

Welcome to my class

Presented by.....
Shajeda Yeasmin
Instructor(Computer)
Bangladesh Sweden Polytechnic Institute.
Kaptai. Rangmati.

Name of the Subject & code

Data Structure & Algorithm

Subject Code : 28542

অধ্যায়-৩

(Arrays, Pointers and Strings)



অ্যারে (Array)

একটি সাধারণ ভেরিয়েবলের নামের আওতায় মেমরিতে পরপর সংরক্ষিত একই টাইপের কতগুলো ডেটার সমষ্টিকে অ্যারে বলা হয়। অথবা, অ্যারে হল একই টাইপের কতগুলো ভেরিয়েবলের সেট।

অ্যারে ভেরিয়েবল একই নামে, একই টাইপের একাধিক ডেটা সংরক্ষণ করতে পারে। সাধারণ ভেরিয়েবল ঘোষণার মত ব্যবহারের পূর্বে ডেটা টাইপসহ অ্যারে ভেরিয়েবল ঘোষণা করতে হয়।

অ্যারে ঘোষণার ফরম্যাট হল:

`DataType ArrayName [ArraySize];`

উদাহরণ: `int Roll [15];`

অ্যারে দুই প্রকার।

যথা :

১) একমাত্রিক অ্যারে (One Dimensional Array) ।

২) বহুমাত্রিক অ্যারে (Multi Dimensional Array ।

একমাত্রিক অ্যারে :

যে অ্যারের একটি মাত্র কলাম বা একটি মাত্র রো থাকে তাকে একমাত্রিক অ্যারে বলে।

যেমন: `int Roll [5];`

দ্বিমাত্রিক অ্যারে :

যে অ্যারের একাধিক কলাম ও একাধিক রো থাকে তাকে দ্বিমাত্রিক অ্যারে বলে।

যেমন: `int Roll [5][6];`

এখানে প্রথম সংখ্যাটি সারি এবং দ্বিতীয়টি কলামের সংখ্যা নির্দেশ করে।

পয়েন্টার কি?

পয়েন্টার হচ্ছে একটা ভ্যারিয়েবল যার ভ্যালু হচ্ছে আরেকটি ভ্যারিয়েবল এর মেমরি লোকেশন। পয়েন্টার একটা ডেটা, অ্যারে বা ভ্যারিয়েবল এর কম্পিউটার মেমরি লোকেশন রিপ্রেজেন্ট করে বা পয়েন্ট করে। অন্যান্য ভ্যারিয়েবল এর মত পয়েন্টার ভ্যারিয়েবল ব্যবহার করার আগে কম্পিউটার/ কম্পাইলারকে বলতে হবে এটা একটি পয়েন্টার ভ্যারিয়েবল।

পয়েন্টার এমন একটা অবজেক্ট যেটা অন্য একটি ভেরিয়েবল, কম্পিউটারের মেমরির যে অ্যাড্রেসে আছে, সেই এড্রেস কে পয়েন্ট করে। আমরা যত ভেরিয়েবল ব্যবহার করি, সেগুলো সবই কিন্তু কম্পিউটারের মেমরিতে একটি নির্দিষ্ট জায়গায় সঞ্চিত হয়। আমরা একটু আগে যে অ্যারে দেখলাম, সেটাও কিন্তু কোন না কোন এড্রেসে সংরক্ষিত আছে।

String.

স্ট্রিং সাধারণত স্ট্রিং বলা হয়ে থাকে। পাইথনে যে কোন সেনটেন্সকেই স্ট্রিং হিসেবে ব্যবহার করা যায় সিঙ্গেল(' '), ডাবল(" ") কিংবা ট্রিপল("""" """) কোটেশন এর মাধ্যমে। আমাদের পাইথন কনসোলে যদি নিচের মত করে বাক্য লিখে এন্টার চাপি তাহলে আউটপুটে সেই বাক্যকে দেখতে পারবো।

```
>>> "We love python!
```

```
""We love python!>>>
```

```
'The most popular general purpose programming language'
```

```
'The most popular general purpose programming language'
```


Mention different dimension of array with diagram.

7

অ্যারে দুই প্রকার।

যথা :

১) একমাত্রিক অ্যারে (One Dimensional Array) ।

২) বহুমাত্রিক অ্যারে (Multi Dimensional Array) ।

আবার বহুমাত্রিক অ্যারে দুই প্রকার

যথাঃ

১) দ্বিমাত্রিক অ্যারে (Two Dimensional Array) ।

২) ত্রিমাত্রিক অ্যারে (Three Dimensional Array) ।

একমাত্রিক অ্যারে :

যে অ্যারের একটি মাত্র কলাম বা একটি মাত্র রো থাকে তাকে একমাত্রিক অ্যারে বলে।

যেমন: `int Roll [5];`

দ্বিমাত্রিক অ্যারে :

যে অ্যারের একাধিক কলাম ও একাধিক রো থাকে তাকে দ্বিমাত্রিক অ্যারে বলে।

যেমন: `int Roll [5][6];`

এখানে প্রথম সংখ্যাটি সারি এবং দ্বিতীয়টি কলামের সংখ্যা নির্দেশ করে।

ত্রিমাত্রিক অ্যারে:

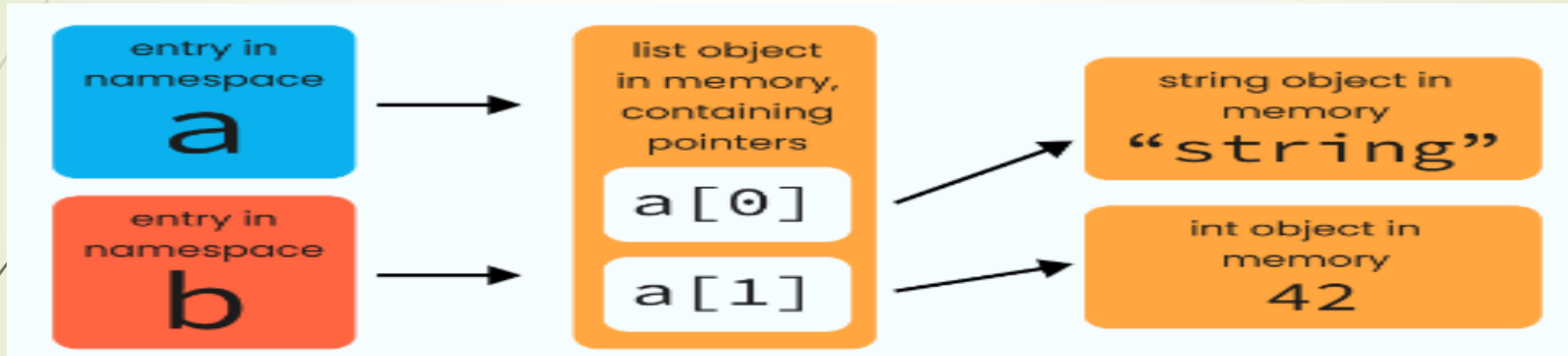
মূলত ত্রিমাত্রিক অ্যারে হলো অ্যারের অ্যারে।

যেমনঃ `int Roll [4][3][2];`

এখানে প্রথম সংখ্যাটি সারি ও দ্বিতীয় সংখ্যাটি কলাম এবং তৃতীয় সংখ্যাটি অ্যারের সংখ্যা।

Explain the initialization of Pointer.

পয়েন্টারগুলিকে নাম হিসাবে বিবেচনা করা যেতে পারে - অর্থাৎ, পাইথনের নেমস্পেসে এন্ট্রিগুলি - যা পাইথনের মেমরির বস্তুর সাথে মিলে যায়। উপরের উদাহরণে, পয়েন্টারটি হল `my_string` , এবং মেমরিতে থাকা বস্তুটি হল "Hello World!" মান সহ স্ট্রিং।



পয়েন্টার ঘোষণায়, আমরা শুধুমাত্র পয়েন্টার ঘোষণা করি কিন্তু এটি আরম্ভ করি না। একটি পয়েন্টার ঘোষণা করতে, আমরা এর নামের আগে (*) ডিরেফারেন্স অপারেটর ব্যবহার করি। `int *ptr;` এখানে ঘোষিত পয়েন্টার কিছু র্যান্ডম মেমরি ঠিকানা নির্দেশ করবে কারণ এটি আরম্ভ করা হয়নি।

Explain String declaration and initialization.

পাইথনে খুবই গুরুত্বপূর্ণ ডেটা টাইপ হলো স্ট্রিং। একগুচ্ছ ক্যারেক্টার বা কিছু ওয়ার্ডের সিকুয়েন্সকে সাধারণত স্ট্রিং বলা হয়ে থাকে। পাইথনে যে কোন সেনটেন্সকেই স্ট্রিং হিসেবে ব্যবহার করা যায় সিঙ্গেল(' '), ডাবল(" ") কিংবা ট্রিপল(""" """) কোটেশন এর মাধ্যমে। আমাদের পাইথন কনসোলে যদি নিচের মত করে বাক্য লিখে এন্টার চাপি তাহলে আউটপুটে সেই বাক্যকে দেখতে পারবো।

```
>>> "We love python!"
```

```
'We love python
```

```
'The most popular general purpose programming language
```

```
"The most popular general purpose programming language"
```

লক্ষণীয়,

ইনপুট দেয়ার সময় ডাবল বা সিঙ্গেল কোটেশন যাই ব্যবহার করা হোক না কেন, আউটপুটের সময় সিঙ্গেল কোট দিয়ে সেই স্ট্রিং কে দেখায়।

স্ট্রিং অপারেশন

কনক্যাটেনেশন (Concatenation)

ইন্টিজার বা ফ্লুটের মত, স্ট্রিংকেও যোগ করা যায় যাকে কনক্যাটেনেশন বলা হয়।

```
>>> "Spam" + 'eggs'
```

```
'Spameggs'
```

```
>>> print("First string" + ", " + "second string")
```

```
First string, second string
```

তাই বলে কোন নাম্বারের সাথে স্ট্রিং যোগ করা যাবে না,

```
>>> "2" + "2"
```

```
'22'
```

```
>>> 1 + '2' + 3 + '4'
```

রিপিটেশন (Repetition) যোগের মত স্ট্রিং নিয়ে গুনও করা যায়, একে রিপিটেশন বলে। তবে এই গুন হতে হবে একটি স্ট্রিং এর সাথে একটি ইন্টিজার নাম্বারের। স্ট্রিং এবং স্ট্রিং এর মধ্যে নয় অথবা ফ্লুট টাইপের ডাটার সাথে নয়। উদাহরণ,

```
>>> print("spam" * 3)
```

```
spamspamspam
```

স্ট্রিং ফরম্যাটিং

স্ট্রিং ডাটার সাথে স্ট্রিং টাইপের ডাটাকে যুক্ত করে সুন্দর স্ট্রিং আউটপুট তৈরি করতে format মেথড ব্যবহার করা হয়। এর মাধ্যমে একটি স্ট্রিং এর মধ্যে থাকা কিছু আর্গুমেন্টকে রিপ্লেস বা সাবস্টিটিউট করা যায়। format মেথডের মধ্যের প্রত্যেকটি আর্গুমেন্ট দিয়ে এর সামনে থাকা স্ট্রিং এর মধ্যের প্লেস হোল্ডার গুলোকে রিপ্লেস করা হয়। প্লেস হোল্ডার গুলো {} এর সাথে ইনডেক্স বা নাম ব্যবহার করে ডিফাইন করা হয়। একটি উদাহরণ দেখলেই বিষয়টি পরিষ্কার হয়ে যাবে

```
message = "If x = {x} and y = {y}, then x+y = {z}".format(x = 20, y = 300, z = 20+300)
print(message)
```

আউটপুট,

If x = 20 and y = 300, then x+y = 320

join মেথড একটি স্ট্রিং ওয়াল লিস্টের (লিস্ট নিয়ে পরবর্তীতে আলোচনা করা হয়েছে) স্ট্রিং গুলোকে একত্রিত করে কিন্তু মাঝখানে নির্ধারিত একটি সেপারেটর ব্যবহার করে। যেমন উপরের উদাহরণে, apple, orange, pineapple এই তিনটি ভ্যালুকে একত্রিত করা হয়েছে কিন্তু তাদের মধ্যে কমা , সেপারেটর ব্যবহার করে।

```
print("Hello ME".replace("ME", "world"))
#prints "Hello world"
```

replace মেথডের মাধ্যমে একটি সাব স্ট্রিং কে খুঁজে সেখানে অন্য কিছু রিপ্লেস করা যায়। যেমন উপরের উদাহরণে - ME রিপ্লেস করে world বসানো হয়েছে।

```
print("This is a sentence.".startswith("This"))  
# prints "True"  
print("This is a sentence.".endswith("sentence."))  
# prints "True"
```

startswith, endswith মেথডের মাধ্যমে কোন একটি ব্যাক্যর শুরু বা শেষ নির্দিষ্ট কোন সাবস্ট্রিং দিয়ে হয়েছে কিনা তা চেক করা যায়।

upper() মেথড স্ট্রিংয়ের সবগুলো ক্যারেক্টারকে uppercase এ পরিবর্তিত করে। একইভাবে

lower() মেথড স্ট্রিংয়ের সবগুলো ক্যারেক্টারকে lowercase এ পরিবর্তিত করে।

```
print("This is a sentence.".upper())  
# prints "THIS IS A SENTENCE."  
print("AN ALL CAPS SENTENCE".lower())  
#prints "an all caps sentence"
```

```
print("a, e, i, o, u".split(", "))  
#prints "['a', 'e', 'i', 'o', 'u']"
```

Write an algorithm for traversing in array

Algorithm for Traversing an Array:

Step 01: Start.

Step 02: [Initialize counter variable.] Set $i = LB$.

Step 03: Repeat for $i = LB$ to UB .

Step 04: Apply process to $arr[i]$.

Step 05: [End of loop.]

Step 06: Stop.

Write an algorithm for inserting element of array.

Algorithm::

Here LA is a Linear Array with N elements and K is a positive integer such that $K \leq N$. This algorithm inserts an element ITEM into the Kth position in LA.

1. [Initialize counter.] Set $J := N$.
2. Repeat Steps 3 and 4 while $J \geq K$.
3. [Move Jth element downward.] Set $LA[J+1] := LA[J]$.
4. [Decrease counter.] Set $J := J - 1$.
[End of Step 2 loop.]
5. [Insert element.] Set $LA[K] := \text{ITEM}$.
6. [Reset N.] Set $N := N + 1$.
7. Exit.

Algorithm to Delete an element from an Array:

Step 01: Start

Step 02: [Initialize counter variable.] Set $i = \text{pos} - 1$

Step 03: Repeat Step 04 and 05 for $i = \text{pos} - 1$ to $i < \text{size}$

Step 04: [Move i th element backward (left).] set $a[i] = a[i+1]$

Step 05: [Increase counter.] Set $i = i + 1$

Step 06: [End of step 03 loop.]

Step 07: [Reset size of the array.] set $\text{size} = \text{size} - 1$

Step 08: Stop

ধন্যবাদ