Práctica 11

DOCENTE	CARRERA	CURSO
MSc. Vicente Enrique Machaca Arceda	Escuela Profesional de Ingeniería de Software	Fundamentos de Lenguajes de Programación

PRÁCTICA	TEMA	DURACIÓN
11	Programación Funcional	2 horas

1. Datos de los estudiantes

- Grupo: 03
- Integrantes:
  - Jonathan Aguirre Soto

2. Ejercicios

Implemente funciones en Javascript que:

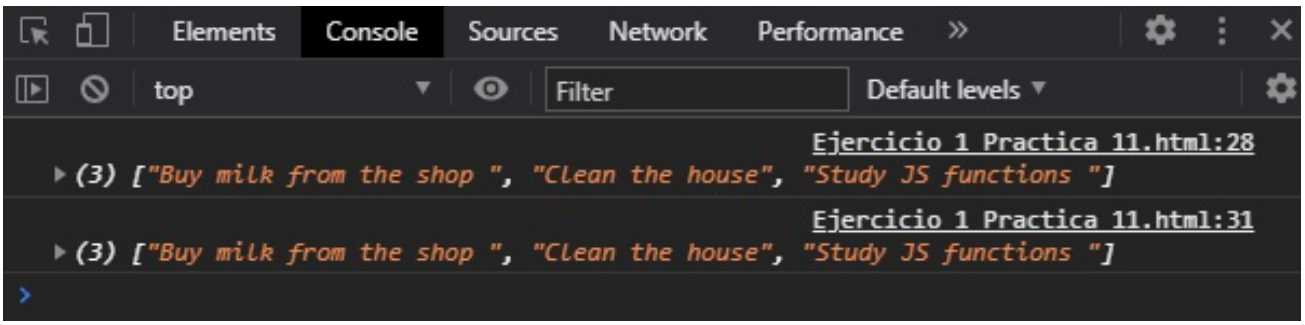
1. Muestre solo el nombre (name) del conjunto de datos mostrado linea abajo. Implemente una versión iterativa y una funcional (puede usar map, filter o reduce). **(3 puntos)**

```
let tasks = [
{
 name      :  Buy  milk from the shop  ,
 duration      : 20 ,
 priority      : 1
},
{
 name      :  Clean  the house   ,
 duration      : 120 ,
 priority      : 3
},
{
 name      :  Study  JS functions   ,
 duration      : 180 ,
 priority      : 1
}
];
```

```
<html>
<body>
<script>
 let tasks = [
```

```
    {'name' : 'Buy milk from the shop ',
     'duration' : 20,
     'priority' : 1},
    {'name' : 'Clean the house',
     'duration' : 120 ,
     'priority' : 3},
    {'name' : 'Study JS functions ',
     'duration' : 180 ,
     'priority' : 1}
];

function iterative (h){

   let name = [];

   for (var i = 0;i < h.length; i++) {

      name.push(h[i].name);
   }

   return name;
}

console.log(iterative(tasks));

const y = tasks.map(x => x.name);
console.log(y);

</script>
</body>
</html>
```

- Ejecución del programa.



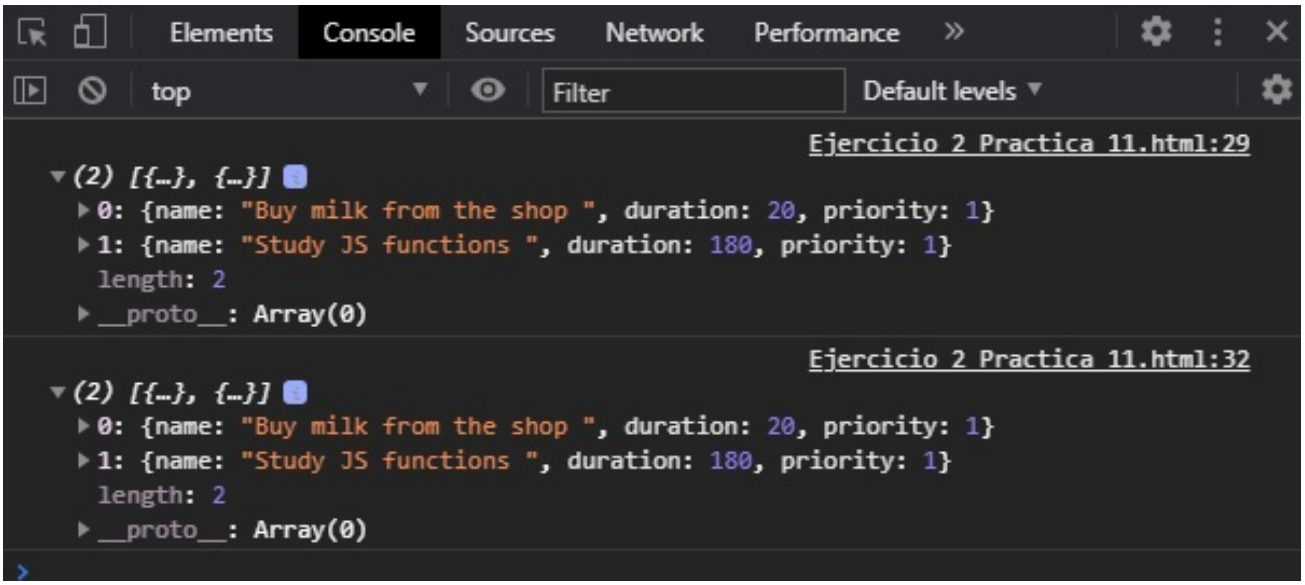
2. Con los datos del ejercicio anterior, ahora muestre las tareas con prioridad (priority) 1. Implemente una versión iterativa y una funcional (puede usar map, filter o reduce).**(3 puntos)**

```
let tasks = [
  {
   name      :  Buy  milk from the shop  ,
   duration      : 20 ,
   priority      : 1
  },
  {
   name      :  Clean  the house   ,
   duration      : 120 ,
   priority      : 3
  },
  {
   name      :  Study  JS functions   ,
   duration      : 180 ,
   priority      : 1
  }
  ];
```

```
<html>
<body>
<script>
 let tasks = [
    {'name' : 'Buy milk from the shop ',
     'duration' : 20,
     'priority' : 1},
    {'name' : 'Clean the house',
     'duration' : 120 ,
     'priority' : 3},
    {'name' : 'Study JS functions ',
     'duration' : 180 ,
     'priority' : 1}
 ];

 function iterative (h){

    let priority=[];

    for (var i = 0; i < h.length; i++) {

       if(h[i].priority == 1)
                           priority.push (h[i]);
    }

    return priority;
 }

 console.log(iterative(tasks));

 const y = tasks.filter(x => x.priority === 1);
 console.log(y);

</script>
```

```
</body>
</html>
```

- Ejecución del programa.



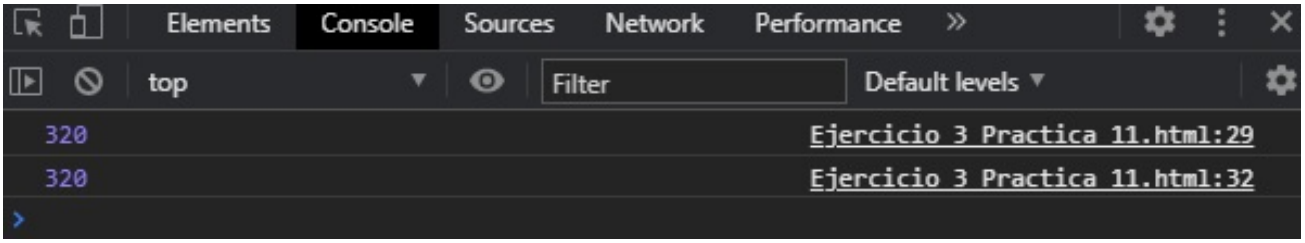
3. Con los datos del ejercicio anterior, muestre la cantidad total de tiempo que tomarán todas las tareas. Implemente una versión iterativa y una funcional (puede usar map, filter o reduce). **(3 puntos)**

```
let tasks = [
  {
   name      :  Buy  milk from the shop  ,
   duration      : 20 ,
   priority      : 1
  },
  {
   name      :  Clean  the house   ,
   duration      : 120 ,
   priority      : 3
  },
  {
   name      :  Study  JS functions   ,
   duration      : 180 ,
   priority      : 1
  }
  ];
```

```
<html>
<body>
<script>

 let tasks = [
    {'name' : 'Buy milk from the shop ',
     'duration' : 20,
     'priority' : 1},
    {'name' : 'Clean the house',
     'duration' : 120 ,
     'priority' : 3},
    {'name' : 'Study JS functions ',
     'duration' : 180 ,
     'priority' : 1}
 ];

 function iterative (h){

    let tiempo = 0;

    for (var i = 0;i < h.length; i++) {

       tiempo = tiempo + h[i].duration;
    }

    return tiempo;
 }

 console.log(iterative(tasks));

 const y = tasks.map(x => x.duration).reduce((total, tiempo) => total + tiempo);
 console.log(y);
```

```
</script>
</body>
</html>
```

- Ejecución del programa.



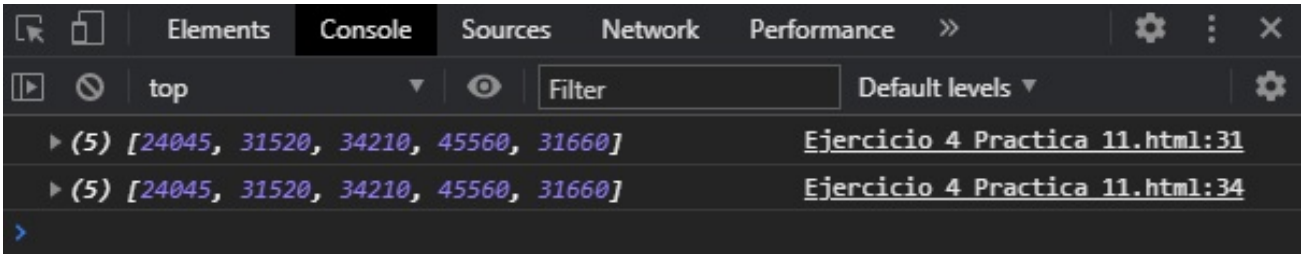
4. Muestre el precio (price) de los vehiculos de tipo "suv". Implemente una versión iterativa y una funcional (puede usar map, filter o reduce). **(3 puntos)**

```
const vehicles = [
  { make : Honda    , model : CR - V , type : suv    , price : 24045 },
  { make : Honda    , model : Accord    , type : sedan    , price : 22455 },
  { make : Mazda    , model : Mazda 6   , type : sedan    , price : 24195 } ,
  { make : Mazda    , model : CX -9    , type : suv    , price : 31520 },
  { make : Toyota    , model : 4 Runner    , type : suv    , price : 34210 },
  { make : Toyota    , model : Sequoia    , type : suv    , price : 45560 },
  { make : Toyota    , model : Tacoma    , type : truck    , price : 24320 },
  { make : Ford    , model : F -150    , type : truck    , price : 27110 },
  { make : Ford    , model : Fusion    , type : sedan    , price : 22120 },
  { make : Ford    , model : Explorer    , type : suv    , price : 31660 }
  ];
```

```
<html>
<body>
<script>

 const vehicles = [
 { make : 'Honda ', model : 'CR -V', type : 'suv', price :24045 },
 { make : 'Honda ', model : 'Accord ', type : 'sedan ', price :22455 },
 { make : 'Mazda ', model : 'Mazda 6', type : 'sedan ', price :24195 },
 { make : 'Mazda ', model : 'CX -9 ', type : 'suv', price :31520 },
 { make : 'Toyota ', model : '4 Runner ', type : 'suv', price :34210 },
 { make : 'Toyota ', model : 'Sequoia ', type : 'suv', price :45560 },
 { make : 'Toyota ', model : 'Tacoma ', type : 'truck ', price :24320 },
 { make : 'Ford ', model : 'F -150 ', type : 'truck ', price :27110 },
 { make : 'Ford ', model : 'Fusion ', type : 'sedan ', price :22120 },
 { make : 'Ford ', model : 'Explorer ', type : 'suv', price :31660 }
 ];

 function iterative (h){

    let price = [];

    for (var i = 0;i < h.length; i++){
```

```
        if(h[i].type == 'suv')
                            price.push(h[i].price);
    }

    return price;
  }

  console.log(iterative(vehicles));

  const t = vehicles.filter(x => (x.type == 'suv')).map(y => y.price);
  console.log(t);

 </script>
 </body>
 </html>
```

- Ejecución del programa.



5. Muestre el score total (pilotingScore + shootingScore) de los usuarios Force (isForceUser: true). Implemente una versión iterativa y una funcional (puede usar map, filter o reduce). **(4 puntos)**

```
var personnel = [
      {
      id: 5 ,
      name : " Luke Skywalker ",
      pilotingScore : 98 ,
      shootingScore : 56 ,
      isForceUser : true ,
      },
      {
      id: 82 ,
      name : " Sabine Wren ",
      pilotingScore : 73 ,
      shootingScore : 99 ,
      isForceUser : false ,
      },
      {
      id: 22 ,
      name : "Zeb Orellios ",
      pilotingScore : 20 ,
      shootingScore : 59 ,
      isForceUser : false ,
      },
      {
      id: 15 ,
      name : " Ezra Bridger ",
      pilotingScore : 43 ,
      shootingScore : 67 ,
      isForceUser : true ,
      },
      {
      id: 11 ,
      name : " Caleb Dume ",
      pilotingScore : 71 ,
      shootingScore : 85 ,
      isForceUser : true ,
      }
    ];
```

6. Enlace de Github:
`https://github.com/Jona2010/Fundamentos-de-Lenguaje-de-la-Programaci-n`