wawiwa

Funciones

Ya hemos aprendido sobre funciones, qué son y por qué las usamos.
En este capítulo ampliaremos sobre funciones

Hay 4 maneras distintas de escribir funciones en Javascript:
1.) Function declaration (declaración de función)
2.) Function expression (expresión de función)
3.) Arrow function (función de flecha)
4.) Immediate function (función inmediata)

wawiwa

Declaración de función

Declaración de función es la manera tradicional de definir una función.
Empezamos a declararla usando la clave *"function"*.
Después escribimos el nombre de la función y después los parámetros.

```
<html>
   <body>
    <script>
        // Declaración de función
        function add(a, b) {
            console.log(a + b);
        }
        // Llamando a la función
        add(4, 5);
    </script>
   </body>
</html>
```

wawiwa

Funciones de expresión

Las funciones de expresión son una manera distinta de definir una función en JavaScript.

Aquí definimos una función usando una variable y guardando el valor retornado en esa **variable**.

```
<html>
   <body>
    <script>
        // Expresión de función
        const add = function(a, b) {
            console.log(a + b);
        };

        // Llamando a la función
        add(4, 5);
    </script>
   </body>
</html>
```

wawiwa

Funciones de expresión

Esta función puede ser **anónima** - no tiene que tener un nombre.
Por ejemplo, la función “square” puede ser definida así:

```
const square = function(number) {
     return number * number;
};

var x = square(4); // x obtiene el valor de 16
```

wawiwa

Funciones de expresión

Sin embargo, un nombre ***puede*** ser dado a una función de expresión.

Darle un nombre le permite a la función referirse a ella misma y también hace más fácil identificar la función en los trazos del debugger:

```
const factorial = function fac(n) {
    if (n > 0 && n <= 1) {
        return 1;
    } else {
        return n * fac(n - 1);
    }
}

console.log(factorial(3)); // = 6
```

Factorial Formula

$$n! = n \times (n-1) \times (n-2) \times \ldots \times 1$$

$$1! = 1$$
$$2! = 2 \times 1 = 2$$
$$3! = 3 \times 2 \times 1 = 6$$
$$4! = 4 \times 3 \times 2 \times 1 = 24$$
$$5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$$

wawiwa

Funciones de expresión

En js, una función puede ser definida basada en una condición. Por ejemplo, la siguiente función define myFunc solo si num es igual a `0`:

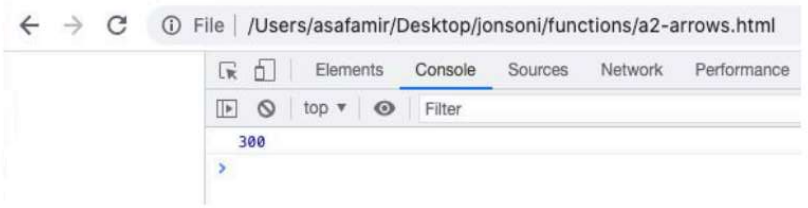
```
var myFunc;
var num = 0;
if (num === 0) {
    myFunc = function() {
        return num++;
    };
    myFunc();
}
console.log(num); // 1
```

wawiwa

Función de flecha

Las funciones de flecha han sido introducidas con la versión ES6 de js: Es usado para **acortar** el código.

```
<html>
    <body>
        <script>
            const calcBonus = (bonus) => bonus + 100;
            const finalBonus = calcBonus(200);
            console.log(finalBonus); // 300
        </script>
    </body>
</html>
```



wawiwa

Función de flecha

```
<html>
    <body>
        <script>
            const priceCar = (sits) => sits * 1000;
            const finalPrice = priceCar(4);
            console.log(finalPrice);
            // Borra los paréntesis de los argumentos
            const priceCar = sits => sits * 1000;
        </script>
    </body>
</html>
```

wawiwa

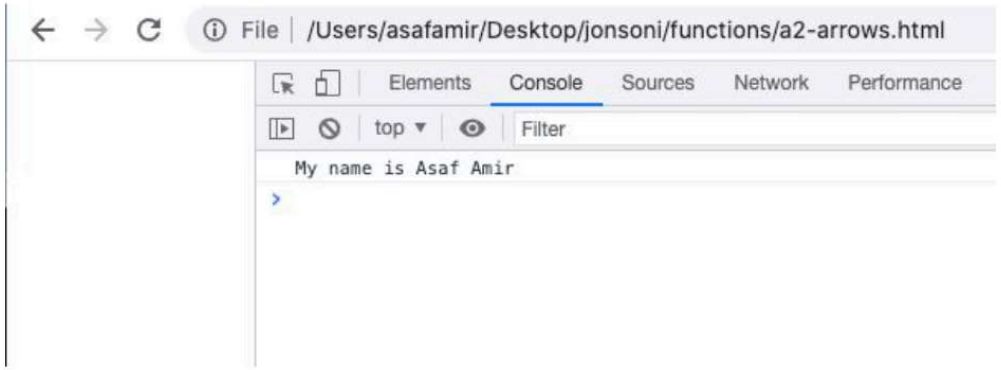
Función de flecha

```
<html>
   <body>
       <script>
           const priceCar = sits => {
               return = sits * 1000;
           };
           const finalPrice = priceCar(4);
           console.log(finalPrice); // 4000
       </script>
   </body>
</html>
```

wawiwa

Función de flecha

```
const getFullName = (fname, lname) => {
   return "My name is " + fname + " " + lname;
};
console.log(getFullName("Asaf", "Amir"));
```



wawiwa

Función inmediata

Una función inmediata es la que se ejecuta tan pronto como se define.

```
<html>
    <body>
        <script>
            (function(a, b) {
                console.log(a + b);
            })(3, 5);
        </script>
    </body>
</html>
```

wawiwa

¿Qué se va a imprimir?

```
<script>
    max();

    function max() {
        console.log(a > b ? a : b);
    }

    var a = 5, b = 6;
</script>
```

wawiwa

¿Qué se va a imprimir?

```
<script>
    max();

    function max() {
        console.log(a > b ? a : b); // undefined
    }

    var a = 5, b = 6;
</script>
```

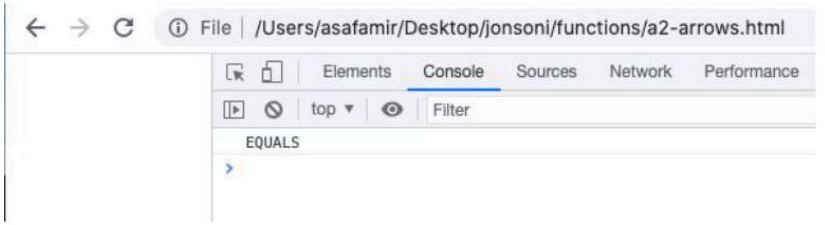
wawiwa

Funciones inmediatas – Continuación

```
var theBiggest = (function(n1, n2) {
    if (n1 > n2) return n1;
    else if (n1 < n2) return n2;
    else return "EQUALS";
})(a,b);

var a = 6, b = 7;

console.log(theBiggest); // EQUALS
```



wawiwa

Hazlo tú mismo 1

A. Escribe una función de **Declaración** que acepte 3 nombres. El nombre retorna el número más grande.

```
function maxBetween3Numbers(n1, n2, n3) {}
```

B. Agrega un botón con la leyenda “find max”.

a. Agrega un párrafo con id="max”.

b. Escribe un script que reciba 3 números de la ventana emergente.

c. El script mostrará en id="max” el número más grande (**debes** usar la función de la parte A).

d. Invoca las funciones al presionar el botón

wawiwa

Hazlo tú mismof 2

A. Escribe una función de **expresión** que acepte 3 números. La función retorna el número más grande.

```
function maxBetween3Numbers(n1, n2, n3) {}
```

B. Add a button with the caption find max.

a. Agrega un párrafo con id="max".
b. Escribe un script que reciba 3 números de la ventana emergente.
c. El script mostrará en id="max" el número más grande (**debes** usar la función de la parte A).

wawiwa

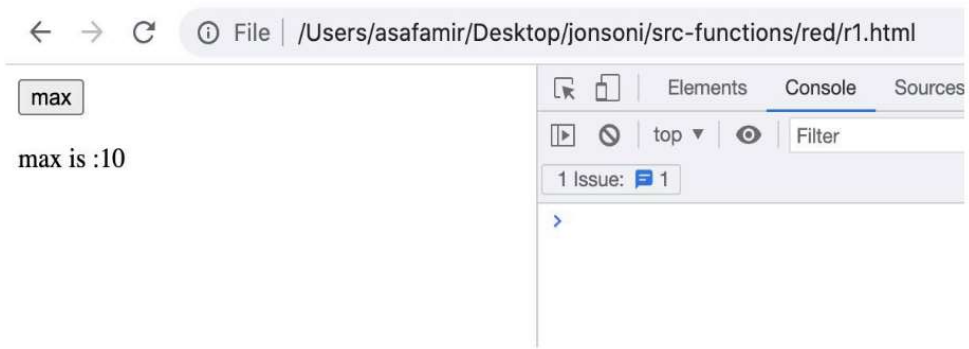
Hazlo tú mismo 3

A. Escribe una función de **flecha** que acepte 3 números. La función retorna el número más grande.

```
function maxBetween3Numbers(n1, n2, n3) {}
```

B. Add a button with the caption find max.

a. Agrega un párrafo con id="max".
b. Escribe un script que reciba 3 números de la ventana emergente.
c. El script mostrará en id="max" el número más grande (**debes** usar la función de la parte A).



wawiwa

Hazlo tú mismo 4

A. Escribe una función de **declaración** que acepte un arreglo y retorne el número más grande.
B. Invoca la función de A usando arr = [1, 6, 2, 9] y alerta el número más grande.

wawiwa

Hazlo tú mismo 5

A. Escribe una función de **expresión** que acepte un arreglo y muestre en alerta con el número más grande.

B. Invoca la función de A usando arr = [1, 6, 2, 9] y alerta el número más grande.

wawiwa

Hazlo tú mismof 6

A. Escribe una función de **flecha** que acepte un arreglo y muestre en alerta con el número más grande.
B. Invoca la función de A usando arr = [1, 6, 2, 9] y alerta el número más grande.

wawiwa

Hazlo tú mismo 7

A. Escribe una función de **declaración** que acepta un arreglo y un número. La función retornará true si el número existe en el arreglo.
B. Invoca la función de A usando arr = [1, 6, 2, 9] y num = 6, y alerta si el número existe en el arreglo.

wawiwa

Hazlo tú mismo 8

A. Escribe una función de **expresión** que acepta un arreglo y un número. La función retornará true si el número existe en el arreglo.

B. Invoca la función de A usando arr = [1, 6, 2, 9] y num = 6, y alerta si el número existe en el arreglo.

wawiwa

Hazlo tú mismo 9

A. Escribe una función de **flecha** que acepta un arreglo y un número. La función retornará true si el número existe en el arreglo.

B. Invoca la función de A usando arr = [1, 6, 2, 9] y num = 6, y alerta si el número existe en el arreglo.

wawiwa

Hazlo tú mismo 10

A. Escriba una función de **declaración** que acepta como parámetro 3 números y muestra el número más chico en alerta .
B. Llama la función de A con n1 = 3, n2 = 5, n3 = 9 y alerta el número más chico.

wawiwa

¿Preguntas?