

# ENGG1003 - Thursday Week 2

Data types, and introduction to arrays

Steve Weller

University of Newcastle

4 March, 2021

# Lecture overview

## 1 variables and data types §2.2

- ▶ principles
- ▶ live demo

## 2 arrays in Python §2.3

- ▶ principles
- ▶ live demo

# 1) variables and data types

- variable names – make them descriptive
- camelCase
- snake\_case
- matter of preference/style/taste
  - ▶ experiment, find what works best for you

# Assignment

- $x = 2$
- $x = x + 4$
- $x += 4$  is short for  $x = x + 4$

# The type of a variable

- types seen so far:
  - ▶ `int`
  - ▶ `float`
  - ▶ `str`
  - ▶ another (final?) type will be introduced next lecture
- explain “floating point” terminology—think of float as real number (fractional part, not an integer)
- mention “objects” only in passing

# The type of a variable (ctd.)

# Type function

- §2.2.4 and §2.2.5
- built-in function `type`
- type conversion
- automatic type conversion

# Operator precedence—OMIT??



Division—quotient and  
remainder—OMIT??

# Live demo of variables and data types

## 2) Arrays in Python

- simple arrays used in Monday's lecture
  - ▶ height of a ball was computed for each millisecond
  - ▶ time stored in array `t`
  - ▶ height stored in array `y`
- arrays we use in this course are imported from `numpy` library
- for each array, all array *elements* must be of the same type
  - ▶ eg: all `int`, or all `float`

# Array creation and array elements



- array *index* used to identify array elements
  - ▶ Python uses *zero-based indexing*
  - ▶ indices start at zero: 0, 1, 2, ...
- four common ways of creating arrays:
  - ▶ `linspace`
  - ▶ `zeros`
  - ▶ `array`
  - ▶ `copy`

# #1 Linspace

- have seen `linspace` function already
- `t = np.linspace(0, 1, 1001)` creates 1001 coordinates between 0 and 1, inclusive at both ends

0	1	2				1000
0	0.001	0.002	...			0.998 0.999 1

- `t` is the name of the array
- array indices are `0, 1, 2, ...`
- array elements: `t[0], t[1], t[2], ..., t[1000]`

- pp47-48 screenshot
- p47: create array with linspace
- p48 check array type, and array element type

## #2 Zeros function

```
In [1]: from numpy import zeros

In [2]: x = zeros(3, int)           # get array with integer zeros

In [3]: x
Out[3]: array([ 0,  0,  0])

In [4]: y = zeros(3)               # get array with floating point zeros

In [5]: y
Out[5]: array([ 0.,  0.,  0.])

In [6]: y[0] = 0.0;  y[1] = 1.0;  y[2] = 2.0    # overwrite

In [7]: y
Out[7]: array([ 0.,  1.,  2.])

In [8]: len(y)
Out[8]: 3
```

- array of int or array of float
  - ▶ but cannot mix int and float type!
- `len(y)` is *length* of array `y`

## #3 Array function

- create an array of zeros



# Index out of bounds

- show error when access out of bounds—use PyCharm screenshot
- contrast with C

## #4 Copying an array

- **BE VERY CAREFUL** with naive/obvious copy method
- `copy` function creates new array and copies values
  - ▶ use this method!

# Slicing an array

- needs a figure showing boxes

# Live demo of Python arrays