

{desafío}
latam_

Arreglos y archivos _

Sesión Presencial 1



Itinerario

Activación de conceptos

Desarrollo Desafío

Panel de discusión

Activación de conceptos

Arrays

Posibles usos:

- En una aplicación web podemos traer los datos de la base de datos en un array y luego mostrar los datos en la página.
- Al traer los datos desde una API podría venir una colección y podríamos guardarlo dentro de un arreglo.
- Traer información guardada en uno o más archivos.
- Crear gráficos.

Índices

Los índices van de cero hasta $n - 1$, donde n es la cantidad de elementos del arreglo.

```
a = [1, 2, 'hola', 'a', 'todos']
```

```
a[0] # 1
```

```
a[1] # 2
```

```
a[4] # todos
```

Índices fuera de rango

`a[-1]` # => lee el último el elemento

`a[-2]` # => Lee el penúltimo elemento

`a[10000000]` # => o cualquier índice fuera de rango devuelve nil

Membresía

Saber si un elemento se encuentra dentro de un arreglo

```
1  a = [1, 2, 3, 4, 5]
2  a.include? 4 # => true
3  a.include? 6 # => false
```

Agregar elementos

```
1 a = [1, 2, 3, 4, 5]
2 a.push 6
3 a.push 7
4 print a
```


Borrar elementos

```
1 a = [1, 2, 3, 4, 5, 6, 7]
2 a.delete(7)
3 print a
```

Contar elementos

```
1 a = [1, 2, 3, 4, 5]
2 print a.count
```

Distintas formas de iterar

Con times

```
1 a = [1, 2, 3, 4, 5]
2 a.count.times do |i|
3   puts i
4 end
5
```

Con while

```
1 a = [1, 2, 3, 4, 5]
2 i = 0
3
4 while i < a.count
5   puts a[i]
6   i += 1
7 end
```

Con .each

```
1 a = [1, 2, 3, 4, 5]
2 a.each do |e|
3   puts e
4 end
```

Filtrando datos

Paso a paso

```
a = [100, 200, 1000, 5000, 10000, 10, 5000]
n = a.count
filtered_array = []
n.times do |i|
  if a[i] >= 1000
    filtered_array.push a[i]
  end
end
print filtered_array
```

Con .select

```
a = [100, 200, 1000, 5000, 10000, 10, 5000]
a.select{|x| x >= 1000 }
```

Leer archivo

```
data = open( 'data' ).read  
# => "1,2,3,4,5,6,7,8,9,10\n"
```

Leer archivo con múltiples líneas

```
data = open('archivo2').readlines
```

Leer datos con múltiples líneas y limpiar

```
original_data = open('archivo2.txt').readlines
lines = original_data.count
array = []
lines.times do |i|
  array << original_data[i].to_i
end
```

Es posible ocupar **.map**, por ejemplo: `lines.map{ |x| x.to_i }`

Desafío

Panel de discusión

{desafío}
latam_

*Academia de
talentos digitales*

www.desafiolatam.com