2023

Java script

BY SADEK
AHMED

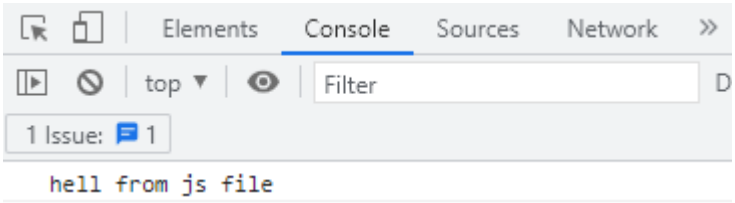
Java Script

- يتم كتابة كود java script فى صفحة الhtml داخل script tag الموجود فى الheader او فى ملف خارجى ووضع مسار الملف فى src attribute فى script tag
- يمكن وضع script tag فى الbody ويفضل وضعه فى النهاية
- يمكن عمل comment فى java script بطريقتان وهما :
  1) Line comment وهو comment لسطر واحد ويتم عن طريق كتابة //
  2) Multiply line comment وهو comment لاكثر من سطر ويتم عن طريق كتابة /* */ ويوضع الcomment فى المنتصف
- الwindow هو object فى java script يحتوى على الكثير من function ويتستخدم لتحكم فى tab او الصفحة
- Window.alert(“ ”) هو امر يقوم باظهار رسالة عند الدخول الى الصفحة
- الalert لم يعد يستخدم لانه يتسبب فى ايقاف تحميل الصفحة
- يجب وضع ; فى نهاية كل سطر
- الdocument هو object فى java script وهو يمثل صفحة الhtml
- Document.write(“ ”) هو امر للكتابة داخل صفحة الhtml سواء كان ب text او html script
- Write function لم تعد تستخدم

```
document.write("<h1> header 1 <h1>")
```

- Console هو object فى java script يستخدم فى التحكم فى الconsole الموجود فى المتصفح
- Console.log(“ ”) يستخدم لطباعة رسالة فى console

```
console.log("hello from js file")
```



- Console.error(“ ”); هو امر يقوم ايضا بطابعة رسالة فى الconsole ولاكن فى شكل error
- Console.table(); يقوم بانشاء table داخل console وغالبا ما يستخدم معه array
- فى ال console.log() يمكن اضافة بعض التنسيقات ل الtext المكتوب ولاكن يفصل بين الtext و التنسيق ب , وتوضع علامة %c قبل الكلام المراد اضافة له النص ويمكن اضافة اكثر من تنسيق باضافة %c اخرى وكتابة تنسيق اخر ويفصل بينه وبين الاخر ب ,
- الconsole الموجود فى المتصفح يتبع ما يسمى ب web API
- ECMAS هى منظمة عالمية مسؤلة عن وضع المعايير للغات البرمجة
- ECMAScript6 وهنا المقصود بها لغة java script الاصدار السادس

Data type

- Typeof(); هو امر يقوم بارجاع نوع المتغير او الdata

1. String هو نوع من data ويقصد به نص ويوضع بين " " او بين ‘ ’
2. Number ويشمل الصحيح والعشرى ولا يجب وضعه داخل " "
3. Array وهى مجموعة من العناصر داخل [ ] يفصل بين كل عنصر والاخر ب ,
   - الarray هى object فى java script
   - يمكن ان تكون العناصر التى بداخلها من اى نوع
4. Object ويتكون من مجموعة من العناصر (keys) و القيم الخاص بهم وكل عنصر معه القيمة الخاص به يفصل بينهم : وفصل بين كل عنصر والاخر ب , ويوجد جميع العناصر داخل { } كتالى
   {name : “ahmed” , age : 20 }
   - يمكن وضع الdata type المختلفة فى الobject
5. Boolean وهى data type تكون لها قيمتان فقط وهى true او false
6. Undefined
7. Null وهى object

Variables

- عملية انشاء متغير تسمى declare وهى تتم عن طريق كتابة var ثم اسم المتغير ثم = ثم قيمة المتغير كتالى : var name = “ahmed”;
- يمكن عدم كتابة كلمة var ولاكن يفضل جدا كتابتها
- عند انشاء عنصر ب id فى html بذالك فهو ينشى متغير يمثل tag كاملا
- يمكن كتابة اسم المتغير فى الconsole لمعرفة قيمته

- هناك معايير لكتابة اسم المتغير مثل :
  - عدم وضع مسافة فى اسم المتغير
  - لايجب ان يبدا اسم المتغير ب رقم
  - يمكن استخدام _ فى اسم المتغير وفى اى مكان
  - يمكن وضع $ فى اسم المتغير فى اى مكان
  - لايمكن استعمال اى special char اخر فى اى مكان
  - الJavaScript هى لغة sensitive اى انها تفرق بين الحروف الكبيرة والصغيرة
  - لايمكن استخدام keywords الخاص باللغة فى كتابة اسم المتغير
- هناك keywords اخرة لانشاء متغير مثل let و const

var	let	const
يجوز عمل declare مرة اخرى	لايجوز عمل declare مرة اخرة	لايجوز عمل declare مرة اخرة
عند الوصول الى المتغير قبل declare يقوم بعرض undefended	عند الوصول الى المتغير قبل declare يقوم بعرض رسالة error تشير الى ان المتغير لم يعرف بعد	عند الوصول الى المتغير قبل declare يقوم بعرض رسالة error تشير الى ان المتغير لم يعرف بعد
عند انشاء متغير تقوم بحفظه فى window object	لاتقوم بوضع المتغير فى الwindow	لاتقوم بوضع المتغير فى الwindow

- عند طباعة " او ' فى الكلام يلزم عمل واحد من التالى :
  1) عندما نريد طباعة " نضعها فى ' '
  2) عندما نريد طباعة ' نضعها بين " "
  3) نضع \ قبل " او '
- عندما نريد طباعة \ نضع قبلها \ اخرى
- يمكن استخدام \ لتكملة فى السطر التالى
- \n يقوم بعمل سطر جديد
- Concatenation يحدث فى java script عن طريق +
- فى ECMAScript 6 قدمت طريقة افضل ل concatenate وهى افضل فى عمل concatenate بين data type المختلفة وفى تنظيم الكود من حفظ مسافة وامكانية الكتابة فى اكثر من سطر وتتم عن طريق ان المحتوى يوضع بين علامة تسمى back tick ` ` وهى تاتى من حرف ذ فى الكيبورد ولعمل concatenation بداخلها نكتب ${ } ونضع الvariable بداخلها
- تستخدم علامة back tick فى كتابة اكواد html ووضعها فى variable بسبب كونها لاتتجاهل المسافات والاسطر الزائدة

Arithmetic operators

- NAN هو رمز معناه not a number ويقوم بظهور فى حالة حدوث عمليات حسابية بشكل خاطئ او عمليات غير معروف نتائجها
- NAN نوعها number
- لعمل اس فى java script نستخدم **
- يوجد فى الjava script مصطلح post و pre فى الincrement و decrement (-- , ++)
- Unary plus وهو استخدام اشارة الجمع (+) مع عنصر واحد
- Unary negation وهى استخدام اشارة الطرح (-) مع عنصر واحد
- عند استخدام unary plus مع عنصر رقم او رقم ولاكن string يقوم برجاع الرقم بينما فى unary negation يقوم بارجاع الرقم ولاكن باشارة مختلفة
- فى حالة استخدام كلا الunary مع string من كلام يقوم بارجاع القيمة NAN
- استخدام كلا unary مع الارقام من النوع الhexadecimal يقوم بارجاعهم بالشكل العشرى المعروف مع تغير الاشار فى حالة كونو unary negation
- Null و false و " " عند طباعتهم مع unary تكون قيمتهم ب 0 بينما true تكون ب 1
- يوجد طريقة اخرى لتحويل الرقم الذى يكون string الى number باستخدام constructor وهو number كتالى Number("100")
- يمكن جعل العمليات الحسابية فى هذا الشكل a=a+2 كتالى a+=2 وهكذا فى باقى العمليات

Number method

- tostring() هو function يستخدم لتحويل الرقم الى string ولاكن يكتب بشكل مختلف كتالى (100).tostring()
- tofixed() هو function يستخدم لتقريب الاعداد العشرية الى اقرب عدد معين من الارقام العشرية وعدد هذه الارقام يوضع داخل () ويكتب كما يكتب to string بجانب ان القيمة العائدة منه تكون string
- Parseint() هو function يستخدم لارجاع قيمة الصحيحة من عدد موجود فى شكل string ويختلف عن الnumber() فى انه يقوم بارجاع int فقط ويكتب بالشكل الطبيعي للfunction حيث يوضع الرقم داخل ()
- Parsefloat() مثل الparseint ولاكن يرجع قيمة float
- Isinteger() هو function يوجد داخل Number object يقوم بارجاع قيمة true فى حالة كون القيمة الموجودة داخل () رقم صحيح و false فى حالة عكس ذالك ويكتب كتالى Number.isinteger(100)

- يجب مراعة ان الNumber object يبدا ب N capital
- isNAN() وهو function يقوم بالتحقق من كون القيمة الموجودة داخل ال () ستقوم بارجاع NAN ام لا وهو ايضا من الfunction الخاص Number object
- يوجد مجموعة من الfunction الموجودة فى Math object كتالى

result	example	The using	The function
20	Math.round(20.2)	يقوم بتقريب الى اقرب رقم صحيح	**round()**
21	Math.ceil(20.2)	يقوم بتقريب الى اعلى رقم صحيح	**ceil()**
20	Math.floor(20.9)	يقوم بتقريب الى اصغر رقم صحيح	**floor()**
2	Math.min(10,2,5,11)	يقوم باعطاء اصغر قيمة من القيم المعطاه	**min()**
11	Math.max(10,2,5,11)	يقوم باعطاء اكبر قيمة من القيم المعطاه	**max()**
8	Math.pow(2,3)	يقوم بعمل رقم مرفوع لاس	**pow()**
Any number	Math.random()	يقوم بانشاء رقم عشوائى	**random()**
22	Math.trunc(22.5)	يقوم بتجاهل الرقم العشرى ويرجع الصحيح	**trunc()**

- يرجى مراعة ان الMath object يبدا M capital

String methods

- String هو عبارة عن array من characters
- يمكن الوصول الى حرف فى الstring عن طريق الindex وهو الposition الخاص بالحرف فى الstring ويدا الindex من 0 ويكتب كتالى name[2] حيث ان name هو المتغير الذى يحمل الstring و 2 هو الindex وهذا يعنى الحرف لثالث فى الstring
- هناك function يقوم بالوصول الى حرف فى الstring وهو charAt() ويوضع الindex بين ال ()
- length هى method تقوم بارجاع عدد عناصر الarray وتكتب كتالى name.length
- الspace داخل الstring يتم حسابه على انه char
- trim() هو function يقوم باذالة المسفات الموجودة فى اول ونهاية الstring
- toUpperCase() هو function يقوم بتحويل الstring الى حروف capital
- toLowerCase() هو function يقوم بتحويل الstring الى حروف small
- indexOf() هى function تقوم بالبحث عن كلمة او حرف داخل string وتقوم بارجاع الindex وفى حالة الكلمة تقوم بارجاع اول index تبدا منه الكلمة
- الindexOf باخذ قيمتان الاولة وهى الحرف او الكلمة المراد البحث عنها (اجبارى) ويمكن ان ياخذ قيمة ثانية وهى الindex الذى سيبدا منه فى البحث عن الكلمة او الحرف (اختياري) فى حالة عدم وجود ثانى قيمة سيبدا البحث من بداية الstring
- lastIndexOf() هو function مثل الindexOf ولاكنه يبدا البحث من نهاية الstring
- slice() هو function يقوم بقطع جزء من string وياخذ قيمتان الاولة وهى الindex الذى سيبدا منه (اجبارى) والثانية وهى الindex الذى سيتوقف عنده (اختيارى) وفى حالة عدم كتبته سيستمر الى النهاية
- القيمة الثانية فى slice() لايقوم الfunction باخذها ولاكنه ياخذ ما قبلها
- عند وجود الindex بالسالب يكون العد من النهاية ويبدا العد ب -1
- repeat() هو function يقوم بتكرار variable او عدد او string عدد من المرات ويتم وضع عدد المرات داخل ال()
- split() هو function يقوم بتقسيم الstring وارجاع اجزائه فى array

result	example	Forms of split()
['ahmed']	Let x=”ahmed”; x.split();	**فى حالة عدم اضافة قيم للfunction يقوم باخذ string كامل ووضعه فى مصفوفة**
['a', 'h', 'm', 'e', 'd']	Let x=”ahmed”; x.split(“”);	**فى حالة اضافة "" بدون اضافة شى بداخلها يقوم بتقسيم جميع الcharacter ووضعها فى array**
['ahmed', 'Mohamed']	Let x=”ahmed Mohamed”; x.split(“ ”);	**فى حالة اضافة character داخل ال "" يقوم بقص العناصر التى بين هذا الcharacter ولا يضعه معهم**
['ahmed']	Let x=”ahmed Mohamed”; x.split(“ ”,1);	**يمكن اضافة قيمة ثانية لل function وهى اقصى عدد للاجزاء التى سترجع فى الarray**

- الفرق بين الsplit و الslice ان الslice يقوب بقص جزء واحد من الstring وارجاعه بينما split يقوم بقص اكثر من جزء ويرجعهم فى array
- substring() هو فنكشن مثل الslice(نفس القيم ونفس الوظيفة) ولاكن مع مجموعة اختلافات بسيطة مثل انه فى حالة كتابة القيمة الاولة (strat) اكبر من القيمة الثانية (end) سيقوم بعمل swap لهما وثانى فرق انه لايتعامل مع الارقام السالبة بحيث يعتبرها 0
- يوجد طريقة فى substring تحاكى طريقة الارقام السالبة وهى ان نضع length-n كا index

- Substr() هو function يختلف عن substring و slice حيث انه ياخذ اول قيمة هى index البداية وثانى قيمة هى عدد الcharcter التى سياخذها
- Substr يقبل الارقام السالبة
- Include() وهى function تقوم بفحص string من كونه يحتوى على جزء معين وترجع قيمة Boolean وتكتب كتالى name.include("ah")
- يمكن اضافة قيمة ثانية لinclude وهى قيمة الindex التى سيبدا منها فى البحث عن الجزء المطلوب
- startsWith() وهو function يقوم بارجاع قيمة Boolean حيث انه يبحث فى ما اذا كان الstring يبدا بحرف او كلمة معينة ام لا ويكتب مثل include
- يمكن اضافة قيمة ثانية لstartswith() وهى قيمة الindex التى سيبدا البحث منها
- Endswith() تقوم بفحص اذا كان الstring ينتهى بحرف او بكلمة معين ام لا ويمكن اضافة له قيمة ثانية وهى قيمة الlength التى سينتهى عندها
- فى الcomparison operator (== , < , > , <= , >= , !=) يقومو بمقارنة الvalue غير مكترثين بنوع data
- بينما فى مقارنات الidentical (=== , !==) تكون المقارنة فى value و data type
- فى java script الnot يتم من خلال علامة ! بينما الand تتم من خلال && و الor يتم من خلال عملية ||
- If function تستخدم لعمل مجموعة من المهام فى حالة تحقق من شرط معين و syntax الخاص بها كتالى if (condition){ action }
- else if تملك نفس ال syntax الخاص ب if بينما الelse لاتمتلك ال() لانها لاتحتاجه لانها بدون شرط
- Ternary operator هى طريقة اخرى تقوم نفس عمل الif وتكون الsyntax الخاص بها كتالى condition ? action if true : action if false
- هناك مايسمى بnullish operator وهو استخدام علامة ال|| او ؟؟ فى استخدام الvariable حيث تبعا لقيمة الvariable يتم الاختيار بينه وبين قيمة او متغير اخر بحيث عند استخدام || يقوم بتجاهل طرف اذا كان null او undefined او false value ولاكن ؟؟ تقوم بتجاهل فى حالة كون القيمة null او undefined فقط
- False value هى القيم التى تشبه الfalse مثل 0 او الstring بدون حرف
- الsyntax الخاص بswitch كتالى

```
switch(/*expression*/){
    case /*value 1*/ :
        /*action*/
         break;
    case /*value 2*/ :
        /*action*/
         break;
    default:
        /*action*/
}
```

- يلزم كل case ينتهى ب break لانه فى حالة عدم وضعها سيقوم بذهاب الى جميع الحالات الاخرة
- Default case يتم تنفزها فقط فى عند عدم تنفيذ باقى الcases
- يمكن وضع الdefault case فى اى مكان فى switch ولاكن يجب وضع break

Array

- انشاء الarray يشبه انشاء variable ولاكن يتم وضع فيه اكثر من قيمة داخل [ ] ويفصل بينهم ب , كتالى let names=["ahmed","Mohamed"];
- يتم استخدام كل element فى الarray باستخدام الindex
- يمكن وضع array داخل array اخرى (nested array)
- يمكن تغير قيمة فى الarray عن طريق كتابة اسم الarray بالindex ثم علامة = ثم القيمة الجديدة كتالى names[0]="ali"
- يمكن استبدال nested array بعنصر او العكس
- isArray() هى function موجودة داخل Array object يقوم بتحقق من كون القيمة داخل ال()هى Array ام لا كتالى Array.isArray(names)
- يمكن اضافة عنصر الarray فى java script ويتم عن طريق كاتبة الindex التالى واضافة له القيمة
- فى حالة اضافة عنصر فى index ليس التالى يتم جعل العناصر التى لم تضاف كا empty
- لاضافة عنصر جديد فى الarray بدون الحاجة الى عد عناصر الarray فى كل مرة نقوم بوضع الindex ب length بسبب كون length دائما يذيد عن اخر index ب 1 وهذه الطريقة هى الافضل
- unshift() هو function يقوم باضافة عناصر الى اول الarray وتكتب العناصر بداخل ( )
- push() هو function يستخدم ابضا فى اضافة عناصر الى الarray ولاكن فى النهاية وتكتب العناصر داخل ()
- shift() هو function لازالة اول عنصر فى الarray ولايكتب شئ داخل () بالاضافة الى انه يقوم بارجاع القيمة المحذوفة لاستخدمها
- pop() هى ايضا function لازالة عنصر من المصفوفة ولاكن من النهاية ويقوم ايضا بارجاعه ولاياخذ قيم داخل ()
- indexOf() function و lastIndexOf() يمكن استخدامهما ايضا فى الarray
- include() function ايضا يتم استخدمها فى الarray
- indexOf او lastIndexOf عندما لاتوجد القيمة المراد البحث عنها تقوم بارجاع -1
- sort() هى function تستخدم لترتيب الarray بشكل ابجدى (يشمل الارقام)

- reverse() هى function تقوم بعكس ترتيب الarray
- slic() function يتم استخدمها ايضا فى الarray
- splice() هى function تقوم بحذف او استبدال جزء فى الarray
  - فى حالة انها اخذت قيمتان فان اول قيمة هى index الخاصة باول العناصر المحذوفة وثانى قيمة هى عدد العناصر الذى ساتحذف ابادا من اول قيمة
  - فى حالة انها اخذت ثلاث قيم او اكثر فان اول قيمتان كما هما والثالثة و فيما فوق هى القيم التى ستضاف
- concat() هى function تقوم بعمل ربط بين اكثر من array وعدد الarray الموضوعة داخل () غير معروف كما يمكن اضافة عناصر بجانب الarray ويكتب الfunction كتالى all_array=array1.concat(array2,array3,"element")
- Join() هو function يقوم باخذ array واراجه فى شكل نص مفصول بينه بفاصل معين يوضع بين ( ) بحيث
  - عند ترك ال() فارغة يفصل بينهم ب ,
  - عند وضع string بدون character("") يقوم بجمع عناصر الarray بدون فاصل

Looping

- يتم كتابة الsyntax الخاص بfor loop كتالى

```
for(/*initialization*/ , /*condition*/ , /*increment or decrement*/){
    /*the code*/
}
```

- يمكن استخدام break فى الloop وهى تجعله يتوقف
- Continue يمكن استخدمها ايضا داخل الloop وهى تجعله يتخطى خطوة
- يمكننا استخدام مايسمى بlabel فى الloop وهو يستخدم كاسم للloop
- يضاف قبل الloop ثم توضع علامة : ثم الloop كتالى

```
frist_loop:for(){}
```

- عندما يتم استخدامه فانه ياستخدم مع continue و break مثل ;continue frist_loop او ;break frist_loop
- الloop while لا يختلف كثيرا فى الsyntax عن for loop حيث ان فى while داخل ال() يوضع الcondition فقط

Function

- الfunction هى مجموعة اكواد تؤدى مهمة معينة وتاتى اهميتها فى عدم كتابة الكود اكثر من مرة
- Built in function هى الfunction القادمة مع اللغة بينما user defined function هى الfunction التى تصنع بواسطة الusers
- الsyntax الخاص بانشاء function يتم اولا بوضع كلمة function ثم اسمه ثم () ثم {} كتالى

```
function name_of_function(/*parameter*/){
    /*the code*/
}
```

- الparameter هى المتغيرات التى ستحمل القيم التى سيعطيها الuser للfunction
- توضع الparameter داخل () ويمكن ان لايحتاج الfunction لparameter
- بعد انشاء الfunction يلزم عمل call له لكى يتم تنفزيه
- الparameter ما هى الاvariable ولاكن عند انشائها لاتوضع كلمة let قبلها
- يمكن ارجاع القيمة من ال functionعن طريق كلمة return وهى توضع فى اخر سطر وتوضع بعدها القيمة التى سترجع
- اى code فى الfunction بعد جملة الreturn لا يتم تنفيزه
- عند وضع الreturn بدون قيمة بعدها تقوم بارجاع undefined
- وعند وجود parameter بدون اضافة قيمة بها فانها تاخذ قيمة default بundefined
- يمكن اضافة قيمة default يقوم الfunction باستخدامها فى حالة عدم وضع قيمة لها وتتم عن طريق وضع قيمة له داخل ال() عند انشاء الfunction كتالى

```
function name(p1="default value"){}
```

- عند احتياج عدد غير معروف من الparameter فى الfunction يتم اللجوء الى ما يسمى ب reset parameter وهو يكون array من الparameters
- يتم انشاء الreset parameter عن طريق وضع ثلاثة من علامة . قبل اسم الparameter
- الreset parameter يكون فى شكل parameter واحد ولاكن فى الحقيقة يكون عبارة عن array من الparameter
- الreset parameter يلزم ان يكون اخر parameter
- يمكن وضع function call قبل الfunction declaration بدون مشكلة
- يمكن وضع الfunction بداخل variable ولاكن تكون بدون اسم وعند استخدامها نستخدم اسم الvariable كتالى

```
let variable= function(){}
```

- عند وضع الfunction فى variable يلزم ان يكون الcall بعد عملية الdeclaration

- يطلق على الfunction الموجود داخل variable اسم Anonymous function
- يمكن اضافة function بداخل function (nested function) ويمكن استخدامها مع return كا value
- Global variable هى متغيرات قابلة للاستخدام بواسطة جميع الاكواد
- Local variable هى متغيرات قابلة للاستخدام بواسطة الاكواد الموجودة فى نفس الscope
- فى حالة وجود local و global بنفس الاسم يتم استخدام الlocal من الاكواد الموجودة مع نفس الscope الخاص ب local ويتم تجاهل الglobal
- فى حالة استخدام variable تم انشائه بواسطة var ويوجد منه global و local سيتم عمل over writing اى ان قيمته ستتغير ولن يحدث ذالك فى let
- هناك طريقة لكتابة الfunction تسمى الarrow function وهى تتم باضافة function الى variable ويتكون الsyntax الخاص بها كتالى

```
let names=(/*parameters*/)=>/*what will return*/;
```

- الhigher order function هى الfunction التى تاخذ function كا parameter
- map() هى higher order function تعتبر كا method لتعامل مع الarray ولاكنها تتعامل مع نسخة من الarray وليست الarray الحقيقية
- الmap تاخذ قيمتان الاولى وهى الfunction التى سيكتب فيها ما سيحدث للarray وثانى قيمة وهى ما يسمى ب this parameter
- الfunction التى تاخذها الmap كا قيمة تاخذ 3 parameter الاولى تشير الى الelement والثانية تشير الى الindex والثالثة تشير الى الarray
- الparameter الاول فى الfunction التى تاخذها الmap هو الاجبارى الوحيد بنما باقى الparameter اختيارى ويتم كتابة الmap كتالى

```
let newArray=array.map(function(el,ind,arr){
    /*the code that will happen on the array and will save on newArray by return statement */
},10);
```

  - الnewArray هو الarray الذى ستحفظ فيه نتيجة الmap function
  - الarray هو الarray الذى ستطبق عليه الmap function
  - الthis parameter هى 10 الthis parameter
  - اول parameter فى الfunction الموجودة داخل الmap (el) دائما يدل على الelements الخاصة ب الarray وهو اجبارى
  - ثانى وثالث parameter فى الfunction الموجودة داخل الmap (ind,arr) يدلان على الindex و array المستخدمة
  - القيم التى سترجع فى return هى من ستضاف الى الnewArray
  - يمكن استخدام شكل الarrow function بدلا من الشكل الطبيعى للfunction الموجودة داخل الmap
  - يمكن وضع function جاهزة (تم انشئها مسبقا) فى الmap
- فى حالة التعامل مع string بالmap يلزم وضعه فى array ويفصل بين كل حرف والاخر عن طريق استخدام split("") ويلزم وضع بداخلها "" لكى تفصل بين كل حرف والاخر
- الfilter() هو higher order function مسؤل عن تطبيق عمليات على العناصر الarray والتحقق من شرط معين فى حالة كون الشرط true يقوم بارجاع العنصر
- يمتلك الfilter() نفس syntax الخاص ب map ولاكن فى جملة الreturn نضع الشرط واذا تحقق يستم ارجاع الelement
- reduce() هى ايضا higher order function مثل الmap اى انها تتعامل مع الarray ولاكنها تختلف فى الوظيفة والsyntax
- وظيفة الreduce مثل map ولاكنها تقوم بارجع قيمة واحدة فى النهاية مثلما ان filter هى map ولاكنها ترجع العنصر فى حالة تحقق شرط
- يختلف syntax الخاص ب الreduce كتالى

```
let variable=array.reduce(function(acc,el,ind,arr){
    /*return statement with the operations that will happen on accumulator */
},10);
```

  - فى الreduce يتم اضافة parameter جديد الى الfunction الموجودة بداخله وهو الaccumulator وهو القيمة التى سيتم تنفذ عليها العمليات وارجعها فى النهاية وهو اجبارى
  - القيمة الثانية فى reduce function والتى تمثل الـ10 فى هذا المثال هى الinitialvalue وهى القيمة التى ستعطى الى الaccumulator فى بداية عمل الfunction
  - فى حالة عدم وجود الinitialvalue سيتم اضافة اول قيمة فى الarray المستخدم الى الaccumulator وبذالك فى بداية الfunction يصبح الelement parameter يشير الى ثانى عنصر و الindex parameter يحمل قيمة 1
  - القيمة النهائية فقط للaccumulator هى من سيتم ارجعها من الreduce function
  - فى كل مرة تحدث العملية التى فى الreturn فهى تحفظ فى الaccumulator على سبيل المثال الجملة التالية return acc+el ففى كل مرة تقوم الreduce بدخول الى عنصر جديد فى الarray تقوم بجمعه على الacc وحفظه فى الacc اى انها مثل كتابة acc=acc+el وفى النهاية تقوم برجاع قيمة الacc النهائية
  - الparameter الاجبارى الوحيد فى الreduce هو الaccumulator ولايلزم كتابة الelement parameter او استخدامه كما فى هذا المثل عند كتابة return acc+1 حيث انه فى كل مرة من مرات العناصر الموجودة فى الarray سيقوم يذياة الacc ب1 ولن يستخدم الel
- forEach() هى ايضا higher order function لها نفس الsyntax الخاص بmap و filter ولاكنها تختلف عنهم فى انها لاتحتاج الىarray جديدة كما انها لاتقوم بارجاع شئ (undefined)
- الforEach تقوم بالمرور على كل عنصر فى الarray وتقوم باستخامه فى action معين ولاكنها لاتغير قيمته او انها تقوم باكسابه خاصية معينة
- يجب ان تكون الE فى forEach حرف capital

Object

- الobject هو واحد من انواع الdata الذى يستخدم فى الكثير من الاشياء ومنها انشاء نوع بينات جديد
- يتكون الobject من شيئان وهما
  1. Properties وهى data
  2. Methods وهى function
- الwindow و console و document هم امثلة على الobject
- Location هو مثال لnested object وهوobject موجود داخل object اخر وهو الwindow
- الsyntax الخاص بobject كتالى

```
let object_name={
    //properties
    name: "ahmed",
    age: 19 ,
    //methods
    say_hello : function(){console.log("hello ahmed")}};
```

  - توضع الproperties و methods داخل ال{ } وينتهى الobject ب ;
  - ترتيب الproperties او الmethod داخل الobject غير مهم حيث يمكن وضعهم فى اى مكان
  - يكتب العنصر (properties او method) ثم توضع علامة : ثم قيمته
  - يفصل بيت كل عنصر والاخر ب علامة ,
- لعمل access على عنصر فى الobject نكتب اسم الobject ثم علامة . ثم اسم العنصر
- يمكن اضافة اسم العنصر الخاص بالobject داخل " " وذالك فى حالة تسميته بالطرق الخطئ لتسمي (مثل وضع مسافة فى الاسم وهكذا)
- يمكن عمل access على عناصر الobject من خلال وضعه داخل ال[ ] وهذه الطريقة مفيدة فى حالة ان اسم العنصر موضوع داخل " "
- يمكن استخدام ال[ ] فى حالة اننا نريد عمل access على عنصر فى الobject ولاكن اسم العنصر موجود كا قيمة فى variable على سبيل المثال Let x="age" و الage هو اسم عنصر فى الobject فيمكننى ان اقوم بعمل access عليه كتالى object_name[x]
- عند عمل access من داخل الobject يلزم ايضا استخدام طرق الaccess (. او [ ])
- يمكن عند انشاء الobject ان نتركه فارغ
- يمكن اضافة عنصر الى الobject من خارجه عن عمل access له باى طريقة ثم اضافة = ثم قيمته وفى حالة كان العنصر المضاف موجود فان قيمته تتغير
- يمكن انشاء object بطريقة new keyword كتالى

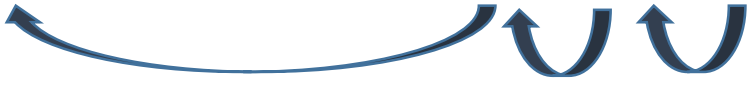
```
let object_name=new Object();
```

  - Object() هنا هو constructor يقوم بانشاء object
- This هى pointer فى الjava script تشير فى الوضع الطبيعي الى الwindow object
- الthis فى الوضع الطبيعي تشير الى الobject الذى يوجد فيه الfunction الذى يتم استخدامه او الvariable الذى يحمل الfunction
- توجد طريقة اخرة لانشاء object وهى عن طريق create method وهى function داخل الObject object حيث تكتب كتالى let x=Object.create() ويمكن ان نتركها فارغة اى اننا نضع { } فارغة داخل ال( ) او اننا نضع نموذج داخل ال( )
- النموذج الموضوع داخل ال( ) هو عبارة عن object اخر وهذه العملية تشبه الى حد كبير عملية الوراثة فى اللغات الاخرى
- يمكن للobject الذى وضع نسخة من قيم object اخر بداخله استخدام جميع عناصر الobject الاخر والتعديل عليها
- يفضل عند انشاء object استعمال this بدل اسم الobject عند استعامل عناصره بداخله لانه ف حالة حصول عملية الوراثة للobject تستخدم العناصر بشكل سليم
- هناك طريقة اخرى لانشاء object وتحقيق فيها الوراثة عن طريق assign method وهى تختلف عن الcreate حيث انها تختلف فى الparameter
- الparameter الخاصة بالassign method تبدأ بtarget object وهو الobject الذى ستضاف اليه جميع العناصر من الobjects الاخرى قبل ان تضاف الى الobject النهائى وباقى الobject تضاف بعده كتالى

```
let fob=Object.assign(tob,ob1);
```

  - tob هو الtarget object هو مثل اى Object يحتوى على عناصر
  - يتم اضافة كل الobject التى بعد target object اليه ثم يوضع الtarget object فى fob
  - فى حالة انه يوجد عناصر بنفس الاسم فى الobject الاخرى وموجودة ايضا فى tob يتم تغيير قيمتها فى tob ووضعها بقيمة الذى يسبقه كتالى

```
let fob=Object.assign(tob,ob1,ob2);
```



  - فى كل مرة تحدث هذه العملية يتم اضافة العناصر الموجودة فى object فى الذى يليه وذا تواجدت نفس العناصر يتم اخذ الموجود فى اول object
  - يمكن وضع الtarget object فارغ حيث يوضع مكانه { } فقط

Dom

- Stands for document object model
  - عند انشاء الصفحة يقوم الbrowser بانشاء Model لها عبارة عن object وهو الdocument
  - Dom تستخدم لتحكم فى الdocument عن طريق الjava script

Get the Element

- للوصل الى العنصر الموجود فى html نستخدم واحد من الmethod الموجودة داخل document object ومنها
  1. getElementById() وهو method تقوم بالوصول الى العنصر عن طريق الid attribute
     - يقوم بارجاع عنصر واحد فقط بسبب ان الid لايمكن ان يتكرر
  2. getElementsByTagName() تستخدم للوصول الى جميع العناصر باستخدام tag name هو اسم الtag الموضوع بين <> فى html
     - الtag name يوضع بين " " فى ()
     - getElementsByTagName سيقوم بارجاع array موجودة فيها العناصر ويمكن استخدمها بواسطة الindex
  3. getElementsByClassName() ويستخدم للوصول الى العناصر بالقيمة الموجودة فى class attribute
     - سيقوم برجوع بعنصر او اكتر فى شكل array
  4. querySelector() يستخدم ايضا للحصول على العنصر ولاكن باى selectors مثل id او class وغيرها
     - توضع علامات مميزة داخل " " قبل وضع القيمة مثل علامة . توضع قبل الclass او # قبل الid وهكذا
     - تقوم بارجاع عنصر واحد فقط حتى وان كان الselector موجود فى اكتر من عنصر
  5. querySelectorAll() مثل querySelector ولاكن يمكنها ارجاع اكثر من عنصر
  6. title هو ايضا method فى document object ويقوم بارجاع بعنوان الصفحة
  7. Body يقوم بارجاع body tag كامل
  8. Form سيقوم بارجاع كل الform الموجودة فى الصفحة ويمكن استخدام الindex لاستخدام واحد منهم
  9. Links تقوم بارجاع جميع الlink الموجودة فى الموقع
  10. Images تقوم بارجاع جميع الصور الموجودة فى الصفحة
- يمكن الاتيان بالعنصر وتخزينه فى variable ومن ثم استخدامه
- innerHTML هو function يستخدم لرجاع الhtml الموجود داخل العنصر كما هو (tags و text)
- textContent هو function يستخدم لارجاع text الموجود فى العنصر كما يظهر فى الصفحة ولا يقوم بارجاع tags
- يمكن استخدام الInnerHTML و textContent بجانب الاتيان بمحتوى العنصر تعديل محتوى العنصر حيث تضاف علامة = ثم يضاف المحتوى الجديد حيث ان المحتوى القديم يتم اذالته وتتم كتالى
  - فى innerHTML سيتم وضع المحتوى بين " " ويتم معاملته على انه html code مثل الكتابة فى الsource code تماما
  - فى textContent سيتم معاملة المحتوى الموجود بين " " على انه نص حتى لو كان يحتوى على html code

Get the Attribute

- للوصول الى اى attribute فى tag نقوم بالوصول الى العنصر اولا ثم نضع علامة . ثم الattribute ويمكننا تغير قيمته عن طريق اضافة = ثم القيمة وهذه الطريق تضيف attribute ان لم يكن موجود ويمكن استخدمها فى اضافة اى شكل من اشكال الselectors مثل الid و الclass
- getAttribute() هو function يستخدم مع العنصر لكى ياتى بقيمة attribute معين ويوضع الattribute داخل الـ( ) بين الـ" "
- setAttribute() هو function يستخدم مع العنصر لتعديل على قيمة attribute وياخذ قيمتان الاولة هو الattribute والثانية هى القيمة الجديدة
- attributes هو function يقوم بارجاع كل الattribute الموجودة فى العنصر
- hasAttribute() هو function يقوم باختبار من وجود attribute فى عنصر ام لا وياخذ قيمة واحدة وهو الattribute داخل () ويوضع بين " "
- removeAttribute() تقوم بحذف attribute من عنصر
- hasAttributes() تقوم يتحقق من وجود اى attribute او لا لذالك فهى لاتاخذ قيم
- createElement() هو function موجود فى document object يستخدم لانشاء tag جديد (غير موجودة) او قديمة من الhtml كتالى :

```
document.createElement("AAA");
```

  - بعد كتابة هذا السطر يمكن استخدام tag جديدة كانت غير موجودة وهى<AAA> </AAA>
- يمكن وضع الtag الجديد فى variable لضافة بعض الخصائص له كتالى

```
let new_tage =document.createElement("AAA");
```

- createAttribute() هى ايضا function موجودة فى الdocument object تقوم بانشاء attribute جديد
- يمكن استخدام setAttribute() ايضا لانشاء attribute غير موجود و وضع له قيمة
- setAttributeNode() هو function فى document object تقوم بجعل الattribute كا node
- createTextNode() هو function فى document object يقوم بجعل الtext كا node
- appendChild() هو امر يستخدم لوضع text او variable يحتوى على element داخل اى tag ولاكن الtext يلزم ان يكون Node كتالى

```
document.write(`<div id="div"></div>`); //create div
```

```
let text=document.createTextNode("hello ahmed"); //make a text node
let new_element=document.createElement("aaa"); //create a <aaa></aaa>
new_element.appendChild(text); //put the text in aaa element
document.getElementById("div").appendChild(new_element); //put aaa in the div
console.log(document.getElementById("div"));
```

- النتيجة الخاص بهذا الكود كتالى : <aaa>hello ahmed</aaa>

- createComment() هو function فى document object يستخدم لعمل comment ويمكن وضعه فى متغير او عمل عليه appendChild

Children

- عند وجود عنصر بداخل عنصر فى الhtml العنصر الموجود بالداخل يسمى child
- يوجد function وهو children وهو يقوم بالتيان بجميع العناصر الموجودة داخل عنصر كتالى element.children حيث هنا سيقوم بارجاع جميع العناصر الموجودة بداخل العنصر الذى تم تخزينه فى element variable
- يمكن استخدام الindex مع الchildren لانه يقوم بارجاع array من العناصر
- childNodes هو function يقوم بارجاع جميع العناصر الnode (text ,element ,comment) الموجودة داخل element معين ويمكن استخدام فيها الindex ايضا
- fristChild هو function يقوم بارجاع اول child غير مكترس بنوعه اى ممكن ان يكون text او comment او element
- lastChild هو function يقوم بارجاع اخر child غير مكترس بنوعه اى ممكن ان يكون text او comment او element
- fristElementChild هو function يقوم بارجاع اول element child
- lastElementChild هو function يقوم بارجاع اخر element child

Events

- الevent يمكن كتابتها فى html او فى JavaScript كتالى :
  - فى الhtml يكتب كا attribute كتالى <button onclick=”console.log(clicked)”>button</button>
  - فى الjava script يتم استخدمها على عنصر ويضاف اليها function كتالى element.onclick=function ( ) { }
- onclick هو event يقوم بعمل فنكشن فى حالة الضغط على العنصر
- oncontextmenu هو event يقوم يتنفيذ function فى حالة الضغط على العنصر بالذر اليمين للفارة
- onmouseenter هو event يقوم بتنفيذ امر فى حالة الوقوف على العنصر ب الفارة
- onmouseleave هو event يقوم بتنفيذ امر فى حالة كانت الفارة على عنصر ثم اذيلت من عليه
- onload هو event يقوم يقوم بتنفيذ امر فى حالة اكتمال تحميل عنصر معين
- onscroll هو event يقوم بتنفذ امر فى حالة حدوث scroll فى عنصر معين وغالبا ما تستخدم مع الwindow
- onresize هو event يقوم يتنفيذ امر معين فى حالة تغير حجم العنصر
- onfocus هو event يقوم بتنفيز امر معين فى حالة الضغط على حقل ادخال
- onblur هو event يقوم بتنفيز امر معين فى حالة الخروج من الكتابة فى حقل ادخال
- onsubmit هو event يقوم بتنفيذ امر معين فى حالة الضغط على ذر submit
- يوجد الكثير من الevents ولاكن السابقين كانو الاشهر
- امر preventDefault() يقوم بايقاف الحدث الذى مخصص الى العنصر بشكل طبيعي
- غالبا ما يوجد function من كل event مثل انه يوجد function يسمى click() وهو يقوم بعمل click على العنصر
- عند كتابة نفس الevent لنفس العنصر اكتر من مرة فان ما سيتم هو استخدام اخر event
- addEventListener() هى function يتم اضافتها لعمل event على عنصر معين و تاخذ قيمتان الاولة وهى الevent بدون on ويوضع بين " " والقيمة الثانية هى الfunction التى ستتم عند حدوث الevent
- تستخدم addEventListener فى حالة اننا نريد تطبيق اكثر من event من نفس النوع على نفس العنصر
- افضل استخدام لـaddEventListener وهو اضافة event لعنصر ليس موجود بعد فى الصفحة

Class list

- فى الclass attribute يمكن كتابة اكثر من class كتالى <div class=”class1 class2 class3”> عن طريق وضع مسافة بين كل class
- classlist هى function تقوم بارجاع object من الclasses الخاصة بالعنصر كتالى element.classlist
- يمكن استخدام length بعد الاتيان بـclass list لمعرفة عدد الclasses
- contain() هو function يستخدم مع class list لتحقق من وجود class معين فيها ام لا ويرجع قيمة Boolean ويوضع الclass بين " " فى ( )
- Item() يستخدم للاتيان بـclass من الclass list عن طريق الindex ويكتب الindex بين " " داخل ال( )
- add() هو function يستخدم مع الclass list لاضافة class او اكثر ويوضع كل class بين " " ويفصل بين كل class والاخر بعلامة ,
- remove() هو function يستخدم مع الclass list لازالة class او اكثر ويوضع كل class بين " " ويفصل بين كل class والاخر بعلامة ,

- ()toggle هو function يستخدم مع الclass list ويضاف له قيمة واحدة بين " " هذه القيمة اذا لم تكن موجودة فى الclass list سيقوم باضافتها واذا كانت موجودة سيقوم باذالتها
- يتم استخدام الclasses فى عمليات كثير على العناصر وليس فقط تحديدها

CSS styling

- يمكن استخدام style كا function مع العنصر لارجاع object عناصره هى تنسيقات الcss
- Color هى function يمكن استخدمها مع الstyle لاضافة اللون الى العنصر عن طريق وضع علامة = ثم اللون بين " "
- يمكن اضافة اكثر من عنصر css معا بستخدام cssText وهى function تستخدم كتالى element.style.cssText =”color: green ,opacity:0.9” حيث ان كل عنصر يوضع وتوضع بعده قيمته وبينهما علامة : و توضع علامة , بين كل عنصر والاخر
- ()removeProperty هى function تستخدم مع الstyle وتقوم بحذف css property وتوضع هذه الproperty بين " " فى ( )

Deal With Element

- ()before هو function يستخدم مع العنصر لاضافة text او عنصر اخر قبله فى الصفحة ويوضع الtext او العنصر فى ( )
- ()after هو function يستخدم مع العنصر لاضافة text او عنصر اخر بعده فى الصفحة ويوضع الtext او العنصر فى ( )
- () prepend هو function يستخدم مع العنصر لاضافة text او عنصر اخر فى البداية داخل العنصر فى الصفحة ويوضع الtext او العنصر فى ( )
- ()append هو function يستخدم مع العنصر لاضافة text او عنصر اخر فى النهاية داخل العنصر فى الصفحة ويوضع الtext او العنصر فى ( )
- ()remove هو function يستخدم لازالة العنصر تماما من الصفحة

Traversing

- العناصر الchild بنسة لبعضهم يطلقون عليهم sibling (اشقاء)
- nextSibling هو function يستخدم مع عنصر لكى يقوم بارجاع الشئ التالى بعد هذا العنصر (comment ,text ,element)
- nextElementSibling هو function يستخدم مع عنصر لكى يقوم بارجاع العنصر التالى بعد هذا العنصر (just element)
- previousSibling هو function يستخدم مع عنصر لكى يقوم بارجاع الشئ السابق ل هذا العنصر (comment ,text ,element)
- previousElementSibling هو function يستخدم مع عنصر لكى يقوم بارجاع العنصر السابق ل هذا العنصر (just element)
- parentElement هو function يستخدم مع العنصر الchild لكى يقوم بارجاع العنصر الparent

Cloning

- ()cloneNode هو function يستخدم مع العنصر لكى يقوم باخذ نسخة منه وياخذ قيمة واحد وهى true او false وهى تحدد اذا كان سينسخ العناصر التى بداخله ايضا ام لا وفى حالة عدم كتابة قيمة سيتم التعامل معها كا false by default
- الcloneNode يقوم بنسخ الattribute الخاصة بالعنصر ايضا كما هى وهنا يمكن ان يحدث خطا وهى وجود نفس الid فى كلا العنصرين الاصلى و النسخة لذالك يفض تغير الid الخاص بالنسخة او العنصر

Bom

- Stands for browser object model

- الdocument object هى جزء من الwindow object
- اى global variable او object او function هو عنصر من window object
- فى حالة الكتابة فى الglobal scoop يمكن عدم وضع كلمة window عند استخدام عنصر منها
- alert() تقوم باخراج رسالة فى حالة الدخول الصفحة ولاتستخدم لانها توقف تحميل الصفحة ولاكن يمكن استخدمها احيانا فى حالة اظهار رسال للuser عند اغلاقه للصفحة
- Confirm() مثل الalert ولاكنها تستخدم لسؤال وتقوم بارجاع قيمة true او false
- Prompt() هى ايضا مثل الalert ولاكنها تقوم باظهار رسالة وتطلب ادخال اجابة وتقوم بارجاع هذه الاجابة
- لا يتم استخدام كلا من الalert او confirm او الprompt لانه يوجد بدائل افضل بكثير
- setTimeout() هى higher order function تستخدم لتنفيز function بعد مرور وقت معين بmile second تاخذ قيمتان الاولة وهى الfunction والثانية هى الوقت ويمكن كتابتها كتالى

```
setTimeout(function() {},1000)
```

  - يمكن اضافة parameter او اكثر الى الsetTimeOut ويتم يوضعهم بعد الوقت
  - القيمة الثانية هى الوقت ويكتب بmile second اى ان 1000 تساوى ثانية واحدة
- clearTimeout() هو function عند تشغيله يقوم بايقاف settimeout معين وهو ياخذ قيمة واحدة وهى الvariable الذى يحمل الsetTimeOut
- setInterval() هى function مثل الsetTimeOut ولاكنها تقوم بتكرار الكود مرارا وتكرارا بفاصل زمنى بينهم ويكتب تمام مثل setTimeOut
- لايتم ايقاف setInterval الى بعد استخدام clearInterval() وهو function ياخذ قيمة وهى الvariable الذى يحمل الsetInterval
- Open() هى function فى الwindow object يقوم بفتح صفحة وياخذ تقريبا اربع قيم وجميعهم اختيارى وهم كتالى :
  1. اول قيمة تاخذ الURL الخاص بالصفحة المراد فتحها ويوضع بين " " ويمكن تركها فارغة وهذا سيؤدى الى فتح صفحة فارغة تماما
  2. ثاني فيمة وهى الtarget ويمكن تركها فارغة وتكون قيمتها الdefault ب _blanck
  3. القيمة الثالثة وهى كثير من الoption الخاص بالصفحة مثل حجم الwindow وكثير من الخصائص
  4. القيمة الرابعة وهى الخاص بhistory replacement وتاخذ قيمة true او false
- Close() وهى ايضا function موجودة فى الwindow تستخدم لاغلاق الصفحة ولاكن الصفحة التى فتحت من الopen function
- Stop() هى function تستخدم لايقاف تحميل الصفحة
- Print() هى function تستخدم لعمل print للصفحة كاملة
- Focus() هو function يستخدم مع متغير يحمل صفحة لعمل focus عليها ويتكب كتالى win1.focus() حيث win1 هو variable يحمل صفحة
- scrollTO() او scroll() هم functions يقومو بعمل نفس الشئ وياخذو نفس القيم وهى قيم الx و الy ولاكن الفرق ان scroll غير مدعومة فى متصفح safari الخاص بشركة apple
- scrollto و scroll يقومو بذهاب الى مكان معين فى الصفحة بينما يوجد function اخرى تقوم بعمل scroll ايا كان المكان فى الصفحة وهى scrollBy()
-

Location object

- هو object موجود داخل الwindow object ويستخدم فى الاتيان بobject خاص بكل المعلومات الخاص بlink الموقع
- يمكن كتابته window.loaction او location مباشرة
- href هو عنصر فى location object يقوم بارجاع لينك الصفحة
- عند اضافة قيمة لhref سيقوم المتصفح بذهاب الى الموقع الخاص بهذه القيمة وتكون هذه القيمة عبارة عن link وعند الذهاب الى الموقع سوف تسجل هذا فى الhistory الخاص بالbrowser
- يمكن ان تكون هذه القيمة id خاص بعنصر عن طريق اضافة قيمة وهى تبدا ب#/ ثم الid كتالى /#id1
- Host هو عنصر فى الlocation يقوم بارجاع اسم الموقع بجانب الport المفتوح منه
- hostName هو عنصر ايضا موجود فى الlocation يقوم بارجاع اسم الموقع فقط
- يمكن اضافة قيمة للhost او للhost name ومنها نقوم بتغيير اسم الموقع فقط بدون تغير المسار
- Protocol عنصر فى location يقوم بارجاع الprotocol المستخدم فى الموقع وغالبا ما يكون http او https
- يمكن اضافة قيمة ايضا للprotocol لتغير الprotocol المستخدم ولاكن الكثير من المواقع تجبر الuser على استخدام protocol واحد فقط
- Hash هو عنصر فى location يقوم بارجاع الhash القادم منه الuser
- Reload() هى function داخل الlocation تقوم بعمل reload للصفحة
- Replace() هى function داخل الlocation تقوم بذهاب الى Link معين ويوضع داخل ال() فى شكل string اى انه بين " " ولاكن عند استخدامه الصفحة الحالية لايتم تسجلها فى الhistory اى انه لانستطيع الرجوع اليها وهذا عكس تغيير قيمة الhref
- Assign() هى function فى الlocation تستخدم مثل الreplace ولاكنها تضع الصفحة الحالية عند استخدامها فى الhistory اى انها مثل تغيير قيمة الhref

History

- الhistory هى object فى الwindow تستخدم لتحكم فى الhistory
- Length يقوم بارجاع عدد الصفحات التى تمت زيارتها من ضمنهم الصفحة الحالية
- ()Back هى function فى الhistory تستخدم للرجوع الى الصفحة السابقة وتكتب كتالى history.back()
- ()Forward هى function فى الhistory تستخدم لذهاب الى الصفحة التالية فى الhistory وفى حالة عدم وجود صفحة تالية تقوم بارجاع undefined
- ()Go هى function فى الhistory تستخدم لذهاب الى صفحة فى الhistory حسب الposition وهو قيمة عددية توضع داخل () وفى حالة الposition كان 0 يقوم بعمل reload وفى حالة كان الposition ب سالب واحد فا السالب يدل على الرجوع الى الوراء والواحد يدل على الرجوع صفحة وفى حالة كان الرقم موجب يدل على التقدم الى الامام

Local storage

- Local storage هو object موجود داخل الwindow object يقوم يتخزين عناصر والقيم الخاص بها لاستخدمها بعد ذالك ويمكن ان تكون هذه القيم من اختيار الuser
- ()setItem وهو function فى الlocalStorage يقوم باضافة عنصر والقيمة اخاص به ويكتب كتالى

```
window.localStorage.setItem("name","ahmed");
```

  - يمكن ازالت الwindow من الكود
  - اول قيمة تاخذها setItem هى العنصر وثانى قيمة تاخذها هى قيمة العنصر
- الlocalStorage هو object لذالك يمكن استخدام طرق اضافة او اذالة عنصر عن طريق استخدام علامة . او [ ]
- ()getItem هى function تستخدم للاتيان بقيمة عنصر فى الlocalStorage ويوضع اسم العنصر بين" " فى ( )
- ()removeItem هى ايضا function فى localStorage تستخدم لاذالة عنصر ويوضع العنصر فى ال( ) بين " "
- ()clear هى function فى الlocalStorage تستخدم لازالة جميع العناصر فى local storage
- ()Key هو function فى الlocalStorage يستخدم للاتيان بعنصر فى localStorage بستخدام الindex والذى يوضع فى ( )
- عند اغلاق الصفحة فان الlocalStorage تظل كما هى
- يمكننا رؤية جميع الLocalStorage من خلال خانة الapplication فى المتصفح

Session Storage

- هو مثل الlocal storage ولاكن كل البينات تظل فقط لنفس الجلسة وليس دائما
- الجلسة تنتهى عندما تغلق tab
- عند وجود اكثر من tab فان كل واحدة تكون بsession مختلفة
- يمكن استخدام فيها جميع الfunction التى فى local storage ولاكن مع الsessionStorage object

Set data type

- الset هى نوع من البينات يشبه الarray وهو عبارة عن object يقوم يتخزين عناصر ولاكن بدون تكرار
- لانشاء الset نقوم بستعمال new keyword ثم الSet constructor ويوضع array بين ال( ) كتالى

```
let theSet=new Set([1,1,2,2,3]);
```

  - سيقوم الset بتخزين ثلاث قيم فقط وهم 1 و2 و3 لانه لايقوم بتخزين التكرار
  - الsize فى الset كا length تماما ففى هذا المثال قيمته ب3
- لايمكن الحصول على العنصر فى الset بستخدام الindex
- يمكن اضافة عنصر فى الset عن طريق استخدام ()add function وتستخدم مع الset وتوضع العناصر فى ( ) كتالى theSet.add(4)
- عند اضافة عنصر موجود يتم تجاهله
- يمكن حذف عنصر من set عن طريق استخدام ()delete function ويوضع العنصر المراد حذفه بين ( )
- يمكن حذف جميع العناصر فى set عن طريق clear() function
- ()has هى الfunction التى تستخدم للبحث عن عنصر فى الset وتقوم بارجاع قيمة true او false ويوضع العنصر فى ( )
- يمكن استخدام forEach مع الset
- يوجد نوع ايضا يسمى الweakSet وهى نوع من الdata مثل الset ولاكن يختلف عنه فى بعض الاشياء مثل انه لايقوم يتخذين غير الobject
- يتم انشاء الweakSet ولاكن بدلا من وضع كلمة set توضع WeakSet والعناصر داخل الarray تكون object
- الweakSet لاتحتوى على خاصية الsize
- لايمكن استخدام الforEach مع الweakSet

Map data type

- الmap هى data type تشبه الobject ولاكن توجد اختلافات بينهم كما الset تشبه الarray ولاكن بختلافات بينهم
- يتم انشاء الmap من خلال new مع الMap constructor كتالى

```
let theMap=new Map();
```

- الmap تحتوى على key و value مثل الobject
- يتم اضافة الkeys والvalue بين ( ) فى شكل array كل key وvalue يكونان على شكل array ايضا كتالى

```
let theMap=new Map([ ["key1","value1"] , ["key2","value2"] ]);
```

  - يلزم عن استعمال هذه الطريقة ان لا يوضع الkey كا symbol
- يمكن استعمال set() function مع الmap لكي تقوم باضافة key و value للmap ويكون الkey هو اول قيمة لset function والvalue هى الثانية
- الobject يحتوى على default keys عند اشائه من الprototype (الوراثة) بينما الmap تكون بدون default keys
- يمكن انشاء object بدون default keys من الprototype عن طريق وضع null فى create function كتالى let ob=Object.create(null)
- فى الobject يمكن ان يكتب الkey من string او من symbol ولاكن فى الmap يمكن ان يكتب من اى شئ
- فى الobject عند كتابة key كا symbol او كا string لايقوم بتفرقة بينهم اذا كانو نفس الkey ولاكن فى الmap يتم التفرقة
- فى الMap يتم ترتيب العناصر على حسب الادخال الذى تم بينما فى الobject لا يحدث ذالك
- فى الmap توجد الsize التى تقوم بارجاع عدد عناصر الmap بينما فى الobject لايتم ذالك
- الmap يمكن استخدام معها الloop بشكل مباشر بينما الobject لا
- الmap افضل من الobject من ناحية السرعة
- للاتيان بقيمة العناصر فى الmap نستخدم الget() function مع الmap ونضع الkey فى ( ) كتالى

```
console.log(theMap.get("key1"));
```

  - يلزم وضع الkey كما هو موجود فى الmap
- لحذف عنصر فى الmap نقوم بستخدام الdelete() function مع الmap ونضع الkey كما هو فى map داخل ال( )
- عند وضع كود الdelete فى الطباعة داخل الconsole يتم ارجاع قيمة Boolean تدل على نجاح او فشل الحذف
- clear() هى function تستخدم لازلة جميع العناصر فى الmap ولاتوضع لها قيمة
- has() هى function تستخدم مع الmap لكى تتاكد من وجود عنصر وتقوم بارجاع قيمة true او false ويوضع الkey فى ( )
- يوجد data type اخرى وهى weakMap وهى مثل الmap ولاكن توجد اختلافات بينهم مثل ان weakMap تقوم بتخزين الobject فقط
- يتم انشاء الweakMap عن طريق استخدام new مع الWeakMap constrictor

Array-extra information

- from() هو function من الArray constructor يقوم بانشاء array من القيم المعطاة اليه
- تستخدم فى عمل الفلاتر
- الfrom() تاخذ ثلاث قيم
  1. Iterable وهو الذى يحتوى على القيم التى سيتم وضعها او التى سيتم تعديلها ثم وضعها
  2. Map function وهى تستخدم لعمل تعديل على الiterable قبل ارجاعه
  3. This
- الiterable يمكن ان يكون string او array او set
- فى حالة كتابة الiterable سيقوم الfrom بارجاع الarray من الiterable بدون تغير عليها
- copyWithin() هى function فى الArray constructor تقوم بنسخ جزء من الarray وتقوم بوضعه مكان جزء اخر وبذالك فان حجم الarray لا يتغير
- الcopyWithin تاخذ ثلاث قيم
  1. Target وهو الindex الذى سيتم وضع النسخة فيه
  2. Start وهو مكان بداية اخذ النسخة
  3. End هو مكان نهاية النسخة وهو لا يشمل اخر عنصر
- عند كتابة الtarget فقط سيتم اعتبار البداية هى اول عنصر و النهاية تكون الى اخر عنصر بما فيهم العنصر الخير كتالى

```
let arr1=[1,2,3,4,5,6];
let arr2=arr1.copyWithin(2);
```

  - هنا سيقوم بعمل نسخ للarray من اول عنصر الى اخره ويضعه مكان العنصر الخاص بـ2 index وسيصبح شكل الarray كتالى [1,2,1,2,3,4,5,6] ولاكن بسبب ان حجم الarray لايمكن ان يذيد سيتم اذالة اخر عنصرين وتكون النتيجة [1,2,1,2,3,4]
- عندما يكون الstart هو الend لن يقوم باخذ العنصر
- some() هو function فى الArray constructor يقوم يذهاب الى عناصر الarray بشكل تدريجى ويقوم بتحقق من وجود شرط ويقوم بارجاع true او false
- الsome فى حالة تحقق الشرط على واحد من عناصر الarray تقوم بارجاع true وتنتهى

- الsome تاخذ قيمتان اولهم هى الfunction التى ستختبر الشرط وثانى قيمة هى الthis parameter الذى يعود على الfunction
- الfunction الموجودة داخل الsome تاخذ ثلاث قيم وهى element و index و array والelement هو الاجبارى الوحيد
- تكتب الsome كتالى

```
let arr1=[1,2,3,4,5,6];
let res=arr1.some((ele) => ele>3);
```

  - سيقوم بذهاب الى عنصر عنصر داخل الarr1 وسيقوم باختبار الشرط الى ان يصل الى 4 وسيتحقق الشرط ويقوم بوضع true فى res وسينتهى الfunction بدون الذهاب الى 5 او 6
  - يمكن استخدام الthis عن طريق وضع قيمتها ب 3 واستخدام this فى الfunction بعتبار انها تشير الى 3
- every() هى function مثل الsome تماما من ناحية الكتابة ولاكن الاختلاف فى ان الevery تقوم بذهاب الى جميع العناصر فى الarray وتتاكد من شرط معين وفى حالة تحققه فى جميع العناصر سيتم ارجاع true
- Spread operator هو operator يتم من خلال وضع ثلاث علامات من . قبل الstring او الarray و الobject ويقوم بفصل عناصر الarray او الstring او الobject
- الspread operator يمكن استخدامه فى عمل concatenate بين الstring

Regular Expression

- الregular expression هى طريقة تقوم بتعريف اللغة على معيير كتابة شئ معين
- الpattern هو الشكل المراد البحث عنه فى الstring او يمكن القول بانه المعيار
- يكتب الpattern من خلال وضع / / وبينهم الpattern ويضاف بعدهم الmodifiers وهى عبارة عن رموز تحدد بعض الاعدادات pattern كتالى :
  - i تقوم بتحديد ان الpattern بكون insensitive بمعنى انه لايفرق بين الحروف الكبيرة او الصغيرة
  - g تقوم بتطبيق الpattern على كل اشكاله وليس اول شكل فقط
  - m تجعل الpattern يتم تطبيقها على جميع الاسطر
- يكتب كتالى

```
let string="google.com";
let pat=/com/i;
console.log(string.match(pat));
```

- يمكن استخدام طريقة اخرة لكتابة الpattern من خلال الnew مع RegExp() وهو constructor ياخذ قيمتان وهما الpattern و الmodifiers
- يمكن تخزين الregular expression فى variable
- match() هى function يمكن استخدامها على string ووضع الpattern داخل ال( ) وتستخدم لمعرفة وجود الpattern فى الstring
- يمكن استخدام اكثر من pattern فى جملة واحدة كتالى

```
let pat=/(com|org|net)/i;
```

  - فى هذه الحالة سيقوم بارجاع مكان com او org او net التى تظهر لديه فى البداية
  - لن يقوم بارجاعها جميعا بسبب عدم استخدام g فى الmodifiers
- يمكن جعل الpattern عبارة عن range وليست جملة يتم البحث عنها ويوضع الrange كتالى

```
let string="235879";
let pat=/[2-5]/ig;
console.log(string.match(pat));
```

  - تم وضع الrange بين [ ] وتم تحديد البداية والنهاية وبينهم –
  - بسبب وضع الg سيتم ارجاع جميع العناصر التى تحقق الpattern
  - سيقوم هنا بارجاع فى الconsole جميع العناصر التى حققت الpattern فى شكل array
  - يمكن عكس الrange بستخدام علامة ^ مكان البداية قبل كتابتها كتالى [^2-5]
- لايشترط ان يكون الrange عبارة عن ارقام وانما يمكن ان يكون حروف ايضا
- يمكن ان يكون الpattern عبارة عن كلمة وrange معا
- يختلف الrange فى الحروف الكبيرة عن الصغيرة اذا لم نضيف الi modifier
- يمكن اضافة اكثر من range كتالى [0-9a-zA-Z] ولعمل ولعكسها يتم اضافة ^ واحدة فى البداية
- يوجد ما يسمى ب character classes وهى مجموعة من الcharacter توضع فى الpattern لارجاع اشياء معينة كتالى :
  1. علامة . تقوم بارجاع جميع الcharacter وتكتب كتالى /./g
  2. /w تقوم بارجاع جميع الcharacter من a الى z ومن A الى Z ومن 0 الى 9 و _ وتكتب كتالى //w/g
  3. /W وهى عكس /w فى ما ترجعه
  4. /d سيقوم بارجاع اى character من 0 الى 9
  5. /D هو عكس /d فى ما يرجعه
  6. /s سيقوم بارجاع المسفات فقط
  7. /S سيقوم بارجاع اى شئ غير المسافات
  8. \b وتوضع بعدها او قبلها كلمة لارجاع جميع الكلمات التى تبدا او تنتهى بهذه الكلمة

9. \B وهى ترجع اى شي عكس الb/
- يمكن دمج الcharacter class ب string او بrange
- Test() هى function تستخدم لارجاع قيمة true او false وتستخدم لتحقق من وجود الpattern فى الstring وتكتب مثل الmatch
- يوجد بعض الcharacter وتسمى quantifier characters وتستخد لتحديد كمية الcharacter المضاف قبله مثل
  1. + توضع بعد الcharacter او بعد character classes لتدل على انه واحد او اكثر
  2. * توضع بعد الcharacter او بعد character classes لتدل على انه صفر او اكثر
  3. ؟ توضع بعد الcharacter او بعد character classes لتدل على انه صفر او واحد
- يوجد quantifiers اخرى تستخدم لتحديد اعداد الcharacter بشكل اكثر مثل
  1. {x} تقوم بتحديد ان الcharacter او character class الذى قبلها يوجد منه عدد x
  2. {x,y} تقوم بتحديد ان الcharacter او character class الذى قبلها يوجد منه عدد ما بين x الى y
  3. {x,} تقوم بتحديد ان الcharacter او character class الذى قبلها يوجد منه عدد x او اكثر
- علامة الـ$ يمكن استخدمها كا quantifier وياتى قبلها character من اى نوع او string لتدل على انهاء الpattern بهذا الcharacter
- ^ تستخدم وبعدها character من اى نوع لتدل على ان الpattern بيدا ب هذا الcharacter وهى تختلف ندما نستخدمها مع الrange
- توجد اهمية كبيرة للregular expression فى الfiltering او فى اشياء اخرى مثل استخدمها مع replace() function

OOP

- الconstructor هو function ولاكنه يختلف عن الfunction العادى فى مجموعة من الاشياء مثل
  1. عند عمل اسم الfunction الافضل ان يبدا ب حرف capital
  2. عند انشاء اى data او function فيه توضع this ثم علامة . ثم المتغير او او الfunction
- نستخدم الconstructor مع الnew operator لانشاء مجموعة من الobject من خلاله تحميل نفس الdata والfunction
- عند التعديل على الconstructor يتم التعديل على جميع الobject التى تم انشائها منها
- توجد طريقة احدث لكتابة الclass غير طريقة الconstructor وتكتب كتالى

```
class class_name{
    constructor(/*parameters*/){
        /*the data*/
    }
}
```

  - لاستخدام الclass لانشاء object فهذه العملية تتم ايضا بستخدام الnew كما فى الconstructor
- لتاكد من ان object معين ماخوذ من class معين نستخدم key word وهى instanceof وتقوم بارجاع قيمة Boolean وتكتب كتالى

```
ob1 instanceof class1;
```

- عند اضافة الfunction الى متغير فانها تظل بذالك property
- عند انشاء method داخل الclass نقوم بكتابتها مثل اى function ولاكن بدون كتابة كلمة function
- عن انشاء الdata فى الclass تتم بدون كتابة كلمة var او let
- الproperty و data داخل الconstructor هم فقط ما نقوم بوضع معهم this
- Static property and method هم عبارة عن data او method يتم استخدمهم فقط من قبل الclass وليس من الObjects وتتم عن طريق الكتابة خارج الconstructor ووضع كلمة static قبل انشاء الdata او الMethod
- عند استخدام واحد من الstatic data or method داخل الconstructor يتم وضعها من خلال عمل access عليها من الclass
- كل code فى الconstructor يتم تنفذه بمجرد استخدام انشاء object من خلاله
- Inheritance هى عملية الوراثة وتعنى ان class يمكنه اخذ عناصر class اخر والتعديل او الاضافة عليها
- يتم الinheritance يتم من خلال كتابة كلمة extends ثم اسم الclass الموروث منه كتالى

```
class ClidClass extends ParentClass{}
```

  - الclass الموروث منه غالبا ما يشار اليه ب الparent
  - الclass الاخذ للوراثة غالبا ما يشار اليه ب الchild
- تتم الوراثة لجميع الmethod او data الموجودة خارج الconstructor بينما ما داخل الconstructor لا يحدث له وراثة
- لعمل وراثة لما داخل الconstructor نقوم باستخدام super() function وتوضع بداخل ال( ) جميع الMethod او الdata المراد ورثتها
- عناصر الclass بشكل default تكون public اى انه يمكن عمل عليها access بشكل طبيعى من خارج الclass
- العنصر الprivate هو عنصر لايمكن لاى كود عمل access عليه الا من خلال الcode الموجود داخل الclass
- لعمل عنصر private نقوم بوضع علامة # قبل اسم العنصر ونقوم بعمل له declarer ثم يمكننا استخدامه كما نشاء فى الclass ولامن ايضا ب#
- الprototype هى الالية التى من خلالها تتم الوراثة
- يمكن اضافة عنصر فى الprototype عن طريق تحديد الclass ثم عمل الaccess على الprototype ثم عمل access على العنصر المراد اضافته ثم وضع قيمة له

Modules import and export

- الmodule هوواحدة برمجية تتكون من مجموعة من الاكواد تستخدم فى المساعدة فى المشاريع
- لعمل export للfunction او لdata لكي يتمكن اى ملف اخر من استخدم هذه الdata او الfunction نقوم بوضع كلمة export قبل كل الdata او الfunction حتى قبل كلمة Let و function
- لستدعاء الdata او الfunction ناتى فى الfile الذى نكتب فيه حاليا ثم نضيف كملة الimport كتالى

```
import {/*the data and function*/} from "the link or the place of the export file";
```

  - داخل ال{ } يتم وضع الdata والfunction التى سيتم اخذها
  - داخل ال" " يتم وضع مسار الfile الذى سيتم اخذ منه الdata و الfunction وذا كان فى نفس المجلد نقوم بوضع /. ثم اسم الfile
  - يفصل بين كل عنصر والثانى داخل ال{ } ب ,
  - يمكن اعادة تسمية العناصر التى تم استيرادها عن طريق الkey word وهى as حيث انها توضع بعد العنصر المراد اعادة تسيمته ثم يوضع الاسم الجديد بعدها
- يلزم عند استخدام الmodule وضعه فى ال html فى الscript tag وةضع مساره بالاضافة الى وضع Type attribute له بmodule
- يمكن بدل وضع كلمة export قبل كل data او function يمكننا استخدام كلمة export{ } ونضع بداخلها العناصر ونستطيع استخدام فيها الas
- يمكننا بدل كتابة جميع الdata او الfunction فى الimport استخدام علامة * لانها تدل على الكل كتالى

```
Import * from "the link or the place of the export file";
```

JSON

- Stands for javascript object notation

- JSON هى طريقة لمشاركة البينات بين الclient side و الserver side بستخدام الobject
- الJSON تم انشائها من الjavascript وهى بديل لXML
- امتداد ملفات الJSON تكون .json
- يمكن تحويل الJSON object الى javascript object
- الJSON object يحتوى فقط على مجموعة من بينات وهم الstring و number و الobject و الarray و ال Boolean و null فقط ولايحتوى على اى function او اى عمليات حسابية
- لانشاء JSON object نقوم فقط بوضع البينات داخل { }
- ويتكون الJSON object من key و value فقط والkey يجب ان يكون بين " "
- لايمكن كتابة comment فى الJSON object
- الAPI اختصار ل application programming interface وهى واجه وسيطة بين الfront end و الback end من خلالها يتم مشاركة البينات وتكون البينات فى شكل JSON object
- بسبب اختلاف اهمية البينات التى يمكن ان يتم مشاركتها تم تقسيم الAPI الى نوعان public API و private API
- الJSON عندما ياتى من الbackend يكون عبارة عن string لزالك عند تحويله الى javascript object يتم ذالك من خلال ()parse وهو function فى الJSON constructor ياخذ الstring كا قيمة لتحويلها الى object
- ليكى يتم ارسال البينات الbackend يلزم ان تكون JSON ايضا اى انها يججب ان تكون string لذالك يتم استخدام الstringify()وهو function موجود فى الJSON constructor ياخذ قيمة واحدة وهى الobject لكى يقوم بتحويلها الى string

Request and response

- لانشاء request نقوم باستخدام new keyword مع الXMLHttpRequest() constructor ويمكن تخزينه فى valiable
- Open() هى function فى الXMLHttpRequest() constructor تستخدم لعمل الrequest حيث تاخذ اكثر من قيمة واول قيمة وهى نوع الrequest مثل الGET و الPOST وتوضع بين " " والقيمة الثانية وهى الAPI المراد عمل عليه request
- لن يتم ارسال الrequest الا بعد استخدام الsend() function

```
let x=new XMLHttpRequest();
x.open("GET","https://www.web-site.com");
x.send();
```

- يوجد بعض الproperty فى الrequest والتى لها بعض الدلائل مثل
  1. Status وهى key فى الrequest object تقوم بارجاع رقم وهذا الرقم يدل على حالة الrequest مثل 0 وهو عدم الوصول الى الserver او 404 الذى يدل على عدم ايجاد الصفحة برغم من وجود الserver او 403 الذى يدل على وجود الصفحة ولاكن عدم وجود اذن بدخل اليها او 200 الذى يدل على اكتما الrequest بنجاح
  2. Readystate وهى key فى الrequest object تقوم بارجاع رقم هذا الرقم يدل على مستوى تقدم الrequest حيث ان 0 يدل على ان الrequest لم يتم انشائه و 1 يشير الى وجود اتصال مع الserver و 2 يشير الى ان الserver قام باستلام الrequest و 3 يشير الى ان الrequest يحدث له معالجة و4 يشير الى ان الrequest انتهت والرد جاهز

- لايفضل التاكد من صحة الrequest من خلال الreday state لان رقم 4 يدل دائما على ان الrequest انتها حتى وان كان بشكل غير سليم لذالك يفضل الاعتماد على الstatus

3. Onreadystatechange هى event موجودة فى الrequest object تقوم بتنفيز امر فى حالة تغير الready state ويفضل استخدمها فى حالة لتحقق من صحة الrequest مثل الstatus او غيرها قبل بدا استعمال البينات القادمة بعد ارسال الrequest
4. ResponseText هو preporty فى الrequest تقوم بارجاع الresponse فى شكل text والذى يكون عبارة عن JSON

- يمكن استعمال الresponse ةعمل عليه loop ولاكن اولا يجب ان يكون فى شكل objects وليس string لذالك يمكننا استخدام parse()

Promise

- الpromise هو عبارة عن object يستخدم لتنفيز لتنفييز functions اعتمادا على تنفيز function
- لانشاء promise نستخدم Promise constructor مع الnew key word وهو ياخذ function كا parameter وهذه الفنكشن تاخذ two function كا parameter وفيه واحد من هذان الfunction سيتحقق ويكون الsyntax كتالى

```
const p=new Promise((fun1,fun2)=> {
    if(/*condition*/){
        fun1("frist value")
    }else{
        fun2("secound value")
    }
})
```

- الfun1 و fun2 يجب ان يحدث واحد منهم ولايحدث الاخر
- كل من fun1 و fun2 عبارة عن function ولاكن لايتم استخدامهم غير تخذين قيمة معين
- Then() هى method فى الpromise تاخذ قيمتان وكلاهما function وكل واحدة فهم ستتم اذا تحققت التى تناظرها فى الpromise وكلا منهما تاخذ قيمة تكون هي الموضوعة بداخل المناظرة لها فى الpromise
- يمكن عمل اكثر من then معا
- يمكن ان تحتوى الthen على function واخدة فقط ولاكنها ستتم فى حالة تحقق اول function فقط لذالك يمكن استخدام الcatch وهى مثل الthen ولاكنها تاخذ function واحدة وتتم فقط فى حالى تحقق ثانى function
- غكرة الpromise كل عمل اكثر من function بشكل متتابع بطريقة افضل فى التنفيز والقرائة
- Fetch() هى function نقوم بارجاع الresponse ثم يمكن استعمال عليها الthen و الcatch وتاخذ الURL كا arjument
- لتحويل الresponce الى json فاننا نستخدم function json().