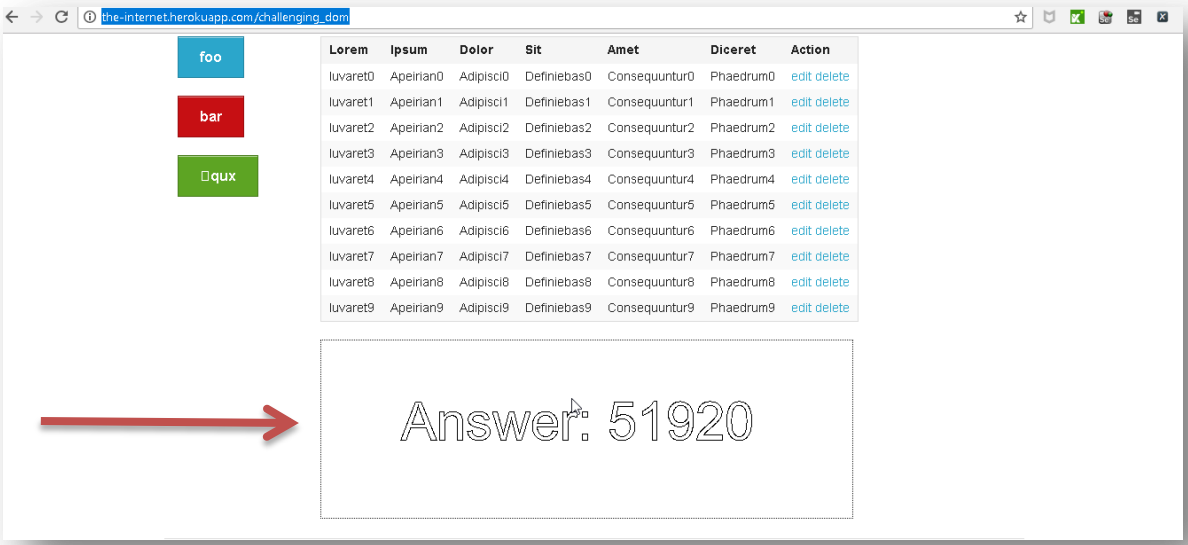
¿Cómo obtener texto desde una imagen en CANVAS?

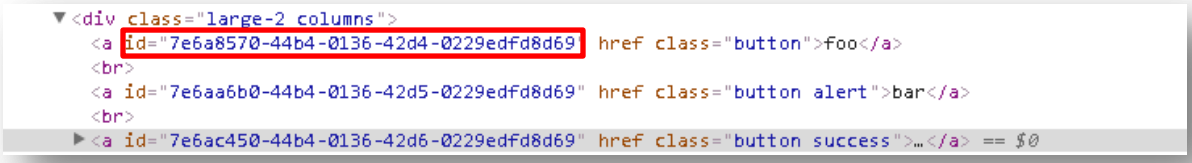


En la página http://the-internet.herokuapp.com encontraremos en la opción Challenging DOM, una página, que al presionar uno de los tres botones de colores de la izquierda, obtendremos en la "caja" de debajo, un texto, en el ejemplo de la imagen anterior es, "Answer 51290". Cada vez que se presiona uno de los botones, aparecerá un nuevo texto de manera aleatoria. En esta guía aprenderemos a cómo obtener el texto que aparecerá en el CANVAS, tomándola directamente desde el String del script que lo ejecuta.

¡VEAMOS UNA DE LAS POSIBLES SOLUCIONES!

Para empezar analizaremos alguno de los botones que ejecuta el cambio de texto en el canvas.

Al inspeccionar algunos de estos botones, nos encontraremos una no muy grata sorpresa. Encontraremos que tienen un id que cambia cada vez que interactuamos con dicho botón.



Sin embargo, nos permitirá identiicarlo por su atributo **class.**

Para tener en cuenta: En este objeto podemos ver que aparentemente hay dos atributos, **id** y "**href class**". Sin embargo "**href class**" son dos atributos, escritos de esta forma debido a que comparten el mismo nombre, es decir **href** = "**button**" y **class** = "**button**".

Una vez identificado el objeto, de la siguiente forma:

```
@FindBy(className="button")
      public WebElementFacade btnBoton;
```

Procederemos a obtener el texto del canvas. Al analizar el HTML encontraremos que el canvas es dibujado gracias a un JavaScript que dibuja el texto descrito dentro del script. En esta solución obtendremos el script, y una vez lo tengamos, tomaremos el String que nos interesa capturar.

```
public void obtenerCanva()
{
    List<WebElement> elJS= getDriver().findElements(By.xpath("//script[text()]"));
    String elTextoEnCanvas = elJS.get(1).getAttribute("innerHTML");
    elTextoEnCanvas=elTextoEnCanvas.substring(140, 153).replace("'", "");
    System.out.println(elTextoEnCanvas);

}
```

Con el método de la imagen anterior solucionaremos el ejercicio en mención.

1. El método nos traerá una lista donde están contenidos los scripts que existen dentro del HTML.

```
List<WebElement> elJS= getDriver().findElements(By.xpath("//script[text()]"));
```

2. De la lista de scripts, tomaremos el script que nos interesa, en este caso es el segundo ítem de la lista, por lo tanto le enviamos como parámetro el “1”, y a ese elemento le obtendremos un atributo llamado “innerHTML”, de tal manera que nos traiga un String que contiene todo el texto que realiza toda “la magia”.

```
String elTextoEnCanvas = elJS.get(1).getAttribute("innerHTML");
```

Al ejecutarse esta instrucción, nuestro String “elTextoEnCanvas” quedará con la siguiente información:

“**var** canvas_el = document.getElementById('canvas');

var canvas = canvas_el.getContext('2d');

canvas.font = '60px Arial';

canvas.strokeText('Answer: 51920',90,112);”

Señalado en el cuadro rojo, podemos ver que “viaja” la información que nos interesa imprimir por pantalla.

3. Al identificar el texto que nos interesa, usaremos un editor de texto como *notepad++*¸ y localizaremos la posición inicial y final para realizar un Substring, es decir, cortar el String y tomar solo lo que necesitamos.

```
elTextoEnCanvas=elTextoEnCanvas.substring(140, 153).replace("'", "");
```

Y hacemos un **replace** del carácter comilla sencilla (‘) por cadena vacía, de tal forma que solo nos quede un texto limpio.

4. Por último, imprimimos por pantalla el resultado de nuestro String final.

```
System.out.println(elTextoEnCanvas);
```

¡AHORA INTÉNTALO TÚ!