Arrays Asociativos – Parte II

Repaso de la clase anterior

- Vimos que los arrays asociativos son arrays de objetos.
- Una forma de declarar un objeto es la siguiente:

  ```
  const miObjeto = { propiedad1: “Dato”, propiedad2: 1 };
  ```

- Los arrays asociativos son arrays como los que ya habíamos visto, sólo que se componen de objetos.
- `const array = [miObjeto];  // un array con un solo elemento`

Referencia a cada elemento del array

- Dado el array:

```
const arr = [
    { name: “Tony”, age: 17 },
    {name: “Bob”, age: 20}
];
```

- Para acceder al nombre del segundo elemento:

  puedo hacerlo de dos maneras distintas:

  arr[1].name

  arr[1][“name”]

Objetos

- Cada elemento de nuestro objeto puede tener cualquier tipo de dato, desde los datos primitivos, hasta funciónes, otros objetos, o incluso arrays.

```
const gato = {
    nombre: "Tom",
    edad: 3,
    color: "white",
    vacunasAlDia: false,
    maullar: () => {
        console.log("MIAU")
    },
    pedirComida: () => {
        console.log("miauuuuu miauuuu");
    },
    comidasFavoritas: ["Sushi", "Carne", "Arroz"]
}
```

Analizando el objeto anterior

- El objeto llamado gato contiene strings, enteros, booleanos, pero también guarda funciónes y un array de strings.
- Para acceder a las funciónes lo hago de la siguiente forma:

```
> gato.pedirComida();
  miauuuuu miauuuu
<- undefined
> gato["maullar"]();
  MIAU
<- undefined
>
```

Al final de la llamada a la propiedad, debo colocar las llaves (), para ejecutar la función.

Funciones en un objeto

```
> gato.pedirComida
<· () => {
            console.log("miauuuuuu miauuuu");
        }
> gato.pedirComida()
  miauuuuuu miauuuu
<· undefined
```

Arrays dentro de un objeto:

```
> gato.comidasFavoritas[0]
<- "Sushi"
> gato["comidasFavoritas"][1]
<- "Carne"
>
```

Debo indicar la propiedad del objeto, y luego el índice del array, como lo hacíamos con los arrays comunes

Recorrer un array de objetos

```
const estudiantes = [
    {
        nombre: "Antonio Perez",
        cursos: ["Ingeniería", "Intro a la Prog"],
        solicitarEntrevista: function() {
            console.log("Solicitud enviada...")
        }
    },
    {
        nombre: "Ana Barrios",
        cursos: ["Arquitectura", "Curso de Excel"],
        solicitarEntrevista: function() {
            console.log("Solicitud enviada...")
        }
    },
    {
        nombre: "Jose Rodríguez",
        cursos: [],
        solicitarEntrevista: function() {
            console.log("Solicitud enviada...")
        }
    }
];
```

1- Mostrar cantidad de cursos por estudiante:

- Una forma de resolver lo pedido:

```
function cantidadDeCursos() {
    for (const e of estudiantes) {
        const total = e.cursos.length;
        console.log("El estudiante " + e.nombre + " cursa " + total + " cursos");
    }
}
```

El resultado:

```
> cantidadDeCursos();
  El estudiante Antonio Perez cursa 2 cursos
  El estudiante Ana Barrios cursa 2 cursos
  El estudiante Jose Rodríguez cursa 0 cursos
< undefined
>
```

2- En caso que el estudiante no tenga cursos, pedirle una entrevista

- Una posible solución:

```
function pedirEntrevistaEstudiantesSinCursos()
{
    for(const e of estudiantes) {
        if (e.cursos.length == 0) {
            console.log("Solicitando entrevista a " + e.nombre);
            e.solicitarEntrevista();
        }
    }
}
```

- Al llamar a esta función:

```
> pedirEntrevistaEstudiantesSinCursos();
  Solicitando entrevista a Jose Rodríguez
  Solicitud enviada...
<- undefined
> estudiantes[2].nombre
<- "Jose Rodríguez"
> estudiantes[2].cursos.length
<- 0
```

Ejercicio:

- Crear una función que recorra el array de estudiantes

  Y solicite una entrevista a aquellos estudiantes que estén cursando la carrera (o curso) de Ingeniería

Próxima clase:

- Arrancamos con Programación Orientada a Objetos
- Veremos una breve introducción, y comenzamos con los conceptos de clases, constructor, instancias, etc.