

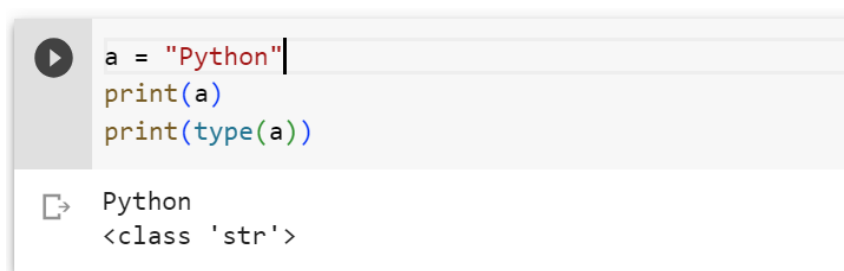
1. দিন - ১৩ (Monisha)

1.1. স্ট্রিং

- 1.1.1. upper() ফাংশনের মাধ্যমে আপার কেসে পরিণত করা
- 1.1.2. lower() ফাংশনের মাধ্যমে লোয়ার কেসে পরিণত করা
- 1.1.3. replace() ফাংশনের মাধ্যমে স্ট্রিং রিপ্লেস করা
- 1.1.4. split() ফাংশনের মাধ্যমে স্ট্রিংকে স্প্লিট করা
- 1.1.5. find() ফাংশনের মাধ্যমে সাব-স্ট্রিং-এর ইন্ডেক্স বের করা
- 1.1.6. count() ফাংশনের মাধ্যমে একটি সাব-স্ট্রিং কত বার আছে তা বের করা
- 1.1.7. String Concatenation
- 1.1.8. অনুশীলনঃ ফার্স্ট নেম এবং লাস্ট নেম নিয়ে একসাথে পুরো নাম প্রিন্ট করা।

1.1 স্ট্রিং:

স্ট্রিং python এর একটি ডাটা টাইপ। শূন্য বা তারচেয়ে বেশি সংখ্যক অক্ষর বা চিহ্ন দিয়ে স্ট্রিং তৈরি হয়। স্ট্রিং এর শুরু এবং শেষ হয় ডাবল অথবা single কোটেশন এর মাধ্যমে।



```
a = "Python"
print(a)
print(type(a))
```

Python
<class 'str'>

1.1.1 upper() ফাংশনের মাধ্যমে আপার কেসে পরিণত করা

স্ট্রিং এর সকল অক্ষর বড় হাতের লেখার জন্য upper() ফাংশন ব্যবহার করা হয়। python এর একটি built-in ফাংশন যা স্ট্রিং এর জন্য ব্যবহার করা হয়।

```
print(a.upper())
```

PYTHON

1.1.2. lower() ফাংশনের মাধ্যমে লোয়ার কেসে পরিণত করা

স্ট্রিং এর সকল অক্ষর বড় হাতের লেখার জন্য lower() ফাংশন ব্যবহার করা হয়। এটি python এর একটি built-in ফাংশন যা স্ট্রিং এর জন্য ব্যবহার করা হয়।

```
print(a.lower())
```

python

1.1.3. replace() ফাংশনের মাধ্যমে স্ট্রিং রিপ্লেস করা

যে কোন স্ট্রিং এর মান বা অক্ষর কে অন্য কোন মান বা অক্ষর দিয়ে স্থানান্তর করা হয় replace() ফাংশন ব্যবহার করে। replace() ফাংশন দুইটি আর্গুমেন্ট গ্রহণ করে, প্রথম আর্গুমেন্ট হচ্ছে যে মান বা অক্ষর পরিবর্তন করা হবে, ২য় আর্গুমেন্ট হচ্ছে যে মান বা অক্ষর দ্বারা পরিবর্তন করা হবে।

```
b="Fake Python"
print(b)
|
#Using replace function:
b_new=b.replace('Fake','Real')
print(b_new)
```

Fake Python
Real Python

1.1.4. split() ফাংশনের মাধ্যমে স্ট্রিংকে স্প্লিট করা

split() ফাংশন স্ট্রিং এর সকল মান বা অক্ষর কে লিস্টের একক মান এ পরিণত করে ।

```
print(b.split())
```

```
['Fake', 'Python']
```

এখানে separator নির্দিষ্ট করে দেয়া যেতে পারে । ডিফল্ট separator - white space.

```
c="Hi I am here#to#learn#Python"
print(c.split('#'))
```

```
['Hi I am here', 'to', 'learn', 'Python']
```

1.1.5. find() ফাংশনের মাধ্যমে সাব-স্ট্রিং-এর ইন্ডেক্স বের করা

python এ find() মেথড ব্যবহার করে স্ট্রিং এর সাবস্ট্রিং এর ইন্ডেক্স নির্দেশ করা হয় ।

```
c="Hi I am here to learn Python"
print(c.find('Hi'))
```

```
0
```

স্ট্রিং এর সাবস্ট্রিং অনুপস্থিত থাকলে ইন্ডেক্স হবে -1 ।

```
▶ c="Hi I am X.I am here to learn Python"  
print(c.find('C++'))
```

-1

1.1.6. count() ফাংশনের মাধ্যমে একটি সাব-স্ট্রিং কত বার আছে তা বের করা

একটি স্ট্রিং এ একটি ভ্যালু কতবার উপস্থিত তা নির্ণয় এর জন্য count() ফাংশন ব্যবহার করা হয়। ফাংশনটি একটি ইনটিজার ভ্যালু return করে।

```
▶ c="Hi I am X.I am here to learn Python"  
print(c.count('I'))
```

↗ 2

স্ট্রিং এ অনুপস্থিত ভ্যালুর জন্য 0 return করবে।

```
▶ c="Hi I am X.I am here to learn Python"  
print(c.count('C++'))
```

0

1.1.7. String Concatenation

স্ট্রিং concatenation() মেথড ব্যবহার করে ২ টি স্ট্রিং ভ্যালু যোগ করা হয়। এখানে "+" চিহ্ন ব্যবহার করে একটি স্ট্রিং ভ্যালু অন্য স্ট্রিং ভ্যালুর সাথে যোগ করা হয়।



```
a = " Python"
b = " is"
c = " awesome"

result = a+b+c
print(result)
```



```
Python is awesome
```

অনুশীলন: ফার্স্ট নেম এবং লাস্ট নেম নিয়ে একসাথে পুরো নাম প্রিন্ট করা।

স্ট্রিং concatenation() মেথড ব্যবহার করে ফার্স্ট নেম এবং লাস্ট নেম নিয়ে একসাথে পুরো নাম প্রিন্ট করা সম্ভব

|

```
[29] first_name = "Sheikh"
      last_name = "\tHasina"

      full_name = first_name + last_name
      print(full_name)
```

```
Sheikh Hasina
```