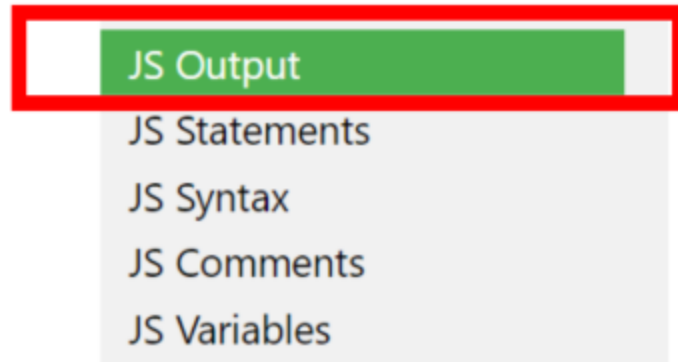


Trabajando la [visualización de datos](#): (*seguimos en el mismo apartado del tutorial que en la tarea anterior*)



JSTarea3.html - Sobre una copia de la tarea anterior realiza los cambios oportunos para mostrar los datos empleando diferentes formas de visualización de los mismos:

- usando la propiedad **innerHTML** de un elemento HTML (la forma más común de mostrar datos; ver más abajo la aclaración *innerHTML* vs *textContent*)
- utilizando **document.write ()** (no es la forma en que el DOM funciona conceptualmente, y es una manera fácil de crear errores; normalmente evitaremos su uso en favor de las funciones de manipulación de DOM;)
- usando un “cuadro de alerta” con **window.alert ()**
- recuerda el [uso de la consola](#): escribe la salida que se pide también **en la consola del navegador** usando **console.log()**

truco: si en lugar de poner console.log(foo), pones console.log({foo}), además del valor, verás el nombre de la variable

```
> foo = 42
< 42
> console.log(foo);
42
< undefined
> console.log({foo});
▶ {foo: 42}
< undefined
> |
```

innerHTML vs textContent

Aunque comúnmente se usan de manera intercambiable, hay una gran diferencia entre las propiedades `textContent` e `innerHTML` cuando se trabaja con Elementos en JavaScript, y es importante que la

conozcas. La principal diferencia es que **cuando se utiliza la propiedad innerHTML, el navegador del cliente interpretará el contenido como un marcado HTML y aplicará ese cambio** a la ventana DOM.

A continuación se muestra código que usa la propiedad innerHTML y su comportamiento:

```

<body>
  <!-- Both used to add content to a element/tag -->
  <h1>textContent vs innerHTML - JavaScript DOM - dcode</h1>
  <div id="my-div"></div>

  <script>
    const myDiv = document.getElementById('my-div');

    myDiv.innerHTML = 'Hey my name is <strong>Dom</strong>';
  </script>
</body>

```

textContent vs innerHTML - JavaScript DOM - dcode

Hey my name is Dom

Por otro lado, la **propiedad textContent** solo cambiará el texto, conservando cualquier marca HTML

```

<body>
  <!-- Both used to add content to a element/tag -->
  <h1>textContent vs innerHTML - JavaScript DOM - dcode</h1>
  <div id="my-div"></div>

  <script>
    const myDiv = document.getElementById('my-div');

    myDiv.textContent = 'Hey my name is <strong>Dom</strong>';
  </script>
</body>

```

textContent vs innerHTML - JavaScript DOM - dcode

Hey my name is **Dom**