Week 3 Tutorial

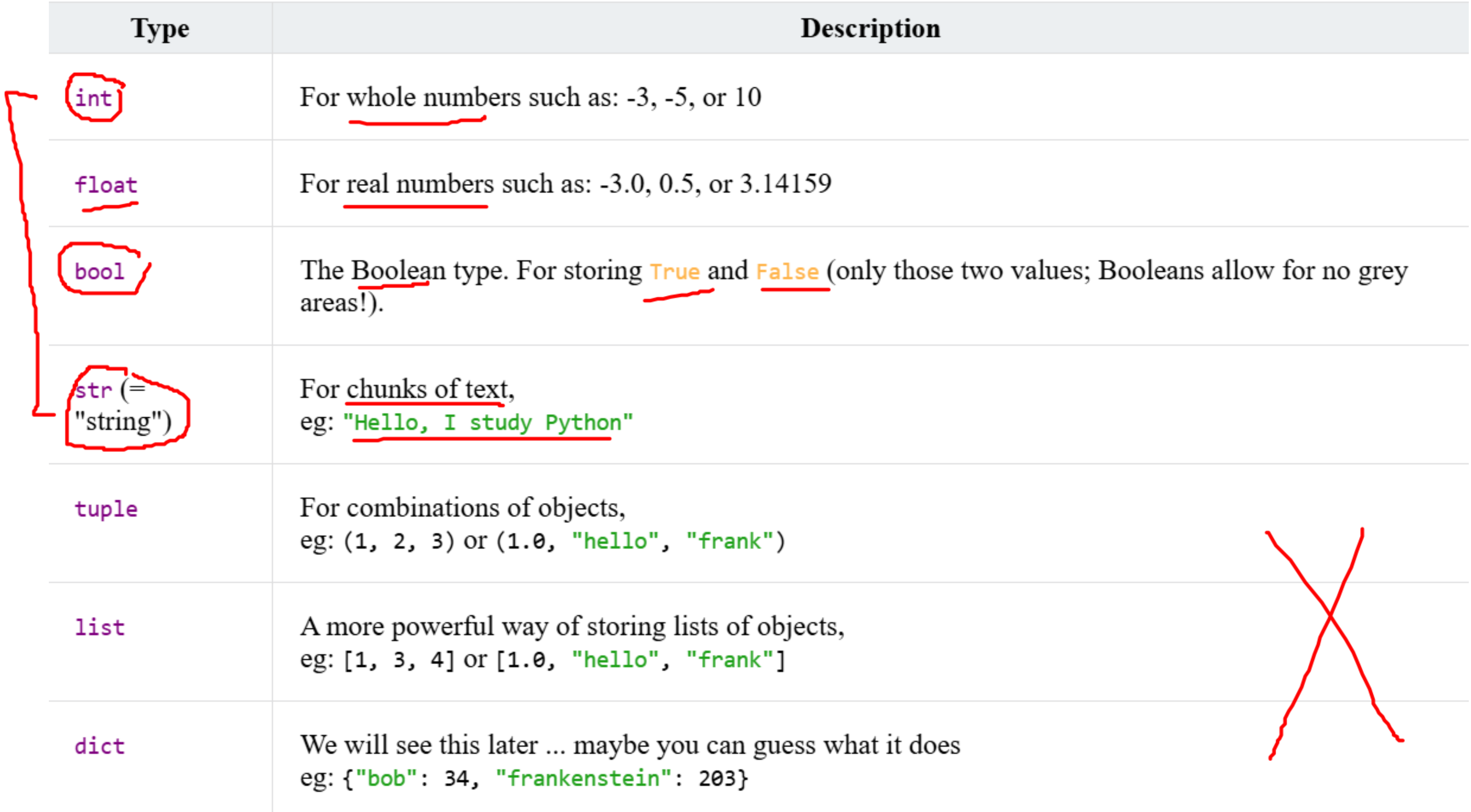
COMP10001 – Foundations of Computing

Semester 2, 2025

Clement Chau

- Python basics and types
- Variables and Strings
- Conditionals and Sequences

Revision, data types!

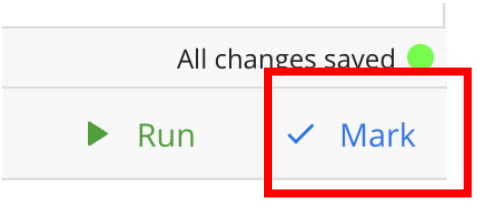


Type	Description
`int`	For whole numbers such as: -3, -5, or 10
`float`	For real numbers such as: -3.0, 0.5, or 3.14159
`bool`	The Boolean type. For storing `True` and `False` (only those two values; Booleans allow for no grey areas!).
`str` (= "string")	For chunks of text, eg: `"Hello, I study Python"`
`tuple`	For combinations of objects, eg: `(1, 2, 3)` or `(1.0, "hello", "frank")`
`list`	A more powerful way of storing lists of objects, eg: `[1, 3, 4]` or `[1.0, "hello", "frank"]`
`dict`	We will see this later ... maybe you can guess what it does eg: `{"bob": 34, "frankenstein": 203}`

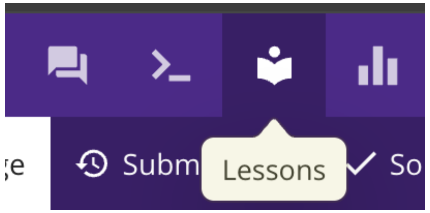
Ed Lessons

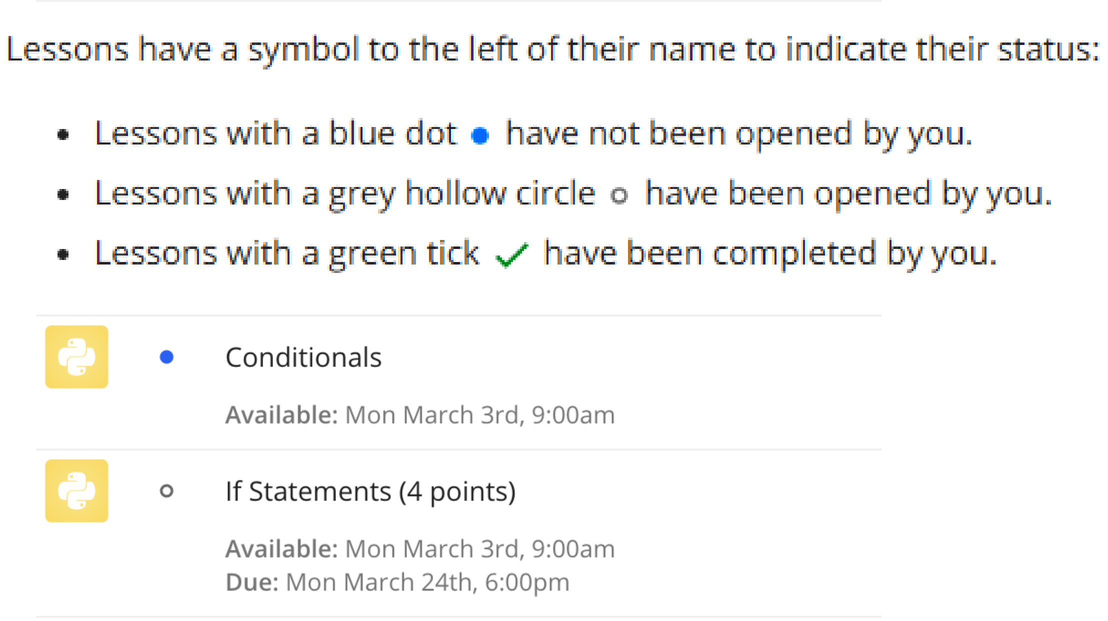
<u>Canvas > COMP10001_2025_SM2 > Ed Discussion > Lessons</u>

- Worksheets 1 and 2 due : Monday 11 August 6 pm
- Worksheets 3, 4, and 5 due : Monday 18 August 6 pm



Remember to press 'Mark' !



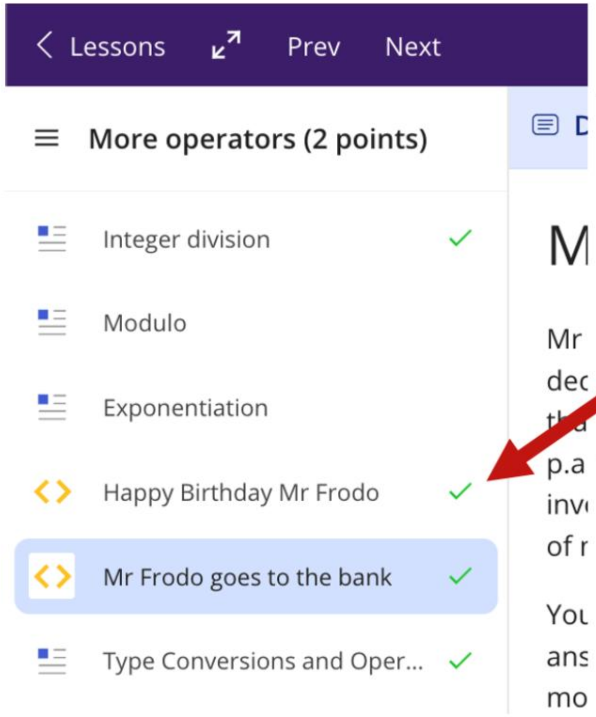


Lessons have a symbol to the left of their name to indicate their status:

- Lessons with a blue dot ● have not been opened by you.
- Lessons with a grey hollow circle ○ have been opened by you.
- Lessons with a green tick ✓ have been completed by you.

● Conditionals
Available: Mon March 3rd, 9:00am

○ If Statements (4 points)
Available: Mon March 3rd, 9:00am
Due: Mon March 24th, 6:00pm



More operators (2 points)

- Integer division ✓
- Modulo
- Exponentiation
- Happy Birthday Mr Frodo ✓
- Mr Frodo goes to the bank ✓
- Type Conversions and Oper... ✓

Every Code Challenge, or Quiz with a Green Tick is worth 1 point. If you do not have a green tick (all tests passing/fully correct) you will not get 1 point. Reattempting the worksheet will not remove your green tick.

Agenda

1. Week 3 Discussion – **Tutorial sheets** (~ 55 mins)
2. Q&A for **Ed worksheet 1 - 5** (~ 55 mins)

At the end of this workshop, you should:

- *be familiar with **Types, Strings, Conditionals, and Sequences.***

*The tutorial sheets correspond to **Ed Worksheets 1 - 4.***

Tutorial Sheet – Week 3

In groups of 2-3, work through the questions on the Week 3 tutorial sheets.

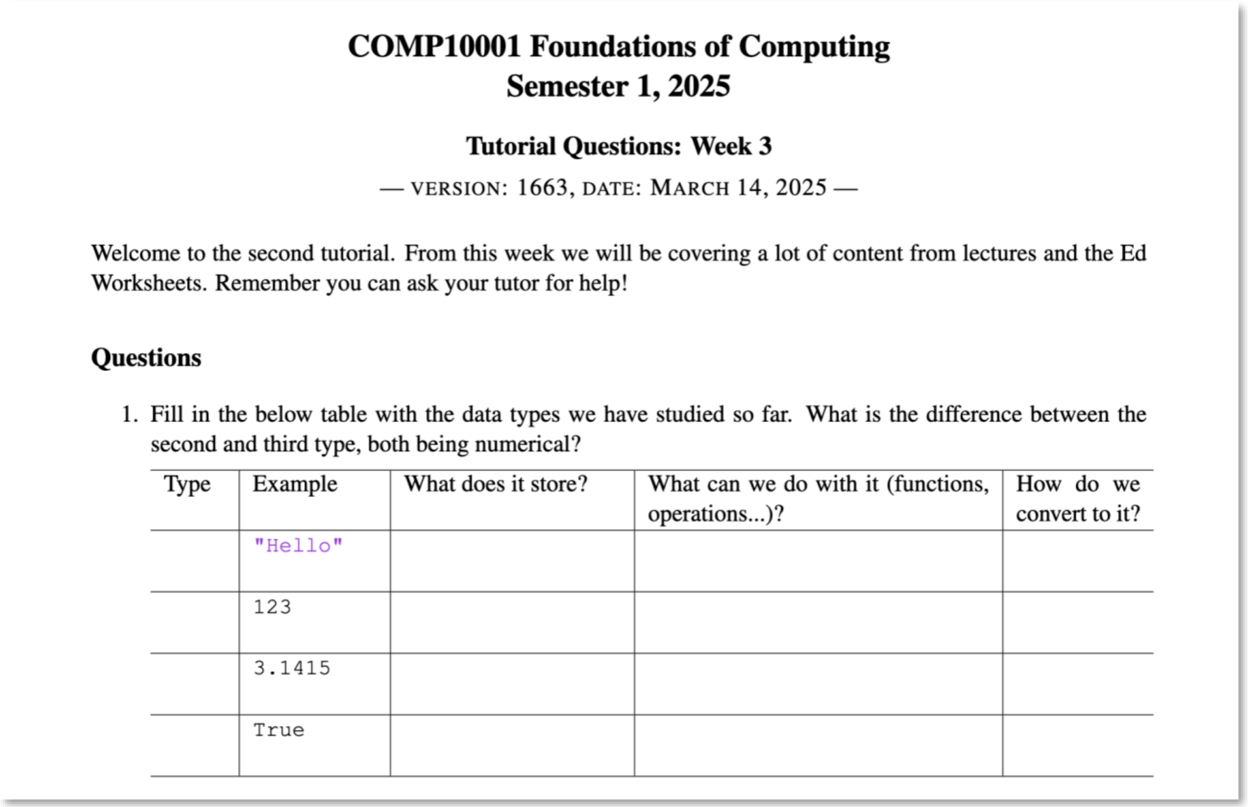
Please use pen & paper.

We will review questions from Q1 and Q4.

Then, move on selected questions

between Q6 – Q11.

Tutorial solution will be released Friday night.



COMP10001 Foundations of Computing
Semester 1, 2025

Tutorial Questions: Week 3

— VERSION: 1663, DATE: MARCH 14, 2025 —

Welcome to the second tutorial. From this week we will be covering a lot of content from lectures and the Ed Worksheets. Remember you can ask your tutor for help!

Questions

1. Fill in the below table with the data types we have studied so far. What is the difference between the second and third type, both being numerical?

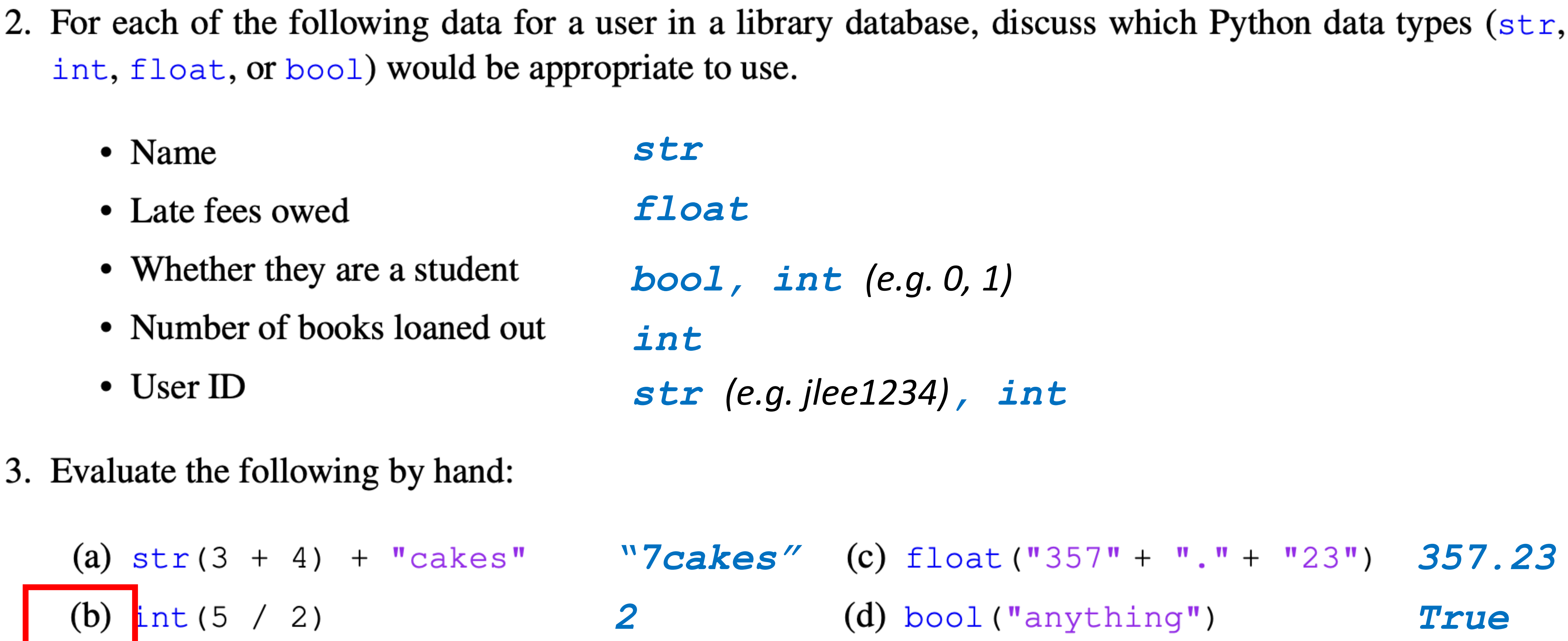
Type	Example	What does it store?	What can we do with it (functions, operations...)?	How do we convert to it?
	`"Hello"`			
	`123`			
	`3.1415`			
	`True`			

Tutorial Sheet – Week 3

1. Fill in the below table with the data types we have studied so far. What is the difference between the second and third type, both being numerical?

Type	Example	What does it store?	What can we do with it (functions, operations...)?	How do we convert to it?
str	"Hello"	A sequence of characters	len(), input(), print(), slicing, indexing, .lower()	str()
int	123	A whole number (integer)	Arithmetic operations, counting & numbering, indexing and slicing	int()
float	3.1415	A number containing a fractional part	Arithmetic operations, mathematics & real world measurements	float()
bool	True	A truth value (T/F)	result of truth tests, used in conditional statements	bool()

Tutorial Sheet – Week 3



2. For each of the following data for a user in a library database, discuss which Python data types (`str`, `int`, `float`, or `bool`) would be appropriate to use.

• Name	***str***
• Late fees owed	***float***
• Whether they are a student	***bool, int*** *(e.g. 0, 1)*
• Number of books loaned out	***int***
• User ID	***str*** *(e.g. jlee1234)****, int***

3. Evaluate the following by hand:

(a)	`str(3 + 4) + "cakes"`	***"7cakes"***	(c)	`float("357" + "." + "23")`	***357.23***
(b)	`int(5 / 2)`	***2***	(d)	`bool("anything")`	***True*** *non-empty str*

Division operator:

Ed lessons > Worksheet 2 > More operators > Integer division [Link]

Tutorial Sheet – Week 3

4. Evaluate the following by hand, given the assignments `a = 1, b = 2, c = 2.0`:

(a) `a / a` **`1.0`** *(not 1)*

(b) `b + b` **`4`**

(c) `b + c` **`4.0`**

(d) `a / b` **`0.5`**

(e) `a // b` **`0`** *(floor division)*

(f) `a % b` **`1`** *(1÷2=0 (quotient) with a remainder of 1)*

(g) `a + b / c` **`2.0`**

(h) `(a + b) / c` **`1.5`**

5. What is the output of the following? Why?

(a) `123 + 123` **`246`** *(Arithmetic addition)*

(b) `"123" + "123"` **`“123123”`** *(String concatenation)*

(c) `"123" + 123` **`TypeError`**

(d) `3 * 4` **`12`** *(Arithmetic multiplication)*

(e) `"3" * 4` **`“3333”`** *(String multiplication)*

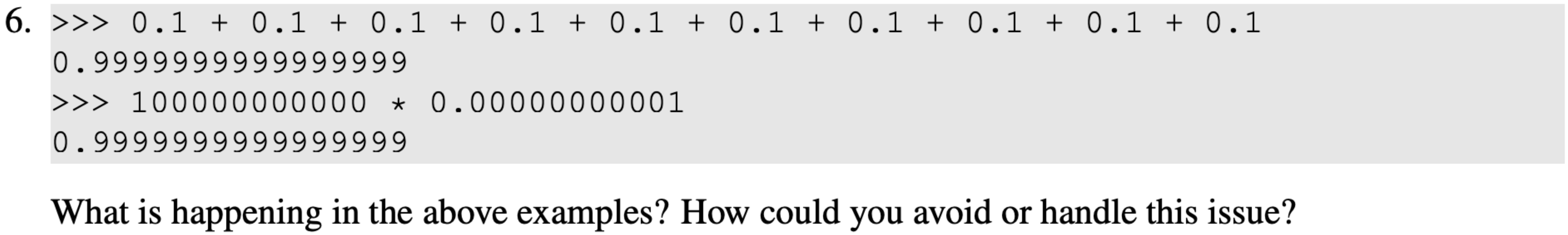
(f) `"3" * "4"` **`TypeError`**

***Modulo (% :** computes the **remainder** of an integer division)*

Ed lessons > Worksheet 2 > More operators > Modulo [Link]

Tutorial Sheet – Week 3

In groups of 2-3, work through the questions from Q6 – Q11.



6.
```
>>> 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1 + 0.1
0.9999999999999999
>>> 100000000000 * 0.00000000001
0.9999999999999999
```

What is happening in the above examples? How could you avoid or handle this issue?

base 10 (0,1,2,3,…,9) <> base 2 (0,1)
Floating-point numbers are represented in computer hardware as base 2 (binary) fractions.
Unfortunately, most decimal fractions cannot be represented exactly as binary fractions.

One way to avoid it is to use `int` *instead of* `float`
Or use `round()` *function*

https://docs.python.org/3/tutorial/floatingpoint.html
https://docs.python.org/3/library/functions.html#round

Tutorial Sheet – Week 3

7. Evaluate the following truth expressions:

(a) `True or False` ***True***

(b) `True and False` ***False***

(c) `False and not False or True` ***True***

(d) `False and (not False or True)` ***False***

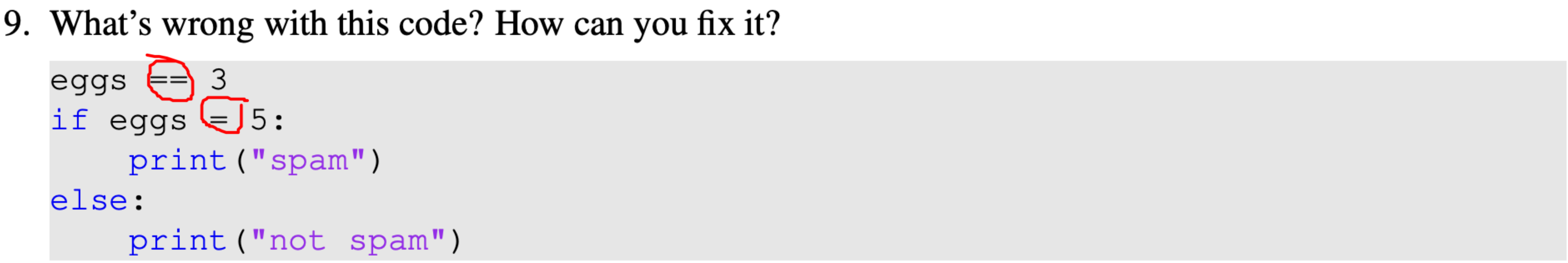
Logical Operators : *Ed lessons > Worksheet 3 > Conditionals > Logical Operators: Combining Truth [Link]*

8. For each of the following if statements, give an example of a value for `var` which will trigger it and one which will not.

	Trigger	***Not trigger***
(a) `if 10 > var >= 5:`	5,6,7,8,9	*(e.g.)* 10
(b) `if var[0] == "A" and var[-1] == "e":`	*(e.g.)* "Apple"	
(c) `if var in ("VIC", "NSW", "ACT"):`	*(e.g.)* "VIC"	*(e.g.)* "vic"
(d) `if var:`	*This condition will convert `var` into a boolean value.* *non-zero/non-empty*	

Conditional Blocks : *Ed lessons > Worksheet 3 > If Statements > Conditional Blocks [Link]*

Tutorial Sheet – Week 3



9. What’s wrong with this code? How can you fix it?

```
eggs == 3
if eggs = 5:
    print("spam")
else:
    print("not spam")
```

This programmer has confused the assignment (=) and equality (==) operators.

assignment(=) : does not evaluate to anything

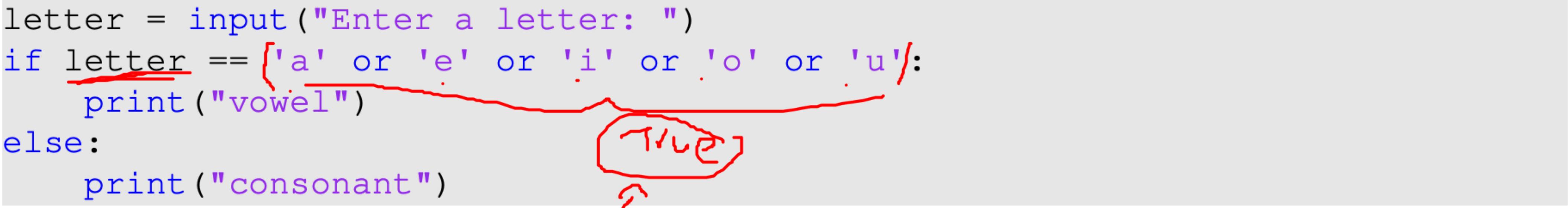
equality (==) : Relational Operators (equal to)

```
eggs = 3 # value assign
if eggs == 5: #if assigned eggs is equal to 5
```

__A Common Mistake:__ Ed lessons > Worksheet 3 > Conditionals > A Common Mistake [Link]

Tutorial Sheet – Week 3

10. What’s wrong with this code? How can you fix it?



```
letter = input("Enter a letter: ")
if letter == 'a' or 'e' or 'i' or 'o' or 'u':
    print("vowel")
else:
    print("consonant")
```

True

Logical operators separate conditions,
so the logical statement
`letter == 'a’ or True or True or True or True`
will always evaluate to `True`

```
if letter in 'aeiou’:
```

```
if letter in ('a', 'e', 'i', 'o', 'u'):
```

['a', 'e']

Tutorial Sheet – Week 3

11. Evaluate the following given the assignment s = "python"

(a) s[1] *'y'*

(b) s[-1] *'n'*

(c) s[1:3] + s[3:5] *'ytho'*

(d) s[10] *IndexError*

(e) s[10:] *''* *(w/o space)*

(f) s[-4:-2] *'th'*

(g) s[:-4] *'py'*

(h) s[::2] *'pto'*

[start:stop:step]

(i) s[::-1] *'nohtyp'*

step by -1
(backward)

str	p	y	t	h	o	n
index	0	1	2	3	4	5
	-6	-5	-4	-3	-2	-1

←

s[-1]

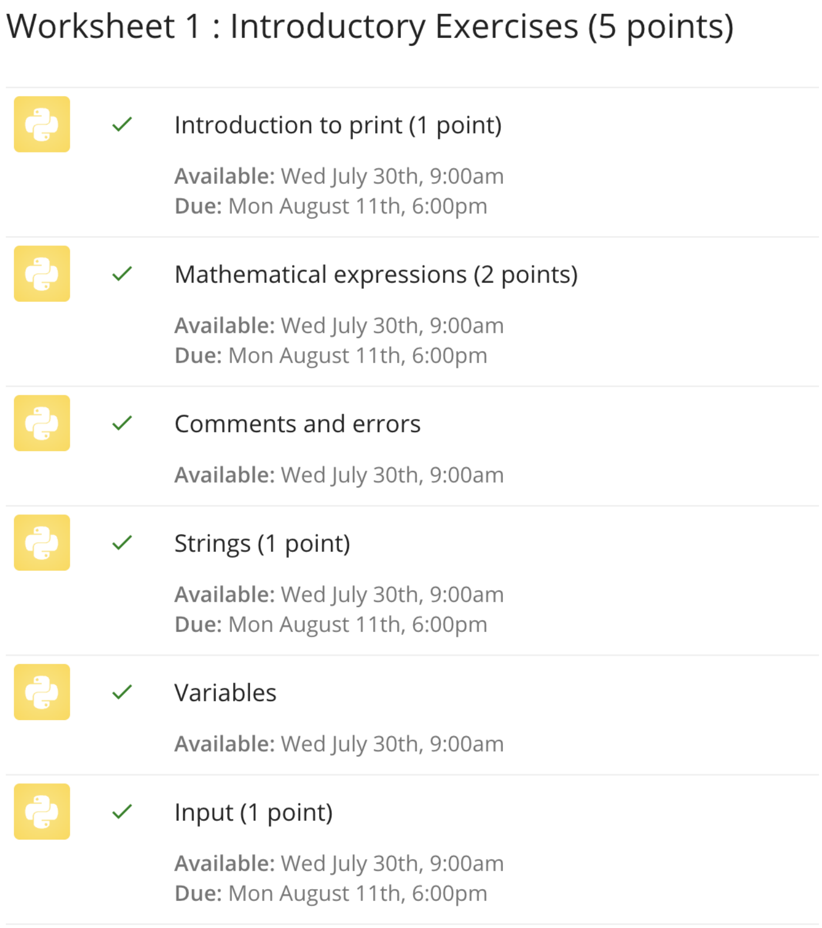
(d) s[10] : *indexing operation > out of range and return IndexError*

(e) s[10:] : *slice operation > nothing to slice and return an empty string*

Ed worksheet 4: *Ed lessons > Worksheet 4 > Substring slicing > Changing the Step Size and Direction [Link]*

Ed worksheets 1 & 2

- Worksheets 1 and 2 due : Monday 11 March 6 pm



Worksheet 1 : Introductory Exercises (5 points)

✓ Introduction to print (1 point)

Available: Wed July 30th, 9:00am
Due: Mon August 11th, 6:00pm

✓ Mathematical expressions (2 points)

Available: Wed July 30th, 9:00am
Due: Mon August 11th, 6:00pm

✓ Comments and errors

Available: Wed July 30th, 9:00am

✓ Strings (1 point)

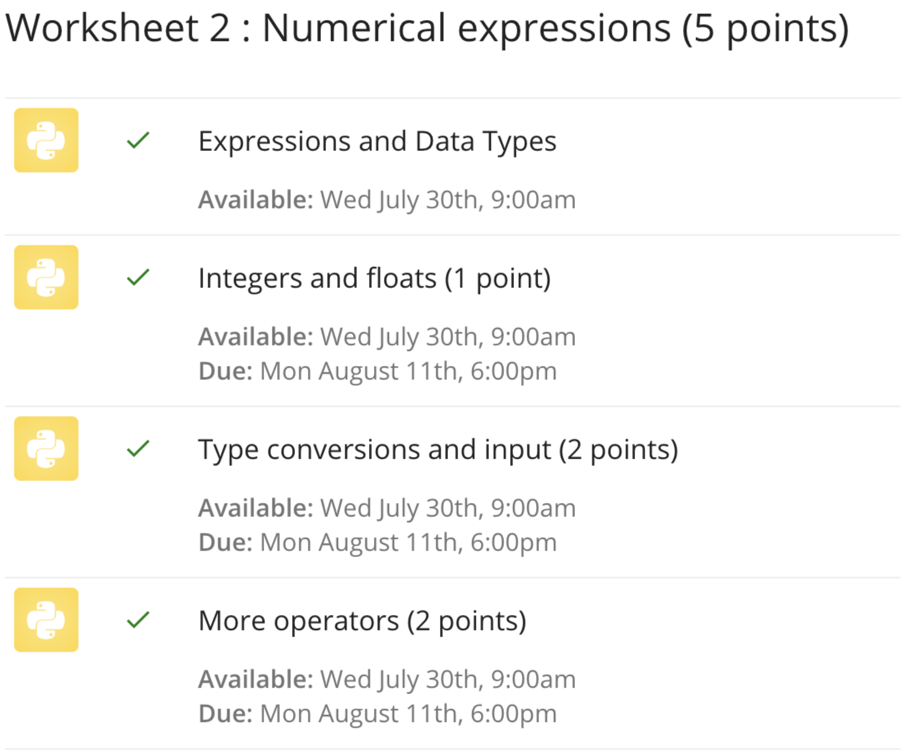
Available: Wed July 30th, 9:00am
Due: Mon August 11th, 6:00pm

✓ Variables

Available: Wed July 30th, 9:00am

✓ Input (1 point)

Available: Wed July 30th, 9:00am
Due: Mon August 11th, 6:00pm



Worksheet 2 : Numerical expressions (5 points)

✓ Expressions and Data Types

Available: Wed July 30th, 9:00am

✓ Integers and floats (1 point)

Available: Wed July 30th, 9:00am
Due: Mon August 11th, 6:00pm

✓ Type conversions and input (2 points)

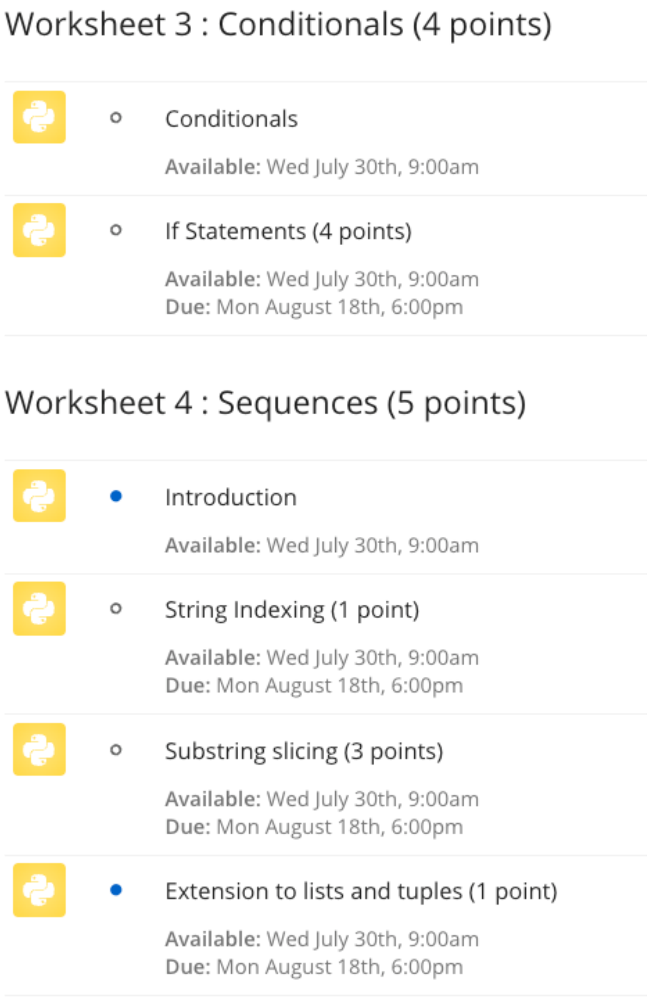
Available: Wed July 30th, 9:00am
Due: Mon August 11th, 6:00pm

✓ More operators (2 points)

Available: Wed July 30th, 9:00am
Due: Mon August 11th, 6:00pm

Ed worksheets 3, 4 & 5

- Worksheets 3, 4, and 5 due : Monday 18 August 6 pm



Worksheet 3 : Conditionals (4 points)

○ Conditionals

Available: Wed July 30th, 9:00am

○ If Statements (4 points)

Available: Wed July 30th, 9:00am
Due: Mon August 18th, 6:00pm

Worksheet 4 : Sequences (5 points)

● Introduction

Available: Wed July 30th, 9:00am

○ String Indexing (1 point)

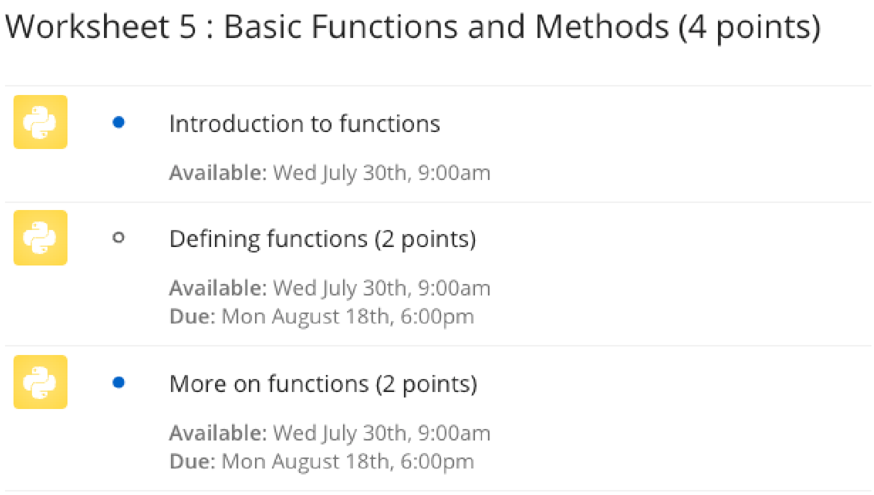
Available: Wed July 30th, 9:00am
Due: Mon August 18th, 6:00pm

○ Substring slicing (3 points)

Available: Wed July 30th, 9:00am
Due: Mon August 18th, 6:00pm

● Extension to lists and tuples (1 point)

Available: Wed July 30th, 9:00am
Due: Mon August 18th, 6:00pm



Worksheet 5 : Basic Functions and Methods (4 points)

● Introduction to functions

Available: Wed July 30th, 9:00am

○ Defining functions (2 points)

Available: Wed July 30th, 9:00am
Due: Mon August 18th, 6:00pm

● More on functions (2 points)

Available: Wed July 30th, 9:00am
Due: Mon August 18th, 6:00pm

Scan here for slides