# Resumen de Word: Desestructuración de Arreglos en JavaScript

Es momento de trabajar con la desestructuración de arreglos o la asignación desestructurante de arreglos también.  
Lo mismo que vimos en el video pasado, pero con arreglos ahora, creemos un arreglo sencillo, que se llama personajes,  
que va a ser igual a arreglo. Bueno, vamos a definir un par de valores como Goku, Vegeta y Trunks, y tenemos esos tres personajes.  
  
Supongamos que queremos imprimir los resultados en consola como lo haríamos tradicionalmente. Bueno, tenemos los personajes.  
Si necesitamos el primero, sabemos que en Javascript los arreglos empiezan en la posición cero, entonces accedemos con  
`personajes[0]`. Si queremos también a Vegeta y a Trunks, podemos acceder `personajes[1]` y `personajes[2]`, respectivamente.  
  
Sin embargo, estar llamando `personajes[0]`, `personajes[1]`, etc., es tedioso y podría simplificarse. Aquí es donde la  
desestructuración de arreglos resulta útil, ya que permite extraer elementos de forma más directa.  
  
Para extraer valores de un arreglo:  
Creamos una constante `const [p1, p2, p3] = personajes;` para acceder a Goku, Vegeta y Trunks respectivamente.  
Ejemplo: `console.log(p1); // Goku`  
  
Ignorar valores no deseados:  
Si queremos solo el segundo y tercer valor, usamos una coma `const [, p2, p3] = personajes;`, ignorando el primero.  
  
También podemos usar desestructuración en funciones:  
Si tenemos `function retornaArreglo() { return ["ABC", 123]; }`, usamos desestructuración `const [letras, numeros] = retornaArreglo();`.  
  
Usar funciones con set:  
Creamos `useState`, que retorna un arreglo `[valor, setValor]`. Luego desestructuramos `[nombre, setNombre] = useState("Goku");`.  
Este método es fundamental en frameworks como React para gestionar el estado.