

# PROGRAMACIÓN EN NUEVAS TECNOLOGÍAS 2

**Javascript, ES6**

# ES5, ES6, ES2016, ES2017, ES.NEXT

- ECMAScript vs JavaScript
- ¿Que soportan la mayoría de los entornos?
- Transpilers (Babel, TypeScript, CoffeeScript, etc.)
- ¿Que sintaxis debería usar?

# CLOSURES

- Funciones declaradas dentro del ámbito léxico.
- Tienen acceso a variables declaradas en la función padre.
- Es posible gracias al sistema de ámbitos de Javascript.

# FUNCIONES DE PRIMER NIVEL

- Funciones tratadas de la misma manera como si fuesen variables.
  - Pueden ser asignadas a variables, valores de un array o incluso valores de un objeto.
  - Puede ser pasado como argumento de otra función
  - Puede ser retornados desde funciones
- Permiten la creación de funciones de alto nivel.
  - Tomando una o más funciones como argumentos, o incluso devolviendo una función.
  - Ejemplos: `map()`, `filter()`, `reduce()`, `pop()`

# ASINCRONICIDAD

- Javascript es un lenguaje sincronizado, ejecutado en un solo hilo.
- Una función que toma mucho tiempo correr va a causar problemas de performance y eventualmente congelar la página.
- Javascript tiene funciones que actúan de forma asíncrona.

# ASINCRONICIDAD

- Ejecución en Pilas
- Browser APIs
- Función de encolado: Function queue
- Ciclo de eventos.

# ASINCRONICIDAD

- Ejemplo de funciones asíncronas
  - `setTimeout()`
  - `XMLHttpRequest()`, `jQuery.ajax()`, `fetch()`, `axios`.
  - Llamadas a base de datos (persistencias locales).

# CALLBACKS

- Flujo de control con llamadas asíncronas.
- Ejecuta la función una vez la llamada asíncrona retorna un valor.



# PROMESAS (PROMISES)

- Solución al problema de callbacks
- Te permite escribir código que asume que un valor es retornado dentro de una función completada.
- Solo necesita un solo manejador de errores.

# REFERENCIA

- [https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global\\_Objects/Array](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Global_Objects/Array)
- [https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/Spread\\_syntax](https://developer.mozilla.org/es/docs/Web/JavaScript/Reference/Operators/Spread_syntax)