

Java Script

High-level: ~~high level langs obj-as. C -> low level of code~~ ~~high level langs obj-as. C -> low level of code~~

Garbage-collected:

Interpreted: \Rightarrow just-in-time compiled

Multi-paradigm: ① Procedural



{ part 1 of poly part }

② Object-Oriented (OO)

part 2

③ Functional Programming (FP)

part 3

Prototype array & push \rightarrow higher. \rightarrow Array.prototype.push. Heritage

First-class functions: \rightarrow functions are variables, can be passed as arguments

Dynamic

\rightarrow runtime-type. var-type \rightarrow dynamic

single-threaded

\rightarrow one thread at a time, no parallel threads

Non-blocking event loop \leftarrow non-blocking I/O, callbacks, promises, promises

JS ENGINE



node.js: v8 \rightarrow JS engine that handles JS script processing

compilation

can happen way after comp.

Source code $\xrightarrow{\text{Step 1 compilation}}$ Machine code $\xrightarrow{\text{Step 2 exe}}$ Program running

interpreted:

Source code $\xrightarrow{\text{Slow - Step 1}}$ Program running
 $\xrightarrow{\text{exe line by line}}$ $\xleftarrow{\text{conversion to machine code}}$

Just in time

Source code $\xrightarrow{\text{Step 1 compilation}}$ Machine code $\xrightarrow{\text{Step 2 exe}}$ Program running
 $\xrightarrow{\text{a file isn't created}}$ $\xleftarrow{\text{immediately}}$

enter the engine

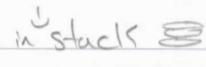
code \rightarrow Parsing

optimization

AST \downarrow

AST \rightarrow Machine code

in stack



Compilation \longrightarrow Exe

JS : Data Structure

Destructuring: Break a complex data structure to a less complex, like variable
let/const [var names] = array ← array/object → let/const {var names} = object

Destructuring Arrays

- I) 1. create an array of 5 items ✓ 2. destructure them according to the formula above ✓ 3. print in the console ✓
- II) 1. create an array of a menu in a resto ✓ 2. extract the first and the main course and log them ✓ 3. extract the first and the third (you need to leave a hole for that) ✓ 4. Switch the place of the third and the first element in the array → it doesn't change the array, only the var assignments ✓
- III) create a function that get as args 2 index numbers and return an array, → destructure the function in 1 line of code ✓
- IV) create a 2D array and destruct it. not to z, [s, g] but to z, s, g ✓
- V) create an array of 3 elements, ext the 4th element. what do u get? ✓
VI) to avoid this, give the null a default value.

Destructuring Objects

- I) create an object and destructure it ✓
- II) Destruct but this time customize the var names for ex extract name as firstName ✓
- III) create an object with a property that has multiple values
ext and give the properties default values in case they don't exist (for this I need to try and ext a property that doesn't exist) ✓
- IV) Mutating variables. Create 2 variables a=1, b=2 create an object with variable a, b → destructure the object using a and b. did u manage? solve it by wrapping
- V) Nested objects: create an object inside an object and destr it. ✓

seperately

Spread operator: unpacking all the array elements at once and writing them
create an array, add it 2 elements at the beginning (create another array)

- I) do the same with the spread operator

- II) log the array with one line, without using a loop
★ useful only where I need element, element, element...

- III) Copy: copy an array with 1 line of code ✓

- IV) Merge: merge 2 arrays together ✓

IV: write a function that get 3 arguments, write an array with 3 elements
pass the array into the function! do the same with the ...
the elements comes from user input ✓

V: create an object, create another object and copy the first object to
the second object while adding some properties ✓

Rest Operator: do the opposite of spread operator. syntax ...

I) Create an array of 5 elements, destructure it with variables and the rest
operator ✓

II) Create an object with 3 subobjects, put the middle one in a variable and all
the rest ... in a rest object ✓

III) Create a function that take 2 or more arguments and sum them

Short Circuiting (& ||)

I) log to the console 3 || 6, 6 || 3, 'red' || 3 why?
true:
false: "" - empty string, undefined, null, 0

II) Avoid assigning a variable with an undefined value using || and not if null...
||: will return the first trutty value or the last value if all of them are falsy
&: will return the first falsy value or the last value if all of them are trutty

Looping An Array: for (const item of array) do something

I) change the for loop you created to for...of loop and
II) Create another for loop with the index of the item (a of array.entries())
destructure the item to id and content

Looping an Object

I) Create an object loop over its keys (of object.keys(objectName)) and print
them. Create a nested object and loop over it
II) -|| Loop over the values
III) Loop over an object

Sets : collection of unique values

- I) Create a new set with an array that includes duplicates. the array should have 6 items and 3 duplicates ✓
- II) Pass the set a string as a parameter. find out how many letters there R? ✓
- III) Pass an object. is it possible? No ✓
- IV) For one of the created sets: 1) get its size 2) look if an item exist - test with item that exist and that doesn't exist ✓
 - 3) add item to a set - twice ✓ 4) delete one element ✓
 - 5) try to retrieve the first element of a set, is it possible? No ✓
 - 6) loop over a set and print its values ✓
- V) convert a set to array ✓

Maps: map values to keys. the key can be of any type

- I) Create an empty Map and add items, add a string and a number keys set an array value ✓
- II) the set method returns the map, use it for your advantage to set 2 entries, 1, 2, 3, 4 ✓
- III) read a value/s by providing a key ✓
- IV) find out if a map has certain key? value, if yes, delete it
find out the size of a map ✓
- V) Convert an object into map, convert an array into map, map to an array ✓
- VI) iterate the map and print the key, and the value ✓
- VII) Convert map to an array ✓
- VIII) Get all the keys and all the values ✓

Strings:

- I) get the letter of the position x , get the length of a string .
 - get the index of a letter , the last index of a letter , the index of a word in a phrase
 - get the index of letter that doesn't exist
- II) slice a string, extract part of it . use the slice method with 1 args and with 2 args. Create a sentence with n words, extract the first word ✓
 - without using hard code index number , extract the last word . use slice() if this is not empty
- III) use slice with negatives parameters . Convert dAVID to David
- IV) clean an email address that comes from input , delete spaces and [enter] \n
- V) Replace: convert 234,99€ to 234.99 \$ 2) replace all the occurrence ✓
- VI) of words in a sentence . try replaceAll(), try Regex() → instead of 'string' use ...
VII) Verify if a word includes a seq of letters, if it starts if it ends with seq of letters

VI) take an email address and divide it into 2, before and after the @ and put it in 2 var. don't use the slice . 2) create a string "a+ very + nice" and divide it by the + 3) create a String with first name and last name and put it in 2 var ✓

VII) take the 2 variable and create a new string out of them, with "/" between each word . 2) Create a method that takes first middle and family name and transform the first letter to uppercase. try it with f,m,l name and only with f name ✓

VIII padding: create a string with a credit card number. mask it so it will show only the latter 4 numbers ✓
to replicate, repeat or string 3 times ✓

Functions

Default parameter

- I) create a function with a default parameters using short circuiting - the old way
- II) create the same thing - the new (ES6) way. test by giving it zero parameters, 1 parameter, skipping the middle par, only the last par

Value vs Ref

- I) create a function that get as arg: number, string, object^{array}. inside the function change every one of the values of those args. print them (the originals). what happened?
 - the value of the string is passed as an arg, so the original doesn't change
 - the ref of the obj is passed as an arg, so the original change

First-Class vs High-Order

first class: function = value (let $x = y \Rightarrow \text{return } y$), function = object, object has methods

High-order: a function that receive another function as arg → add(x) → a function that return another function

High-order function
count() { return () => 3 }
callback function
returned functions

CallBack Function

- I) create a function that take a string → to lower case . another → to upper
- create a function that take a fn and a string as argument and return the output String

- II) use for Each on an array

Function that return a function

- I) create a function that returns a function (the inner function do something and has a parameter, as the case for the first function. call the second function)
- II) rewrite the functions you wrote using arrow functions, do the same with jonas Functions

Call , Apply , Bind: manually set the this. Keyword in a regular function call, the this is undefined. it is defined when the the function is called on the object. this inside a method (function of an object) will work

this inside a function won't work → undefined

Making it work: 3 options 1. call 2. apply 3. bind

call: first arg - the obj that the this is ref to, then the rest of the arguments

apply: do the same. doesn't get the args as plain text, but as an array // less used

bind: return a method where the this is defined. it binds a certain this to a method, for good

I) write 2 objects that has the same properties, write an external method that uses this property. call this method. what happened. fix it with call(), fix it with apply by passing a predefined array, with a small trick, pass the array to the call method and make it work ✓

✓ create 2 methods, one for each object - using bind. ✓ II) create a bind method with a predefined, fix parameter ✓

✓ create HTML button with class=button, add a js event listener and assign it to the function you wrote ✓

✓ create a function to calculate something. test it, use the bind to fix one of the parameters → creating more specific function from more general fn

✓ rewrite the previous method - with 1 fn returning another function ✓

IIFE: Immediately Invoked Function Expression ↗ ני'י'ק א'ז'יו

function that exec only once. create one like that

Closures

create a function with variable and after create an inner function

create a variable that stores the returned function, and call it

console.dir(varname). look at the scopes

OOP: Object Oriented Programming In JS

Intro

model: describe a real word/abstract features

interaction between objects happen through a public interface (API)

API: methods outside of the object that serves to communicate with the object

class: a blueprint for creating objects based on the rules of the class

instance: an object that was created from the class

object state: = data

4 OOP Principles: 1) Abstraction 2) Encapsulation 3) Inheritance 4) Polymorphism

OOP in JS

Java: class $\xrightarrow{23!}$ instance JavaScript: prototype $\xleftarrow{\text{by}} \text{object}$

implement OOP: 1) constructor fn 2) classes 3) object.create()

const fn: technique to create object from a function

classes: it's not the classic Java class. Behind the scene it just a constructor fn

create(): the easiest way to link an object to a prototype. not widely used

Constructor function

the function is called by the keyword new function(). then, 4 thing happen

- 1) New {} is created // empty object
- 2) assignment of this. to the {} // this $\xrightarrow{\text{point}}$ {}
- 3) {} is link to a prototype
- 4) the function return the {}

I) create a const fn of user with name, password as parameters. at the body of the fn, print(this). call the function. look at this in the console

II) inside the cons assign the name and password to this. print this.

III) create a user. check if it's instance of User

Object.create(): assign a prototype to an object/instance
get as parameter \rightarrow the prototype. Create a new object with that prototype

Prototypes: Inheritance of methods / properties

Each function in JS has a property called **prototype** (also const fn)

Every object that is created by a const fn will get access to all the methods (and properties) that we define on the prototype property

- I) Use the user from before. Using the prototype property add a method that prints the user's name, password and date created .test
- II) display the prototype of an instance and of its object (daron... User)
- III) add a property to the User object not via the constructor, via the prototype check via the method `hasOwnProperty` for the new property, then check for one of the constructor property

Prototype on Built In Objects

- I) create an array and display its prototype
- II) display the prototype of a function (all the functions) without creating it
- III) create a new method that filters all the strings out of an array.
apply the method like this: `myArray.onlyNonC()`. \Rightarrow don't do it in the real life

Classes

- 1) classes are not hoisted. I can't use it before declaring it
- 2) classes are first class citizens: I can pass them into functions, and return them
- 3) classes are exec in strict mode, even if I didn't activated it

Getters & Setters

- I) Create an object with name, sex, birthyear. try to access and print them
create a get method to print "It's Mr/Ms name". Write a method (not a get) that does the same thing. Use both of them. Check the obj prototype
- 1.) create a method that calculate the age 2.b) create a get for the same thing
- II) create a setter for the name, Capitalize the first letter, create a new object and print it. How is the name written
- II) A stack overflow might occur. fix it and print `obj.name`

Static Methods

Are created on the constructor and not on the prototype.

- I) Create a static method on 1 of your objects. 1) inside a class 2) outside a class \Rightarrow in both of the cases `console.log(this)`

→ OOP: Inheritance & Encapsulation

Inheritance: using constructor fn

- I) Create an animal object constructor with 2 params. add to it a method eat(). Create a lion constructor. Implement inheritance inside the constructor (the lion should have 3 parameters). Is lion obj instance of Animal?
- II) Call an animal method on the lion. It works? If not - fix it. Check if your lion instance is instance of Lion? of Animal?
- III) console.dir the Lion.constructor / Lion.prototype.constructor is it Lion or Animal? You want it to point to Lion. Fix it

Inheritance: Classes

- I) Create a lion class that extends Animal, create the constructor
Create a lion and check if it's instance of Animal. Check the lion constructor
- II) Create 2 methods for the lion

Inheritance: Object.create() → Linking Objects together

- ① We don't use new to create an object → const childInstance = Object.create(child)
- ② We don't need a constructor method → Child.method* = function (att1, att2...)
- ③ We don't mess with the prototype → const Child = Object.create(Parent)
the Parent must be an object
*Creating a child instance will call this method on the instance after ② has been done.

Asynchronous JavaScript

Async, Ajax, API

Synchronous: The code is exe line by line, each line waits for the previous line to finish

Async: exe doesn't wait for async task to finish its work. The code is non-blocking

AJAX: Asynchronous JavaScript And Xml. Allow us to communicate with remote web servers in an asynchronous way.

API: piece of software that can be used by another software and allow apps to talk to each other and exchange info/data

Web API: App running on a server that receives request for data, and send data back (response)

XML: once upon a time... today no API use XML → they use JSON

JSON: JavaScript object converted to a string

My API: I have methods inside a public interface, objects made of a class are self-contained piece of software, and other pieces of software (on other machine) can interact with

Search for api: GitHub public API. The CORS should be Yes/Unknown (access) making an api call:

1) Old school: new XMLHttpRequest → events + callback functions

2) Modern way: → using promises

XHttpRequest / Old Way

1) create a new object 2) use open(method, api endpoint)

3) use send (send the request to the endpoint). I can't put the result in a var, it's asynch

4) addEventListener('load', callback function) → // what to do with the data

5) the response is represented by this.responseText

6) convert the json(string) to an object (just raw) with JSON.parse()

7) the json is an array - I can destructure it

=> the next

I) Create an html page that will show the flag, name, continent, population, language and currency of Israel ✓

II) Create a method that accept a country name dynamically and present the country's info. call this method 21 times with different countries ✓

III) refresh the page several times, the next chapter is how to control the order of the response ✓

IV) create a method that represent the country that you give in parameter and its neighbors (borders property) ~~✓~~

call Back Hell: A lot of nested call backs in sequence

I) create a method that will present the country that it received as argument but also its neighbors (property borders: [])

II) call the fn on island, australie (island)

III) create a set timeout nested sequence (6) → you reproduce a callBack hell

Promises and Fetch API: modern way

1) const req = fetch(url) // building a promise

Promise: An object that is used as a place holder for a future result of an async operation ⇒ like a container for a future value.

call back hell: avoided by chaining promises for a sequence of asyn operations

Lifecycle: pending → settled

pending: before the future value has arrived

settled: asyn task has finished → fulfilled: success ⇒ the value is available
→ rejected: failed ⇒ An error happened

2) then(callback fn) // consuming a promise

the callback function get 1 arg: the return value of the fn that called it
the result is the http response, the body of it is a stream, to access it I need
to apply the method .json(), it's available on all the responses that come
from a fetch() building

.json(): return a promise, not a raw data. So I need to .then() it

data: is the resolved value of the json promise

I) do exercise II in the previous page using promises

Chaining Promises

then return a promise so we can use .then() on it and create a promise
chain that will replace the call back sequence
promise.then(return promise).then(return promise)....

II) implement the country and its neighbors

Handling Rejected Promises

2 ways to handle a rejected promise. If I catch the exp the exc don't continue

1) pass a second call back fn to the then(). the arg will be the error

2) adding .catch() at the end with the call back fn .catch(err=...)

* the fetch() fail only when there is no internet connection

I) change the api url to something that won't work - run your code. Solve the problem with handling rejected promises. (print - promise was rejected)
does it work? did the promise fulfilled?

Throwing Errors

I) create a condition if the property ok of the response is false, throw new Error('...').

Async\Await: using function

a better and easier way to build/consume promises. Then, callback fn

- 1) Create a normal function but with the word `async` before. The function will keep running in the background. When it's done it returns a promise
- 2) Before every promise I write `await` (it will wait for the promise to be fulfilled)
- 3) Store it in a var if you want to use it

Errors: we can't use `.catch()` like before. We will use `try...catch`

E) Create an `async` fn of challenge 1. use `catch`, `test`, `t`. call the method 5 times

Returning values from Async

The `async` fn return no matter what I want to return (string, number...)

in the form of a promise. If I will assign the return value to a variable I will assign it with an pending promise \Rightarrow since the value isn't ready yet

Arrays

V

a sample of simple array methods, forEach, Data Transformations, map(), filter(), reduce(), chaining (?)

Simple array methods

- I) create an array and take a slice of it, take the 3 last ele, the 3 first, the 2 in the middle, print it, print the original array ✓
- create a copy of the array with ... and with slice ✓
- II) Delete elements from the array. the same elements from ex I ✓
- II) reverse the order of the array (create one 1-7), print the original ✓
- IV) array1+array2 -> ✓
- II) transform your array into string with 1 method ✓

forEach(): Can't break out of the loop

In each iteration the function forEach will pass the current element to the callback

- I) create a forEach(), manipulate the element of every iteration ✓
 1. in every iteration add 3 to the element and print its index ✓
 3. in every iteration print that you are going to delete element x in index y -> and delete it ✓

Data Transformation

- Map: takes an array, loops over it, in every itr it do some operation → and the output of this operation is applied to a new array. Map the values to a new array. It's like forEach but it creates a new array
- Filter: filter the array, if element pass a certain condition it is passed to a new array
- Reduce: reduce the array to one single value (a single value, not an array)

map()

- I) create an array with numbers. These are money sums. Use map() to map to a new array. operation: convert from euro to sterlin. Do the same with forEach()
- II) do forEach() ex I but with map instead of printing, move it to the new arr

filter()

- I) create an array of numbers and strings, filter all the strings out. 2) create an array of truly/falsey values, filter the falsy values. Do the same with forEach() and delete them from the original arr ✗

reduce()

- I) Create an array with [redraw, Deposits]. use the reduce to calculate the balance
remember → reduce(accumulator, element, index, array). do the same with forEach
- II) Use the same array to calculate the max ✓

Map, foreach(), Set, foreach()

- I) Create a map. loop with for each and display the key and value ✓
- II) Create a Set. - " " - " " - " " - ✓

Doing challenge 2, I ✓

find()

- I) Create an array and find 1 element according to a condition; for example, the first element that starts with an 'g' or less than -50 ✓
- II) Create an array of 5 objects. Search for one of them and print it ✓

findIndex(): Return the index

When I want to delete an element from an array (splice), but for using the splice method I need the index. findIndex() comes to the rescue

- I) Take an array and use the findIndex to find the index of the smallest element

Some(), every()

- I) return true/false if my array contains an element that fulfills a condition.
- II) Create an array and find out if you have an element that starts with "p", an element that's bigger than zero. try to use includes() for this ✓
- every(): it's like some but it returns true only if all the elements fulfill the condition
- III) check if all your elements in your arr are bigger than 0 ✓

flat(), flatMap()

- I) Create a nested array and flat it. try to use the [...] rest operator. What is the result?
- II) Create a deeper nested array 3D array and flat it ✓
- III) try to flat an array of objects
- IV) Create an object {name: balance: [.....]} with 3 names inside. map all the balances and result with 1 array, calculate its sum
- V) do it with flatMap() ✓

sort()

1) Create an array of string, sort it ✓

2) [numbers] → sort it, is it ok? if not - fix it with callback function ✓

fill(), Array.from()

1) Create an empty array with 6 places ✓

2) Fill it with 20 ✓

3) delete all the values, fill it with [, , , , 0, 1, 2] (at the end), then [, , 2, , , ,]

4) create an array using Array.from . 7 places, filled with 1 . 4B) create an array using Array.from [1, 2, ... 10] 4C) create an array with 100 random dice numbers

5) create an object and the create an array of this object ✓

array.Array.from(): receive an object

1: object with length property of length: x 3

2: some callback function () => 'to fill with'

Module

Modulariser le code : diviser le code, débogage

import, export, debug, ...
java -> opt export

nomé : export

par défaut : export default what ;

import: function, class...

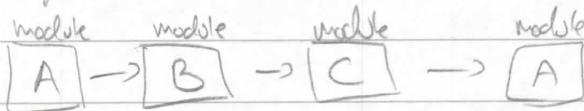
import what from 'chemain'

chemain is not chemin -> ref

String = chemin

in the html it's MUST <script type="module">

Dependance



dependance cyclique

dependency is some 3rd party code that your application depends on

Promise

negative result - positive result

callback hell

le moyen de faire ça

① pending

fulfill

② fulfill

waiting —

③ return

client



Jax_rs

ça va être

Etat initial de chaque promises: waiting...

immutable et pas un "set promise" -> immuable

constructor : 2 parameters

Promise = new Promise((resolve, reject) => { })

promise methods

function function
↑ ↑

.resolve (.value).then (resolved, rejected)

.then(): 2 argument 1: resolve 2: rejected

catch() => pour gener les echec et les erreurs. 1 argument

resolve() => return the new object Promise qui est resolu avec la valeur donnee
if it's followed by then

if not: the object will be filled with the value

reject(): new Error, return the promises Object that was rejected

all(): 1 argument an array! of promises

? map JS 3.5. ✓ Array-as promise -> GP not => promise (js de l'appli) block
Array-as promise B B of the iterator ->

race(): Array of promises resolve each promise -> 1st result

with promises: we don't stop the app and wait for the result

Print

window.print(). I can do it as an action to a button

Redirection

to the same website or another

window.location = "http:...."

- - - - - replace

Animation

setInterval (function () {
action }, timeout);

(timeout) ms x B, after which B runs
clearInterval(); a func to tell if 3s. 315

refresh rate, 60FPS

parseInt (sofx) → 50 . it stops to parse when it sees a String