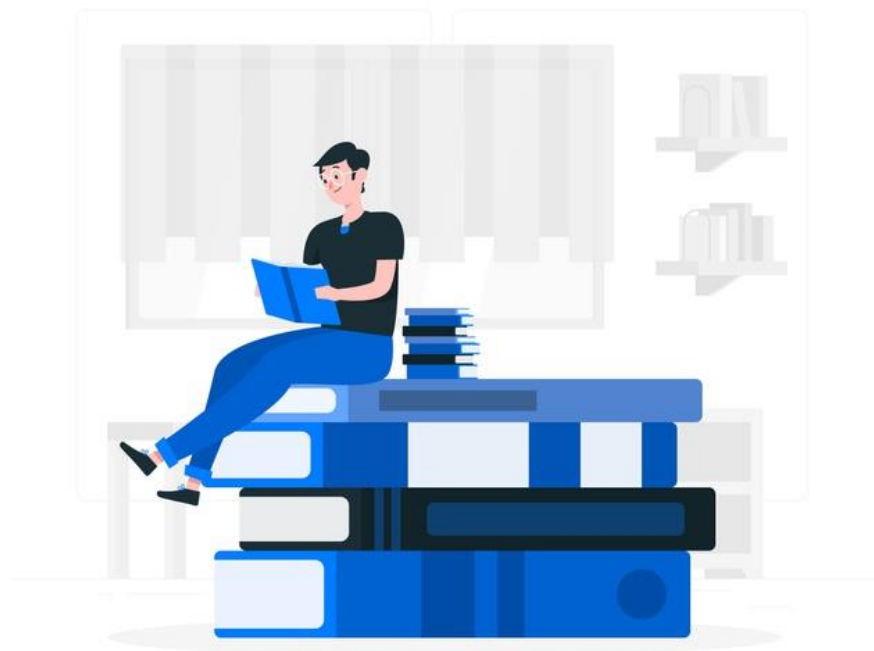
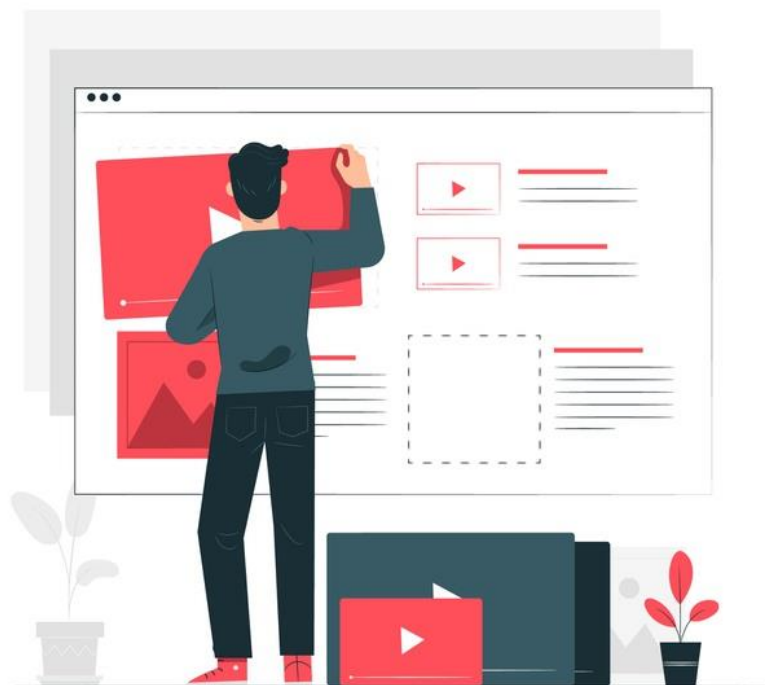


Khoá học: Java Developer

# Bài 02: Array và String



# Nội dung



1. Mảng (Array)
2. Chuỗi (String)
3. Các thao tác với chuỗi
4. Vòng lặp foreach

# Mảng

Array

- Mảng là một tập hợp các phần tử cùng kiểu dữ liệu và cùng một tên.
- Để truy xuất đến các phần tử khác nhau, ta dùng chỉ số (index). Chỉ số được đánh từ 0.
- VD: Đây là một tập hợp điểm lập trình của 6 sinh viên.
- VD: Đây là một tập hợp tuổi của 4 người.

float diemLT[]	7.5	4.0	8.2	10	6.5	3.8
Index =	0	1	2	3	4	5

int tuoi[]	12	46	7	23
Index =	0	1	2	3

# Các thao tác với mảng

## Khai báo mảng



### I. Chỉ khai báo, không khởi tạo

Không khai báo số lượng phần tử

- `int[] a;`

Có khai báo số lượng 5 phần tử trong mảng.

- `string[] c = new string[5];`

### II. Khai báo và khởi tạo

- `int[] d1 = {2,3,4,5,6};`
- `int[] d2 = new int[] {2,3,4,5,6};`

Khai báo mảng số nguyên gồm 5 phần tử chứa các giá trị 2, 3, 4, 5, 6.



# Các thao tác với mảng

Truy xuất số lượng/giá trị phần tử trong mảng

## I. Sử dụng chỉ số (index) để gọi phần tử. (index tính từ 0 )

- `int[] a = {4, 3, 5, 7};`

Khai báo mảng a gồm 4 phần tử

- `a[2] = a[1] * 4;`

Giá trị phần tử trị số 2 bằng giá trị phần tử số 1 \* 4 => 3 \* 4 = 12

- Kết quả sau khi thay đổi giá trị: `a = {4, 3, 12, 7}`

## II. Truy vấn số lượng phần tử

- `a.length;`

Sử dụng thuộc tính length để truy vấn số lượng phần tử của mảng

# Các thao tác với mảng

Duyệt mảng

- Hai vòng lặp thường được sử dụng để duyệt mảng là for và for-each.

```
int [] a = {4, 3, 5, 9};
```

## Cú pháp for

```
for ( int i=0; i<a.length; i++) {  
    Console.WriteLine(a[i]);  
}
```

## Cú pháp for-each

```
foreach ( int xElement in a) {  
    Console.WriteLine(xElement);  
}
```

# Các thao tác với mảng

Duyệt mảng

- **Ví dụ:** Tính tổng các số chẵn có trong mảng a.
- **Diễn giải:**
  1. Lấy giá trị từng phần tử từ mảng với vòng lặp for-each
  2. Kiểm tra giá trị phần tử có phải là số chẵn không?
  3. Nếu là số chẵn thì cộng vào tổng.

```
int[] a = {9, 3, 8, 7, 3, 9, 4, 2};  
double tong = 0;  
foreach (int xElement in a) {  
    if (xElement % 2 == 0) {  
        tong += xElement;  
    }  
}  
Console.WriteLine("Tổng: " + tong);
```

# Chuỗi

## String

- **String** lưu trữ một chuỗi ký tự.
- Trong C#, khai báo:
  - **string**: kiểu dữ liệu
  - **String**: Lớp tiện ích có các phương thức xử lý chuỗi.

```
char[] hiArray = {'H', 'e', 'l', 'l', 'o'};  
string greeting = "Hello";
```

```
Console.WriteLine(hiArray);  
Console.WriteLine(greeting);
```

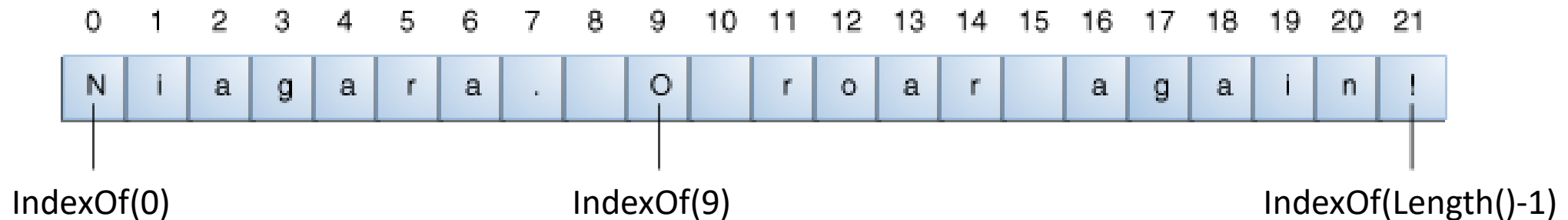
Ký tự đặc biệt	Hiển thị
\t	Ký tự tab
\r	Về đầu dòng
\n	Xuống dòng
\\	\
\"	"



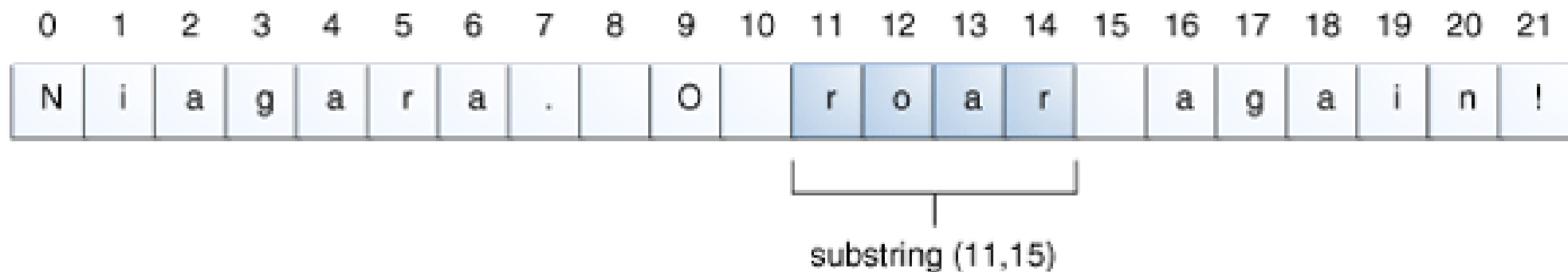
# Thao tác với chuỗi

Object

```
string palindrome = "Niagara. O roar again!";
```



```
char aChar = palindrome.IndexOf(9);
```



```
string roar = palindrome.Substring(11, 15);
```

# Thao tác với biến kiểu string

Phương thức (Methods)	Mô tả (Description)
<b>.ToLower()</b> <b>.ToUpper()</b>	Chuyển nội dung chuỗi sang in thường Chuyển nội dung chuỗi sang in HOA.
<b>.Trim()</b> <b>.TrimStart()</b> <b>.TrimEnd()</b>	Tự động loại bỏ các ký tự trắng dư thừa Tự động loại bỏ các ký tự trắng dư thừa ở đầu chuỗi Tự động loại bỏ các ký tự trắng dư thừa ở cuối chuỗi
<b>.Length()</b>	Lấy độ dài chuỗi
<b>.Substring(StartIndex, Count)</b>	Lấy chuỗi con Tách và xoá bỏ chuỗi con ra khỏi chuỗi hiện tại
<b>.Contains(text)</b>	Kiểm tra chuỗi text có là chuỗi con của chuỗi hay không
<b>.IndexOf(index)</b>	Lấy ký tự tại vị trí chỉ định
<b>.Replace(find, replace)</b>	Tìm kiếm và thay thế tất cả
<b>.Split(separator)</b>	Tách phần tử theo 1 ký hiệu cho trước. Kết quả trả về một mảng các phần tử đã cắt.

# Thao tác với biến kiểu string

Phương thức (Methods)	Mô tả (Description)
<b>.CompareTo(text)</b>	So sánh chuỗi và text. Nếu 2 chuỗi giống nhau sẽ trả về 0.
<b>.CopyTo()</b>	Sao chép 1 phần các ký tự ở chuỗi gốc sang một mảng ký tự khác
<b>.Insert()</b>	Chèn thêm nội dung vào chuỗi ở vị trí chỉ định
<b>.Remove(StartIndex, Count)</b>	Tách và xóa bỏ chuỗi con ra khỏi chuỗi hiện tại
<b>.PadLeft()</b> <b>.PadRight()</b>	Canh lề trái / phải