به نام خدا

-------------------------------------------------------------------------------------------------------------------------

Data types in js :

-number

-string

-boolean

-BigInt

-object

-Array

-null

----------------------------------------------------------------------------------------------------------------------------------------------------------

How to convert datatypes in js :

-Number() => داده را به تایپ نامبر تبدیل میکند . => ساده ترش اینه که یه + قبل اسم داده بزاریم .

-String() => داده رو به استرینگ تبدیل میکنه

-Boolean() => داده رو به تایپ بولین تبدیل میکنه

----------------------------------------------------------------------------------------------------------------------------------------------------------

Logical operators :

\*isNaN() => checks that is data a number or not .

== ---> بررسی برابری

< or > => بررسی کوچکتری یا بزرگتری

=== 🡪 بررسی برابری مقدار و تایپ داده

----------------------------------------------------------------------------------------------------------------------------------------------------------

Bitwise operator :

! 🡪 عملگر نقیض

&& 🡪 and

|| 🡪 Or

----------------------------------------------------------------------------------------------------------------------------------------------------------

Ternary condition :

جهت نوشتن شرط تک خطی

برای مثال :

Let accessAlowed = (age>18) ? true : false ;

دستور بعد از ؟ در صورت صحیح بودن شرط اجرا میشود و دستور بعد از : در صورت نادرست بودن شرط اجرا خواهد شد .

----------------------------------------------------------------------------------------------------------------------------------------------------------

Switch(x)

Case "firs condition for x " :

….

Breake ;

Case "second condition for x " :

….

Breake ;

Case "third condition for x " :

….

Breake ;

Default : 🡪 در صورتی که هیچ کیسی درست نباشد این دستور اجرا خواهد شد .

....

Breake;

----------------------------------------------------------------------------------------------------------------------------------------------------------

Declaration functions :

Index 🡪 از صفر شروع میشه

Length 🡪 طولش رو میگیره

charAt() 🡪 character at (here) 🡪 به عنوان ورودی شماره ایندکس رو میگیره

concat () 🡪 ورودی ها رو بهم میچسبونه

trim() 🡪 اسپیس های ابتدا و انتها رو پاک میکنه

toLowerCase() 🡪 همه حروف ورودی رو کوچیک میکنه

toUpperCase() 🡪 همه حروف دیتای ورودی رو بزرگ میکنه

search() 🡪 یک استرینگ رو میگیره و ایندکس حرف اول رو در صورت وجود به عنوان خروجی بر میگردونه

indexOf() 🡪 مانند متود سرچ عمل میکنه

includes() 🡪 دیتای ورودی رو میگیره و اون رو بررسی میکنه و برای وجود یا عدم وجودش یک بولین بر میگردونه

slice(start?,end?)

substr(start? , length?)

substring(start? , end? )

----------------------------------------------------------------------------------------------------------------------------------------------------------

Math methods :

PI 🡪 π

Pow(2,4) 🡪 2 رو به توان 4 میرسونه

Sqrt() 🡪 جذر

Abs() 🡪 قدر مطلق

Min() 🡪 مینیموم دیتا های ورودی رو ریترن میکنه

Max() 🡪 ماکسیمم دیتا های ورودی رو ریترن میکنه

Floor() 🡪 جزء صحیح

Trunc() 🡪 قطع کردن عادی

Round() 🡪 گرد میکنه

Ceil() 🡪 همیشه به سمت بالا گرد میکنه

----------------------------------------------------------------------------------------------------------------------------------------------------------

Assignment operators :

+= 🡪 num +=5 🡪 num = num + 5

-= 🡪 num -=5 🡪 num = num – 5

\*= 🡪 num \*=5 🡪 num = num \* 5

/= 🡪 num /= 5 🡪 num = num / 5

\*\*= 🡪 num \*\*= 5 🡪 num = num \*\* 5

Number ++ 🡪 number += 1

++number 🡪 number += 1

Number -- 🡪 number -= 1

--number 🡪 number-= 1

----------------------------------------------------------------------------------------------------------------------------------------------------------

Loops :

For(begin point ; condition ; step ){

//loop body

}

While (condition ) {

//codes so called loop body

}

Do{

//body

}while( condition ) 🡪 حداقل یکبار اجرا میشه در هر صورت

----------------------------------------------------------------------------------------------------------------------------------------------------------

Arrays :

آرایه ها کمک میکنند که چند تا موجودیت رو تو یه متغیر تعریف کنیم . در یک ارایه از هر نوعی در کنار هم میتوان استفاده کرد .

["Ali" , 10 , true ]

Arrays methods :

Push() 🡪 اضافه کردن یک آیتم به انتهای ارایه

Pop() 🡪 حذف کردن آخرین آیتم آرایه . این متود پارامتری به عنوان ورودی نمی گیرد .

Shift() 🡪 پارامتری نمیگیرد و اولین آیتم آرایه را حذف میکند .

unshift() 🡪 یک پارامتر را دریافت کرده و اون رو به ابتدای آرایه اضافه میکند .

foreach() 🡪 فانکشنی رو از ما میگیره و برای هر عضو از ارایه مد نظر اون رو اجرا میکنه

includes() 🡪 سرچ میکنه داخل ارایه ، همیشه کارساز نیست ،نمیتونه به عنوان ورودی یک ابجکت بگیره

some() 🡪 مانند فورایچ یک فانکشن میگیره و ترو یا فالس برمیگردونه ، وقتی به یک ترو برسه متوقف میشه

Every () 🡪 یک شرط رو به شکل همگانی چک میکنه و نتیجه رو برمیگردونه ، در صورتی ک به حتی یک شرط فالس برسه متوقف میشه

Splice() 🡪 با توجه به ایندکس ورودی هر آیتمی رو میشه حذف یا اضافه کرد .

Splice(start , count , items we need to add/remove(optional))

findIndex() 🡪 ایندکس ورودی رو در آرایه به ما برمیگردونه

map() 🡪 این متود همواره یک آرایه به ما بر میگردونه . یک فانکشن خاص رو روی همه اعضا اعمال میکنه و نتیجه رو در غالب یک آرایه برمیگردونه .

Filter() 🡪 یک فانکشن به عنوان فیلتر ورودی میگیره و اعضایی که ازون فیلتر رد میشن رو در غالب آرایه به ما برمیگردونه .

isArray() 🡪 چک میکنه آیا ورودی یک آرایه هست یا نه

indexOF() 🡪 ورودی که بهش میدیم رو در آرایه یا استرینگ سرچ میکنه و ایندکسش رو بهمون برمیگردونه ، در صورت وجود نداشتن آیتم ورودی در آرایه ، عدد -1 ایندکس خروجی خواهد بود .

Join() 🡪 تک نک آیتم هارو بهم میچسبونه .

Reverse() 🡪 کل آرایه رو برعکس میکنه .

Split () 🡪 تقریبن بر عکس متود جوین میتونه عمل کنه .

Find() 🡪 داده ای که بهش میدیم رو جست و جو میکنه و اطلاعات کل اون آبجکت یا آرایه رو بهمون برمیگردونه . تا زمانی که به یک ترو برسه اجرا میشه

Eval() 🡪 یک استرینگ به شکل ورودی میگیره و اگه یه کد باشه اون رو اجرا هم میکنه ازین جهت امنیت پایینی داره .

----------------------------------------------------------------------------------------------------------------------------------------------------------

Objects :

Let user = new object() // " object constructor " syntax .

Let user = {} // " object literal " suntax .

----------------------------------------------------------------------------------------------------------------------------------------------------------

Primitive data types :

-number

-bigInt

-Boolean

-symbol

-Null

-undefined

-string

Reference:

` -Object

-array

-Function

----------------------------------------------------------------------------------------------------------------------------------------------------------

Anonymous Functions :

Example : user.filter(یک تابع بی نام )

IIFE : Immediately Involked Function Expression 🡪 تابع خودخوان

Syntax :

(function(){

//codes

})()

----------------------------------------------------------------------------------------------------------------------------------------------------------

کد نویسی اصولی :

* زمانی که میخوایم یک آرایه رو پیمایش کنیم بهتره فقط و فقط از متود هاش استفاده کنیم برای مثال استفاده از foreach برای حلقه های for و while و do while .
* از eval استفاده نشه .
* همیشه برای ورودی های هر فانکشن مقدار default در نظر بگیرید .
* نام گذاری اصولی فانکشن ها بر اساس عملکردشون
* هر فانکشن همیشه فقط یک کار انجام بده نه چند کار

----------------------------------------------------------------------------------------------------------------------------------------------------------

Set Interval :

setInterval(function () {} , delay time (ms))

این تابع هر ms یکبار اجرا میشه .

clearInterval(interval name )

----------------------------------------------------------------------------------------------------------------------------------------------------------

False and truthy values :

-Boolean() 🡪 مقداری رو میگیره و فالسی و تروثی بودن اون رو برمیگردونه

-true , false

-undefined / NaN 🡪 false value .

[ ] / { } 🡪 true

----------------------------------------------------------------------------------------------------------------------------------------------------------

setTimeout :

زمانی که میخایم یک فانکشی بعد از یک مدت زمان معین اجرا بشه

Syntax :

setTimeout(function () { } , time(ms) , … arguments )

----------------------------------------------------------------------------------------------------------------------------------------------------------

Flag in JS :

فلگ ها یک مقدار بولین هستند و در شرط ها کاربرد دارند .

----------------------------------------------------------------------------------------------------------------------------------------------------------

Let and const :

ما میتونیم متغیر هایی که با var تعریف کردیم رو هم آپدیت بکنیم هم دوباره تعریف بکنیم اما متغیر هایی که با let ساخته میشن رو فقط یکبار میتونی تعریف بکنی (مزیت ) .

Let از قانون hoistins پیروی نمیکنه .

متغیر هایی که با var تعریف میشن همیشه گلوبال هستن حتی اگه تو یه اسکوپ باشن .

اگه متغیری رو با const تعریف کنیم دیگه مقدارش آپدیت نمیشه و ثابته .

اگه متغیری با const تعریف بشه باید تو همون خط مقدار دهی بشه .

DOM in JS :

Document Object Model

----------------------------------------------------------------------------------------------------------------------------------------------------------

Methods for selecting an element :

-getElementById(" " ) 🡪 یونیک و برای انتخاب یک المنت با آیدی خاص

-getElementsByClassName( " " ) 🡪 همیشه آرایه برمیگردونه

-grtElementsByTagName( " " ) 🡪 همیشه آرایه برمیگردونه

-queryselector(' ' ) 🡪 ورودیش مثل سلکتور های سی اس اس هست و اولین المنت که اون آیدی یا کلاس رو داشته باشه انتخاب میکنه .

-querySelectorAll( ' ' ) 🡪 همه المان های ورودی رو برمیگردونه و یک آرایه هست .

----------------------------------------------------------------------------------------------------------------------------------------------------------

نحوه دسترسی به attribute ها و تغییر اون ها :

getAttribute(' ' ) 🡪 اسم اون صفت رو میگیره

setAttribute('name' , " value" ) 🡪 اضافه کردن صفت

----------------------------------------------------------------------------------------------------------------------------------------------------------

Access style of element :

Elementname.style.css\_property = "value"

----------------------------------------------------------------------------------------------------------------------------------------------------------

Access to the content of an HTML Element :

1.elementName.innetHTML

2.elementName.innerText

3.elementName.textContent

برای دسترسی به محتوا و value تگ هایی که تو فرم ها استفاده میشن از .value استفاده میکنیم .

----------------------------------------------------------------------------------------------------------------------------------------------------------

Build elements of HTML with JS :

-createElement('element TagName ')

----------------------------------------------------------------------------------------------------------------------------------------------------------

How to append elements in DOM :

-append(' ' )

parentElements.append(' ')

----------------------------------------------------------------------------------------------------------------------------------------------------------

Events in JS :

-keypress

-keydown

-keyup

-change 🡪 معمولا برای تگ های انتخاب کردن انجام میشه

-onblur

-onfocus

-ondblclick

----------------------------------------------------------------------------------------------------------------------------------------------------------

addEvenetListener( , )

دو ورودی میگیره اولی اسم ایونت بدون on و ورودی دوم فانکشنی که میخایم اجرا بشه .

----------------------------------------------------------------------------------------------------------------------------------------------------------

classList

methods 🡪

add()

remove()

toggle(class name ) 🡪 اگه اون کلاس رو داشته باشه حذفش میکنه و اگه نداشته باشه بهش اضافه میکنه .

----------------------------------------------------------------------------------------------------------------------------------------------------------

has Attribute and remove Attribute :

void 🡪 یعنی چیزی رو بر نمیگردونه

----------------------------------------------------------------------------------------------------------------------------------------------------------

Third way of adding an event :

targetID.Event = function

----------------------------------------------------------------------------------------------------------------------------------------------------------

Event object \*

اطلاعات ورودی از کاربر رو به شکل یک object در اختیار ما قرار میده .

Id.addEventListener('event' , function(event){

//codes

}

----------------------------------------------------------------------------------------------------------------------------------------------------------

preventDefault()

برای جلوگیری از رفتار دیفالت و پیشفرض یک عنصر از متود preventDefault() استفاده میکنیم .

TargetId.preventDefault()

Example :

-form 🡪 ریلود شدن

-a 🡪 لینک شدن

----------------------------------------------------------------------------------------------------------------------------------------------------------