

## UF3.NF1.JavaScript Funcions

Crea un projecte al teu IDE que es digui JSFuncions<nom><cogno<sup>oooo</sup>ms> i realitza els següents exercicis.

Treballa amb un repositori al Git per pujar cadascun dels exercicis amb un commit i adjunta en l'entrega la url.

A cada exercici has de crear una funció que realitzi la lògica demanada:

1.- Passa el teu nom i cognoms com a paràmetres i els mostri per consola.

```
1 usage new *
2 function nomicognom(nom, cognoms) :void {
3     console.log("El nom és: " + nom);
4     console.log("Els cognoms són: " + cognoms);
5 }
6 let nom :string = "Roger";
7 let cognoms :string = "Faixeda";
8
9 nomicognom(nom, cognoms);
```

El nom és: Roger	Funcions.js:2
Els cognoms són: Faixeda	Funcions.js:3

2.- Passa un valor per paràmetre i mostri de quin tipus és.

```
11 //Exercici 2
12
13 usage new *
14 function funcio2(parametre) :void {
15     console.log("El valor del parametre és " + typeof parametre)
16 }
17 parametre = 123
18
19 funcio2(parametre)
```

```
Run  Funcions.js x
C:\Program Files\nodejs\node.exe C:\Users\rfaix\Documents\1rDAM_RogerFaixeda\LEA\UF2\Funcions.js
(1) El nom és: Roger
(1) Els cognoms són: Faixeda Diez
(2) El valor del parametre és number
Process finished with exit code 0
```

### 3.- Mostri la data actual

```
21 //Exercici 3
22 1 usage new *
23 function dataActual() :void {
24     const data :Date = new Date();
25     const opciones :{...} = { year: 'numeric', month: 'long', day: 'numeric' };
26     const datanova :string = data.toLocaleDateString( locales: 'es-ES', opciones);
27     console.log("(3) La data actual és: " + datanova);
28 }
29
30 dataActual();
```

"C:\Program Files\nodejs\node.exe" C:\Users\rfaix\Documents\1rDAM\_RogerFaixeda\LEA\UF2\Funcions.js  
(1) El nom és: Roger  
(1) Els cognoms són: Faixeda Diez  
(2) El valor del parametre és number  
(3) La fecha actual es: 2 de noviembre de 2023  
Process finished with exit code 0

### 4.- Declara una variable dins d'una funció i mostra el seu valor fora d'aquesta. Què passa? Perquè?

No funciona ja que la variable només existeix dins la funció.

```
31 dataActual();
32
33 //Exercici 4
34 no usages new *
35 function variable() :void {
36     const valor1 :number = 10
37 }
38 console.log("El valor és: " + valor1)
```

### 5.- Passa't un String, retorni el número de vocals que té

```
39 //Exercici5
40 1 usage new *
41 function funcioString(cadena){
42     const vocals :RegExp = /[aeiouAEIOU]/g
43     const comptadorVocals = cadena.match(vocals)
44     return comptadorVocals.length
45 }
46 const cadena :string = "Hola"
47 const numVocals = funcioString(cadena)
48 console.log("(5) La cadena " + cadena + " te " + numVocals + " vocals.");
49
50 (5) La cadena Hola te 2 vocals.
```

6.- Passa 3 enters com a paràmetres (el tercer ha de tenir un valor per defecte) i retorni la suma dels tres.

```
49 //Exercici 6
   1 usage new *
50 function fun6(num1, num2, num3 : number = 0) {
51     return num1 + num2 + num3
52 }
53 const resultat = fun6( num1: 1, num2: 2, num3: 3)
54 console.log("(6) La suma dels valors és: " + resultat)
```

(6) La suma dels valors és: 6

7.- Ha de cridar la funció de l'exercici 3 i retornar el resultat de la funció de l'exercici 6.

```
56 //Exericci7
   1 usage new *
57 function fun7(){
58     dataActual()
59     return resultat
60 }
61 console.log("(7) El resultat de la funcio 6 és: " + fun7())
```

Run Funcions.js

(2) El valor del parametre és number  
(3) La data actual és: 2 de noviembre de 2023  
(5) La cadena Hola te 2 vocals.  
(6) La suma dels valors és: 6  
(3) La data actual és: 2 de noviembre de 2023  
(7) El resultat de la funcio 6 és: 6

Process finished with exit code 0

8.- Passa't dos enters, retornar true si un d'ells és igual a 30 o si la seva suma és igual a 30.

```
63 //Exercici 8
   2 usages new *
64 function fun8(a, b) : boolean {
65     let boolean
66     boolean = a === 30 || b === 30 || (a + b === 30)
67     return boolean
68 }
69 fun8( a: a = 15, b: b = 15)
70 console.log("(8) És: " + fun8(a, b))
```

(8) És: true

9.- Repeteix l'exercici anterior però com a funció anònima.

```
72 //Exercici 9
73
74 1 usage new *
75 let fun9 = function (c : number = 15, d : number = 15) : boolean {
76     let boolean
77     boolean = c === 30 || d === 30 || c+d === 30
78     return boolean
79 }
79 console.log("(9) És: " + fun9())
(9) És: true
```

Els següents exercicis s'han de realitzar amb funcions fletxa:

10.- Passa't dos valors, retorni cert o falç si el valor i el tipus de dades són el mateix.

```
81 //Exercici 10
82 1 usage new *
83 const fun10 = (num1, num2) =>{
84     return num1 === num2 && typeof num1 === typeof num2;
85 }
86 const resultat1 = fun10( num1: 5, num2: 5); // true, ambos valores son números
87 console.log("(10) Tenen el mateix valor: " + resultat1);
88
(10) Tenen el mateix valor: true
```

11.- Passa't un valor, realitzi l'arrel quadrada i retorni el valor amb dos decimals.

```
89 //Exercici 11
90 1 usage new *
91 const fun11 = (num) : number => {
92
93     const arrel : number = Math.sqrt(num);
94     const decimals : string = arrel.toFixed( fractionDigits: 2);
95     return parseFloat(decimals);
96 }
97 resultat = fun11( num: num = 30);
98 console.log("(11) L'arrel quadrada de " + num + " amb dos decimals és: " + resultat);
99
(11) L'arrel quadrada de 30 amb dos decimals és: 5.48
```

12.- Passa't un String, el retorni en minúscules.

```
99 //Exercici 12
100 1 usage new *
101 const fun12 = (cadena) => {
102     return cadena.toLowerCase();
103 }
104 resultat = fun12( cadena: "Introdueix una frase")
105 console.log("(12) La frase és: " + resultat)

(12) La frase és: introduceix una frase
```

13.- Passa't dos valors, retorni el que no és null.

```
106 //Exercici 13
107 1 usage new *
108 const fun13 = (valor1, valor2) : any =>{
109     if (valor1 == null) {
110         return valor2
111     } else {
112         return valor1
113     }
114 }
115 resultat = fun13( valor1: valor1 = "hola", valor2: valor2 = null)
116 console.log("(13) El valor no null és: " + resultat)

(13) El valor no null és: hola
```

14.- Defineix dos funcions:

- **posicio(frase, paraula):** se li passa'n dos Strings (frase, paraula). Ha de mirar en quina posició apareix la paraula en la frase i retornar quantes posicions queden entre aquesta i l'última posició. (Exemple: frase: 'Hola com estàs', paraula: 'com'. Ha de retornar un 9)
- **dividir(frase, paraula):** declara i crida la funció posicio. Amb el número retornat, obté els caràcters que hi han des d'aquesta posició fins al final de l'String. (A l'exemple, retornaria 'estàs')

```
117 //Exercici14
118
119 const frase :string = "Hola com estàs";
120 const paraula :string = "com";
121
122 2 usages new *
123 const fun14a = (frase, paraula) => frase.length - frase.indexOf(paraula)
124 console.log("(14 a) " + fun14a(frase, paraula));
125
126 1 usage new *
127 const fun14b = (frase, paraula) => frase.slice(fun14a(frase, paraula), frase.length)
128 console.log("(14 b) " + fun14b(frase, paraula))

(14 a) 9
(14 b) estàs
```

15.- Explica què fa la següent funció i fes-ne un cas de prova:

$(x) \Rightarrow x + x + x$

```
128 //Exercici 15
129 x = 2
    1 usage new *
130 let fun15 = (x) => x+x+x
131 console.log("(15) Suma 3 vegades el valor x: " + fun15(x))
132
(15) Suma 3 vegades el valor x: 6
```