

mat|r

Bienvenidos a la Clase N°03

En unos minutos comenzamos...

Clase 03

Introducción a mat|r script

mat|r project

Ecosistema 360 para la creación de experiencias digitales

2019

Expositor

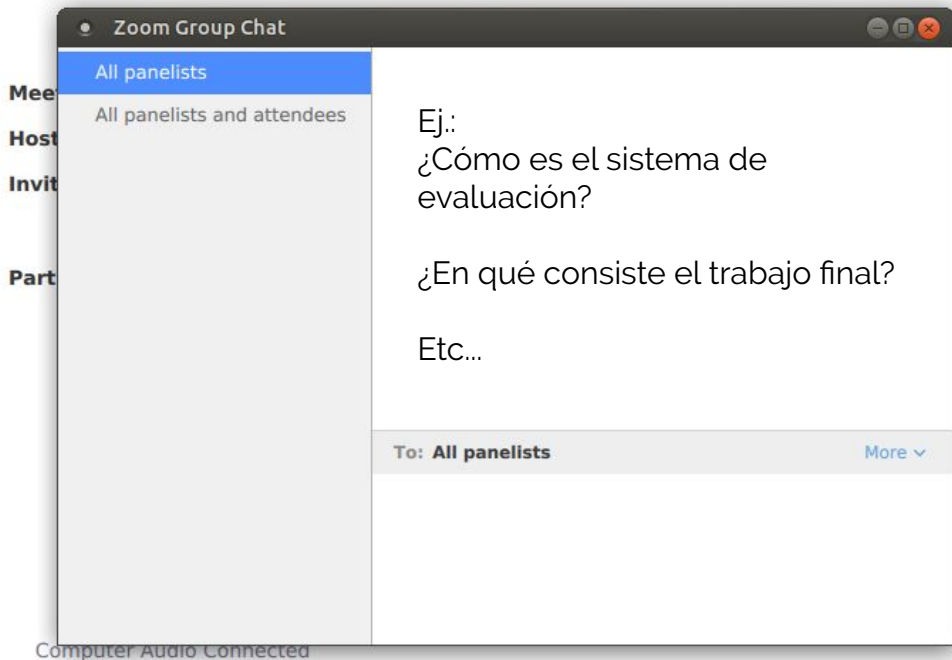


Agustina Dinamarca

Capacitadora mat|r

Se dedica a representar el producto en actividades que tienen como objetivo educar a usuarios sobre qué es mat|r, para qué sirve y cómo usarlo a través del dictado de webinars y capacitaciones, y el desarrollo de contenido educativo.

Usa el chat durante la clase si necesitas hacer alguna consulta sobre la dinámica y organización del curso



Recording...

Question and Answer

Open(0) Answered(0)

Usa Q&A durante la clase si necesitas hacer preguntas sobre el contenido que se está dictando

No open questions

Ej.:
¿Qué tipo de componentes visuales hay?
¿Qué tipo de dato soporta cada uno?
Etc...

Allow anonymous questions



DESAFÍO MAT|R DE LA SEMANA (:

¡PON A PRUEBA TUS CONOCIMIENTOS E INGENIO!

DESARROLLA UNA MAT|R APP PARA CALIFICACIONES

Ayuda:

- Un solo layout en el que el usuario tiene que colocar las notas de las evaluaciones por asignatura y un botón que calcule nota promedio e indique si se exige o no en las mismas.



**¡Espero recibir sus proyectos
por email si hay dudas!**



CONTINUEMOS...

OBJETIVOS DE LA CLASE

- Entender la diferencia entre Array y Map.
- Profundizar sobre sus métodos.



ARREGLOS Y MAPAS

¿QUÉ DIFERENCIA HAY ENTRE ARRAY Y MAP?

ARRAY	MAP
Colección finita de elementos.	
Homogéneo (cada elemento es del mismo tipo entre sí).	
Ordenado (cada valor tiene un índice único (Integer) asociado que indica su orden de aparición en el arreglo).	No ordenado (cada valor tiene una clave única (String) asociada). Hablamos de pares clave-valor.

EJEMPLOS DE ARRAY

```
Array<Integer> varInt = [2, 3, 13, 5, 27]
Array<String> varStr = ["a", "ab", "abc"]
Array<Bool> varBool = [true, false, false, true, false]
Array<Double> varDoub = [2.22, 3.456, -0.12]
Array<Persona> varPer = [po, p1]
```

// donde:

```
Model Persona {
    String nombre
    Integer edad
}
```

```
Persona po = Persona()
po.nombre = "Juan"
po.edad = 34
```

```
Persona p1 = Persona()
p1.nombre = "Pedro"
p1.edad = 21
```

EJEMPLOS DE ARRAY

```
Array<Integer> varInt = [2, 3, 13, 5, 27]
```

```
2 tiene índice 0
```

```
3 tiene índice 1
```

```
...
```

```
27 tiene índice 4
```

¡El índice del primer elemento de un arreglo es el cero 0!

Ej. ¿Cómo acceder al valor del elemento con índice 2?

```
Integer elemento = varInt.get(2) // elemento tiene valor 13
```

Uso un **método de Array** y **le paso el índice** del elemento al cual quiero acceder

EJEMPLOS DE MAP

```
Map<Integer> varInt = {"keyA":2, "keyB":3, "keyC":13, "keyD": 5}
Map<String> varStr = {"A":"a", "B":"ab", "C":"abc"}
Map<Bool> varBool = {"m1":true, "m2":false, "m3":false, "m4":true}
Map<Double> varDoub = {"1":2.22, "2":3.456, "3":-0.12}
Map<Persona> varPer = {"Persona1":po, "Persona2":p1}
```

// donde:

```
Model Persona {
    String nombre
    Integer edad
}
```

```
Persona po = Persona()
po.nombre = "Juan"
po.edad = 34
```

```
Persona p1 = Persona()
p1.nombre = "Pedro"
p1.edad = 21
```

¡Notar que los elementos son pares clave:valor!

¡Las claves son Strings y son únicas!

EJEMPLOS DE MAP

```
Map<Integer> varInt = {"keyA":2, "keyB":3, "keyC":13, "keyD": 5}
```

¡Los elementos se acceden a través de sus claves!

```
Integer elemento1 = varInt.get("keyA") // El valor de elemento1 es 2
```

```
Integer elemento2 = varInt.get("keyB") // El valor de elemento2 es 3
```

Uso un **método de Map** y **le paso la clave** del elemento al cual quiero acceder

DEFINICIONES EN MAT|R SCRIPT

	ARRAY	MAP
Declaración	<code>Array<T> arrayName</code>	<code>Map<T> mapName</code>
Tipo de dato	T puede tomar cualquier tipo.	T puede tomar cualquier tipo.
Constructor	<code>Array<T> arrayName = []</code>	<code>Map<T> mapName = {}</code>
Ejemplo	<code>Array<Integer> arrayName = [10, 33, 25]</code>	<code>Map<Integer> mapName = {"viernes": 1, "martes": 3}</code>

MÉTODOS

LOS TIPOS DE DATO TIENEN ASOCIADOS MÉTODOS PROPIOS.

- Un método es un conjunto de instrucciones, esto es, ~ función (con o sin parámetros, y con o sin un retorno).
- Al ser ejecutado realiza una determinada tarea.
- Determina un tipo de operación que se puede realizar con dicho tipo de dato.
- Se accede a un método de un tipo de dato de la siguiente forma:

Nombre de la variable, seguido de un punto, seguido del método.

Ej. :

```
Array<Integer> arrayInt = [33, 16, 10, 15]  
arrayInt.size() //4
```



MÉTODOS DE ARREGLOS

MÉTODOS DE ARRAYS

<code>Integer size()</code>	Devuelve la cantidad total de elementos en la colección.
<code>T get(Integer indice)</code>	Recorre la colección y retorna una referencia al elemento en el índice indicado. La colección y su tamaño no son modificados.
<code>void remove(Integer indice)</code>	Recorre la colección y elimina el elemento en el índice indicado. Luego de remover el elemento el tamaño de la colección disminuye en uno.
<code>T pop()</code>	Remueve y devuelve el último elemento de la colección. El tamaño de la colección disminuye en uno y el elemento retirado de la misma es devuelto.
<code>Array<T> selected()</code>	Devuelve un Array con los elementos seleccionados en la colección.

EJEMPLO

```
Array<Integer> arr1 = [3, 6, 8, 5, 1]
```

```
arr1.size() // 5
```

```
arr1.get(2) // 8
```

```
arr1.remove(1) // [3, 8, 5, 1] tamaño 4, remueve el elemento de índice  
1
```

```
arr1.pop() // [3, 8, 5] tamaño 3, remueve el último elemento
```

MÉTODOS DE ARRAYS

<code>void setSelected(Array<Integer> indices)</code>	Marca como seleccionado uno o más elementos de la colección.
<code>T shift()</code>	Remueve y devuelve el primer elemento de la colección. El tamaño de la colección disminuye en uno.
<code>void reverse()</code>	Invierte el orden de los elementos de la lista.
<code>void sort()</code>	Compara todos los elementos de la lista y los ordena de menor a mayor. Los elementos con valor nulo de la lista se ordenan al final de la misma. Si T no es un tipo comparable se mantendrá el orden de la colección.
<code>void removeAll()</code>	Elimina todos los elementos de la lista, el tamaño de la lista queda en cero.

EJEMPLO

```
Array<Integer> arr1 = [3, 6, 8, 5, 1]
```

```
arr1.setSelected([2, 4]) // selecciono elementos de índices 2 y 4
```

```
arr1.selected() // [8, 1], obtengo elementos seleccionados
```

```
arr1.shift() // 3, remueve el primer elemento [6, 8, 5, 1]
```

```
arr1.reverse() // [1, 5, 8, 6], invierte el orden de los elementos
```

```
arr1.sort() // [1, 5, 6, 8], ordena de menor a mayor
```

```
arr1.removeAll() // [], remueve todos los elementos, arreglo vacío
```

MÉTODOS DE ARRAYS

<code>Array<T> clone()</code>	Crea una nueva instancia de un array con misma cantidad de elementos, cada uno de sus elementos son nuevas referencias a copias fieles de los objetos del array original.
<code>Integer indexOf(T elemento)</code>	Recorre la lista y devuelve la posición de la primer ocurrencia del elemento parámetro. Si el elemento no se encuentra en la lista retorna el valor -1.
<code>void add(T elemento)</code>	Agrega el elemento parámetro al final de la colección. El tamaño de la colección aumenta en uno.
<code>void set(Integer indice, T elemento)</code>	Reemplaza el elemento de la colección que se encuentra en el índice parámetro por el elemento parámetro. Si el índice no existe lanzará una excepción.

EJEMPLO

```
Array<Integer> arr1 = [3, 6, 8, 5, 1]
```

```
arr1.clone() // [3, 6, 8, 5, 1], clono el arreglo arr1
```

```
arr1.indexOf(8) // 2, obtengo el índice del elemento de valor 8
```

```
arr1.add(7) // [3, 6, 8, 5, 1, 7], agrego al final el elemento 7
```

```
arr1.set(2, 11) // [3, 6, 11, 5, 1], coloco en el índice 2 el elemento 11
```


MÉTODOS DE ARRAYS

Integer push (T elemento)	Inserta el elemento parámetro al final de la lista, el tamaño de la lista crece en uno.
Integer unshift (T elemento)	Inserta el elemento parámetro en la primera posición de la colección, el tamaño de la misma crece en uno y se retorna el nuevo tamaño.
Array<T> concat (Array<T> colección)	Crea y retorna una nueva colección que tiene los elementos de la instancia, agregando al final los de la colección del argumento. Si la colección argumento no es de tipo compatible lanza excepción.

EJEMPLO

```
Array<Integer> arr1 = [3, 6, 8, 5, 1]
```

```
Array<Integer> arr2 = [11, 22]
```

```
arr1.push(14) // [3, 6, 8, 5, 1, 14], agrego el elemento 14 al final
```

```
arr1.unshift(45) // [45, 3, 6, 8, 5, 1, 14], agrego el elemento 45 al  
inicio
```

```
arr1.concat(arr2) // [45, 3, 6, 8, 5, 1, 14, 11, 22], concateno arreglos
```



MÉTODOS DE MAPAS

MÉTODOS DE MAPS

<code>Integer size()</code>	Devuelve la cantidad total de pares en la colección.
<code>T get(String clave)</code>	Retorna el elemento asociado a la clave del parámetro. La colección y su tamaño no son modificados. En caso de no encontrar la clave se produce una excepción.
<code>T get(String clave, T valorPorDefecto)</code>	Retorna el elemento asociado a la clave del parámetro. La colección y su tamaño no son modificados. En caso de no encontrar la clave retorna el 'valorPorDefecto' pasado como argumento.
<code>void remove(String clave)</code>	Elimina el elemento asociado a la clave parámetro. Luego de remover el elemento el tamaño de la colección disminuye en uno. No efectúa modificación alguna en caso de no encontrar la clave.

EJEMPLO

```
Map<Integer> map1 = {"Ana":26, "Lucca":13, "Mateo":33, "Inés":11}
```

```
map1.size() // 4, tamaño del mapa
```

```
map1.get("Lucca") // 13, obtengo el valor asociado a la clave "Lucca"
```

```
map1.get("Pedro", 0) // 0, obtengo valor por defecto 0 si la clave "Pedro"  
no existe
```

```
map1.remove("Mateo") // {"Ana":26, "Lucca":13, "Inés":11}, remuevo elemento  
con clave "Mateo"
```

MÉTODOS DE MAPS

<code>void removeAll()</code>	Elimina todos los elementos del mapa, el tamaño del mapa queda en cero.
<code>Map<T> clone()</code>	Crea una nueva instancia de un mapa con misma cantidad de elementos, cada uno de sus elementos y claves son nuevas referencias a copias fieles de los elementos del mapa original.
<code>Bool hasKey(String clave)</code>	Devuelve true si encuentra el elemento asociado a la clave parámetro, false caso contrario.
<code>Array<String> keys()</code>	Devuelve un array con todas las claves de los pares que contiene.
<code>Array<T> values()</code>	Devuelve un array con todos los valores de los pares que contiene.
<code>void set(String clave, T elemento)</code>	Agrega el elemento parámetro al mapa y lo asocia a la clave. Si la clave ya tenía un elemento asociado lo reemplaza.

EJEMPLO

```
Map<Integer> map1 = {"Ana":26, "Lucca":13, "Mateo":33, "Inés":11}

map1.hasKey("Inés") // true, retorna true o false si existe o no la clave

map1.keys() // ["Ana", "Lucca", "Mateo", "Inés"], retorno arreglo de claves

map1.values() // [26, 13, 33, 11], retorno arreglo de valores

map1.set("Florencia", 14) // agrego elemento de clave y valor especificado

map1.clone() // clono mapa

map1.removeAll() // {}, remuevo todos los elementos
```

OBSERVACIONES DEL CÓDIGO EJEMPLO

Muestra un cuadro de diálogo con título y descripción.

```
void broker.ui.showAlert(String title, String description)
```

Ej.:

```
Integer edad = 21  
String nombre = "Juan"
```

```
broker.ui.showAlert("Nombre:", nombre) nombre es una variable de tipo String.
```

```
broker.ui.showAlert("Edad:", edad.toString()) edad es una variable de tipo Integer. Necesito usar el método "toString" de Integer para transformarlo a String (de 21 a "21").
```




RESUMEN

RESUMEN

- La diferencia entre Array y Map está en que si bien ambos son conjuntos de elementos finitos y homogéneos, el primero es ordenado y segundo no. Los elementos en Map forman pares clave(String)-valor. Cada valor está identificado con una clave ÚNICA.
- Array y Map tienen cada uno diversos métodos propios para operar con ellos.



¿CONSULTAS O
DUDAS?

COMUNÍCATE NOSOTROS



support@matrproject.com



<http://forum.matrproject.com>

mat|r

matrproject.com

—

Contacto

Soledad Peñaloza

Community Manager

soledad.p@matrproject.com

Sergio Farias

Product Development

sergio.f@matrproject.com

Agustina Dinamarca

Capacitadora mat|r

agustina.d@matrproject.com

mat|r

iMuchas Gracias!

mat|r project

Ecosistema 360 para la creación de experiencias digitales