

# **Stings, Type Conversion, Numpy Essentials**

**2018-10-10**

# Basic String Operations

# Basic String Operations

- Compare beginning with template:  
**"This is a string".startswith("string") => False**  
**"This is a string".startswith("This") => True**
- Compare end with template:  
**"This is a string".endswith("string") => True**  
**"This is a string".endswith("This") => False**
- Manipulate content:  
**s = "This is a string"**  
**s[5:7] = "IS"**  
**print(s) => "This IS a string"**

# String Formatting

# String Formatting

- Python 3 introduced a convenient syntax for string creation:

**`f"The value of variable1 is: {variable1}"`**

- Formatting the display of floating point values

**`$ name = "Hemoglobin"`**

**`$ mass = 64458`**

**`$ s = f"{name} has a mass of about {mass/1000:.1f} kDa."`**

=> Hemoglobin has a mass of about 64.5 kDa

**`$ s = f"{mass/1000:10.3}"`**

=> 0.0643 (10 characters and 3 significant digits)

# Type Conversion

# Type Conversion

- Variables can be casted to compatible types  
**float("1.23434")** => 1.23434  
**int("0.932442")** => 0 or **int(round(float("0.932442")))** => 1  
**list("abc")** => ["a", "b", "c"]  
**tuple([1,2,3])** => (1, 2, 3)
- You can check the type of a variable by calling **type(variable)**  
**type(1.2334)** => float  
**type([1,2,3])** => list

# Numpy



# Python's Math Library: Numpy

- Importing and using numpy

```
$ import numpy
```

```
$ numpy.sin(10) => -0.54402111088936989
```

- Contains basic mathematical functions: **sin, cos, sqrt, abs, exp, ...**

- **Numpy arrays:**

- Similar to arrays but all elements have to have the same type

```
A = numpy.array([1, 2, 3])
```

```
B = numpy.array([2, 2, 2])
```

- Fast calculation of matrix operations

```
A + B => array([3, 4, 5])
```

```
A * B => array([2, 4, 6])
```

```
A ** B => array([1, 4, 9])
```

```
sqrt(A) => array([ 1.         ,  1.41421356,  1.73205081])
```