**Algoritmos y Estructuras de Datos**

Son colecciones de valores, las relaciones entre ellos y las funciones u operaciones que se pueden aplicar a los datos.

Nos ayudan a:

1. Solucionar problemas.
2. Guardar datos.
3. Estructuras datos.

Hablemos de un ejemplo con ropa:

Imaginemos que nos dan muchísima ropa de repente, la guardamos de golpe en el placard y luego viene una persona a la pieza y le decimos “alcánzame la **remera** **negra** que esta en el placard a la derecha”, en este momento la persona nos dice “no la encuentro” pero nosotros vamos y decimos “acá estaba”.

Esto significa que nuestra forma de ordenar la ropa solo sirvió para nosotros porque sabemos donde estaba, pero para otras personas es difícil de entender, o directamente no lo entiende.

Entonces, otra forma de ordenar nuestra ropa podría ser:

* Por tipo de prenda (remera, campera, pantalón).
* Por color.

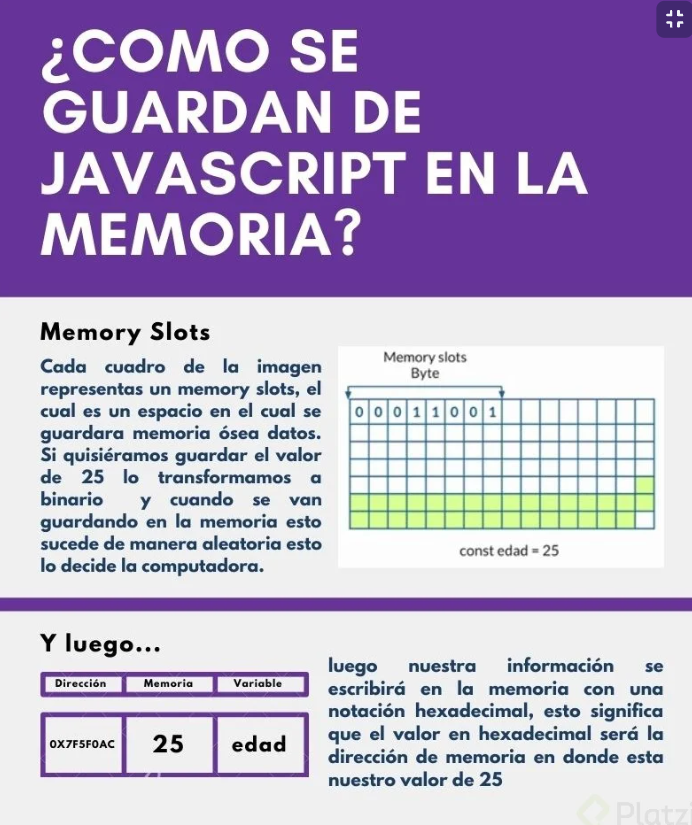
Si nosotros tuviésemos un cajon solo para prendas negras, y da la casualidad que solo tenemos la remera negra, la persona encontraba instantemente lo que le pedimos.

Los algoritmos y estructuras de datos sirven para exactamente lo mismo, nos ayudan a poder guardar, ordenar de forma eficiente y entendible.

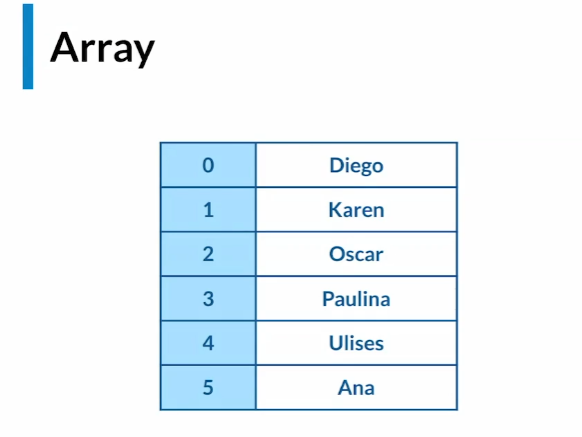
**Cabe aclarar que NO todas las estructuras de datos son buenas para todos los casos, sino que algunas nos sirven más y otras menos.**

Resumen: Nos permiten optimizar el espacio, el orden en que almacenamos la información y nos ayudan en problemas específicos. En algunos lenguajes algunas ya vienen definidas.

Memoria y como se guardan los datos:



Arrays: Los arrays son una colección de información que se puede ver de la siguiente forma:

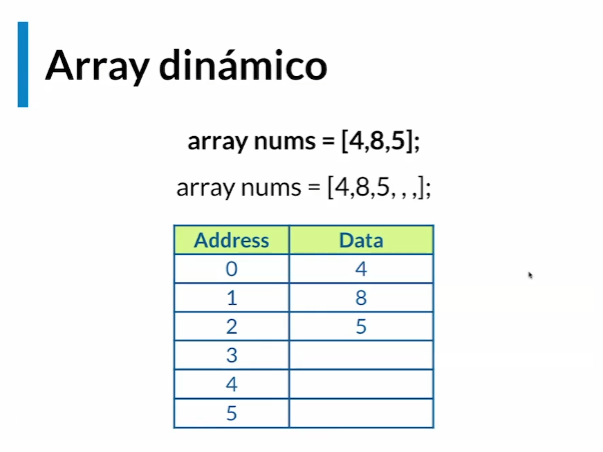


A los arrays, en otros lenguajes de programación se los conoce como listas (Python).

Para acceder a cada uno de los elementos se utilizan los índices que comienzan con el 0.

Hay dos tipos de Array:

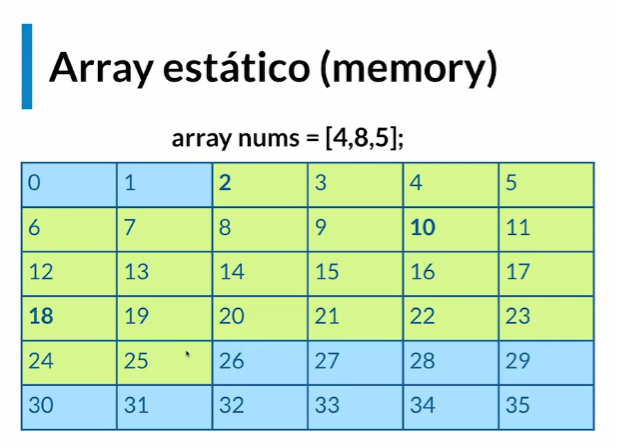
* **Dinámicos**: Son aquellos que se utilizan en el día de hoy, a estos se les puede colocar n cantidad de valores, por lo tanto, **el espacio que ocupe en memoria siempre será mutable**. Los dinámicos, cuantos más valores les agregamos, más estará buscando espacio para almacenar esa información y nuestro programa se tornará pesado.



¿Qué es lo que pasa con los arrays dinámicos? El **array original SE ELIMINA**, **se crea un CLON** **con los datos del ORIGINAL + los datos nuevos**.

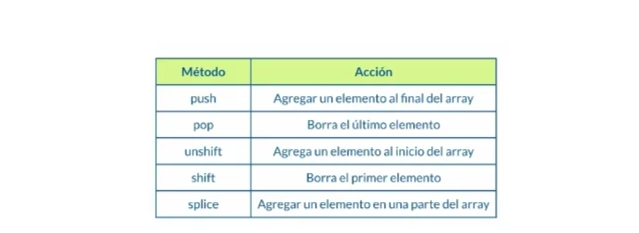
La computadora dice, bueno, inicialmente me manda que tiene 3 valores (4, 8 y 5) pero POR LAS DUDAS le guardo 3 slots más. ¿Qué pasa si le mandamos en vez de 6 valores, 7? La computadora sabrá que excedió la cantidad de espacios que le había calculado y CLONARÁ el viejo array agregándole el valor extra que sería el 6 (array original clonado) + 1 (item nuevo).

* **Estáticos**: Son aquellos que se utilizaban en lenguajes como C, C++, los cuales al momento de definirlos decimos que van a tener solamente “n” valores y luego no se podrán extender, es decir, **su espacio de memoria es inmutable**. Esto hacía que su uso sea eficiente, pues, le decimos a la computadora que x variable no ocupará más de ese espacio y no tendrá que estar buscando nuevos espacios en memoria para guardar un nuevo valor.



Los espacios que ocupa el array son los que están coloreados con verde. En este momento, ya podemos ver que ocupará MÁXIMO ese espacio.

En JavaScript siguientes métodos nos permiten manipular los arrays:



Creando nuestro propio Array (con clases desde 0):