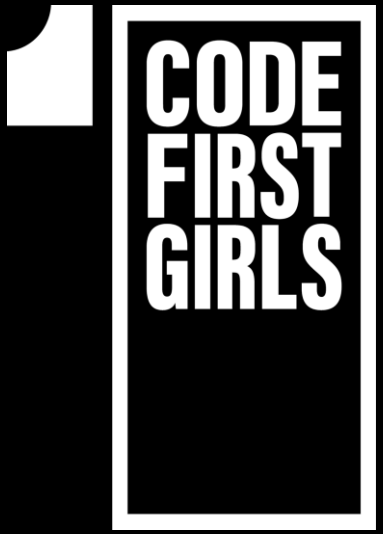
# PYTHON

## LESSON 4




NANODEGREE → FOUNDATION MODULE

# AGENDA

01 Python data types

02 Lists

03 Dictionaries

# LIST

## PYTHON DATA TYPE

- **List:** an ordered collection of values
- List are written inside square brackets and separated by commas

```
# EXAMPLES

lottery_numbers = [4, 8, 15, 16, 23, 42]

student_names = ['Diedre', 'Hank', 'Helena',
'Salome']

person = ['Jess', 32]

teams = [['Jenny', 'Mary'], ['Nigel', 'Steve']]
```

# LIST FUNCTIONS

## DEFINITION

There are functions designed for lists

- len(): the number of items in a list
- max(): The biggest value in a list
- min(): The smallest value in a list

Functions for changing the order of a list

- sorted(): Sorts the
- reversed(): Reverses the order of a list

# EXAMPLE

```
costs = [1.2, 4.3, 2.0, 0.5]

print(len(costs))
print(max(costs))
print(min(costs))
```

# LISTS & FOR LOOPS

**DEFINITION**

DEMO & EXERCISES

to practice writing lists with For Loops

# LIST COMPREHENSION

## DEFINITION

- List comprehensions provide a concise way to create lists.
- It is a programmatically efficient way to create lists
- It is also a frequent question / test for Python tests

# EXAMPLE

The syntax 'formula' for list comprehension is as follows:

```
new_list = [ expression for item in list if conditional ]
```

* 'if conditional' is optional

# DICTIONARIES

## PYTHON DATA TYPE

- **Dictionary:** Stores a collection of labelled items. Each item has a *key* and a *value*

# EXAMPLE

```
person = {
    'name': 'Jessica',
    'age': 23,
    'height': 172
}
```

DEMO & EXERCISES

to practice writing Dictionaries

# TUPLE

## PYTHON DATA TYPE

- **Tuple:** a Tuple is similar to List except that the objects in tuple are IMMUTABLE which means we CANNOT change the elements of a tuple once assigned.

- Tuple syntax uses **( )** brackets

- Tuple behaves very similar to list

# EXAMPLE

```
order = ('croissant', 'coffee', 'juice')
```

# SET

## PYTHON DATA TYPE

**set:** set is an unordered and unindexed collection of UNIQUE items in Python.

- Unordered means when we display the elements of a set, it will come out in a random order.
- Unindexed means, we cannot access the elements of a set using the indexes like we can do in list and tuples.
- Set elements are unique. Duplicate elements are not allowed.
- Set uses **{}**

# EXAMPLE

```
my_set = {1,2,3,4,5, 'hi', 7}
```

# THANK YOU!