Week 5 Tutorial

COMP10001 – Foundations of Computing

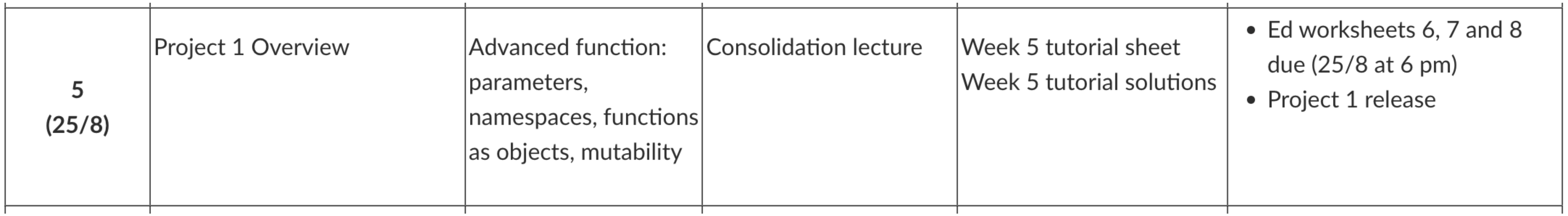
Semester 2, 2025

Clement Chau

- Iterations (For, While)
- Dictionaries
- Sets

Agenda

1. Week 5 Discussion – **Tutorial sheet** (~ 55 mins)
2. One-on-one Q&A for **Ed worksheets** (~ 55 mins)



5 (25/8)	Project 1 Overview	Advanced function: parameters, namespaces, functions as objects, mutability	Consolidation lecture	Week 5 tutorial sheet Week 5 tutorial solutions	• Ed worksheets 6, 7 and 8 due (25/8 at 6 pm) • Project 1 release

Ed worksheets 6, 7 and 8 due (25/Aug, Monday at 6 pm)
Ed worksheets 9 due (1/Sep, Monday at 6 pm)
Mid-Semester Test (11/Sep, Thursday at Lecture time)

Revision: Dictionaries

```
subjects = {
    "COMP10001": "Foundations of Computing",
    "MAST10006": "Calculus 2",
    "MAST10007": "Linear Algebra",
    "SCIE10005": "Today's Science, Tomorrow's World",
}
```

```
print(subjects["COMP10001"])
```

$\Rightarrow$

```
subjects["SCIE10005"] = "TSTW"
```

$\Rightarrow$

Revision: Dictionaries

```
 subjects = {
     "COMP10001": "Foundations of Computing",
     "MAST10006": "Calculus 2",
     "MAST10007": "Linear Algebra",
     "SCIE10000": "Today's Science, Tomorrow's World",
 }

print(subjects.keys())
```

```
⟹  ["COMP10001", "MAST10006", "MAST10007", "SCIE10000"]
```

```
print(subjects.values())
```

```
⟹  [
        "Foundations of Computing",
        "Calculus 2",
        "Linear Algebra",
        "Today's Science, Tomorrow's World"
    ]
```

Revision: Dictionaries

```
subjects = {
    "COMP10001": "Foundations of Computing",
    "MAST10006": "Calculus 2",
    "MAST10007": "Linear Algebra",
    "SCIE10000": "Today's Science, Tomorrow's World",
}
```

```
print(subjects.items())
```

⇒

```
[
    ("COMP10001", "Foundations of Computing"),
    ("MAST10006", "Calculus 2"),
    ("MAST10007", "Linear Algebra"),
    ("SCIE10000", "Today's Science, Tomorrow's World")
]
```

Revision: Sets

```
A = {1, 2, 3}          B = {1, 4, 9}

A | B = A.union(B) = {1, 2, 3, 4, 9}
A & B = A.intersection(B) = {1}

A - B = A.difference(B) = {2, 3}
```

TuteSheet Week 5 – Question 1 (a)

1. Without using a computer, what is the output of the following snippets of code containing loops?

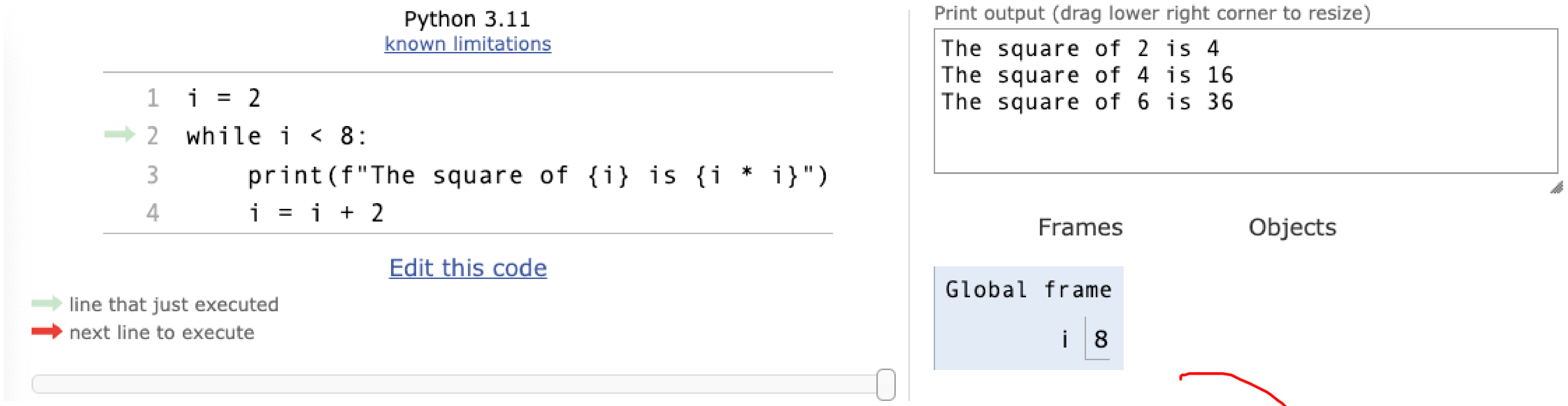
(a)
```
i = 2
while i < 8:
    print(f"The square of {i} is {i * i}")
    i = i + 2
```

```
The square of 2 is 4

The square of 4 is 16

The square of 6 is 36
```

TuteSheet Week 5 – Question 1 (a)



Python 3.11
known limitations

```
i = 2
while i < 8:
    print(f"The square of {i} is {i * i}")
    i = i + 2
```

Edit this code

➡ line that just executed
➡ next line to execute

Print output (drag lower right corner to resize)

```
The square of 2 is 4
The square of 4 is 16
The square of 6 is 36
```

Frames Objects

Global frame

i	8

✓

https://pythontutor.com/

TuteSheet Week 5 – Question 1 (b)

(b)
```
for ingredient in ("corn", "pear", "chilli", "fish"):
    if ingredient.startswith('c'):
        print(ingredient, "is delicious!")
    else:
        print(ingredient, "is not!")
```

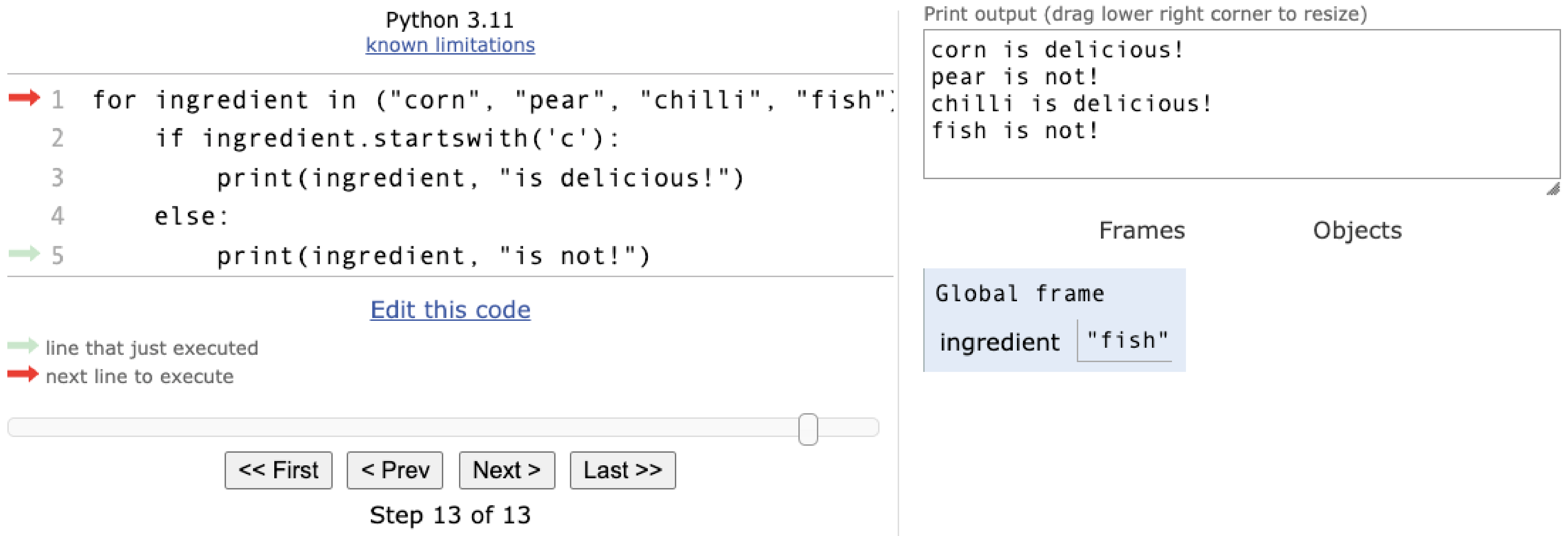
```
corn is delicious!

pear is not!

chilli is delicious!

fish is not!
```

TuteSheet Week 5 – Question 1 (b)



Python 3.11
known limitations

```
for ingredient in ("corn", "pear", "chilli", "fish")
    if ingredient.startswith('c'):
        print(ingredient, "is delicious!")
    else:
        print(ingredient, "is not!")
```

Edit this code

Step 13 of 13

Print output (drag lower right corner to resize)

```
corn is delicious!
pear is not!
chilli is delicious!
fish is not!
```

Global frame

ingredient	"fish"

https://pythontutor.com/

TuteSheet Week 5 – Question 1 (c)

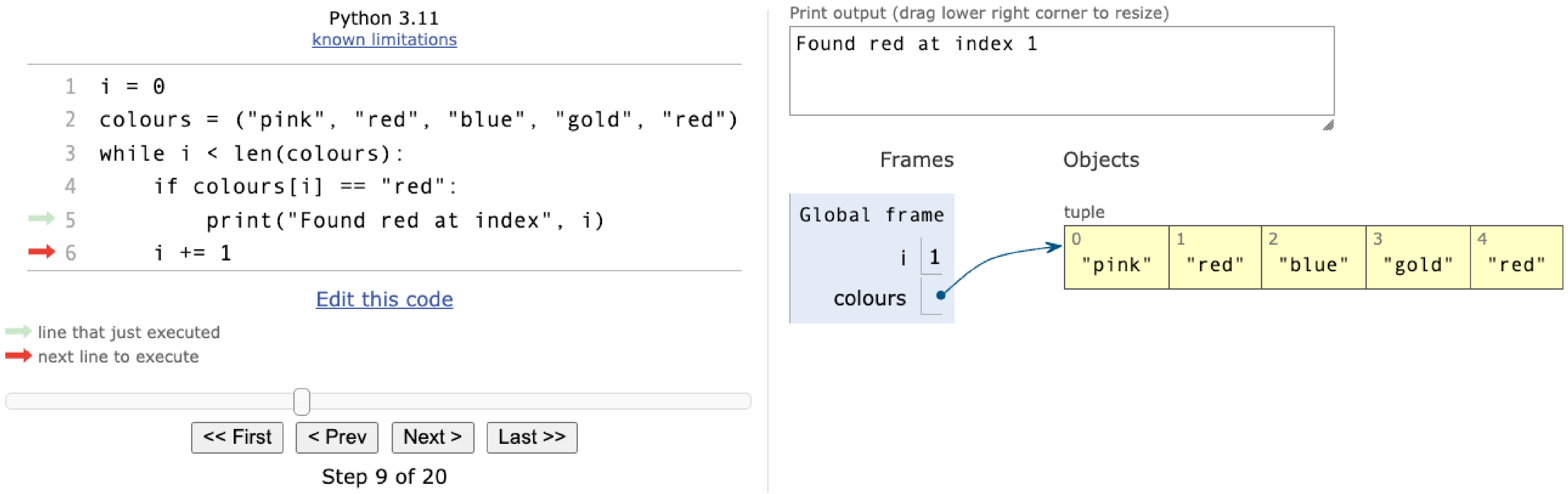
(c)
```
i = 0
colours = ("pink", "red", "blue", "gold", "red")
while i < len(colours):
    if colours[i] == "red":
        print("Found red at index", i)
    i += 1
```

$\longrightarrow$ length 5

```
Found red at index 1

Found red at index 4
```

TuteSheet Week 5 – Question 1 (c)



Python 3.11
known limitations

```
i = 0
colours = ("pink", "red", "blue", "gold", "red")
while i < len(colours):
    if colours[i] == "red":
        print("Found red at index", i)
    i += 1
```

```
Found red at index 1
```

https://pythontutor.com/

TuteSheet Week 5 – Question 1 (d)

(d)
```
input_str = "4+3+2\n3+1+1\n0+4+5+7"
for line in input_str.split("\n"):
    num_sum = 0
    for num in line.split("+"):
        num_sum += int(num)
    print(num_sum)
```

```
9

5

16
```

line

['4+3+2', '3+1+1', 0+4+5+7]

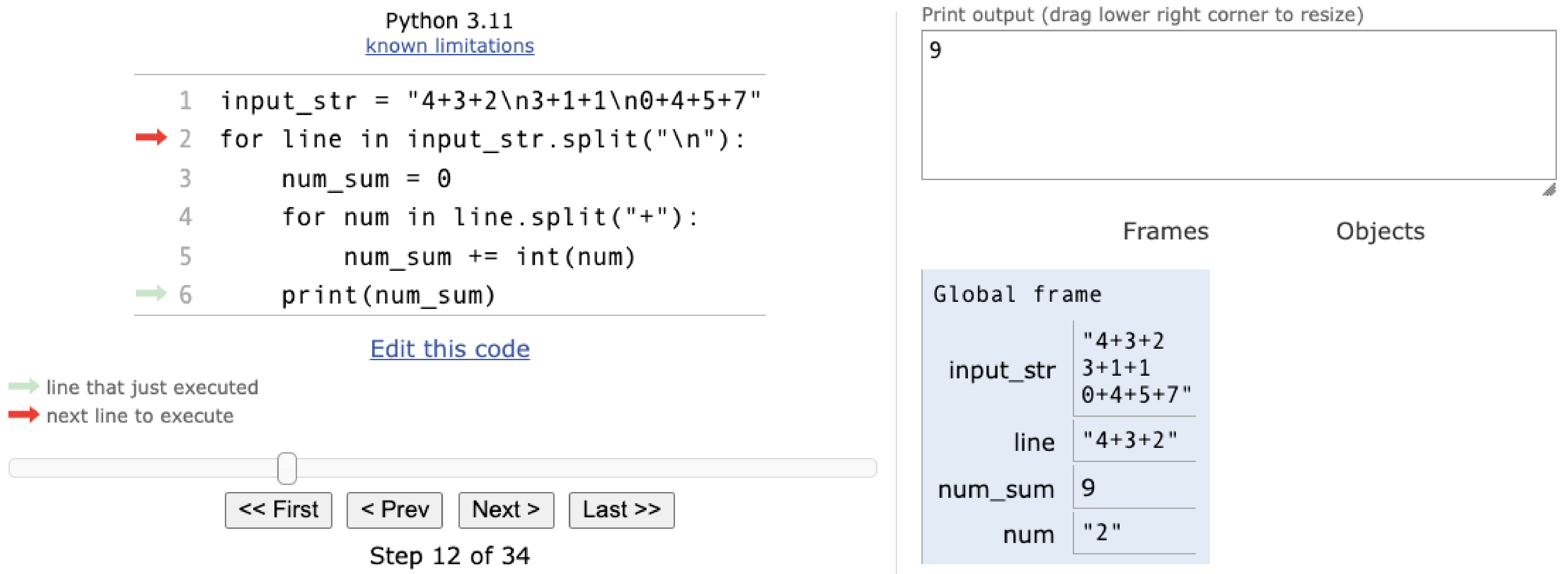
['4','3','2']

9

5

16

TuteSheet Week 5 – Question 1 (d)



Python 3.11
known limitations

```
input_str = "4+3+2\n3+1+1\n0+4+5+7"
for line in input_str.split("\n"):
    num_sum = 0
    for num in line.split("+"):
        num_sum += int(num)
    print(num_sum)
```

Edit this code

Step 12 of 34

Print output (drag lower right corner to resize)

```
9
```

Frames Objects

Global frame

input_str	"4+3+2 3+1+1 0+4+5+7"
line	"4+3+2"
num_sum	9
num	"2"

https://pythontutor.com/

TuteSheet Week 5 – Question 2

2. Consider the following `while` loop and two conversions to `for` loops. Are the two `for` loops equivalent? Why might you choose one over the other?

```
count = 0
items = ('eggs', 'spam', 'more eggs')
while count < len(items):
    print(f"need to buy {items[count]}")
    count += 1
```

```
items = ('eggs', 'spam', 'more eggs')
for count in range(len(items)):
  print(f"need to buy {items[count]}")
```

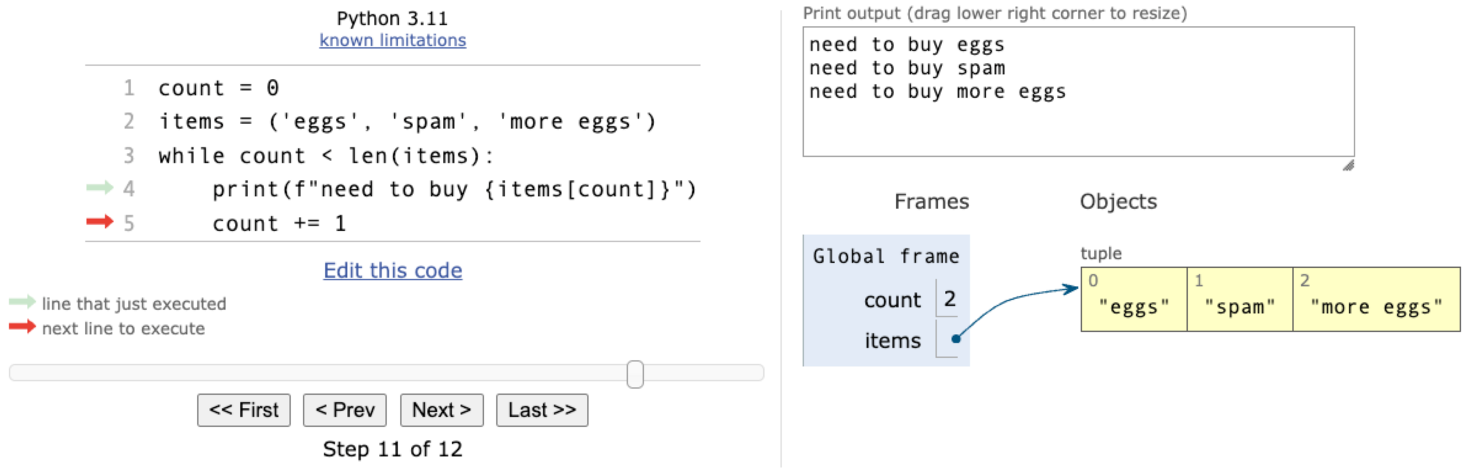
```
items = ('eggs', 'spam', 'more eggs')
for item in items:
  print(f"need to buy {item}")
```

Both are functionally equivalent and will do the same thing.

- *The first uses* ***`range()`*** *to get indices which index items, making it closer to original loop.*
- *The second is cleaner since it iterates through list directly.*

https://www.geeksforgeeks.org/difference-between-for-loop-and-while-loop-in-programming/

TuteSheet Week 5 – Question 2



Python 3.11
known limitations

```python
count = 0
items = ('eggs', 'spam', 'more eggs')
while count < len(items):
    print(f"need to buy {items[count]}")
    count += 1
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 11 of 12

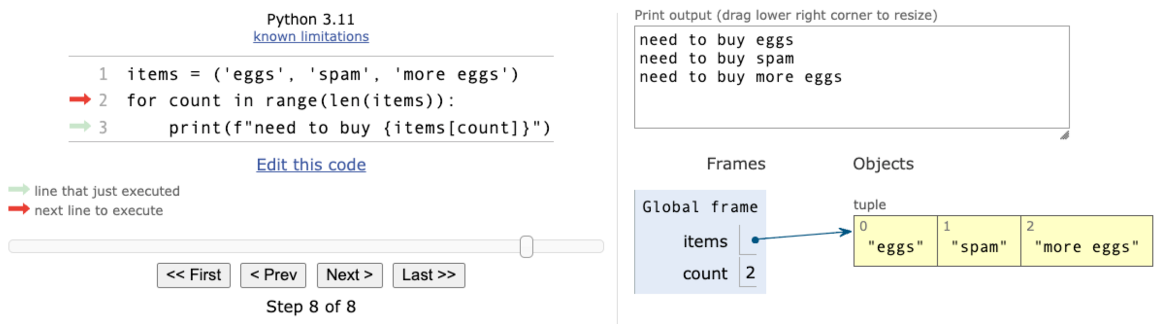
Print output (drag lower right corner to resize)

```
need to buy eggs
need to buy spam
need to buy more eggs
```

Frames / Objects

Global frame: count 2; items → tuple

0	1	2
"eggs"	"spam"	"more eggs"



Python 3.11
known limitations

```python
items = ('eggs', 'spam', 'more eggs')
for count in range(len(items)):
    print(f"need to buy {items[count]}")
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 8 of 8

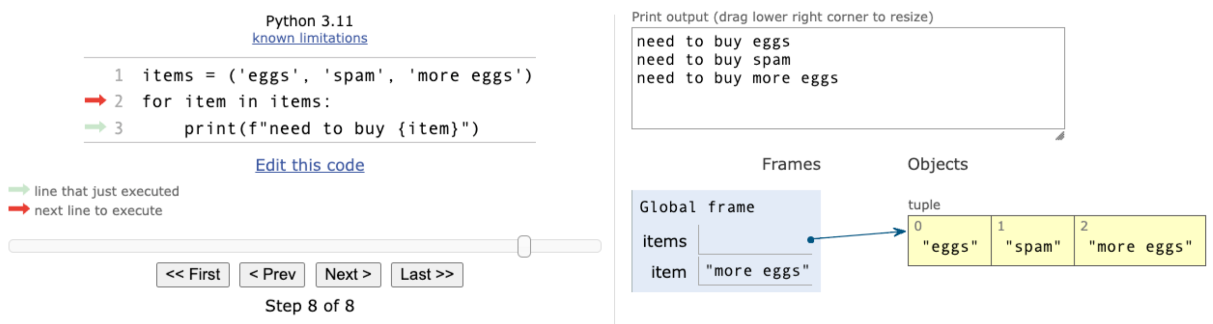
Print output (drag lower right corner to resize)

```
need to buy eggs
need to buy spam
need to buy more eggs
```

Frames / Objects

Global frame: items → tuple; count 2

0	1	2
"eggs"	"spam"	"more eggs"



Python 3.11
known limitations

```python
items = ('eggs', 'spam', 'more eggs')
for item in items:
    print(f"need to buy {item}")
```

Edit this code

line that just executed
next line to execute

<< First | < Prev | Next > | Last >>

Step 8 of 8

Print output (drag lower right corner to resize)

```
need to buy eggs
need to buy spam
need to buy more eggs
```

Frames / Objects

Global frame: items → tuple; item "more eggs"

0	1	2
"eggs"	"spam"	"more eggs"

https://pythontutor.com/

TuteSheet Week 5 – Question 3

3. Rewrite the loops in Questions 1a and 1b, converting `for` loops to `while` loops and vice versa.

(a)
```
i = 2
while i < 8:
    print(f"The square of {i} is {i * i}")
    i = i + 2
```

Convert to **`for`** *loop*

→ 2 4 6 ~~8~~

```
for i in range(2, 8, 2):
        print(f"The square of {i} is {i*i}")
```

(b)
```
for ingredient in ("corn", "pear", "chilli", "fish"):
    if ingredient.startswith('c'):
        print(ingredient, "is delicious!")
    else:
        print(ingredient, "is not!")
```

Convert to **`while`** *loop*

```
ingredients = ("corn", "pear", "chilli", "fish")
i = 0
while i < len(ingredients):
      ingredient = ingredients[i]
      if ingredient.startswith("c"):
            print(ingredient, "is delicious!")
      else:
            print(ingredient, "is not!")
      i += 1
```

TuteSheet Week 5 – Question 3

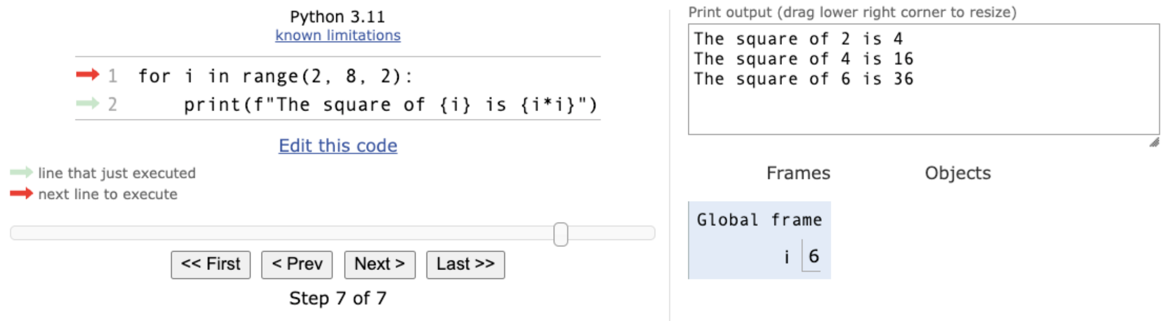
3. Rewrite the loops in Questions 1a and 1b, converting `for` loops to `while` loops and vice versa.

(a)
```python
i = 2
while i < 8:
    print(f"The square of {i} is {i * i}")
    i = i + 2
```

(b)
```python
for ingredient in ("corn", "pear", "chilli", "fish"):
    if ingredient.startswith('c'):
        print(ingredient, "is delicious!")
    else:
        print(ingredient, "is not!")
```

Convert to ***`for`*** *loop*

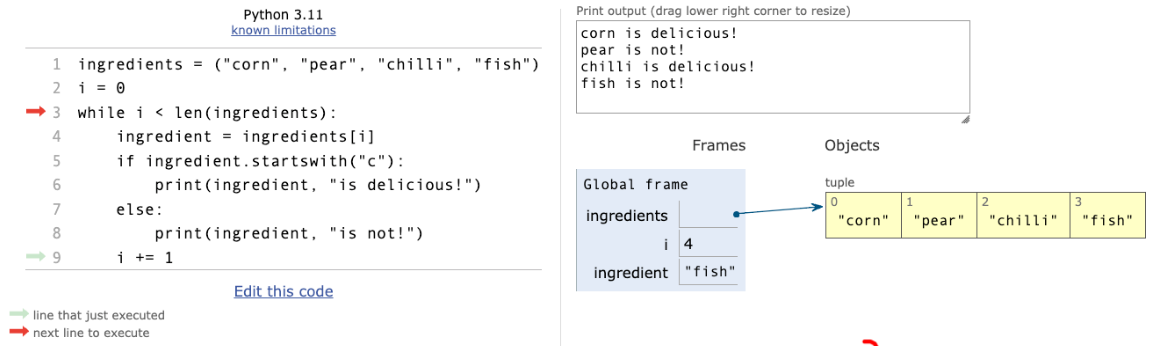
Convert to ***`while`*** *loop*



```python
for i in range(2, 8, 2):
    print(f"The square of {i} is {i*i}")
```

```
The square of 2 is 4
The square of 4 is 16
The square of 6 is 36
```

i	6



```python
ingredients = ("corn", "pear", "chilli", "fish")
i = 0
while i < len(ingredients):
    ingredient = ingredients[i]
    if ingredient.startswith("c"):
        print(ingredient, "is delicious!")
    else:
        print(ingredient, "is not!")
    i += 1
```

```
corn is delicious!
pear is not!
chilli is delicious!
fish is not!
```

ingredients	
i	4
ingredient	"fish"

0	1	2	3
"corn"	"pear"	"chilli"	"fish"

https://pythontutor.com/

TuteSheet Week 5 – Question 4

4. Evaluate the following given the assignment `d = {"R": 0, "G": 255, "B": 0, "other": {"opacity": 0.6}}`. If `d` changes as a result, give its new value. Assume `d` is reset to its original value each time.

(a) `"R" in d.keys()` ***True***

(b) `d["R"]` ***0***

(c) `d["R"] = 255`

d = `{'R': 255, 'G': 255, 'B': 0, 'other': {'opacity': 0.6}}`

(d) `d["A"]` ***KeyError***

(e) `d["A"] = 50` adding key

d = `{'R': 0, 'G': 255, 'B': 0, 'other': {'opacity': 0.6}, 'A': 50}}`

(f) `d.pop("G")` ***255***

d = `{'R': 0, 'B': 0, 'other': {'opacity': 0.6}}`

(g) `d["other"]["blur"] = 0.1`

d = `{'R': 0, 'G': 255, 'B': 0, 'other': {'opacity': 0.6, 'blur': 0.1}}`

(h) `d.items()`

dict_items`([('R', 0), ('G', 255), ('B', 0), ('other', {'opacity': 0.6})])`

TuteSheet Week 5 – Question 5

5. Evaluate the following given the assignment `s1 = {1, 2, 4}` and `s2 = {3, 4, 5}`. If `s1` or `s2` change as a result, give their new value. Assume `s1` and `s2` are reset to their original values each time.

(a) `s1.add(7)`

`{1, 2, 4, 7}`

(b) `s1.add(2)`

s1 does not change (2 is already in the set)

(c) `s2.remove(5)`

`{3, 4}`

(d) `s1 & s2`, or equivalently `s1.intersection(s2)`

`{4}`

The intersection of two sets includes ***only the common elements*** *present in both sets.*

(e) `s1 | s2`, or equivalently `s1.union(s2)`

`{1, 2, 3, 4, 5}`

The union of two sets ***combines all unique elements*** *from both sets.*

(f) `s1 - s2`

`{1, 2}`

The difference between two sets includes elements ***present in the first set but not in the second***.

https://www.geeksforgeeks.org/python-set-operations-union-intersection-difference-symmetric-difference/

Independent Work

- **Project 1 will be released this Friday!**
  - **If you're struggling, please try to find assistance early than later.**
- **Do worksheet 9** on Ed (due **next Monday at 6pm**)
  - Remember that **Ed worksheets contributes to 10% of your total score!**
- **Raise your hand** if you have any questions!

Scan here for annotated slides