ES . NEXT

MORDERN J A V A SCRIPT

TODAY'S AGENDA

- DATA STRUCTURE , COLLECTIONS AND NEW CONTROL STATEMENT
    - SET OBJECT
    - MAP OBJECT
    - FOR ... OF
- ITERABLE
- GENERATORS
- NEW PRIMITIVE DATATYPE (SYMBOL)
- CLASSESS
- MODULARITY
- LAB

Set Object

In mathematical sense , a set of group values that uniqu

It ' s an iterable object

We can pass an array when I ' m creating a set and this will remove the
duplicate items

Method:

. has () / . add () / . delete () / . clear () / . entries () / . keys () / . values ()

Map Object

is an object of key / value pairs both key and value can be either privative
or object

Method:

- . set (key , val) / . get (key) / . delete (key) / . clear () / . has (key) /
  . keys () / . values () / . entries ()

For .... Of

For .... Of statement iterates over property values

It ' s better way to loop over iterable objects

```
var myStr = ""
var myArr = [,,]
var mySet = new Set([,,,,])
var myMap = new Map([[,],[,],[,]])
```

Iterable Objects

Must have @@ iterator method

The implementation [symbol . iterator]()

Can use: for ... of , destructuring , ... spread operator

```
var  arr  =  [1,2,3,4,5,6]
```

ITER A BLE OBJECT → . NEXT ()

{ V A LUE , DONE }

```
var  iter  =  arr[Symbol.iterator]()
```

ITER A TOR OBJECT

Generator Function

function* genfn () { yield 1 , yield 2 , yield 3 }

```
function*  gen()  {

yield  10;
yield  15;
yield  *[11,23,12]
yield  *'ABC'

}
```

```
for(  i  of  gen()  )  {
        console.log(i)
}
```

Symbol

New primitive data type in JavaScript (NEW in ES6)

Unique

Considered as UUID or GUID

Universally Unique Identifier or Globally Unique Identifi

Can be used as object key

SYMBOL (' DESCRIPTI

SYMBOL . FOR (' DESCRIPTION ') =

Symbol

Static Properties

- Symbol . match ()
- Symbol . replace ()

[SYMBOL . REPL A CE](STR , IDX){}

Symbol

Symbol as object property:

Non - enumerable

OBJECT . GETOWNPROPERTYSYM

- A nonymous
- Can ' t convert to JSON object when we use JSON . stringfiy

EXP A MPLE: [SYMBOL . FOR](10) : 123

Class


```
class  className  {

      constructor(p1,p2)  {
        this._p1  =  p1;
        this._p2  =  p2;
  }

    get  p1()  {  return  this.p1;  }

    set  p1(val)  {  this.p1  =  val;  }

    static  staticFn()  {  return  ;  }

    static  get  staticProp()  {  return  ;  }

}
```

Modules

N A MED EXPORT

```
<script  type="module">

  import  {  ......  }  from  "moduleName"

  import  *  as  someName  from  "moduleName"

</script>
```

Modules

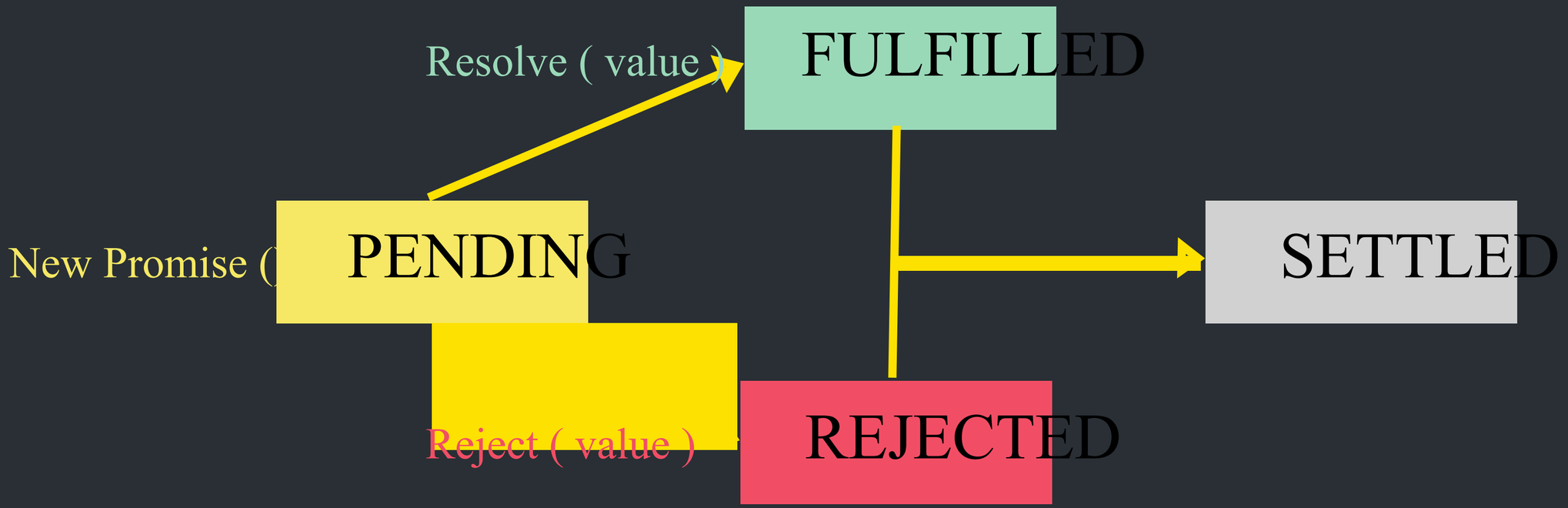
DEF A ULT EXPORT

```
export default class className { }

import className from "moduleName"
```

Promise

Promise is an object representing the eventual completion or failure of an asynchronous operation



Promise

Consuming promise:

THEN ()

Handle the success
of the promise

C A TCH ()

Handle the failure
of the proimse

FIN A LLY ()

Handle after all
chain methods

Promise

Promise Static Methods

PROMISE .ALL ()

Returns either resolved promise if all passed promises are resolved

Or rejected promise if as soon as of these promises is rejected

PROMISE . RACE ()

Returns rejected or resolved promise a soon as one of the passed promises is settled

Promise

Promise Static Methods

PROMISE . REJECT (RE A SON)

Returns rejected promise object with the given reason

PROMISE . RESOLVE (RE SON)

Returns resolved promise object with the given reason

A sync & A wait

A SYNC FUNCTION WRITES PROMISE - B A SED CODE A S BEH A VES IF IT WER SYNCHOROUS CODE

```
async  function  funName  {
          await  new  Promise() .then().catch()
          await      new    Promise() .then().catch()
          await      new    Promise() .then().catch()
}
```

A synchronous Synchronous

WHEN USING A W A IT , THE FUCNTION IS P A USED IN A NON - BLOCKING W A Y UNTILL THE PROMISE SETTLES

Transpilers

TR A NSL A TE / COMPLIRERS

JS

JS ES6

V A NILL A JS

EX: B A BEL JS

~~https://es6console~~.com/

New Features (ES7 , ES8 , ES9)

ECM A SCRIPT 2016 (ES7)

A RR A Y . INCLUDES ()

EXPONENTI A L OPER A TOR **

New Features (ES7 , ES8 , ES9)

ECM A SCRIPT 2017 (ES8)

- STRING . P A DST A RT ()
- STRING . P A DEND ()
- OBJECT . V A LUES (OBJ)
- OBJECT . ENTRIES (OBJ)
- OBJECT . GETOWNPROPERYDESCRIPTOR (CONSTR)

New Features (ES7 , ES8 , ES9)

ECM A SCRIPT 2018 (ES9)

SYMBOL . DESCRPITION

TRY {} C A TCH {}

PROMISE . FIN A LLY ()

A RR A Y . FL A T () / A RR A Y . FL A T

New Features (ES7 , ES8 , ES9)

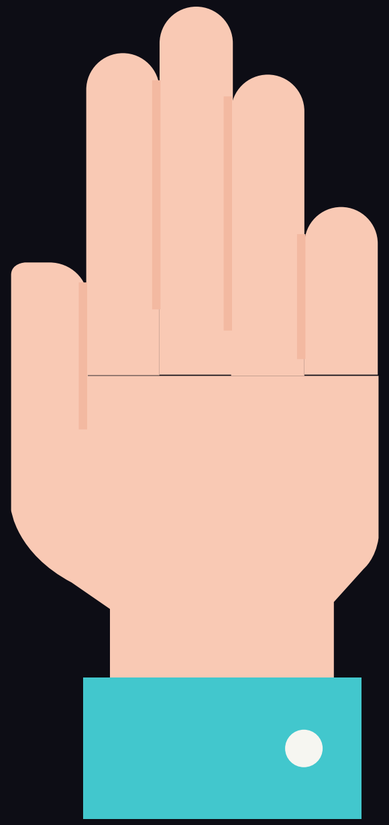
ECM A SCRIPT 2020

BIGINT (NUMBER N)

CH A INING ?.

NULLISH ??

GLOB A LTHIS



TH A NK YOU

A NY QUESTIONS ?

LAB

Create any array of food called 'food':

['Burger', 'Pizza', 'Donuts', 'Pizza', 'Koshary', 'Donuts', 'Seafood','Burger']

01

Create a Set with values of this array

02

Add 'pasta' to the set and log the set to the console

03

Remove 'burger' from the set and log the set to the console

04

Write a function that takes the set as a parameter and clear the set if it has more than 2 items

LAB

Create a class called 'Vehicle'

01

The class has a constructor function that takes 2 parameters (wheels, speed)

02

Create a sub-class 'Bike' that inherits from vehicle and has different default values (wheels: 2 , spead: 'fast enough')

03

Add a static method to bike sub-class to count how many time it's called

Contact me 😎

CODEZETTA
BY NOURHAN SAEED

FeedBack: https://forms.gle/pbiG8YPCQRPgsrdZA

Phone: 01271888031