

A tiny bit more Python

Ruth Anderson
UW CSE 160
Autumn 2021

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)]
```

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 =
```

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 = {v: i + 1 for i, v in enumerate(racers)}
```

Ternary Assignment

A common pattern in python

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

Or

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

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

The diagram shows the code 'flag = "Over" if x > threshold else "Under"'. Three blue brackets are drawn under the code to identify its parts: one under 'Over', one under 'if x > threshold else', and one under 'Under'. Below each bracket is a label: 'Result if true' under 'Over', 'Condition' under 'if x > threshold else', and 'Result if false' under 'Under'.

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

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"
```