A tiny bit more Python

Andrew S Fitz Gibbon
UW CSE 160
Winter 2022

See in python tutor

Enumerate a list

```
lst = [10 ** x for x in range(10)]

for i in range(len(lst)):
    print('value at index', i, 'is', lst[i])
```

index

value

Or:

```
for index, value in enumerate(lst):
     print('value at index', index, 'is', value)
```

Like dict.items()

Enumerate a list

Goal: add each element's index itself

```
lst = [x for x in range(10)]
new_lst = []
for i, v in enumerate(lst):
      new_lst.append(i + v)
```

With a list comprehension:

```
lst = [x for x in range(10)]
new_lst = [i + v for i, v in enumerate(lst)]
```

See in python tutor

Activity: Enumerate a list

Goal: Given a list of participants, in the order they finished a race, create a dictionary that maps their name to their finishing place.

Racers

```
racers = ['Dino', 'Wilma', 'Barney', 'Fred']
race_dict = {'Dino':1, 'Wilma':2, 'Barney':3, 'Fred':4}
```

With a list comprehension:

```
race_dict =
```

Ternary Assignment

A common pattern in python

```
if x > threshold:
     flag = "Over"
else:
     flag = "Under"
```

Or

```
flag = "Under"
if x > threshold:
     flag = "Over"
```

See in python tutor

Ternary Assignment

A common pattern in python

```
if x > threshold:
    flag = "Over"
else:
     flag = "Under"
```

With a ternary expression:

```
flag = "Over" if x > threshold else "Under"
```

Ternary Expression
"Three elements"

Ternary Assignment

```
flag = "Over" if x > threshold else "Under"
```

Result if true

Condition

Result if false

- Only works for single expressions as results.
- Only works for if and else (no elif)

See in python tutor

Ternary Assignment

Goal: A list of 'odd' or 'even' if that index is odd or even.

```
lst = []
for i in range(8):
    if i % 2 == 0:
        lst.append('even')
    else:
        lst.append('odd')
```

or

```
lst = []
for i in range(8):
    lst.append('even' if i % 2 == 0 else 'odd')
```

Ternary Assignment

Goal: A list of 'odd' or 'even' if that index is odd or even.

```
lst = []
for i in range(8):
    if i % 2 == 0:
        lst.append('even')
    else:
        lst.append('odd')
```

Or with a list comprehension!

```
lst = ['even' if i % 2 == 0 else 'odd' for i in range(8)]
```

Get more practice

List Comprehensions:

```
[(x, y) for x in seq1 for y in seq2 if
                        sim(x, y) > threshold]
```

Enumerate:

```
for index, value in enumerate(seq):
    …
```

Ternary If Statement:

```
flag = "Over" if x > threshold else "Under"
```

Bonus: Generator

```
for item in sequence:
```

So… What can `sequence` be?

- `[1, 2, 3]` (or `list`, where `list = [1, 2, 3]`)
- `range(n)`, or `range(n, step)`
- Enumerate, list comprehensions, or more…
- … like maybe a function?

Bonus: Generator

```
for num in go_until(2):
    print(num)

def go_until(max):
    n = 0
    while n < max:
        yield n
        n += 1
```

Bonus: Generator

```
for num in go_until(2):
    print(num)

def go_until(max):
    n = 0
    while n < max:
        yield n
        n += 1
    # A function with no return?!?
```