

পাইথনের চমৎকার ফিচারগুলোর মধ্যে অন্যতম হলো লিস্টের ব্যবহার। লিস্ট প্রোগ্রামিংয়ের একটি গুরুত্বপূর্ণ বিষয়। লিস্ট শব্দের বাংলা অর্থ তালিকা। আমাদের বোধহয় ব্যাখ্যা করার দরকার পড়ে না তালিকা কী জিনিস। পাইথনেও লিস্ট একই কাজ করে। সহজ কথায় লিস্ট হলো কতগুলো আইটেমের একটি তালিকা। অনেক প্রোগ্রামিং ল্যান্ডুয়েজে লিস্ট ডিক্লেয়ার করার সময় বলে দিতে হয় লিস্টের আইটেমগুলোর টাইপ কি হবে, পাইথনে তার দরকার পড়ে না। একটি লিস্টের আইটেমগুলো বিভিন্ন টাইপের হতে পারে।

কিভাবে লিস্ট ডিক্লেয়ার করব। খার্ড ব্রাকেটের ভেতরে কমা দিয়ে একেকটি আইটেম সেপারেট করে দিলেই লিস্ট তৈরি হয়ে যাবে। উদাহরণ :

```
my_list = [1,"a string",45.56]
print my_list[0]
print my_list[1]
print my_list[2]
print my_list
print type(my_list[0])
print type(my_list[1])
print type(my_list[2])
```

প্রথমে কোডগুলো মনোযোগ দিয়ে পড়ুন। বোঝার চেষ্টা করুন এর আউটপুট কী হতে পারে। বরাবরের মতো একটি পাইথন ফাইলে এই কোডগুলো লিখে রান করে দেখুন কী আউটপুট দেখায়। type () ফাংশনটির ব্যবহার আমরা আগেই দেখেছি। আউটপুট দেখে মিলিয়ে নিন আপনি কী আশা করেছিলেন আউটপুট হিসেবে, আর কী এসেছে আউটপুটে। যদি না মিলে বোঝার চেষ্টা করুন কোথায় বুঝতে পারেননি। এই কোড থেকে আমরা কী কী দেখলাম :

* কিভাবে লিস্ট ডিক্লেয়ার করতে হয়।

* লিস্টের আইটেমগুলোর একটি ইনডেক্স ভ্যালু থাকে। এই ইনডেক্স ভ্যালু ব্যবহার করে আমরা n-তম আইটেমের মান বের করতে পারি।

* এই ভ্যালুর মান 0 থেকে শুরু হয়। অর্থাৎ প্রথম আইটেমের ইনডেক্স 0, দ্বিতীয়টির 1, এভাবে n-তম আইটেমের ইনডেক্স (n-1) লিস্ট সম্পর্কে আরও জানার আগে আমরা range () ফাংশনটির ব্যবহার দেখে নেই। এই ফাংশনটির একটি উদাহরণ :

```
print range(0,10)
print range(0,100,10)
```

* এই ফাংশনটি সংখ্যার লিস্ট তৈরি করে। এর সিগনেচার অনেকটা এরকম : range (min, max, step)। এখানে min হলো ন্যূনতম ভ্যালু যেটা থেকে লিস্ট শুরু হবে। max হলো সর্বোচ্চ ভ্যালু যার ঠিক আগের ভ্যালু পর্যন্ত লিস্ট তৈরি হবে। step হলো মধ্যবর্তী ব্যবধান।

* উপরোল্লিখিত কোড রান করলে প্রথমে আমরা পাব 0 থেকে শুরু করে 10-এর ঠিক আগের ভ্যালু অর্থাৎ 9 পর্যন্ত। যদি step না দেয়া হয় তাহলে পাইথন এর ভ্যালু 1 ধরে নেয়। দ্বিতীয়বার আমরা step হিসেবে 10 দিয়েছি। তাই এবার আমরা 0 থেকে শুরু করে প্রতি 10 ঘর পরপর সংখ্যার লিস্ট পাব 90 পর্যন্ত।

* লিস্ট প্র্যাকটিস করার জন্য range () ফাংশনটি ব্যবহার করে দ্রুত লিস্ট তৈরি করুন নেব। আসুন ফেরা যাক লিস্টে। আমরা দেখেছি কিভাবে ইনডেক্স ব্যবহার করে আমরা লিস্টের আইটেমগুলো অ্যাক্সেস করেছি। ধরুন আমাদের লিস্টের সব ডাটা লাগবে না, আমরা একটি নির্দিষ্ট রেঞ্জ নিয়ে কাজ করতে চাই। পাইথন আমাদের সেই সুবিধা দেয় (যা অন্য অনেক ল্যান্ডুয়েজে পাওয়া যায় না)। আসুন দেখি কিভাবে :

এই উদাহরণটি নিজেরা চেষ্টা করার জন্য প্রথমেই একটি লিস্ট তৈরি করে নেই।

```
sl = range(1,11) # A list containing the
integers from 1 to 10
```

পাইথন লিস্ট ও ফাংশন

মৃগাল কান্তি রায় দীপ

পাইথন প্রোগ্রামিং পর্ব-২

আসুন এবার লিস্ট নিয়ে নড়াচড়া করা যাক :
list1to5 = sl[0:5]
print list1to5
list2to7 = sl[1:7]
print list2to7

এই কোড রান করলে দেখা যাবে list1to5 একটি লিস্ট, যার ভ্যালু 1 থেকে 5। sl[0:5] বলতে বোঝানো হয় sl নামের লিস্টের 0-তম আইটেম থেকে শুরু করে 5-তম আইটেমের আগের আইটেম পর্যন্ত আইটেমগুলো নিয়ে তৈরি একটি লিস্ট। এবার নিজে নিজেই বোঝার চেষ্টা করুন list2to7-এর ভ্যালু কি হতে পারে এবং কেন।

এবার নিজে কিছু কাজ করুন :

3 থেকে 9 পর্যন্ত লিস্ট বোঝাতে আমরা কি লিখব?

sl[:5]-এর ভ্যালু কত হবে?

sl[4:]-এর ভ্যালু কত হবে?

sl[:]-এর ভ্যালু কত হবে? কেন?

আমরা range ফাংশনে step-এর ব্যবহার দেখেছিলাম। লিস্টের ক্ষেত্রেও step ব্যবহার করা যায়। যেমন :

```
print sl[0:10:2]
```

```
print sl[0:9:3]
```

অর্থাৎ শেষে আরেকটি কোলন দিয়ে আমরা step ভ্যালুটি নির্দেশ করে থাকি। তাই প্রথম ক্ষেত্রে আমরা 0-তম আইটেম থেকে শুরু করে 2টি আইটেম বাদ দিয়ে দিয়ে 10-তম আইটেমের আগের আইটেম পর্যন্ত যে আইটেমগুলো আছে সেগুলোর লিস্ট পাব। নিজে নিজে বোঝার চেষ্টা করি দ্বিতীয় ক্ষেত্রে কি ঘটছে।

যেকোনো ভ্যালুর আগে মাইনাস চিহ্ন দিলে তার অবস্থান বিপরীত দিক থেকে বিবেচনা করা হয়। তাই শেষ দিক থেকে 5-তম আইটেমের ভ্যালু হবে sl[-5]। এভাবে শেষ দিক থেকে 2-তম আইটেমের আগ পর্যন্ত আইটেমগুলোর লিস্ট পাব : sl[:-2]] step। এর ভ্যালু নেগেটিভ হলে গণনা উল্টোদিকে হবে। যেমন শেষ দিক থেকে 2-তম আইটেমের আগের আইটেম থেকে শুরু করে 3-তম আইটেম পর্যন্ত আইটেমগুলো 2

ধাপ করে পেছালে আমরা যে লিস্টটি পাব তার জন্য আমাদেরকে লিখতে হবে : sl[-2:3:-2]

এভাবে নিজেরা ইচ্ছেমতো লিস্ট তৈরি করে তার বিভিন্ন অংশ আলাদা করার চেষ্টা করি। প্রথমবার দেখে লিস্টের সিনট্যাক্স খুব জটিল মনে হতে পারে। কিন্তু কিছুদিন অনুশীলন করলেই ঠিক হয়ে যাবে।

পাইথনে ফাংশন

আমাদের প্রোগ্রামের যে অংশগুলো বারবার আসে সেগুলোকে আমরা পুনরায় ব্যবহারযোগ্য একক (reusable unit) হিসেবে ব্যবহার করতে পারি ফাংশনের সাহায্যে। গণিতে যেমন দেখেছি

কোনো ফাংশন একটি ইনপুট নিয়ে সেটার ওপর বিভিন্ন ধরনের ম্যাথ করে আউটপুট দেয়, প্রোগ্রামিংয়েও সেই একই ব্যাপার ঘটে। আপনি এক বা একাধিক প্যারামিটার পাস করবেন একটি ফাংশনে, ফাংশনটি প্রসেস করে আপনাকে আউটপুট রিটার্ন করবে। তবে প্রোগ্রামিংয়ের ক্ষেত্রে সবসময় যে ইনপুট থাকতে হবে বা আউটপুট দিতে হবে এমন কোনো কথা নেই।

একটি ফাংশন আসলে কিছু স্টেটমেন্টের সঙ্কলন। যখনই কোনো ফাংশন কল করা হয়, তখন এই ফাংশনের ভেতরে থাকা স্টেটমেন্টগুলো এক্সিকিউট করা হয়। পাইথনে আমরা ফাংশন ডিক্লেয়ার করার জন্য def কিওয়ার্ডটি ব্যবহার করি। আসুন দেখে নিই একটি ফাংশন :

```
def hello():
    print "Hello World!"
    return 0
```

প্রথমে আমরা def কিওয়ার্ডটি লিখেছি। তারপর ফাংশনের নাম "hello" এবং তারপর ()। যদি আমরা ফাংশনটিতে কোনো ইনপুট দিতে চাই সে ক্ষেত্রে প্যারামিটারগুলো এই ()-এর মধ্যে কমা দিয়ে আলাদা করে লিখতে হবে। আমরা দেখতে পাচ্ছি, এই ফাংশনে আমরা কোনো ইনপুট দিচ্ছি না। ফাংশনটি "Hello world!" প্রিন্ট করবে। সি প্রোগ্রামিংয়ের সাথে মিল রেখে (এবং রিটার্ন স্টেটমেন্টের ব্যবহার দেখানোর জন্য) আমরা 0 রিটার্ন করছি। আসলে এই স্টেটমেন্টের কোনো দরকার ছিল না।

এবার আসুন দেখা যাক পাইথনে কিভাবে আমরা ফাংশন প্যারামিটার পাস করব।

```
def sayHello (name):
    print "Hello, "+name+" !"
```

এই ফাংশনটিকে কল করুন এভাবে : sayHello ("Computer Jagat")

রান শেষে "Hello Computer Jagat" লেখা দেখতে পারবেন। অর্থাৎ sayHello-এর পর ""-এর মধ্যে যা বসাবেন তাকে কল করবে `sayHello("Computer Jagat")`

ফিডব্যাক : mkrdip@yahoo.com