

isNaN()

Infinit

Number as objects:

Let y = new Number (123);

MAX\_SAFE\_INTEGER it return the max number form it self

MIN\_SAFE\_INTEGER it return the min number form it self

isInteger();

isSafeInteger()

toString()

toExponential()

toPrecision()

valueOf()

Number()

parseInt()

parseFloat()

number properties :

Number.MAX\_VALUE

Number.MIN\_VALUE

MAX\_SAFE\_INTEGER

MIN\_SAFE\_INTEGER

POSITIVE\_INFINITY + &

NEGATIVE\_INFINITY - &

isArray();

instanceof()

pop() -> deleted the last item from array

shift() -> deleted the first item from array

unshift(“name”) -> add the new first element at the beginning of array

splice(1 index num,2 number of deletd element) - > deleted true index the array elements

splice(index of added element , how ilement you want to remove , “” , “”)

sort() -> is used for sort the array

reverse() -> sort from end

toSorted()

toReversed()

when you want to sort number we can it like below

const points = [40, 100, 1, 5, 25, 10];  
points.sort(function(a, b){return a - b});

delete -> by delete keyword we can delete the I tems ex : delete arrayName[0];

getFullYear() -> when you want to take the full year from the date

join();

copyWithin(the place index , the element you want to capy) -> when we whant to copy one element to another place in array

push()

indexOf() when you want to find the index of one element

Array.indexOf()

lastIndexOf()

find()

const numbers = [4, 9, 16, 25, 29];  
let first = numbers.find(myFunction);  
  
function myFunction(value, index, array) {  
  return value > 18;  
}

findIndex()

findLast();

findLastIndex()

const temp = [27, 28, 30, 40, 42, 35, 30];

let pos = temp.findLastIndex(x => x > 40);

document.getElementById("demo").innerHTML = "The last temperature over 40 was in position " + pos;

Math.min.apply(null , arr);

Array iteration :

forEach()

map()

flatMap() -> ex:

const myArr = [1, 2, 3, 4, 5,6];

const newArr = myArr.flatMap((x) => x \* 2);

document.getElementById("demo").innerHTML = newArr;

filter() - > this function is used for filteration in js

reduceRight()->also using for calculation

reduce()->using for calculation

every()-> هر element چک کوی return only true and false

some() -> -> return the true and false values

from() -> د یو string text نه array جوړوی

keys() -> when you want to print the keys of array

entries() -> when you want to print the key and value both for array

with() -> when you want to update the array elements ex: const myMonths = months.with(2, "March");

... -> when you want to spread the array elements ex :

const q1 = ["Jan", "Feb", "Mar"];

const q2 = ["Apr", "May", "Jun"];

const q3 = ["Jul", "Aug", "Sep"];

const q4 = ["Oct", "Nov", "May"];

const year = [...q1, ...q2, ...q3, ...q4];

document.getElementById("demo").innerHTML = year;

Date:

toDateString()

toUTCString();

toISOString()

Date.parse()->it converd the date to milliseconds.

getFullYear()

setFullYear()

getMonth()

getDate()- > return the day

getDay() -> return the week day

…getMilliseconds(),getTime()->it return the time to milliseconds

The getTimezoneOffset() method returns the difference (in minutes) between local time an UTC time:

setDate()

setFullYear()

setHours(),setMilliseconds(),setMinutes()…

math

round()-> round up قانون

ceil()->return the largest near number

floor()->return the down near number

trunc()->only return the inter part

sign()->only return the sign of number ex -+null

pow(x,y) -> power xy

sqrt(x)->جذر

Abs()->return the positive value only

Sin()-> cos tangant…

Min()

Max()

Log() -> logaritam

‌Boolean()

وقتی کی نمبر را چک میکنیم :

function myFunction() {

let voteable;

let age = Number(document.getElementById("age").value);

if (isNaN(age)) {

voteable = "Input is not a number";

} else {

voteable = (age < 18) ? "Too young" : "Old enough";

}

document.getElementById("demo").innerHTML = voteable + " to vote";

}

## The ?. Operator

let name = car?.type;

parseInt

parseFloat

toExponential()

toFixed()

toPrecision()

## Using String Methods

Search()->it return the position of the match

Replace(“one p “ , “other p”)

Match()

# **JavaScript Errors**

The try statement defines a code block to run (to try).

The catch statement defines a code block to handle any error.

The finally statement defines a code block to run regardless of the result.

The throw statement defines a custom error

<script>  
function myFunction() {  
  const message = document.getElementById("p01");  
  message.innerHTML = "";  
  let x = document.getElementById("demo").value;  
  try {  
    if(x.trim() == "") throw "empty";  
    if(isNaN(x)) throw "not a number";  
    x = Number(x);  
    if(x < 5) throw "too low";  
    if(x > 10) throw "too high";  
  }  
  catch(err) {  
    message.innerHTML = "Input is " + err;  
  }  
}  
</script>  
  
Finallay :

function myFunction() {  
  const message = document.getElementById("p01");  
  message.innerHTML = "";  
  let x = document.getElementById("demo").value;  
  try {  
    if(x.trim() == "") throw "is empty";  
    if(isNaN(x)) throw "is not a number";  
    x = Number(x);  
    if(x > 10) throw "is too high";  
    if(x < 5) throw "is too low";  
  }  
  catch(err) {  
    message.innerHTML = "Error: " + err + ".";  
  }  
  finally {  
    document.getElementById("demo").value = "";  
  }  
}

hello = () => {

return “hello”

}

Hello = () => “hello”

hello = val => "Hello " + val;

hello = (val) => "Hello " + val;

promise

const myPromise = new Promise(function(myResolve, myReject) {

setTimeout(function(){ myResolve("I love You !!"); }, 3000);

});

myPromise.then(function(value) {

document.getElementById("demo").innerHTML = value;

});

Onclick = “this.innerHTML=’other text’”

* 🌱 Frontend : React, Next, Web3.js, Angular, Vue, Nuxt, Jquery, Material UI, Bootstrap, TailwindCSS
* 🔭 Backend : NodeJs, CI, Laravel, Django, Express, Nestjs, PHP,
* 📲 Mobile: Android(Kotlin,Java), React Native, Ionic, Flutter
* 🧩 Database : MySQL, PostgreSQL, MongoDB, Firestore, MariaDB, SQLite, AES Lambda, Spring Cloud
* 💬 Language : HTML, CSS, Javascript, Typescript, Python, PHP, C++, C#, Go, Java, Kotlin
* 🔥 CMS: WordPress, Shopify, Wix, Framer, Webflow