LIST AND RANGE

LISTS

- A **list** contains a collection of values in square brackets, [ ].
- Each value can be a float, int, string or variable.
- Each value in a **list** is called an element.
- Lists can contain:
  - Nested lists
  - Tuples
  - Sets
  - Dictionaries
  - Series
  - DataFrames

MUTABLE

- Elements in a **list** are mutable.
- Elements in a **tuple** or **"string"** aren't mutable.

EXAMPLE

```
num = [10, 20, 30]

num[0] = "hello"

["hello", 20, 30]
```

RANGE

- **range** generates a specified **list** of numbers.
- Used with loops.

In [269]:

```
list(range(10, 110, 10))
```

Out[269]:

```
[10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
```

In [272]:

```
numbers = list(range(10, 150, 10))
numbers
type(numbers)
type([])
```

Out[272]:

list

```
nested_list = [ [1,2,3], [4,5,6], [7,8,9]]
nested_range = [list(range(11)), list(range(10, 21)),
                list(range(20, 31))]
nested_range
```

Out[278]:

```
[[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
 [10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20],
 [20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30]]
```

In [280]:

```
basket1 = [1,2,3,4]
basket1.append(55)
basket1
```

Out[280]:

```
[1, 2, 3, 4, 55]
```

In [282]:

```
basket1.pop()
basket1
```

Out[282]:

```
[1, 2, 3, 4, 55]
```

In [283]:

```
basket1.insert(2, 100)
basket1
```

Out[283]:

```
[1, 2, 100, 3, 4, 55]
```

In [288]:

```
basket1.sort()
basket1
```

Out[288]:

```
[1, 2, 3, 4, 55, 100]
```

In [302]:

```
basket2 =[10, 20, 30, basket1]
basket2[3][5]
```

Out[302]:

```
100
```

In [310]:

```
basket3 = ["apples", "pears", "berries", "bananas"]
basket4 = [basket2, basket3]
```

```
basket4[0][2] + basket4[0][3][-2]
```

Out[310]:

85

In [315]:

```
basket5 = ["a", "a", "a", 1, 1,1,1, 3, 3, 3,3, 3, 50, 50]
basket5.count(50)
```

Out[315]:

2

In [331]:

```
basket6 = ["a", "b", "c", "d", 10, 20, 30,40, True, False,
           -1, "car", -2, "bar", -3, "star"]
print(len(basket6))
basket6[4:8]
basket6[11::2]
basket6[15::-2]
basket6[15:8:-2]
basket6[10:15:2]
```

16

Out[331]:

[-1, -2, -3]

In [334]:

```
basket1[-1] = "Chips"
basket1
```

Out[334]:

[1, 2, 3, 4, 55, 'Chips']

In [ ]: