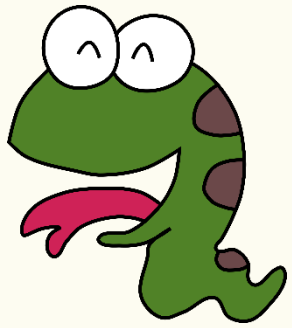
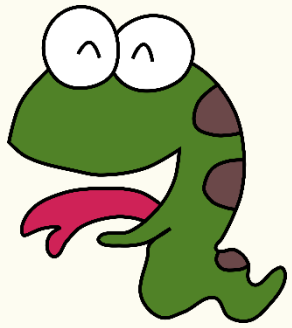


Python Programming

We will be starting shortly

Sit back and relax while you wait!

(Slides for today's lecture are on Scientia)



Python Programming

Warning!

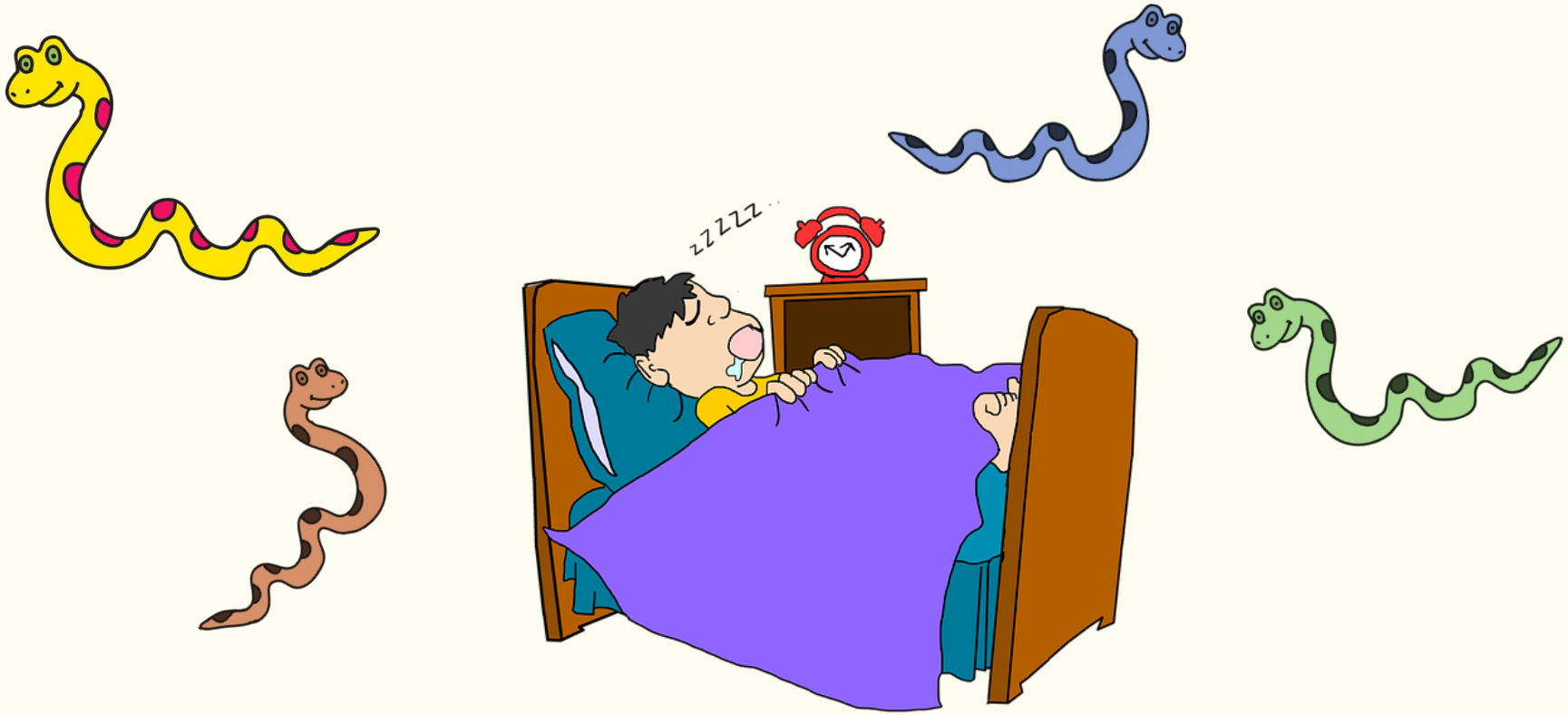
We will start recording this session now!

Also, any messages in the text chat will remain on MS Teams even after the session

Quiz

Who is a snake’s favorite author?

How was your week?



Need more exercises?

`https://edabit.com/challenges`

If you're done with Lesson 8, go ahead with Intro2ML's NumPy tutorial

`https://intro2ml.pages.doc.ic.ac.uk/autumn2021/modules/lab-numpy/introduction`

Your code works correctly!

Now what?

Writing efficient code

Improving the algorithm

Prime number validator (from Lesson 4)

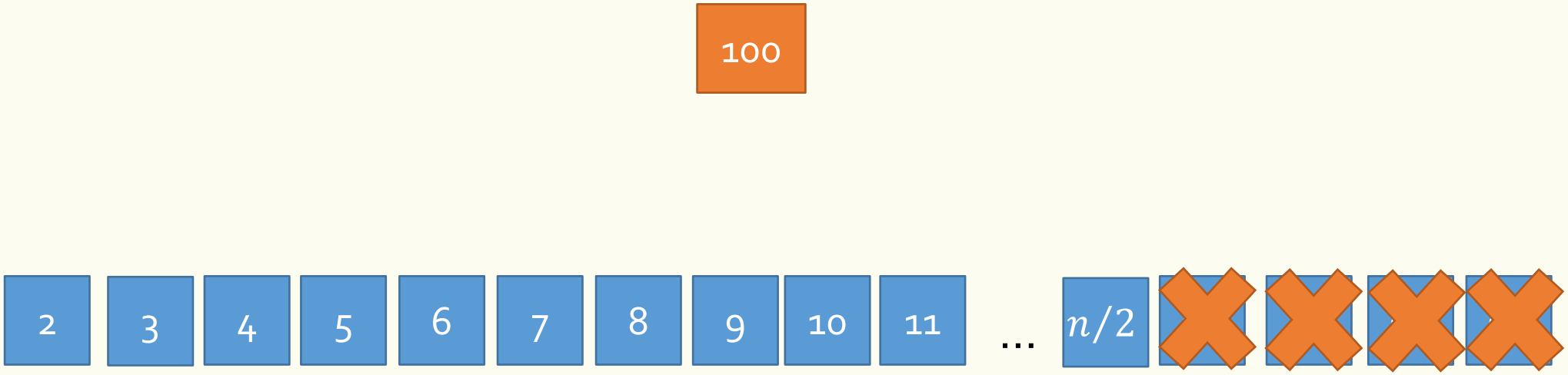
Exhaustive search

Reduce search space?

Factors of 100:

2, 4, 5, 10, 20, 25, 50

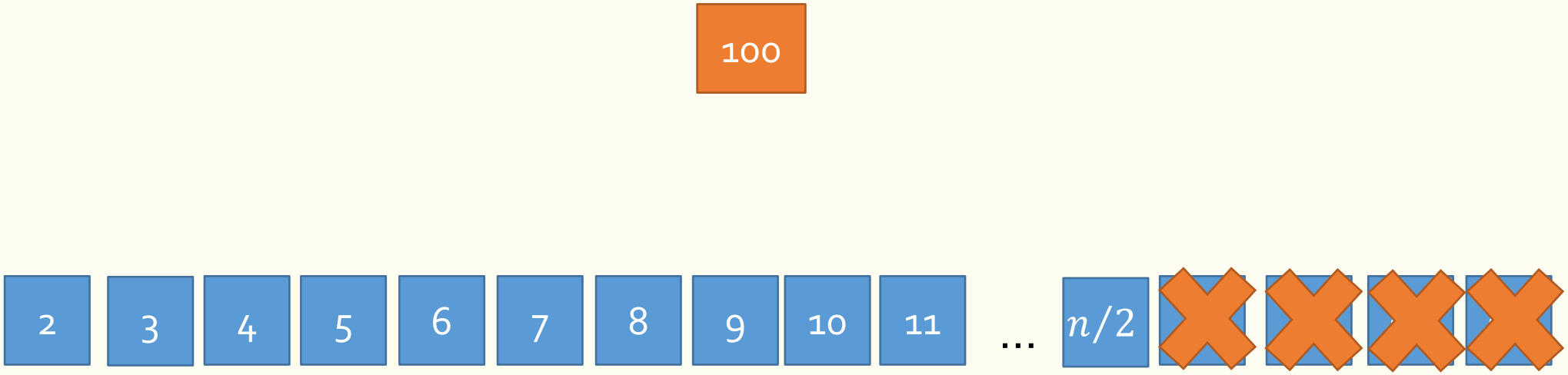
Prime number validator (from Lesson 4)



Factors of 100:

2, 4, 5, 10, 20, 25, 50

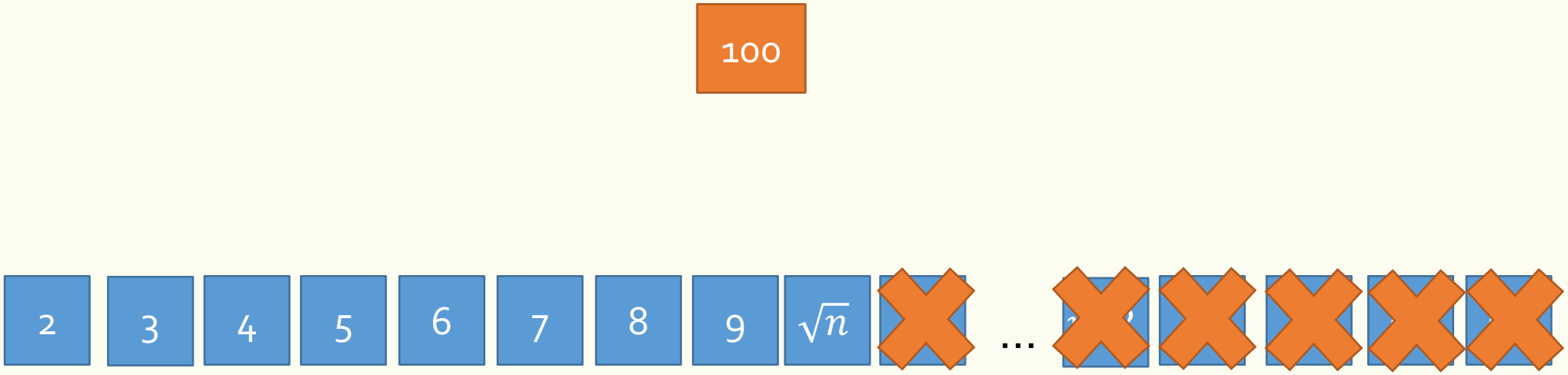
Prime number validator (from Lesson 4)



Factors of 100:

2, **4**, **5**, 10, **20**, **25**, **50**

Prime number validator (from Lesson 4)



Factors of 100:

2, **4**, **5**, 10, **20**, **25**, **50**

Prime number validator (from Lesson 4)

```
n = int(input("Please enter an integer: "))

divisor = 2
is_prime_number = True

while divisor < n:
    if n % divisor == 0:
        is_prime_number = False
        break
    else:
        divisor = divisor + 1

print(n>1 and is_prime_number)
```

Square root estimator (from Lesson 4)

```
n = float(input("Please enter a positive number: "))

step_size = 0.000001
tolerance = 0.0001

root = 0
while root <= n:
    if abs(root*root - n) < tolerance:
        print(root)
        break
    else:
        root = root + step_size
```

How would you make this more efficient?

Writing efficient code

Using efficient operations

Performance tuning #1

- Concatenating a list of strings into a single string

```
items = ["a", "b", "c", "d", "e", "f", "g", "h",
"i", "j", "k", "l", "m", "n", "o", "p", "q", "r",
"s", "t", "u", "v", "w", "x", "y", "z"]
```

```
abcdefghijklmnopqrstuvwxyz
```

Performance tuning #1

- Which will be faster?

```
s = ""
for item in items:
    s += item
```

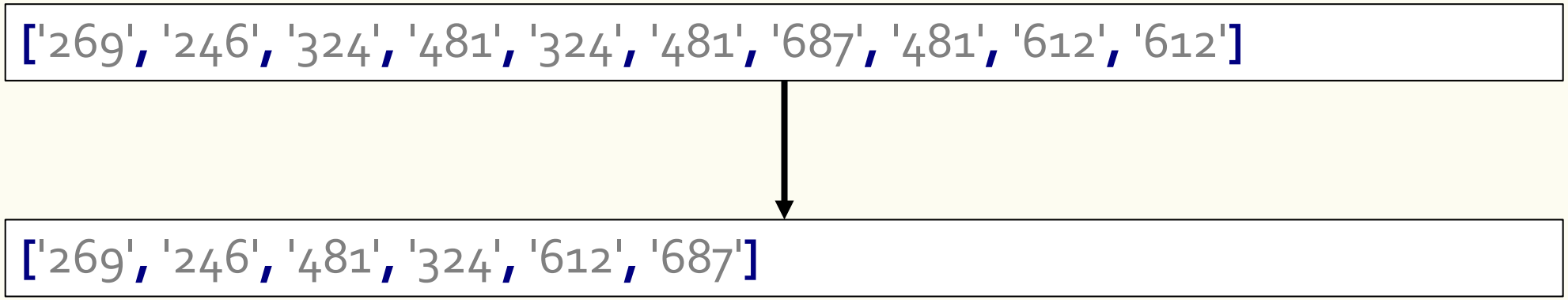
```
s = "".join(items)
```

Writing efficient code

Using optimal data structures

Performance tuning #2

- Filter a list of strings to return another list of unique strings



Performance tuning #2

```
filtered_items = []

for item in items:
   if item not in filtered_items:
      filtered_items.append(item)
```

```
filtered_items = list(set(items))
```

```
filtered_items = list(dict(zip(items, [None]*len(items))))
```

Performance tuning #2

<filter_with_list>,	list size 10,	0.0008445 seconds
<filter_with_dict>,	list size 10,	0.0008762 seconds
<filter_with_set>,	list size 10,	0.0003818 seconds
<filter_with_list>,	list size 100,	0.031647 seconds
<filter_with_dict>,	list size 100,	0.0032972 seconds
<filter_with_set>,	list size 100,	0.0012386 seconds
<filter_with_list>,	list size 600,	0.96294 seconds
<filter_with_dict>,	list size 600,	0.020129 seconds
<filter_with_set>,	list size 600,	0.0093079 seconds
<filter_with_list>,	list size 4500,	10.718 seconds
<filter_with_dict>,	list size 4500,	0.1193 seconds
<filter_with_set>,	list size 4500,	0.03865 seconds
<filter_with_list>,	list size 30000,	85.121 seconds
<filter_with_dict>,	list size 30000,	0.83646 seconds
<filter_with_set>,	list size 30000,	0.22502 seconds

Performance tuning #3

- Is list comprehension faster?

```
s = []
for item in items:
    s.append(item)
```

```
S = [item for item in items]
```

Next week's schedule

Mon 3-4pm	Mon 4-5pm	Tue 9-10am	Wed 9-10am	Thu 11am-1pm
LECTURE Online only	LAB Online only	LAB 219	LAB 219	LAB 221/225

Lecture topic: Searching algorithms

CW1 released on Monday

One on one with Josiah

Mon 11/10 (4PM)		
16:00-16:10	jac202	John Carter
16:10-16:20	am10118	Anagh Malik
16:20-16:30	cu021	Chibudom Onuorah
16:30-16:40	????	Jonathan Hewlett
16:40-16:50	jh3617	Jacob Hughes-Hallett
16:50-17:00	lr4617	Lapo Rastrelli

Tue 12/10 (9AM)		
09:00-09:10	aaa1421	Abdullah Alrumayh
09:10-09:20	ag4916	Agnese Grison
09:20-09:30	aw21	Alistair Weld
09:30-09:40	sh2316	Simon Hanassab
09:40-09:50	aj2221	Alexander Jenkins
09:50-10:00	lmc16	Lucille Cazenave

One on one with Josiah

Wed 13/10 (9AM)		
09:00-09:10	av1017	Avish Vijayaraghavan
09:10-09:20	fn421	Federico Nardi
09:20-09:30	ddg21	Dimitar Georgiev
09:30-09:40	jb1721	Joao Binenbojm de Pereira
09:40-09:50	cp2620	Camille Petri
09:50-10:00	atr17	Alexander Ranne

Thu 14/10 (11AM)		
11:00-11:10	cm2021	Christos Margadji
11:10-11:20	cpc21	Cormac Conway
11:20-11:30	jla21	Jonah Anton
11:30-11:40	mjc121	Matthew Collins
11:40-11:50	mt3215	Maksym Tymchenko
11:50-12:00	sk2521	Sun Jin Kim
12:00-12:10	st321	Sofiya Toteva
12:10-12:20	tap21	Thomas Phillips
12:20-12:30	yo521	Yi Siang Ong

Any feedback for us?

- https://www.menti.com/7qxudnnc3i
- Or go to www.menti.com and enter **1011 6313**