lenght: restituisce indice massimo dell’array oppure dà i numeri dei caratteri della parola

touppercase: tutto in maiuscolo

nome.toUpperCase().

il contrario ètoLowerCase().

[]: restituisce il carattere alla posizione richiesta.

let nome= “Gesualdo”

nome[2]=s

NB parte da 0

indexOf restituisce l’indice di una parola

let nome= “Gesualdo”

nome.indexOf(‘s’) restituisce 2, ossia l’indice di s

CharAt() è uguale ma con i numeri

nome.ChartAt(2)= s

subString() prende un pezzo della stringa con i numeri dell’indice

nome.subString(0,3)=ges

Array

Strutture dati:

let amici=[‘johnny’, ‘pinuccio’, ‘ciruzzo];

anche mix di variabili

let amici=[‘johnny’, 6, true, false,90,’ciro’]

amici.length restituisce la lunghezza dell’array

in questo caso 5(uno in più dell’indice)

per andare ad un indice partiolare

amici[0,1,2...] = ‘inuccio’

cambia variabile all’indice 0,1,2...

Metodi:

Push:

Aggiunge un elemento all’array

let colori= [‘rosso’, ‘blu’];

colori.push(‘rosa’)

aggiunge rosa

Pop:

rimuove l’ultimo elemento dell’array

color.pop

rimuove blu

unshift:

Inserisce un nuovo elemento all’inizio dell’array

colori.unshift(‘rosa’)

rosa diventa primo

shift:

rimuove il primo elemento

colori.shift()

rimuove rosso

IndexOf(parola)

restituisce l’indice della parola

IndexOf(‘blu’)

2

slice:

prende un elemento dall’array

colori.slice()

dà una parte dell’array

colori.slice(0,2)

rosso e blu

2 perchè non è incluso

concat:

concatena 2 array

let arr1= [‘facebook’, ‘instagram’]

let arr2= [‘whatsapp’]

arr1.concat(arr2)

concatena arr1 a arr2

il risultato sarà:

[‘facebook, ‘instagram, facebook’]

array di array

let social=[

[‘facebook’, ‘IG’],

[‘whatsapp’, ‘telegram’]

]

social[0][0]=facebook

social [2][0]= whatsapp

Oggetti:

Struttura dati per immagazzinare dati tipo array

let persona=[‘billy’, 24, ‘Roma’] //con array

let persona={

nome: ‘billy’,

eta: 24,

città: ‘Roma’,

maggiorenne: true,

indirizzo:{città: ‘Roma’, via:’viale del carro 123’}

coloripreferiti:[‘rosso’, ‘blu’]}

coppia chiave-valore

persona.nome=billy

persone.indirizzo.città=Roma

persona[“coloripreferiti”][0]=rosso

persona[‘città’]=Roma

let nome={};

nome.città=’Torino’;

nome.eta=40;

ora nome sarà={città=’Torino’, età=40 }

Oppure

persona= new Object;

persona.cognome=’rossi’;

persona={cognome:’rossi’}

per cancellare chiavi in oggetti= delete persona.cognome

cancella la coppia chiave-valore cognome

Date:

let oggi= new Date();

restituisce la data con orario e fuso orario ed è considerato oggetto

per dare una data precisa facciamo:

let compleanno= new Date(‘mese-giorno-anno-orario’)

oppure con gli slash

oppure

Mese giorno anno

in inglese il mese

compleanno.getDate= da il giorno

getMonth= da il mese in numero

getFullYear= dà l’anno

oppure

setMonth= imposta mese

compleanno.setMonth(0)= gennaio

per moltiplicare/dividere e altro

x=x\*2

x\*=2 //meglio

x-=2 //meglio

x++ come C

let x=15

let y=20

console.log(x>y)=false

console.log(x<y)=true

if else come in C

Ternary operator come if else ma più veloce

let id=100

id === 100 ? console.log(‘id è 100’) : console.log(‘id non è 100’)

? true

: false

Switch case, come nel C

let nome = ‘billy’;

switch(nome);{

case ‘bily’;

fai qualcosa;

break;

case ‘billy’:

fai qualcosa;

break;

default:

fai qualcosa;

break;

}

come nel C

Funzioni, function declarection, come Python:

function nome(parametri){

fai qualcosa

}

function saluta(){

console.log(‘Ciao a tutti’);

}

saluta(); //chiamo funzione

Parametri

function saluta(nome){

console.log(‘ciao’ + “ ” + nome);

}

saluta(‘andrea’)

oppure di più

function saluta(nome, cognome){

console.log(‘ciao’ + ‘ ‘ + nome + ‘ ‘ + cognome)

}

saluta(‘andrea’, ‘canale’)

senza uno dei due dà undefined

oppure facciamo

function saluta(nome, cognome = ‘rossi){

console.log(‘ciao’ + ‘ ‘ + nome + ‘ ‘ + cognome)

}

saluta(‘andrea’)

se mettiamo il cognome prevale quello dato nella chiamata

return: restituisce un valore all’interno della funzione

function saluta(nome, cognome = ‘rossi){

return(‘ciao’ + ‘ ‘ + nome + ‘ ‘ + cognome)

}

console.log(saluta(pino, pinuccio))

dà il return

function expression

let funzione = function(x){

return x\*2

}

per chiamarla usiamo il nome funzione

console.log(funzione(5))

IIFE(funzioni chiamate sul momento)

(function saluta(){

console.log(“Hello World”);

})()

con queste parentesi viene chiamata la funzione

e nelle parentesi passiamo i parametri

Property method o metodi(funzioni dentro un oggetto):

let persona{

valori...

}

possiamo mettere funzioni negli oggetti

let persona{

saluta: function(nome, cognome){

console.log(“Ciao, sono ” + nome + “ “ + cognome)

}

}

persona.saluta()

chiama la funzione

Possiamo anche aggiungerle

persona.ringrazia= function(){

console.log(“Grazie”)

}

persona.ringrazia()

Loop(for e while) come C.

ma:

Do while:

let i = 0;

do{

console.log(i);

i++

}

while(i<15) //fai questo fino al raggiungimento della condizione

Loop su array:

let frutta=[‘banana’, ‘mela, ‘pera’, ‘kiwi’, ‘ananas’]

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

console.log(‘ciao’);

console.log(frutta[i]);

} //printa ciao 4 volte e tutti gli indici di frutta

ForEach, fa delle azioni per ogni elemento dell’array:

frutta.forEach(function(value)){

console.log(value)

} //funzione su 5 elementi cioè quelli dell’array, prende i valori dell’array

oppure value,index per passare anche gli indici

oppure array stampa l’intero array per tutti gli indici

function(value,index,array)

posso anche fare function(pino, pluto, paperino) ed è sempre value, index e array

Map:

frutta.map(function(value,index,array)){return value}

restituisce un array con i risultati

let newfrutta= frutta.map(function(value,index,array)){return value}

è printa un array con i valori, index o array a seconda dal return

For in, usato per gli oggetti:

let persona={

nome=’billy’,

eta=18,

hobby=’calcio’

}

for(let x in persona){

console.log(x)

}

for(chiave in oggetto)

printa le chiavi

se facciamo console.log(persona[x]) stamperà il valore

DOM:

windows=info del browser