Loops

Loops

- Python does not use indices in for loops

  -

```python
animals = ['bird', 'dog', 'cat']

for animal in animals:
    print(f'I have a {animal})
```

Loops

- If we need indices in Python, we can use enumerate

```python
colors = ['blue', 'yellow', 'red', 'green']


for i, color in enumerate(colors):
    print(i, color)
```

- `enumerate` returns an enumerate object which is an iterator that allows us to use a for loop

- You can even change the "index"

```python
for i, color in enumerate(colors, start = 1):
```

Loops

- Dictionaries:

  - The for loop takes the keys

```
animals = {'dog' : 3, 'cat' : 5, 'bird' : 1, 'hamster': 2}

for animal in animals:
    print(f'I have {animals[animal]} {animal}(s)')
```

Loops

- Dictionaries:

    - We can avoid the bracket notation using items

    - items returns an items object, but that is not important to us

```
animals = {'dog' : 3, 'cat' : 5, 'bird' : 1, 'hamster': 2}

for animal, count in animals.items():
    print(f'I have {count} {animal}(s)')
```

Loops

- We use the `break` keyword to finish a loop

- We use the `continue` keyword to finish the current execution and go to the next element in the loop