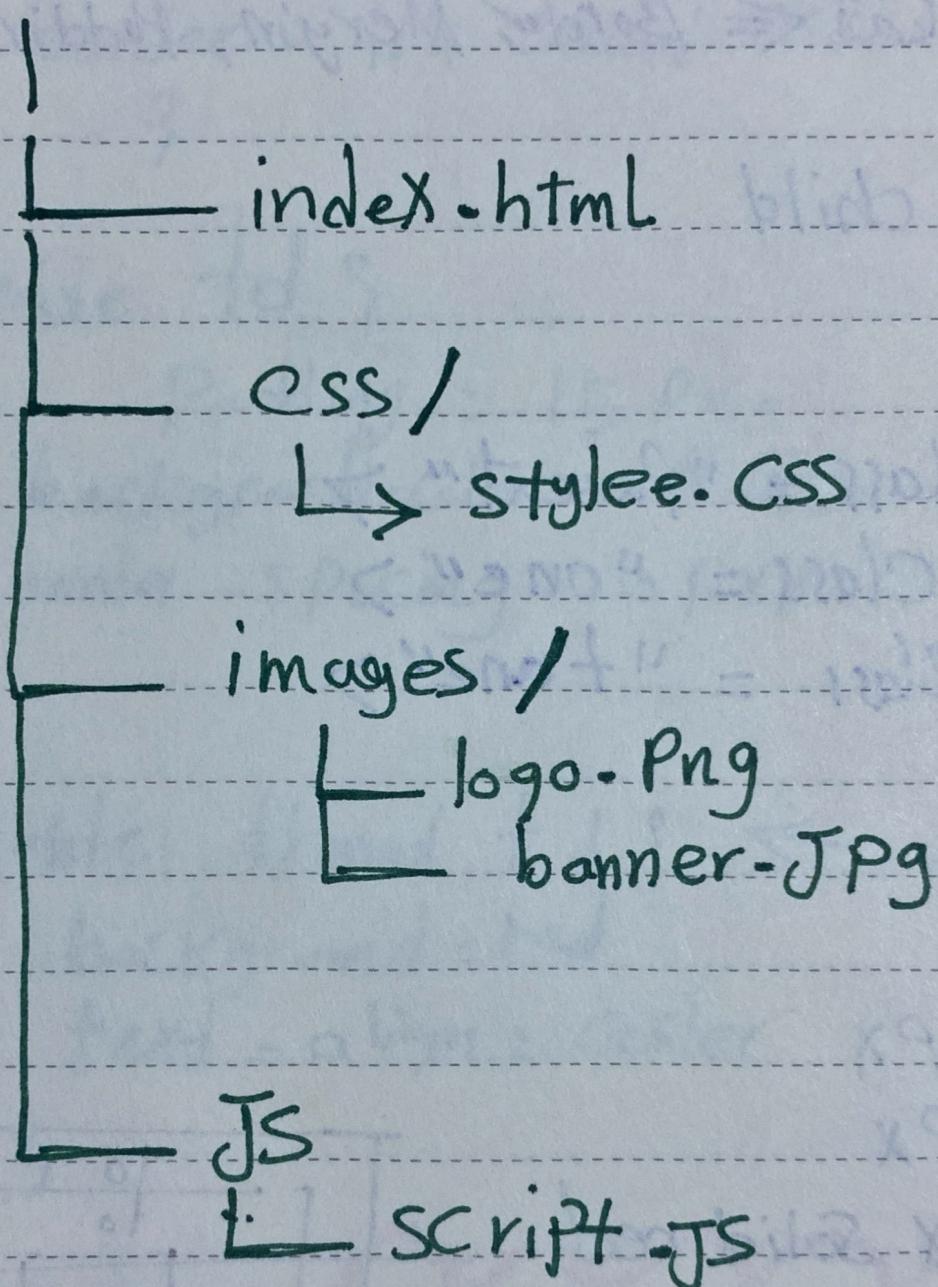
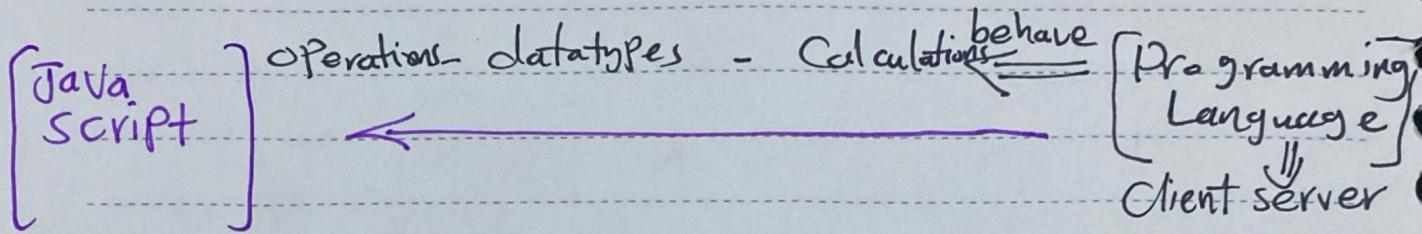


Project -Folder/



Java Script

Compiler
Interpreter
Optimiser



scripting \Rightarrow Not stand alone language (Embedding language)
 language (جazyat) running inside another language (جazyat)
 execute (جazyat) (HTML file) Parse alone run / host
 \hookrightarrow run within browser (interpreted language)

Java script features:

- loosely - not strongly typed Language (type) \Rightarrow Value \neq type
- object based
- interpreted Var Variable ; without initialization

* [Need Validation]

String \Rightarrow "جazyat"

Var x ;

x = "iti" x = iti

(String) \rightarrow

Data types

Primitive types

String

Number

Null

boolean

Object types

Data

Array

PAGE DATE

Primitive :

مُطْبِّقُ العَيْنِ \Rightarrow ثُمَّ إِنْتَ مَا دِيْنَهُ فِيْكَ
بَلْ مَا تَعْنِي الْعَيْنُ الْعَدْيِيْ
- بِطْلِيْ - تَعْرِنُ فِيْكَ

- undefined \rightarrow مُعَدَّلٌ مُتَغَيِّرٌ مُتَحَوِّلٌ مُسْبِطٌ
 - Null \rightarrow المُتَغَيِّرُ لَهُ فِيهِ (فَاضِي)
- أَنْتَ قَاصِدٌ تَقُولُ أَنَّهُ فَاضِي

Non-Primitive:

- object types

(Property method) call | me

Language objects

Browser objects : DOM

HTML objects : DOM

User defined objects

- \leftrightarrow \Rightarrow object

- Id = "WebWrapper" \Rightarrow Property

sensitive

B != b

min 33

* Primitive

No Wrapper type for Null, undefined \Rightarrow بريلاك

There is for String Class String \Rightarrow ابجدر

wrapper object \times غلاف ()

حي في جاوا استرليني

Primitive بعض

ما في جاوا () \Leftarrow (object counterparts) ما في كلاس يختلفها

String \rightarrow الفراغ

Number \rightarrow الممك

Boolean \rightarrow القيمة

عندما تغير تغير معاها

methods

✓ let str = "hello";

Console.log(str.toUpperCase());

behind scene \Rightarrow Primitive(string) "hello"

\Rightarrow new String("hello")

\Rightarrow .toUpperCase() . . . نتائج متعددة

\Rightarrow ترجع الفراغ - ترجع الـ object مؤقت

- toUpperCase() يبقى له خاتمة hello

- جاوا استرليني لا يملك المتغير مساحة مفتوحة

- كل مت خاتمة أليف الفراغ ()

String متغير

- يعطيها مساحة مفتوحة . GC . يومياً لا وتحللت الموارد

وليس لها الفراغ مفتوحة () لـ GC

Feature

2- object based

3- interpreted language

- from top to bottom

- from left to right

Line 1 : ✓

Line 2 : ✓
Syntax error

Line 3 : error

④ event handling

event int
js → event handler (function code)

⑤ totally integrated with HTML ⇒

⑥ Case sensitive

Where to write JS?

1- External

2- Internal

<script>

Code

3- Event handler → event line with

</script>

head ↗

body ↗

<head>

<script src=" " >

</script>

↗

</head>

↑
JS → in head

JS → script source
script without source
No Combination

عن

التنسی حابین

<script src="" >
↓ ignore it → Compiler ignore it
</script>

Var myVar = null;

type of (myVar); // object

object ↗ state (value) heap
 ↳ Identity (address) stack

We use null to restrict object reference

* Var myVar = 10; literal declaration

Variable ↗ type
 ↗ name
 ↳ Value
 ↳ address

Var myVar = new Number(10);
heap ↗ //Constructor or declaration
refer to property

عکس ⇒ مثلاً بـ
عکس ⇒ مثلاً بـ
Number ⇒ 10

myVar = Valueof() ← same place

2 copies for variables:

① global scope / script scope:

[Using var]

Var myVariable;
function / script scope

[global variable]

Using var

initialization ↓
↓

undefined ← assigning

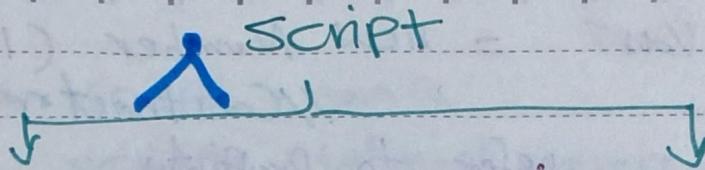
[without using var]

myVariable; ⇒ you have to initialize it

(reference error)

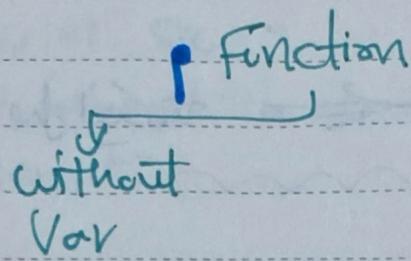
myVariable = 10;

1. Global scope



Using var
↓
initialization ;

without var
initialization لازم



function myfun () {

from my variable = 20j \Rightarrow global scoped script

console.log (

شرط ترتيله فيه

}
myfun()

console.log (from my variable) ↪ function scope

شرط ترتيله الفاتحة

قبله == مكتوب ترتيله الفاتحة
و أنه اندرس

2. local scope

local scope = function scope

in ES5 → we don't have block scope

لَا فِيهِ مَنْعِلٌ جَدُوٌ بِلَوْنِ مَنْتَانٍ بِرَوْهِ عَادِيٍّ / فَمَنْ جَعَهُ الْلَّهُبْ بِسْ

without var → global

Function ↗
↓ with var → Local ✓

local scoped variable

↳ declare variable: inside function

using var

global لَا اسْلَرِيْتَ لَعْرَوْهُ اَيْ مَنْيَرٌ كِبِرٌ مَعْرُوفٌ
JS ➔ default (non-strict mode)

type of → منْتَجِيْرٌ سَاعِيْرَيْلَمْيَرٌ
var mynum = new Number(10); مَلَأَ لَوْ بَوْزَيْهَتْ دَنَابَاتِيْلَمْيَرٌ

object ← وَجَعَاصَارْعَادِيْرَاهِيْلَمْيَرٌ

object ← type بِسْ

real type مَنْتَجِيْرٌ type of object

Number ← Not object بِسْ

Property ctor using Property

→ any function in JS → return id

return ; return false or true

= undefined

type of X

Constructor.name

object or wrapper *

Var X = 10;

X.toFixed();

Number ١

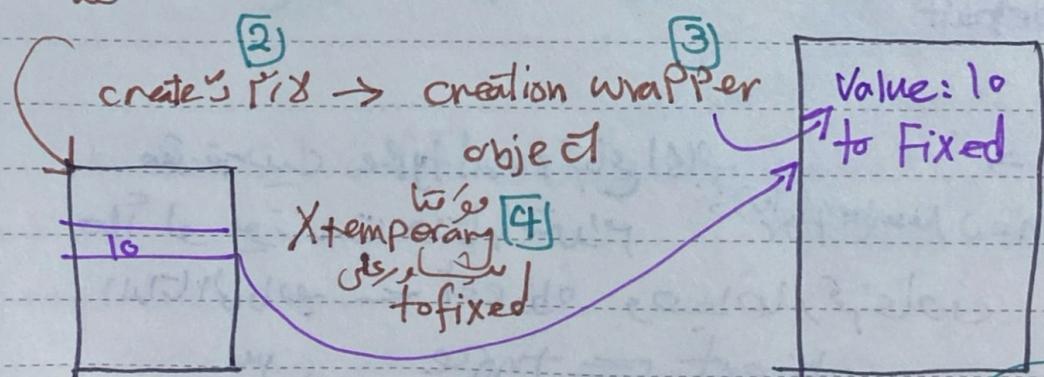
auto or no

واليه مخصوص

no

variables *

methods also for numbers
go back for methods in
objects



Var test = {} ; // object literal declaration

in this case also new consider

X.sayHello = function() { return 'sayHello' ; } ;

creation
using • withy any literal variable = wrapper object
we use it
then destroy

line

36 ~~X =~~ X.sayHello = Function () { return "Say"; }
37 X.sayHello(); → error

38

u should call it at the same line of decreation (36)

var mynum = 1e6 = 1000 000; use

JS → Guardaline 10^6

+ toFixed

(1111.234).toFixed(1) // '1111' → string
(11111.234).toFixed(1) + 50 // 1111150

(100.22).toFixed(5)

100 220

PAGE DATE
001 ⇒ 100 - 000

Number.isInteger(1000 - 55)

Validation → Number
Validation → Integer

احسب المجموع

Number - IsFinite ⇒ ~~isNaN~~ ⇒ Infinity [-]
NAN
(Not a number)

Video 2 min 1:02

Var myVar1 = 10
Var myVar2 = '10'

myVar1 == myVar2 → True

وقت المقارنة [Implicit type]

لما يعادن المقارنة لها صورة باسترجاع

العنوان الثاني حول الـ String to int

(operator) + Plus + - border to Number

→ addition

+ → Concatenating

↓ → Converting

== == ==

→ type
→ values

لـ Implicit conversion

هم لهم وهم فالـ False

Parsing : min 1:08

Parse Int

Parse Float

Number

+

ParseInt

Trim →
user entered

"1234A"

1 2 3 4

It stopped

الى في الامام والى في الوراء

spaces starting from 0 is

char = length -

يتحاول
اجزء العدد

لعمود

first char → digit

second char → digit

stoped ← Co ↓ tail

↳ False Success → 1234

1234

user entered ParseInt("A2134") → NAN (NOT a Number)

~ ~ ParseInt("")

ParseInt ("\\n\\n\\t") → NAN

اجزء العدد ترجح

Parsefloat

methods (dialog)

عووز

User ↗ input (رقم) → prompt

↗ Filter (الدالة) → Confirm

↳kieleb1d1 → alert

* Content ↗ نفع لوجهك

* Prompt ()

// return String { , Null
↓
User clicked ↓
OK Clicked
Cancel
or ESC (Keyboard)

Var result = Prompt()

NAN min 1:23

- Special value Javascript
- Not = anything
- toxic value
- Not = anything

NAN == NaN
False

is isNaN - ایسا نیز نہیں
NAN = Value is !
NAN = - ایسا نہیں وجد نہیں

IF ↓ Check if true

Reasons

- any operation on 'undefined' \Rightarrow NaN
- $0/0 \Rightarrow$ NaN
zero/zero
- ParseInt (true) \Rightarrow NaN
- ParseInt (null) \Rightarrow NaN
- Infinity / Infinity \Rightarrow NaN

isNaN () $\begin{cases} T \\ F \end{cases}$

ParseInt \Leftarrow مبدل للأرقام

لأنه ياعمل حوالياً check
عنوانه ماندخلش في حواريات
ومنها نان \Rightarrow NaN

isNaN ()

// is ~~not~~ not a number

```
do  
var fn = Prompt (" Enter a number ");  
while ( isNaN (fn) );
```

True

حالاً جوه الكوس
وتحرجت صيفيل لعب
عنان

عنان اتأتكم إلّا بعد بيد خله \rightarrow اقم

NaN يُعتبر الألгорیتم \rightarrow + operator

Number

Number

- Trim spaces until operators else
- return if length check = 0
- else digits & operator

IF (length) != 0
return "0"

digits

user entered (" ") \Rightarrow Number

no digits, no space

0 \Leftarrow return "0"
zero

إذا تم تعيين ال變iable

Letters

\rightarrow digits

in

string

\rightarrow NOT digits

Number

\rightarrow NaN

فهـو يـحـلـرـمـ لـعـلـاقـيـ

Number ("") // 0

حسـكـيلـ

ParseInt ("") // NaN

لـعـولـالـيـ

مـعـبـودـ رـقـمـ

Check ChatGPT diff Number()

ParseInt()

Number (NaN) // NaN

Number (null) // 0

Number (undefined) // NaN

ParseInt (undefined) // NaN

Number (null) // 0

ParseInt(null)

// NaN