

Lucas Fernández Aragón

Introducción a JavaScript y TypeScript

Actividad sesión 3

Presentación de la actividad

Objetivos

Los objetivos de esta sesión son adquirir conocimientos acerca de TypeScript y sus funcionalidades más relevantes.

Metodología

Ejercicio 1 (2)

- Crea una interfaz **'Person'** que tenga como atributos *'name'*, *'age'* y *'profession'*. Ahora define una arrow function que tenga como parámetro esta interfaz y que devuelva un array con **el valor de sus propiedades**, esta función tiene que tener **tipado el parámetro de entrada y el return**.

Ayuda -> El array probablemente necesite [union type](#)

```
// input name -> Lucas / age -> 28 / profession -> Full Stack  
// return -> ["Lucas", 28, "profession"]
```

Ejercicio 2 (3)

- Escribe una función llamada *sumOrConcatenate* que acepte **dos parámetros**. Cada uno de estos parámetros podrá ser de tipo *number* o *string*. La función devolverá una suma si los dos parámetros son números, una concatenación con el símbolo ``-`` si son los dos ``strings`` o una cadena concatenada con ``:`` si uno es un *number* y el otro *string*.

```
console.log(sumOrConcatenate(4, 3)); // --> 7  
console.log(sumOrConcatenate(4, "hello")); // --> 4:hello  
console.log(sumOrConcatenate("hello", "world")); // --> hello-world
```

Ejercicio 3 (3)

- Crea dos interfaces, una llamada ``Car`` y otra ``Motorcycle``. La primera tendrá las propiedades ``tires`` (*number*), ``turnOnEngine()`` (función que devuelve void) y ``pressPedal()`` (función que devuelve void).

- La segunda tendrá las propiedades `tires` (number), `turnOnEngine()` (función que devuelve void) y `openThrottle()` (función que devuelve void).
 - Escribe una función que acepte un parámetro que pueda ser `Car` o `Motorcycle` que, primero llame a `turnOnEngine`, y luego si es `Car` llame a `pressPedal` pero si es `Motorcycle` llame a `openThrottle()`.
 - Para la comprobación no valdrá `typeof`, necesitaremos hacer uso de [type predicates](#).
-

Ejercicio 4 (2)

- Crea una `función genérica`, que acepte un array que pueda contener `strings` y `numbers` y devuelva el **mismo array sin el primer elemento**.

```
const strArray: string[] = ['Hello', 'World', 'Im', 'a', 'Full', 'Stack', 'Developer'];
const numArray: number[] = [1, 2, 3, 4, 5, 6, 7];
const mixedArray: Array<number|string> = ['Hello', 'I', 'have', 3, 'tasks'];
const unsupportedArray = [{name: 'Lucas', surname: 'Fernandez'}, 'Hello', 22];

const newStrArray = removeFirstEntry(strArray);
const newNumArray = removeFirstEntry(numArray);
const newMixedArray = removeFirstEntry(mixedArray);
// const newUnsupportedArray = removeFirstEntry(unsupportedArray); --> will fail

console.log(newStrArray); // --> ['World', 'Im', 'a', 'Full', 'Stack', 'Developer'];
console.log(newNumArray); // --> [2, 3, 4, 5, 6, 7];
console.log(newMixedArray); // --> ['I', 'have', 3, 'tasks'];
```

Criterios de evaluación

Los ejercicios se ponderarán con un total de 10 puntos, valiendo 2 puntos el 1 y el 4 y 3 puntos el 2 y el 3. Las entregas posteriores a la fecha de entrega fijada serán penalizadas a criterio del profesor.

Formato y fecha de entrega

El último día de entrega de la actividad será el **20 de Enero de 2023**, entregas posteriores a esa fecha serán penalizadas.