Inputs and Outputs Basics

Take Input From User

`input()` allows flexibility to take the input from the user. Reads a line of input as a string.

Code

PYTHON

```python
username = input()
print(username)
```

Input

```
Ajay
```

Output

```
Ajay
```

Working with Strings

String Concatenation

Joining strings together is called string concatenation.

Code

PYTHON

```python
a = "Hello" + " " + "World"
print(a)
```

Output

```
Hello World
```

Concatenation Errors

String Concatenation is possible only with strings.

Code

PYTHON

```python
a = "*" + 10
print(a)
```

Output

```
File "main.py", line 1
    a = "*" + 10
    ^
TypeError:
can only concatenate str (not "int") to str
```

String Repetition

* operator is used for repeating strings any number of times as required.

Code

PYTHON

```python
a = "*" * 10
print(a)
```

Output

```
**********
```

Code

PYTHON

```python
s = "Python"
s = ("* " * 3) + s + (" *" * 3)
print(s)
```

Output

```
* * * Python * * *
```

Length of String

`len()` returns the number of characters in a given string.

Code

PYTHON

```python
username = input()
length = len(username)
print(length)
```

Input

```
Ravi
```

Output

```
4
```

String Indexing

We can access an individual character in a string using their positions (which start from 0) .
These positions are also called as index.

Code

PYTHON

```python
username = "Ravi"
first_letter = username[0]
print(first_letter)
```

Output

```
R
```

IndexError

Attempting to use an index that is too large will result in an error:

Code

PYTHON

```python
username = "Ravi"
print(username[4])
```

Output

```
IndexError: string index out of range
```

Submit Feedback