

Ejercicios prácticos

Introducción a objetos

Actividades

Todas las actividades de esta guía de ejercicios están hechas para ser resueltas en la **consola del navegador**.

Las actividades que tienen una **(D)** son consideradas desafiantes

1. Agregar al siguiente objeto una propiedad "peso" y asignarle un peso de 45
var Jabalí = {
 color: "marrón",
 altura: 1
}
2. Crear un objeto Rectángulo que tenga las propiedades "alto" con un valor de 10 y "ancho" con un valor de 20.
3. Crear un objeto película que tenga como propiedades "título", "duración", "género" y un arreglo de "actoresPrincipales". Los valores de cada propiedad son a elección
4. En el objeto Jabalí del ejercicio 1, acceder a su propiedad "color" y modificarla por "negro".
5. Modificar los valores del siguiente objeto para convertir a esta fruta en una banana
var Fruta = {
 color: "rojo",
 nombre: "tomate"
}
6. Crear un objeto que represente a una máquina de café. Deberá tener una marca, modelo y capacidad. Además, debe implementar un método prepararCafe que muestre una alerta avisando que está "preparando café".
7. Hacer un objeto heladería que contenga gustosDeHelado, un nombre, y el precioKilo. También debe implementar un método lanzarCampaña que muestre por consola el

precioKilo.

8. Agregar el método `escribirTexto(texto)` al siguiente objeto. Este método deberá mostrar en una alerta el texto pasado por parámetro e incrementar `alertasMostradas` en 1.

```
var MensajeroDeAlertas = {  
    tipoDeAlertas: ["Error", "Aviso", "Bienvenida"],  
    alertasMostradas: 0  
}
```

9. Crear el objeto `RobotCalculador`, que debe tener métodos para realizar operaciones de suma, resta, multiplicación y división. Además deberá tener una propiedad `"nombre"`

10. **(D)** Implementar al objeto `Catapulta` que debe tener un método que permita lanzarObjetos(unObjeto). Este método debe mostrar por consola `unObjeto.nombre + " lanzado"`. Notar que el objeto a lanzarse debe tener un propiedad `"nombre"`.

Por ejemplo, si tenemos al siguiente objeto

```
var Conejo = {  
    nombre: "Pepito"  
}
```

`Catapulta.lanzarObjeto(Conejo)` deberá mostrar por consola `"Pepito lanzado"`

11. **(D)** Al oso hormiguero le gustan las hormigas negras, pero no las rojas. Por eso, se debe implementar al objeto `OsoHormiguero` que tiene un método `degustar(hormiga)` que debe retornar **true** si le gustó la hormiga y **false** en caso contrario.

Disponemos de los siguientes objetos:

```
var HormigaNegra = {  
    tipo: "negra",  
    venenosa: false  
}
```

```
var HormigaRoja = {  
    tipo: "roja",  
    venenosa: true  
}
```

12. Implementar una función que reciba un arreglo de personas, lo recorra y muestre sus edades por la consola.
13. **(D)** Crear una función para encontrar en un arreglo de objetos a aquellos que cumplan con "tipo" y "especie" pasadas por parámetro.

Por ejemplo, dados los siguientes objetos:

```
var ballenaFranca = {  
  tipo: "mamifero",  
  especie: "ballena"  
}
```

```
var ballenaAzul = {  
  tipo: "mamifero",  
  especie: "ballena"  
}
```

```
var osoPardo = {  
  tipo: "mamifero",  
  especie: "oso"  
}
```

```
var animales = [ballenaFranca, ballenaAzul, osoPardo];
```

```
encontrar(animales, "mamifero", "ballena")
```

deberá devolver el arreglo [ballenaFranca, ballenaAzul]

14. **(D)** Crear una función que compare dos objetos Comida según sus ingredientes. Si los ingredientes son los mismos deberá devolver "true", caso contrario "false".

Por ejemplo:

```
var HamburguesaVeggie = {  
  ingredientes: ["soja", "cebolla"]  
}
```

```
var BolaDeSoja = {  
  ingredientes: ["soja", "cebolla"]  
}
```

`comparar(HamburguesaVeggie, BolaDeSoja)` deberá devolver `true`.

Extra: ¿cómo lo solucionarías si los ingredientes no están ordenados? Por ejemplo:
`["soja", "cebolla"]` y `["cebolla", "soja"]`

15. **(D)** Para una librería se necesita separar libros según género. Por eso se nos pide implementar una función que dado un género y una lista de objetos "libro" (que tienen como propiedad al género) se devuelva una lista con los libros pertenecientes a ese género.

Suponiendo que `libro1` y `libro3` tienen como género "Terror".

`filtrarPorGenero([libro1, libro2, libro3], "Terror")` deberá devolver `[libro1, libro3]`

Soluciones

1.

```
var Jabalí = {  
    color: "marrón",  
    altura: 1,  
    peso: 45  
}
```
2. Crear un objeto Rectángulo que tenga las propiedades "alto" con un valor de 10 y "ancho" con un valor de 20.

```
var Rectangulo = {  
    alto: 10,  
    ancho: 20  
}
```
3.

```
var Pelicula = {  
    titulo: "El séptimo sello",  
    duración: 97,  
    genero: "drama",  
    actoresPrincipales: ["Max von Sydow", "Bibi Andersson", "Gunnar Björnstrand"]  
}
```
4.

```
Jabali.color = "negro";
```
5.

```
Fruta.color = "amarillo";  
Fruta.nombre = "banana";
```
6.

```
var MaquinaDeCafe = {  
    marca: "CafeMAX",  
    modelo: "125M",  
    capacidad: 70,  
    prepararCafe: function(){ alert("Preparando café") }  
}
```
7.

```
var Heladeria = {  
    gustosDeHelado: ["Vainilla", "Frutilla", "Chocolate"],
```

```

    nombre: "Heladería Los Hermanos Chantilly",
    precioKilo: 300,
    lanzarCampaña: function(){
        console.log(Heladeria.precioKilo);
    }
}

```

8. var MensajeroDeAlertas = {
 tipoDeAlertas: ["Error", "Aviso", "Bienvenida"],
 alertasMostradas: 0,
 escribirTexto: function(texto) {
 alert(texto);
 MensajeroDeAlertas++;
 }
}

9. var RobotCalculador = {
 nombre: "Robotico",
 sumar: function(nro1, nro2) {
 return nro1 + nro2;
 },
 restar: function(nro1, nro2) {
 return nro1 - nro2;
 },
 multiplicar: function(nro1, nro2) {
 return nro1 * nro2;
 },
 dividir: function(nro1, nro2) {
 return nro1 / nro2;
 }
}

10. var Catapulta = {
 lanzarObjeto: function(unObjeto){
 console.log(unObjeto.nombre + " lanzado");
 }
}

11. Al oso hormiguero le gustan las hormigas negras, pero no las rojas. Por eso, se debe implementar al objeto OsoHormiguero que tiene un método degustar(hormiga) que debe

retornar **true** si le gustó la hormiga y **false** en caso contrario.
Disponemos de los siguientes objetos:

```
var OsoHormiguero = {  
  degustar: function(hormiga){  
    if(hormiga.tipo === "roja") {  
      return false;  
    } else {  
      return true;  
    }  
  }  
}
```

```
12. var edades = function(personas) {  
  for(var i = 0; i < personas.length; i++){  
    console.log(personas[i].edad);  
  }  
}
```

```
13. var encontrar = function(animales, tipo, especie){  
  var resultado = [];  
  var animalActual;  
  for(var i = 0; i < animales.length; i++){  
    animalActual = animales[i];  
    if (animalActual.tipo === tipo && animalActual.especie === especie){  
      resultado.push(animalActual);  
    }  
  }  
  return resultado;  
}
```

```
14. Esta primer solución sólo sirve si los ingredientes están ordenados.  
var comparar = function(comida1, comida2){  
  if(comida1.ingredientes.length !== comida2.ingredientes.length) return false;  
  
  for(var i = 0; i < comida1.ingredientes.length; i++){  
    if (comida1.ingredientes[i] !== comida2.ingredientes[i]) return false;  
  }  
  return true;  
}
```


Esta segunda versión sirve también si están desordenados:

```
var comparar = function(comida1, comida2){  
    if(comida1.ingredientes.length !== comida2.ingredientes.length) return false;  
  
    for(var i = 0; i < comida1.ingredientes.length; i++){  
        if (comida2.ingredientes.indexOf(comida1.ingredientes[i]) === -1 ) return  
false;  
    }  
    return true;  
}
```

```
15. var filtrarPorGenero = function(libros, genero) {  
    var resultado = [];  
    for(var i = 0; i < libros.length; i++){  
        if( libros[i].genero === genero){  
            resultado.push(libros[i]);  
        }  
    }  
    return resultado;  
}
```