

1. Vamos a hacer un programa que le pregunte al usuario la edad de su perro. Como todos sabéis, la edad de las mascotas se calculan multiplicando por 7 los años humanos. El usuario nos dirá la edad de su perro en años humanos. Nuestro programa le imprimirá al usuario el mensaje:

“Tu perro tiene X años en edad de perros”

2. Crear un programa que declare 3 personas con sus nombres y edades (6 variables). El programa pide al usuario el número de la persona que quiere buscar para poder ver su nombre y edad.

Si pone el numero 1, el programa debe imprimir :

“Persona 1 - Nombre: Julia - Edad: 29”

Si pone el número 2, debe mostrar la segunda persona, si pone el 3, la tercera.

3. Crear un programa que recibe la edad de una persona. El programa debe entonces enseñar un mensaje con el nivel de prioridad de atención en un hospital.

El hospital considera que personas de 0 a 15 años tienen nivel de prioridad ALTO, de 16 a 25 nivel MEDIO-ALTO, de 26 a 35 nivel BAJO, de 36 a 55 nivel MÉDIO-ALTO, de 56 a 70 nivel ALTO, de 71 a 120 nivel MÁXIMO.

4. Vamos a hacer un conversor de temperaturas. Vuestra página web va a solicitar al usuario que le diga la unidad de partida (“Celsius” o “Fahrenheit”). Luego le preguntará por la cantidad de grados en esta unidad de partida.

Si el usuario indicó “Celsius”, le devolverá la misma temperatura en grados Fahrenheit. Si el usuario indicó “Fahrenheit”, le devolverá la misma temperatura en grados Celsius. Imprimir la información completa para que el usuario comprenda el resultado.

5. Tenemos en un array el nombre de los alumnos del curso:

```
let alumnos = ["Marta", "Mariandrei", "Dário", "Christian", "Ariel", "Eduardo", "Alejandro", "Samuel", "Alex", "Antonio"]
```

Nuestro programa le preguntará al usuario por un número. Le imprimirá al usuario en la pantalla el nombre del alumno que se sienta en esa posición (NOTA: recordad que los arrays empiezan en 0!!)

6. Ahora vamos a hacer el array un poco más complejo. En lugar de guardar sencillamente el nombre de los alumnos, guardaremos en un array un objeto por cada alumno:

```
let alumnos = [{ 'nombre': "Adolfo", 'edad': 30 }, { 'nombre': "Sergio", 'edad': 31 }, { 'nombre': "Diego", 'edad': 44 }, { 'nombre': "Laura", 'edad': "Deconocida" }, { 'nombre': "Ernesto", 'edad': "36" }, { 'nombre': "Leonardo", 'edad': "Desconocida" }, { 'nombre': "Alex", 'edad': 30 },
```

```
{'nombre': "Samuel", 'edad': 31}, {'nombre': "Alejandro", 'edad': 44}, {'nombre': "Miguel", 'edad': "Desconocida"}]
```

Vamos a hacer lo mismo de antes (preguntarle al usuario por un número). Pero la diferencia, es que esta vez le imprimiremos al usuario en pantalla el nombre **y la edad** del alumno que se sienta en dicha posición.

7. A menudo, los objetos y los arrays con los que trabajaremos se van a volver muy complejos, en tanto que iremos anidando unos dentro de otros para formar estructuras de datos complejas. Un ejemplo de esto lo tenemos en el siguiente array:

```
let countries =  
[{"name": "Spain", "topLevelDomain": [".es"], "alpha2Code": "ES", "alpha3Code": "ESP", "callingCodes": [34], "capital": "Madrid", "altSpellings": ["ES", "Kingdom of Spain", "Reino de España"], "region": "Europe", "subregion": "Southern Europe", "population": 46438422, "latlng": [40.0, -4.0], "demonym": "Spanish", "area": 505992.0, "gini": 34.7, "timezones": ["UTC", "UTC+01:00"], "borders": ["AND", "FRA", "GIB", "PRT", "MAR"], "nativeName": "España", "numericCode": "724", "currencies": [{"code": "EUR", "name": "Euro", "symbol": "€"}], "languages": [{"iso639_1": "es", "iso639_2": "spa", "name": "Spanish", "nativeName": "Español"}], "translations": {"de": "Spanien", "es": "España", "fr": "Espagne", "ja": "スペイン", "it": "Spagna", "br": "Espanha", "pt": "Espanha", "nl": "Spanje", "hr": "Španjolska", "fa": "اسپانیا"}, "flag": "https://restcountries.eu/data/esp.svg", "regionalBlocs": [{"acronym": "EU", "name": "European Union"}, {"otherAcronyms": [], "otherNames": []}], "cioc": "ESP"}]
```

8. Crear un programa libre con Javascript utilizando condicionales (IF) y declaración de variables con diferentes tipos.

9. Comenzar los ejercicios de FreeCodeCamp del módulo “Basic JavaScript” del bloque de “JavaScript Algorithms and Data Structures”.