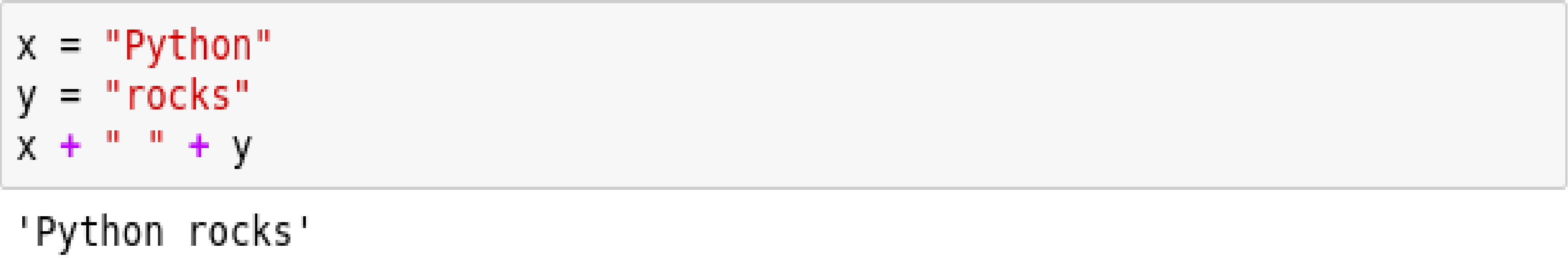
Data Types in python Language

Strings

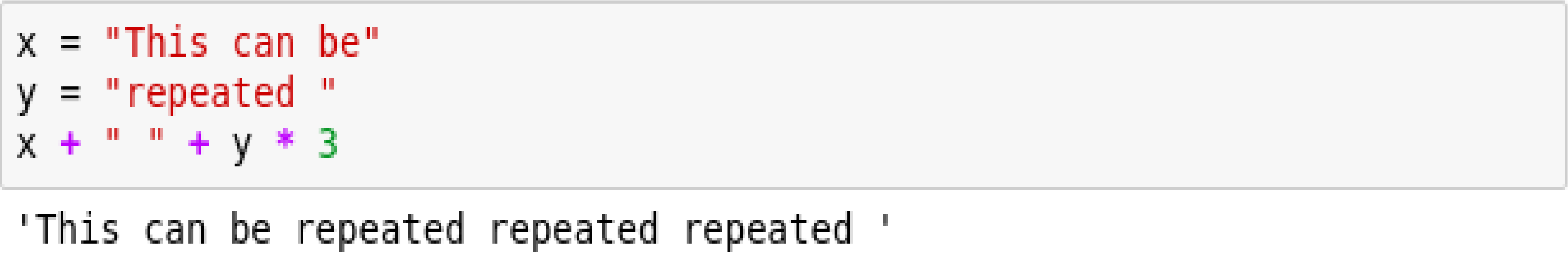
Strings

Strings are collection numbers, letters or symbols enclosed in quotes



```
x = "Python"
y = "rocks"
x + " " + y
```

```
'Python rocks'
```



```
x = "This can be"
y = "repeated "
x + " " + y * 3
```

```
'This can be repeated repeated repeated '
```

Every element in a string has an index number! This is String Indexing

Strings Slicing

Slicing of a string is done whenever we want to cut out a piece of data from a string. Slicing is done using the square brackets **[ ]**

Such that if A = **“Hello Jake and Welcome!”**

- **A[:] = “Hello Jake and Welcome!”**
- **A[1:] = “ello Jake and Welcome!”**
- **A[0:22:1] = “Hello Jake and Welcome!”**
- **A[0:22:2] = “HloJk n ecm”**

Strings Methods

remark = “Mark is a great man indeed”

- len (remark)
- remark.upper()
- remark.lower()
- remark.swapcase()
- remark.title()
- remark.count(‘e’)
- remark.replace('a’,’ b’)
- remark.index(‘g’)
- remark.split()

Lists

Lists

- A list object is an ordered collection of one or more data items which can be of different data types. List can be defined in square brackets.
- Every element in a list is indexed and has an index number.

For example:

```
info = [1, 2, 3, 4, ‘Hello’, True, ‘John’]
```

Lists Slicing

Similar to the string data type, list slicing is done cut out a chunk of data from a set of data in a list.

Such that if names = [“Jake”, “Ben”, “Coy”, “Tom”]

Then names[0] = “Jake”

names[0:3] = [“Jake”, “Ben”, “Coy”]

names[0:3:2] = [“Jake”, “Coy”]

Lists Methods

grades = [“first”, “second”, “third”, “fourth”, “fifth”]

- len (grades)
- grades.sort()
- grades.append(“second”)
- grades.insert(2, “ninth”)
- grades.pop(2)
- grades[1] = “secondth”
- grades.count(“second”)
- grades.index(“fourth”)

Tuples

Tuples

- Tuples are collection of ordered data items which can be of different data types. The elements are unchangeable and they are defined in parenthesis. Every element in a tuple also has an index number.

Given that names = (‘John’, ‘Sam’, ’Sarah’, ’Joy’)

names[0] = ‘John’

names[2] = ‘Sarah’

Tuples Slicing

Given that prices = (29.25, 30, 41.4, 57.0, 30, 56.7, 99)

Then prices[:] = (29.25, 30, 41.4, 57.0, 30, 56.7, 99)

prices[1:] = (30, 41.4, 57.0, 30, 56.7, 99)

prices[1:3] = (30, 41.4)

prices[0:6:2] = (30, 41.4, 57.0, 30, 56.7, 99

Tuples Methods

prices = (29, 30, 41, 57, 30, 56.7, 30)

- len (prices)
- grades.sort() ✗
- grades.append(“ten”) ✗
- grades.insert(2, “six”) ✗
- prices.index(57)
- grades.pop(2) ✗
- grades[1] = “secondth” ✗
- grades.count(30)

Dictionary

Dictionary

Dictionary is a collection of key and value pairs where the keys and values can be of any data type. Dictionaries are made up of KEY: VALUE pairs. In Python, lists and tuples are organized and accessed based on index position. Dictionaries in Python are organized and accessed using keys and values.

```
age = {“key”: value, “key”: value, “key”: value}

grades = {“first”:45, “second”: 56, “third”:67}
```

Dictionary Methods

grades = {“first”:45, “second”: 56, “third”:67}

- len (grade)
- grades[“first”]
- grades.keys()
- grades.values()
- grades.pop(“third”)
- grades[“third”] = 85
- grades.get(“third”)
- grades[“fourth”] = 78