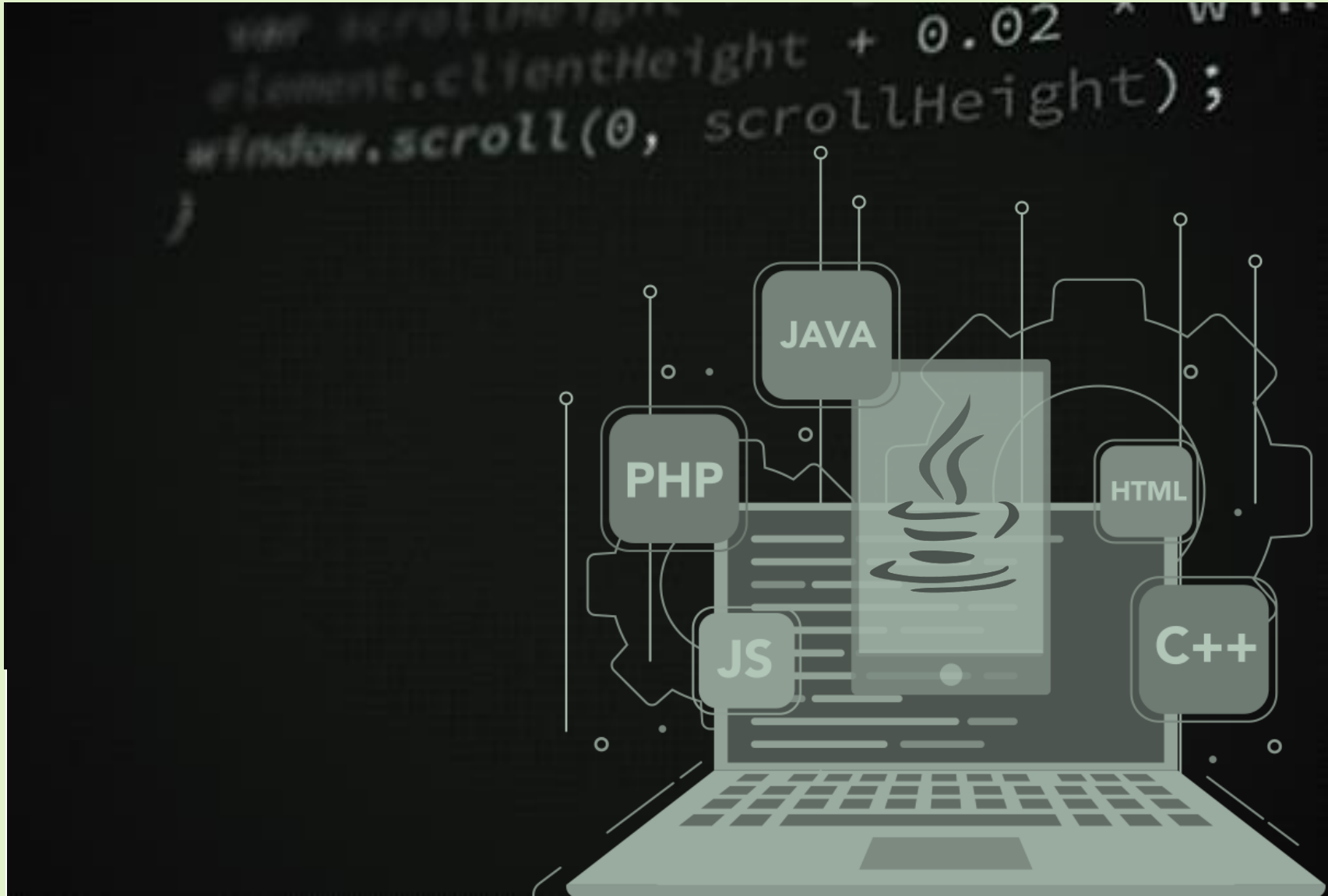
EFA
MORATALAZ

1º CFGS Desarrollo de Aplicaciones Web

PROGRAMACIÓN

DANIEL GONZÁLEZ-CALERO JIMÉNEZ

UT3 – ESTRUCTURAS DE ALMACENAMIENTO ESTÁTICAS



EFA
MORATALAZ

1º CFGS Desarrollo de Aplicaciones Web

PROGRAMACIÓN

INDICE

UT3 – ESTRUCTURAS DE ALMACENAMIENTO ESTÁTICAS

INTRODUCCIÓN

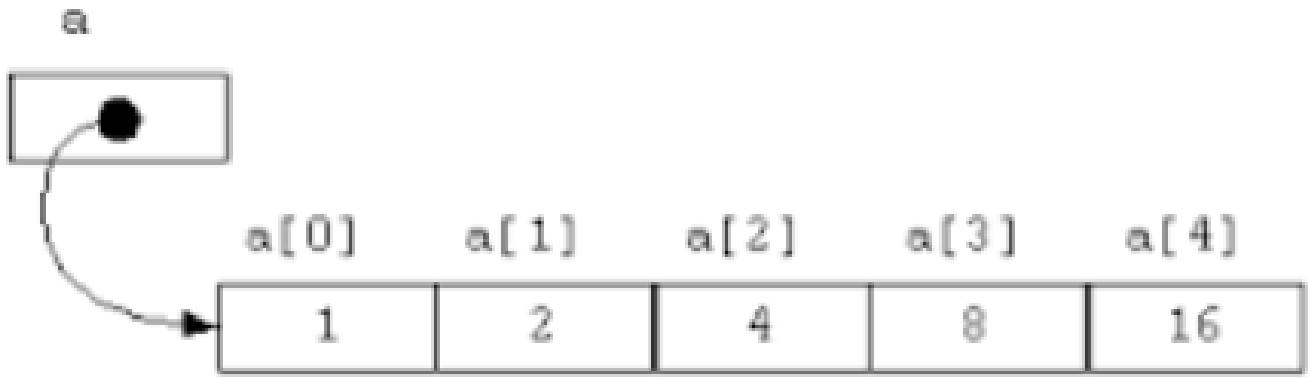
Las estructuras de almacenamiento de datos estáticas se utilizan para **almacenar un conjunto de datos**, lo que nos permite no tener que crear una variable para cada dato.

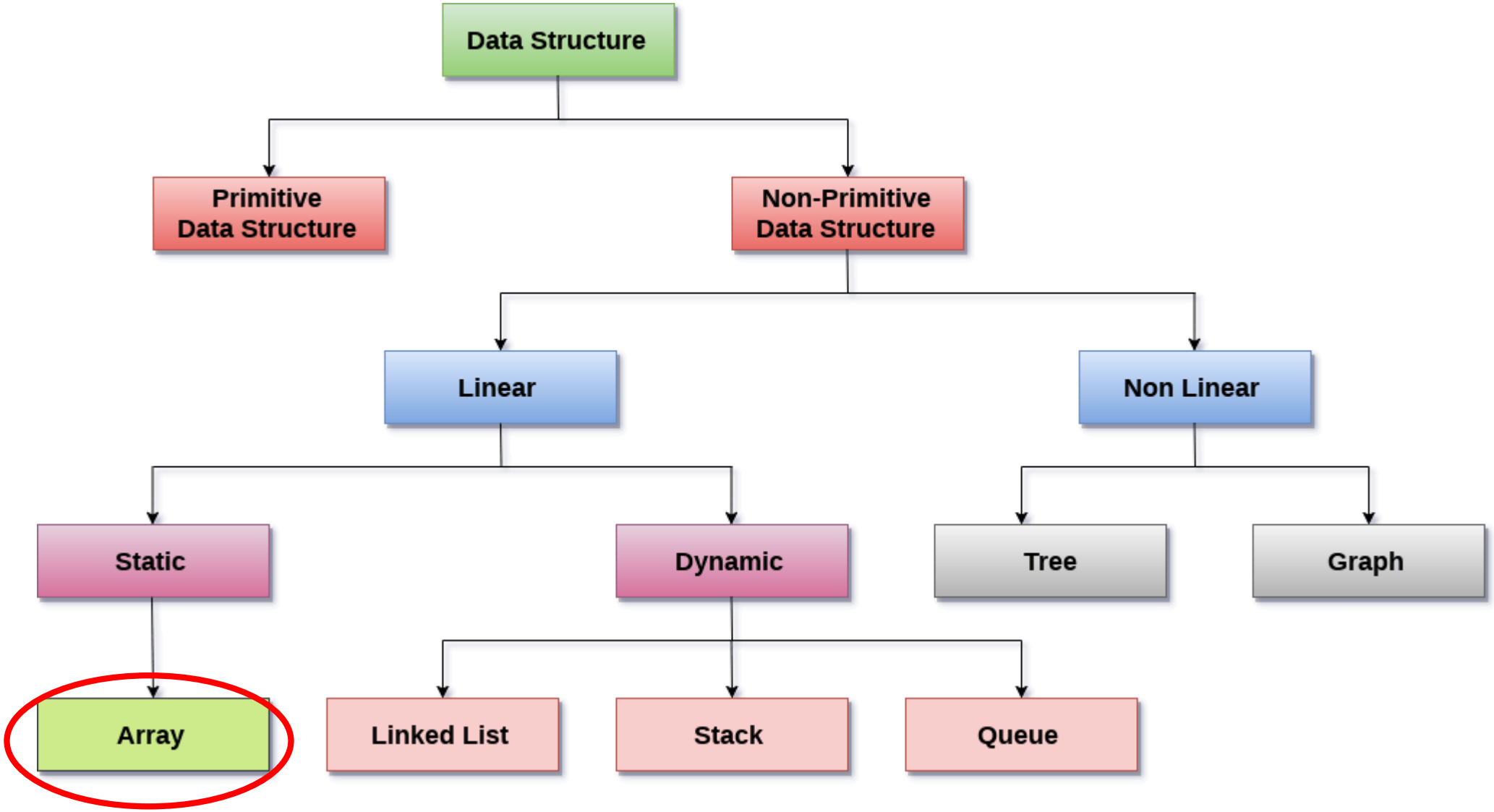
Al igual que las variables, pueden ser de distintos tipos:

- Int
- String
- Double
- Char
- Objetos

….

Para cada estructura de almacenamiento, es posible acceder individualmente a cada uno de los valores que contiene.




Data Structure
Primitive Data Structure
Non-Primitive Data Structure
Linear
Non Linear
Static
Dynamic
Tree
Graph
Array
Linked List
Stack
Queue

ESTRUCTURAS DE DATOS ESTÁTICAS VS DINÁMICAS

Las estructuras de datos estáticas son aquellas en las que el tamaño ocupado en memoria se define antes de que el programa se ejecute y n**o puede modificarse dicho tamaño durante la ejecución del programa.**

Su principal característica es que ocupan solo una casilla de memoria, al igual que una variable.

Una estructura de datos estática puede ser, por ejemplo, los nombres de los equipos de la primera división de fútbol, donde vamos a tener siempre 20 equipos (tamaño 20), siendo imposible incrementar o decrementar el número de equipos durante la ejecución de un programa.

		PTS	PLD	DIF
1	REAL SOCIEDAD	21	10	+5
2	REAL MADRID	20	9	+13
3	SEVILLA FC	20	9	+10
4	ATLÉTICO DE MADRID	18	9	+5
5	REAL BETIS	18	10	+4
6	CA OSASUNA	18	10	+1
7	RAYO VALLECANO	16	10	+5
8	ATHLETIC CLUB	16	9	+4
9	FC BARCELONA	15	9	+5
10	VALENCIA CF	13	10	+2
11	RCD ESPANYOL DE BARCELONA	13	10	0
12	RCD MALLORCA	12	10	-6
13	VILLARREAL CF	11	9	+3
14	RC CELTA	10	10	-2
15	ELCHE CF	10	10	-4
16	GRANADA CF	7	9	-6
17	CÁDIZ CF	7	10	-8
18	DEPORTIVO ALAVÉS	6	9	-9
19	LEVANTE UD	5	10	-9
20	GETAFE CF	2	10	-13

ESTRUCTURAS DE DATOS ESTÁTICAS VS DINÁMICAS

Las estructuras de datos dinámicas son aquellas en las que el tamaño ocupado en memoria se define antes de que el programa se ejecute, pero su tamaño puede variar a lo largo de la ejecución, pudiendo aumentar o disminuir su tamaño.

Si no se establece un tamaño mínimo a la hora de declarar esta estructura, se da por hecho que su tamaño es 0, por lo que, si se quieren insertar nuevos datos, tendrá que aumentar el tamaño previamente.

Una estructura de datos dinámica puede ser, por ejemplo, la cola de un supermercado, ya que no tiene un tamaño prefijado, sino que va cambiando a medida que los clientes se incorporan a la cola o van pasando por caja.

VECTORES/ARRAYS

DECLARACIÓN DE ARRAYS

Para declarar un array se escribe

```
tipo_de_dato[] nombre_del_array;
```

Para declarar un array de enteros escribimos

```
int[] numeros;
```

Para crear un array de 4 número enteros escribimos

```
numeros=new int[4];
```

La declaración y la creación del array se puede hacer en una misma línea.

```
int[] numeros =new int[4];
```

ELEMENTOS DEL ARRAY

Para inicializar el array de 4 enteros escribimos

```
numeros[0]=2;
numeros[1]=-4;
numeros[2]=15;
numeros[3]=-25;
```

Se pueden inicializar en un bucle **for** como resultado de alguna operación

```
for(int i=0; i<4; i++){
        numeros[i]=i*i+4;
}
```

No necesitamos recordar el número de elementos del array, su miembro dato *length* nos proporciona la dimensión del array. Escribimos de forma equivalente

```
for(int i=0; i<numeros.length; i++){
        numeros[i]=i*i+4;
}
```

RECORRER EL ARRAY

Los arrays se pueden declarar, crear e inicializar en una misma línea, del siguiente modo

```
int[] numeros={2, -4, 15, -25};
String[] nombres={"Juan", "José", "Miguel", "Antonio"};
```

Para imprimir a los elementos de array *nombres* se escribe

```
for(int i=0; i<nombres.length; i++){
        System.out.println(nombres[i]);
}
```

FUNCIONES DEL ARRAY

Nombre	Descripción	Parámetros
binarySearch	Busca un valor que le pasamos por parámetro, devuelve su posición. Debe estar ordenado.	Un array y un valor. Los dos del mismo tipo. Estos pueden ser un byte, char, double, float, int, long, short u objecto.
copyOf	Copia un array y lo devuelve en un nuevo array.	Un array y la longitud. Si se pasa del tamaño del array original, rellena los con ceros las posiciones sobrantes. Estos pueden ser un byte, char, double, float, int, long, short u objecto.
copyOfRange	Copia un array y lo devuelve en un nuevo array. Le indicamos la posición de origen y de destino.	Un array, posición origen y destino. Estos pueden ser un byte, char, double, float, int, long, short u objecto.
equals	Indica si dos arrays son iguales.	Dos arrays del mismo tipo.
fill	Rellena un array con un valor que le indiquemos como parámetro.	Un array y el valor a rellenar. Estos pueden ser un byte, char, double, float, int, long, short u objecto.
sort	Ordena el array.	Un array. Estos pueden ser un byte, char, double, float, int, long, short u objecto.
toString	Muestra el contenido del array pasado como parámetros	Un array. Estos pueden ser un byte, char, double, float, int, long, short u objecto.

Ejemplo con array

1) Crear el siguiente array

Dani	Fátima	David	Jesús	Miguel Ángel	Dani Cuenca	Yolanda

2) Mostrar por consola el array con los profesores del departamento de informática

3) Modificar un campo. Si el nombre empieza por D, sustituirlo por “vacio”

4) Mostrar de nuevo por consola el array con los profesores del departamento de informática

vacio	Fátima	vacio	Jesús	Miguel Ángel	vacio	Yolanda

MATRICES

Un array en Java puede tener más de una dimensión. El caso más general son los arrays bidimensionales, también llamados **matrices** o **tablas**.

Una matriz necesita dos índices para acceder a sus elementos. Gráficamente podemos representar una matriz como una tabla de **n filas y m columnas** cuyos elementos son todos del mismo tipo.

	0	1	2	3	4
0	1	2	3	4	5
1	2	4	6	8	10
2	3	6	9	12	15

DECLARACIÓN DE MATRICES

Para declarar una matriz que contenga números enteros:

```
int [][] ventas = new int [4][6];
```

Para declarar una matriz con números decimales:

```
double [][] temperaturas = new double[3][4];
```

Para declarar una matriz de Strings

```
String [][] nombres = new String[3][3];
```

				Promedio de ventas	58.000,00 €		
				% de comisión aplicada	3%		
				Ventas de los comerciales			
	1.450,00 €	50.000,00 €	60.000,00 €	70.000,00 €	80.000,00 €	90.000,00 €	100.000,00 €
Comisiones aplicadas	1%	500,00 €	600,00 €	700,00 €	800,00 €	900,00 €	1.000,00 €
	2%	1.000,00 €	1.200,00 €	1.400,00 €	1.600,00 €	1.800,00 €	2.000,00 €
	3%	1.500,00 €	1.800,00 €	2.100,00 €	2.400,00 €	2.700,00 €	3.000,00 €
	4%	2.000,00 €	2.400,00 €	2.800,00 €	3.200,00 €	3.600,00 €	4.000,00 €
	5%	2.500,00 €	3.000,00 €	3.500,00 €	4.000,00 €	4.500,00 €	5.000,00 €
	6%	3.000,00 €	3.600,00 €	4.200,00 €	4.800,00 €	5.400,00 €	6.000,00 €
	7%	3.500,00 €	4.200,00 €	4.900,00 €	5.600,00 €	6.300,00 €	7.000,00 €
	8%	4.000,00 €	4.800,00 €	5.600,00 €	6.400,00 €	7.200,00 €	8.000,00 €
	9%	4.500,00 €	5.400,00 €	6.300,00 €	7.200,00 €	8.100,00 €	9.000,00 €
	10%	5.000,00 €	6.000,00 €	7.000,00 €	8.000,00 €	9.000,00 €	10.000,00 €

ELEMENTOS DE LA MATRIZ

Inicialización de los elementos de la matriz en varias líneas

```
matriz[0][0] = 2;
matriz[0][1] = 4;
matriz[0][2] = 4;
matriz[1][0] = 6;
matriz[1][1] = 6;
matriz[1][2] = 9;
matriz[2][0] = 8;
matriz[2][1] = 10;
matriz[2][2] = 12;
```

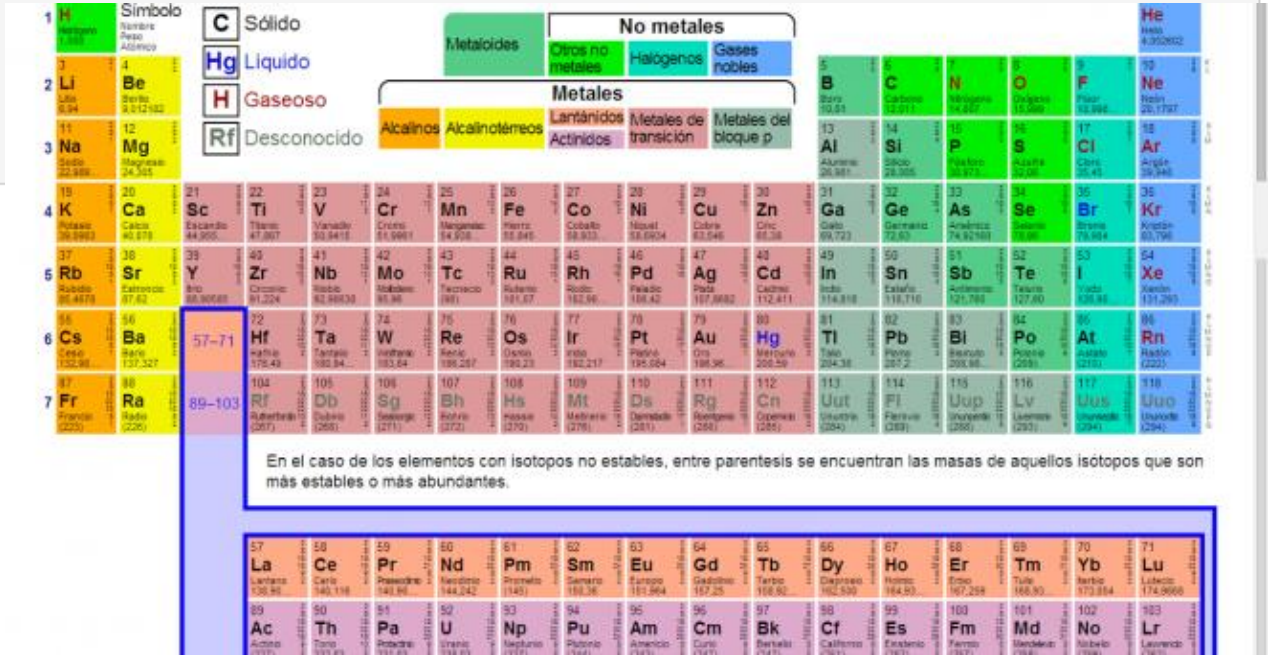
Inicialización de los elementos de la matriz en una sola línea

```
int [][] matriz = {{2,4,4},{6,6,9},{8,10,12}};
```

RECORRER LA MATRIZ

De igual manera, si nos apoyamos en el método .lenght del array podremos listar el contenido de la matriz

```
for (int x=0; x < matriz.length; x++) {
  for (int y=0; y < matriz[x].length; y++) {
    System.out.println (matriz[x][y]);
  }
}
```



Ejemplo con matriz

1) Crear la siguiente matriz

Dani	**Fátima**	**David**	**Jesús**	**Miguel Ángel**	**Dani Cuenca**	**Yolanda**
José Carlos	**Daniel Báñez**	**Daniel Olmo**	**Jose Luis**	**Pedro**	**José Ángel**	**Fran**

2) Mostrar por consola la matriz con los profesores del departamento de informática y otro personal del centro

3) Modificar un campo. Si el nombre empieza por D, sustituirlo por “vacio”

4) Mostrar de nuevo por consola el array con los profesores del departamento de informática

vacio	**Fátima**	**vacio**	**Jesús**	**Miguel Ángel**	**vacio**	**Yolanda**
José Carlos	**vacio**	**vacio**	**Jose Luis**	**Pedro**	**José Ángel**	**Fran**