Assign a Variable

[ ] ↳ *4 cells hidden*

Data Types

- int
- float
- str (string)
- bool (True, False)

[ ] ↳ *8 cells hidden*

Data Structures

- list
- tuple
- dictionary
- set

```
# create a new list
shopping_list = ['egg', 'milk', 'bread']

# update value in a list
shopping_list[0] = 'beer'
shopping_list[1] = 'yoghurt'

print(shopping_list)
```

```
['beer', 'yoghurt', 'bread']
```

```
# list method, add new item at the end of list
shopping_list.append('tomato')
```

```
shopping_list.append('lemon')
print(shopping_list)
```

```
['beer', 'yoghurt', 'bread', 'tomato', 'lemon']
```

```
shopping_list[2:   ]
```

```
['bread', 'tomato', 'lemon']
```

```
new_list = shopping_list + ['omlet', 'book']

print(new_list)
```

```
['beer', 'yoghurt', 'bread', 'tomato', 'lemon', 'omlet', 'book']
```

```
# how many items in this list?
len(new_list)
```

```
7
```

```
# remove bread, index=2
new_list.pop(2)
print(new_list)
```

```
['beer', 'yoghurt', 'tomato', 'lemon', 'omlet', 'book']
```

```
# list is a mutable object
new_list[0] = 'Heineken'
print(new_list)
```

```
['Heineken', 'yoghurt', 'tomato', 'lemon', 'omlet', 'book']
```

```
# tuple is immutable

shopping_tuple = ('egg', 'milk', 'bread', 'egg')
print(shopping_tuple)
```

```
('egg', 'milk', 'bread', 'egg')
```

```
# tuple, we cannot update value in a tuple
shopping_tuple[0:]
```

```
('egg', 'milk', 'bread', 'egg')
```

```
shopping_tuple.count('egg')
```

```
2
```

```
# set is another built-in data structure in Python
shopping_tuple
```

```
('egg', 'milk', 'bread', 'egg')
```

```
# create a new set
shopping_set = set(shopping_tuple)
print(shopping_set)
```

```
{'egg', 'bread', 'milk'}
```

```
print(type(shopping_set))
```

```
<class 'set'>
```

```
my_set = {'Aan','Zue','Ink','ZUE'}
```

```
my_set
```

```
{'Aan', 'Ink', 'ZUE', 'Zue'}
```

```
# dictionary the final data structure
# key-value pairs

customer = {
    "id": 1,
    "fname": "David",
    "lname": "Beckham",
    "nationality": "English"
}

print(customer)
```

```
{'id': 1, 'fname': 'David', 'lname': 'Beckham', 'nationality': 'English'}
```

```
# get value from a dictionary
customer['nationality']
```

```
# dictionary is mutable object
customer['nationality'] = 'British'
print(customer)
```

```
{'id': 1, 'fname': 'David', 'lname': 'Beckham', 'nationality': 'British'}
```

```
# create a new key
customer['team'] = 'Manchester United'

customer['fav_movies'] = ['Batman', 'Superman', 'Marvel']

customer
```

```
{'fav_movies': ['Batman', 'Superman', 'Marvel'],
 'fname': 'David',
 'id': 1,
 'lname': 'Beckham',
 'nationality': 'British',
 'team': 'Manchester United'}
```

```
customer['fav_movies'][-2]
```

```
print(customer) # ordered
```

```
{'id': 1, 'fname': 'David', 'lname': 'Beckham', 'nationality': 'British', 'team': 'Manch
```

```
# delete key
customer.pop('fav_movies')

customer
```

```
{'fname': 'David',
 'id': 1,
 'lname': 'Beckham',
 'nationality': 'British',
 'team': 'Manchester United'}
```

```
# check key is in a dict
'TEAM' in customer
```

```
False
```

```
del customer['fname']
customer
```

```
{'id': 1,
 'lname': 'Beckham',
 'nationality': 'British',
 'team': 'Manchester United'}
```

```

```

Control Flow

- if else
- for

- while

```
# if else
score = 90

if score > 80:
    print("Passed")
else:
    print("Failed")
```

```
Passed
```

```
# for loop + if else
scores = [85, 90, 65, 60, 82]

grades = [] # empty list

for score in scores:
    if score > 80:
        grades.append("Passed") # list method
    else:
        grades.append("Failed")

print(grades)
```

```
['Passed', 'Passed', 'Failed', 'Failed', 'Passed']
```

```
# while loop

count = 0

while (count < 5) :
    print("Hello!")
    count += 1
```

```
Hello!
Hello!
Hello!
Hello!
Hello!
```

Functions

```python
# define a new function

def greeting():
    print("Hello World!")
```

```python
def greeting_user(name):
    return "Hello " + name
```

```python
result = greeting_user("Toy")
```

```python
result
```

```
'Hello Toy'
```

```python
## function to give grade to each student
## function is RE-USABLE

def grading(scores):
    """Give grades to a list of score values for students"""
    result = []
    for score in scores:
        if score > 80:
            result.append("Passed")
        else:
            result.append("Failed")
    return result
```

```python
result = grading([85, 90, 26, 100, 67])
print(result)
```

```
['Passed', 'Passed', 'Failed', 'Passed', 'Failed']
```

✓ 0s completed at 10:23 PM ● ✕