

COMP1021
Introduction to Computer Science

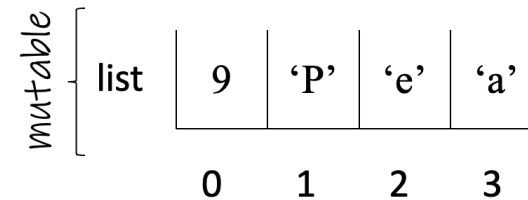
Handling Repeating Patterns

Yongqiang Tian

Slides are adapted from Gibson Lam's and
David Rossiter's

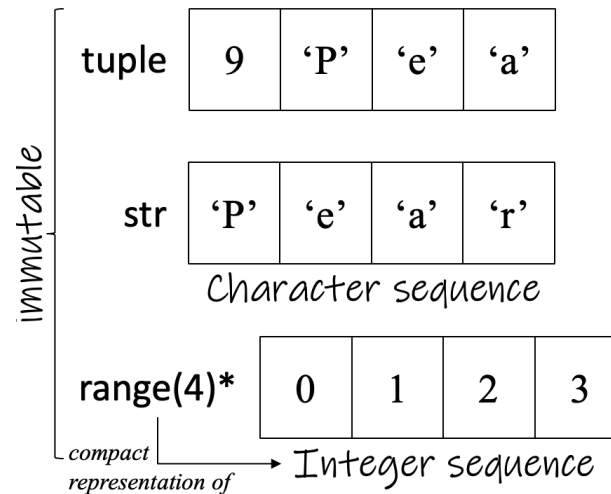
Review

- List/Tuple/String



Mutable	<code>[], for, <u>len</u>, count, index, insert, remove, append, reverse, sort, extend</code>
----------------	---

Immutable	<code>[], for, <u>len</u>, count, index</code>
------------------	--



```
a = [9, 'P', 'e', 'a']
      (9, 'P', 'e', 'a')
      "Pear"
      range(4)

print(a[3])

for e in [9, 'P', 'e', 'a']
          (9, 'P', 'e', 'a')
          "Pear"
          range(4)
    print(e)
```

Review: []

- [] can be used to create a list
 - `list_a = ["John", "Charles", "Vivian"]`
- [] can be used to get element from list/tuple/string
 - `list_a[0]` is the first element!
 - `list_a[-1]` is the last element!

Common mistakes

- `List(range(0))`: `[]`
 - Empty list; no error!

% Modulus Operator

$$5 \% 3 = 2$$

$$\text{Even number} \% 2 = 0$$

$$\text{Odd number} \% 2 = 1$$

```
evenodd.py - //VDRIVE/MYHOME/rossiter/Documents/evenodd.py (3.8.5)
File Edit Format Run Options Window Help
number = int(input("Please give me a number: "))
if number % 2 == 1:
    print("It is an odd number!")
else:
    print("It is an even number!")
```

- If we use other numbers as the divisor we can find repeating patterns with a different size, e.g.:

number 0 1 2 3 4 5 6 7 8 ...

number % 4 0 1 2 3 0 1 2 3 0 ...

Cycles in the repeating pattern

