

Código HTML proporcionado

```
<!DOCTYPE html>
<html>
<head>
  <title>C85</title>
  <link rel="stylesheet" href="style.css">
  <script src="fabric.js"></script>
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>
</head>
<body class="body_backgorund">
  <center>
    <h2>MINECRAFT</h2>
    <div id="head_info">
      <h3>Ancho actual = </h3>
      <h3>Altura actual = </h3>
    </div>
  </center>
  <script src="main.js"></script>
</body>
</html>
```

Enlace del archivo CSS

Enlace del archivo fabric.js

Enlaces bootstrap

Enlace del archivo main.js

El código anterior contiene:

- Enlace para la hoja de estilo
- Enlace para la biblioteca fabric
- Enlace para las clases bootstrap
- Etiqueta body
- Etiqueta center
- Algunas etiquetas de encabezado y etiquetas div
- Enlace para el archivo main.js

Añadir una clase a la etiqueta body -

```
<body class="body_background">
```

Primero, debes añadir una clase a la etiqueta body. Añade la clase `body_background` a la etiqueta body, la cual añade un fondo a la página HTML. El código CSS ya está definido para esta clase en el archivo `style.css`. Solo debes añadir la clase

Añadir las clases bootstrap para la etiqueta -

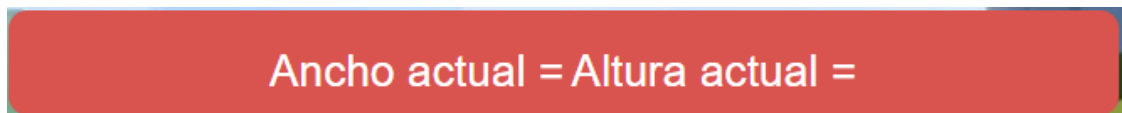
```
<h2 class="btn-primary">MINECRAFT</h2>
```

- **btn-primary** añadirá un color azul para el fondo y un color de letra blanco.
- Output :



Añadir las clases bootstrap para las etiquetas h3 -

```
<div id="head_info" class="btn-danger">
```



Añadir dos etiquetas span dentro de las etiquetas h3 -

```
<h3>Ancho actual = <span id="current_width">30</span></h3>  
<h3>Altura actual = <span id="current_height">30</span></h3>
```

Añade dos etiquetas span dentro de las etiquetas h3 y proporciona un id. Estas etiquetas se utilizan para mostrar el ancho y alto de los bloques (como la pared, el tronco y otros) en la pantalla conforme incrementa o disminuye el tamaño. Establece el texto a 30 para las dos etiquetas, ya que queremos que el tamaño inicial de los bloques sea de 30px.

Ancho actual = 30 Altura actual = 30

Después de añadir todos los elementos HTML -

```

<!DOCTYPE html>
<html>
<head>
  <title>C85</title>
  <link rel="stylesheet" href="style.css">

  <script src="fabric.js"></script>

  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/css/bootstrap.min.css">

  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.0/js/bootstrap.min.js"></script>

</head>

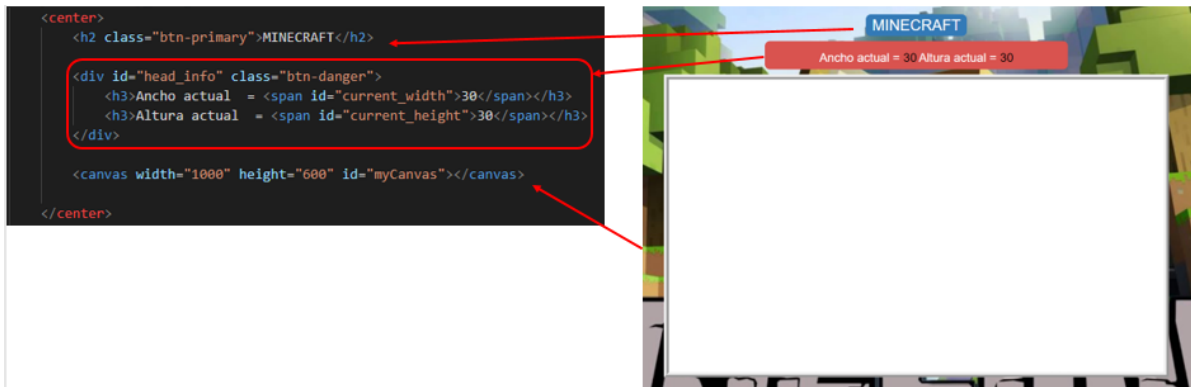
<body class="body_backgorund">
  <center>
    <h2 class="btn-primary">MINECRAFT</h2>

    <div id="head_info" class="btn-danger">
      <h3>Ancho actual = <span id="current_width">30</span></h3>
      <h3>Altura actual = <span id="current_height">30</span></h3>
    </div>

    <canvas width="1000" height="600" id="myCanvas"></canvas>
  </center>
  <script src="main.js"></script>
</body>
</html>

```

Output -



Código main.js -

Primero hay que obtener la referencia para el lienzo y almacenarla en una variable

```
var canvas = new fabric.Canvas('myCanvas');
```

Definir el ancho y el alto de la imagen del bloque

```
block_image_width = 30;
block_image_height = 30;
```

Definir las coordenadas X y Y para la imagen del jugador

```
player_x = 10;  
player_y = 10;
```

Definir una variable para almacenar el objeto de la imagen del jugador

En fabric.js, se almacenan las imágenes en el lienzo como objetos, así que conforme añadimos objetos, también podemos eliminarlos. Esto solo significa que podemos añadir y eliminar imágenes si las utilizamos como objetos.

```
var player_object= "";
```

Función para añadir la imagen del jugador

```
function player_update()  
{  
    fabric.Image.fromURL("player.png", function(Img) {  
        player_object = Img;  
  
        player_object.scaleToWidth(150);  
        player_object.scaleToHeight(140);  
        player_object.set({  
            top:player_y,  
            left:player_x  
        });  
        canvas.add(player_object);  
  
    });  
}
```

Explicación para añadir la imagen con fabric

```
fabric.Image.fromURL("player.png", function(Img) {
```

- **fabric**: Este es el nombre de la biblioteca que utilizamos.
- **Image**: Esto indica que subimos una imagen.
- **fromURL**: Este contiene el enlace de la imagen y la función para subir la imagen.
- **"player.png"**: Esta es la imagen.
- **function(Img)**: Esta es la función que utilizamos para subir la imagen al lienzo.
 - **Img**: Este es el objeto de la imagen que se define por defecto.

Función para añadir las imágenes de los bloques

```
function new_image(get_image)
{
    fabric.Image.fromURL(get_image, function(Img) {
        block_image_object = Img;

        block_image_object.scaleToWidth(block_image_width);
        block_image_object.scaleToHeight(block_image_height);
        block_image_object.set({
            top:player_y,
            left:player_x
        });
        canvas.add(block_image_object);

    });
}
```