Argentina programa 4.0

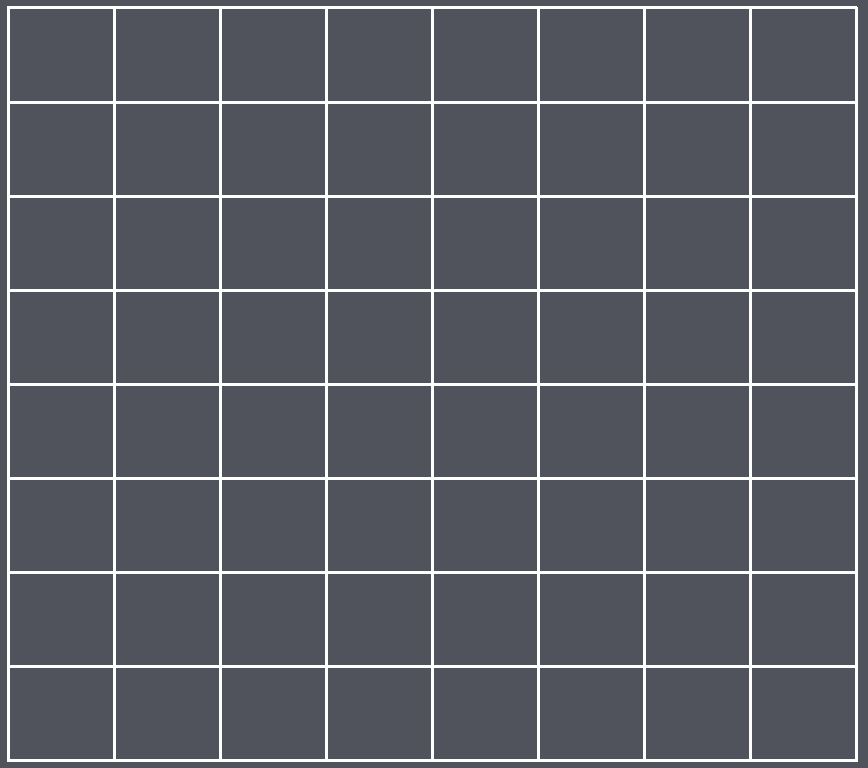
Javascript avanzado y jQuery

Programa “Introducción a la Programación Web”

Javascript avanzado y jQuery

- Programación Funcional

- JQuery



Programación Funcional - JavaScript

JavaScript es un lenguaje de múltiples paradigmas.

Paradigmas de Programación:

- Declarativos(especifica lo que ***quieres hacer, sin decir como***)
- Imperativos(especifica ***cómo hacer algo***)

La programación funcional es un sub-paradigma del paradigma de programación declarativa.

Programación Funcional - JavaScript

Argentina programa 4.0

¿Qué es la programación funcional?

La programación funcional es un paradigma de programación o un estilo de programación que se basa en gran medida en el uso de funciones puras.

Tal como puede haber adivinado por el nombre, el uso de funciones es el componente principal de la programación funcional.

¿Y cuáles son las reglas que llevan a la programación funcional?

Puede ser explicada simplemente siguiendo estas dos reglas en el código:

1. Diseñar el software a partir de funciones puras
2. Evitar mutabilidad y efectos secundarios

Funciones Puras

La misma entrada siempre da la misma salida (idempotencia) y no tiene efectos secundarios.

Ausencia de efectos colaterales; inmutabilidad

No modificamos nada fuera de nuestra función.

Esto supone en última instancia que los resultados son siempre reproducibles a partir de una idéntica entrada.

Funciones de orden superior

Las funciones de orden superior son aquellas que reciben una o más funciones como argumento o bien devuelven funciones como resultado.

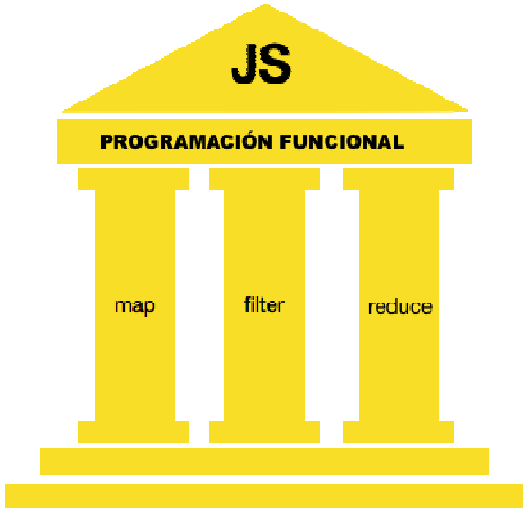
Programación Funcional - JavaScript

Argentina programa 4.0

Funciones de orden superior

En especial nos serán muy útiles conocer los operadores funcionales: map , filter y reduce.

Pilares de la programación funcional en JavaScript, su interés principal está en usarlos sobre arrays.



Javascript - Operadores Funcionales

Argentina programa 4.0

Javascript - Operadores Funcionales

Argentina programa 4.0

Filter

"El método filter() crea un nuevo array con todos los elementos que cumplan la condición implementada por la función dada"

=> El método filter no genera efecto colateral.

Ejemplo:

```
const palabras = ['casa', 'perro', 'electrodoméstico', 'universidad',
'mueblería', 'gato'];

const resultado = palabras.filter(p => p.length > 6);

console.log(resultado);

// expected output: Array ["electrodoméstico", "universidad, "mueblería"]
```

Javascript - Operadores Funcionales

Argentina programa 4.0

Map

"El método map() crea un nuevo array con los resultados de la llamada a la función indicada aplicados a cada uno de sus elementos."

=> El método map no genera efecto colateral.

Ejemplo:

```
var numbers = [1, 5, 10, 15];
var doubles = numbers.map(function(x) {
    return x * 2;
}); // doubles is now [2, 10, 20, 30]
// numbers is still [1, 5, 10, 15]
```

Javascript - Operadores Funcionales

Argentina programa 4.0

Reduce

"El método reduce() ejecuta una función reductora sobre cada elemento de un array, devolviendo como resultado un único valor."

=> El método reduce no genera efecto colateral.

Ejemplo:

```
const array1 = [1, 2, 3, 4];

const reducer = (accumulator, currentValue) => accumulator + currentValue; //
1 + 2 + 3 + 4

console.log(array1.reduce(reducer)); // expected output: 10 (1 + 2 + 3 + 4)

console.log(array1.reduce(reducer, 5)); // expected output: 15
```

Javascript - Operadores Funcionales

Argentina programa 4.0

Find

“El método find() devuelve el valor del primer elemento del array que cumple la función de prueba proporcionada.”

=> El método find no genera efecto colateral.

Ejemplo:

```
const numeros = [5, 12, 8, 130, 44];
const nroMayorADiez = numeros.find(n => n > 10);
console.log(nroMayorADiez);
// expected output: 12
```

Javascript - Operadores Funcionales

Argentina programa 4.0

Some

“El método some() comprueba si al menos un elemento del array cumple con la condición implementada por la función proporcionada.”

Ejemplo:

```
const numeros = [1, 2, 3, 4, 5];
// comprobamos si algún element es par
console.log(numeros.some(n => n % 2 === 0 ));
// expected output: true
```

Javascript - Operadores Funcionales

Argentina programa 4.0

Every

“El método every() determina si todos los elementos en el array satisfacen una condición.”

Ejemplo:

```
const numeros = [1, 2, 3, 4, 5];
console.log(numeros.every(n => n >=3));
// expected output: false
```

Javascript - Funciones anónimas

Argentina programa 4.0

Las funciones anónimas, son funciones que al momento de declararlas no les asignamos un nombre.

```
var saludo = function (){
        // código;
};
```

Podemos asignar estas funciones a una variable. Para ejecutar dicha función debemos hacerlo llamándola por el nombre de la variable.

```
var saludo = function (persona){
    alert("Hola " + persona);
}

saludo("Rodrigo");
```

Javascript - Funciones anónimas

Argentina programa 4.0

Funciones anónimas auto-invocadas / autoejecutables

Éstas funciones, una vez declaradas, se llaman a sí mismas, ejecutando el código definido en ellas.

```
(function (){código;})();
```

Una de las razones por las que son útiles es porque permiten **ejecutar** código **encapsulando** variables al entorno de la función ahorrando posibles errores.

Con función anónima

```
> (function (){
      let alumno = "Juan";
      console.log(alumno);
  })();
  Juan
```

Sin función anónima

```
> let alumno = "Juan";
  console.log(alumno);
  Juan
```

¿Cuál es la diferencia?

Javascript - setTimeout()

El método setTimeout() llama una función o evalúa una expresión después de un número específico de milisegundos.
La función solo se ejecuta una vez.

```
function miFuncion(){
        setTimeout(otraFuncion,3000)
};

function otraFuncion(){
        alert("Somos necesarios, un fragmento de la fatalidad; formamos parte del todo, somos en el todo; no hay nada que pueda juzgar, medir, comparar y condenar nuestra existencia, pues ello equivaldría a juzgar, medir, comparar y condenar el todo. Friedrich Nietzsche");
};

miFuncion(); //llamamos a la función
```

Javascript - setTimeout()

Argentina programa 4.0

Parámetros

setTimeout(a,b);

a = la función que se va a ejecutar pasado el tiempo definido.
b = el tiempo que debe “esperar” antes de ejecutar la función. En milisegundos.

1000 ms = 1 seg
El parámetro “a” es una función previamente definida que podemos llamar o una *función anónima* definida dentro del mismo setTimeout

```
setTimeout(function(){ alert("Hello") }, 3000);
```

Javascript - setInterval()

Argentina programa 4.0

El método setInterval() llama a una función o evalúa una expresión cada determinados intervalos de tiempo.

```
var myVar = setInterval(consoleLogFunc, 3000);


function consoleLogFunc() {
         console.log("Hola!");
};
```

Detener función

```
function detenerIntervalo(){
         clearInterval(myVar)
};
```

Javascript - setInterval()

Argentina programa 4.0

Parámetros

setInterval(a,b);

a = la función que se va a ejecutar después del intervalo de tiempo definido.
b = el tiempo que debe “esperar” antes de ejecutar la función. En milisegundos.

El parámetro “a” es una función previamente definida que podemos llamar o una *función anónima* definida dentro del mismo setTimeout

```
setInterval(function(){ alert("Hello") }, 3000);
```

JQuery - ¿Qué es?

Argentina programa 4.0

Según jquery.com, jQuery es una **biblioteca** de JavaScript rápida, pequeña y rica en funciones.

Permite que cosas como el recorrido y la *manipulación* de documentos HTML, el manejo de *eventos*, la *animación* y *Ajax* sean mucho más simples.

JQuery - ¿Cómo se usa?

Argentina programa 4.0

Podemos empezar a usar jQuery en nuestra página web descargando la librería linkeandola localmente o incluyendola desde un CDN (Content Delivery Network).

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
</head>
```

JQuery - Sintaxis

Argentina programa 4.0

La sintaxis de jQuery está creada para seleccionar elementos HTML y realizar acciones con ellos.

La sintaxis básica es:

```
$(selector).acción()
```

Ejemplos:

```
$('p').hide()
$('.test').hide()
$('#test').hide()
```

Oculta todos los elementos <p>

Oculta todos los elementos con la clase 'test'

Oculta el elemento con id igual a 'test'

Manipulación de elementos: Selectores

Argentina programa 4.0

Syntax	Description	Syntax	Description
`$("*")`	Todos los elementos	`$(this)`	El elemento actual
`$("p.intro")`	Todos los <p> con la clase 'intro'	`$("p:first")`	El primer elemento <p>
`$("ul li:first")`	El primer del primer	`$("tr:even")`	Todos los elementos <tr> par
`$("tr:odd")`	Todos los elementos <tr> impares	`Etc`	Etc

JQuery - Manipulación de elementos

Argentina programa 4.0

Generalmente se escribe el código jQuery dentro de un evento *document ready.*

```
$(document).ready(function(){
        // jQuery aquí…
});
```

Esto se hace para prevenir que el código jQuery corra antes de que el documento esté completamente cargado.
Algunos ejemplos de acciones que pueden fallar si los métodos corren antes de que la página esté completamente cargada:
- Esconder un elemento que no ha sido creado aún.
- Obtener el tamaño de una imágen que aún no cargó.

Esto también permite colocar nuestros <script> en el <head> de la página.
(Ya que nuestro código se va a ejecutar una vez que el documento esté cargado)

JQuery - Manipulación de elementos

Argentina programa 4.0

Algunas de las funciones que nos brinda jQuery para manipular nuestros elementos son:

```
.width()
.height()
.addClass()
.append()
.attr()
.position()
.remove()
```

Muchas de estos métodos sirven tanto para *obtener* como para *definir* propiedades.
Por ejemplo:
`$("#logo").width()` ---> Nos devuelve el ancho de el elemento con id = "logo"
`$("#logo").width(600)` ---> Setea el ancho del elemento "logo" a 600px;

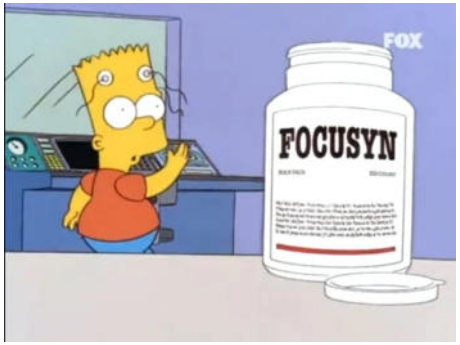
Pueden encontrar más funciones para manipular elementos en la página oficial de jQuery.

JQuery - Manejo de eventos

Así cómo jQuery nos ofrece funciones para solucionar de forma más rápida y sencilla la manipulación de elementos, también nos brinda métodos para el manejo de eventos.

Algunos de ellos son:

```
.click()
.change()
.dblclick()
.focus()
.focusin()
.focusout()
.hover()
```



Para ver la lista completa de eventos que podemos manejar con jQuery, podemos acceder a api.jquery.com/category/events/

Validar un formulario desde el Front-End

Argentina programa 4.0

¿Cómo usamos los métodos del slide anterior?

```
$(document).ready(function(){
  $("#achicarImagen").click(function(){
    var anchoImg = $("#nuestraImagen").width();
    if (anchoImg >= 1200) {
      $("#nuestraImagen").width(300);
    };
  });
});
```

JQuery - Animaciones y otras funciones

Argentina programa 4.0

hide()	Hides the selected elements
queue()	Shows the queued functions on the selected elements
show()	Shows the selected elements
slideDown()	Slides-down (shows) the selected elements
slideToggle()	Toggles between the slideUp() and slideDown() methods
slideUp()	Slides-up (hides) the selected elements
stop()	Stops the currently running animation for the selected elements
toggle()	Toggles between the hide() and show() methods

Referencias

Argentina programa 4.0

- JQuery
- Efectos y Animaciones - JQuery
- Selectores - JQuery
- SetTimeOut - Javascript
- Arrow Functions - Javascript

Argentina programa 4.0

¿Preguntas?

Argentina
programa
4.0

Gracias!