



# **Arrays Asociativos**

# Repaso

- Hasta el momento hemos trabajado con los arrays indexados, es decir, arrays en donde cada uno de sus elementos puede ser referenciado con un índice (número)

- Formas de declarar un array:

```
const arreglo1 = ["Juan", "Pepe"];
```

```
const arreglo2 = new Array("Juan", "Pepe");
```

```
const arreglo2 = Array.of("Juan", "Pepe");
```

- Vimos que los arrays son OBJETOS, no son datos primitivos.

# Arrays indexados

- Como mencionamos, podemos acceder a cada elemento a través de su índice.

```
const arr = ["Juan", "Ana", "Pedro"];
```

```
console.log(arr[0]);    // Juan
```

```
console.log(arr[1]);    // Ana
```

```
console.log(arr[2]);    // Pedro
```

-

# Arrays Asociativos (o Arrays de Objetos)

- Son otro tipo de colección de datos, en donde sus elementos son objetos.
- A la hora de referenciar los elementos de un array, lo hacemos a través de una clave (o key), en lugar de un índice numérico.

Ejemplo básico:

```
const objetoPersona = {  
  nombre: "Joaquin",  
  edad: 18  
};  
const array = new Array(obj);
```

# Array de objetos - Ejemplo 2:

```
> const compu = {  
    nombre: "MiPc",  
    procesador: "Intel Core i7",  
    ram: "16GB",  
    dd: "2TB"  
};  
⏏ undefined  
  
> console.log(compu["nombre"]);  
MiPc  
⏏ undefined  
  
> console.log(compu["dd"]);  
2TB  
⏏ undefined
```

# Referencia a elementos del objeto

```
> const compu = {  
  nombre: "MiPc",  
  procesador: "Intel Core i7",  
  ram: "16GB",  
  dd: "2TB"  
};  
< undefined  
  
> compu.nombre;  
< "MiPc"  
  
> compu["dd"];  
< "2TB"
```

Podemos hacer referencia a cada elemento del objeto de la forma:  
`compu.nombre`

O de la forma `compu["nombre"]`

# Array de objetos

```
> const arrayPersonas = [  
  {nombre: "Pablo", edad: 20},  
  {nombre: "Ana", edad: 21},  
  {nombre: "Maria", edad: 18},  
  {nombre: "Felipe", edad: 25},  
]  
< undefined  
  
> arrayPersonas[1].edad  
< 21  
  
> arrayPersonas[0].nombre  
< "Pablo"  
  
> arrayPersonas[2]["nombre"]  
< "Maria"  
  
>
```

Aquí tenemos un array formado por objetos. Podemos acceder a cada elemento del array con su índice, y a su vez, acceder a cada elemento del objeto con las formas vistas.

# Otro ejemplo mas complejo

```
> const persona = {  
  nombreCompleto: "Juan Pérez",  
  esCasado: false,  
  fechaNacimiento: "1980-02-20",  
  empleo: {  
    nombreEmpresa: "Software UY",  
    cargo: "Senior Backend Developer",  
    cargaHoraria: "Full Time"  
  },  
  titulo: "Analista Programador"  
}  
< undefined  
  
> persona.empleo.cargo  
< "Senior Backend Developer"  
  
> |
```





# Ejercicio:

- Realizar un programa que pida los siguientes datos de 5 estudiantes:
- Nombre completo, Carrera cursando, cantidad de materias aprobadas, y cantidad de materias reprobadas.
- Agrupar todos los datos en un array de objetos.
- Luego, mostrar por consola todos los datos de cada estudiante (usar una línea de impresión por cada estudiante)

# Resolución - Parte 1: Pedir datos

Una posible  
solución:

```
</head>
<body>
  <script>
    // declaro un array vacio para ir agregando los objetos:
    const datos = [];

    for (let i = 1; i <= 5; i++) {
      alert("Estudiante Nro. " + (i));
      const nombre = prompt("Ingrese su nombre");
      const carrera = prompt("Ingrese su carrera");
      const aprobadas = parseInt(prompt("Cantidas de aprobadas"));
      const reprobadas = parseInt(prompt("Cantidas de reprobadas"));
      // con push agregamos un elemento al array
      datos.push({
        nombreCompleto: nombre,
        carrera: carrera,
        cantidadAprobadas: aprobadas,
        cantidadReprobadas: reprobadas
      });
    } // cierre for

  </script>
```

# Resolución - Parte 2: Mostrar datos

```
    } // cierre for

    // ahora recorreremos el array asociativo para mostrar cada elemento:
    for (let alumno of datos) {

        const salida = "Nombre: " + alumno.nombreCompleto
            + ", Carrera: " + alumno.carrera
            + ", aprobadas: " + alumno.cantidadAprobadas
            + ", reprobadas: " + alumno.cantidadReprobadas;

        alert(salida);
    }

script>
```



# Proxima clase:

- Terminamos con la primer parte de arrays del curso.
- Haremos varios ejercicios de lo visto hasta el momento.