```
# Tutorial Examples for lecture week 3
# conditionals and loops
# author: B. Schoen-Phelan
# date: Oct 2020


# some examples taken from or adapted from the ms
python course
# with Susan Ibach and Christopher Harris

# arrays exist but not natively
# we have to import a module
# arrays can only be of one type
# lists can contain items of different types

names = ['Bianca', 'Bryan', 'Susan']
# print(names[:1])
# print(names[-1])
# print(names[:])
# names.insert(-1,'Buddy') #does something interesting
# # do this instead for insert at end:
# names.insert(len(names), 'Buddy2')
# print(names)

# looping
# for name in names:
#     print(name)

# looping a number of times with range
# for i in range(10):
#     print(i, end =" ")
#
# #
# my_list = [10, 20, 30, 40]
# for i in range(len(my_list)):
#     print(my_list[i], end =" ")
#
```

```
# my_sum = 0
# for i in range(1, 11):
#     my_sum += i
# print("Sum of first 10 natural number :", my_sum)

# for i in range(2, 26, 2):
#     print(i, end =" ")

# print(list(range(10)))


# spot the difference in these examples
# version 1
# price = input("how much did it cost? ")
# if float(price) >= 1.00:
#     tax = 0.07
#     print(tax)
# else:
#     tax = 0
#     print(tax)

# version 2: difference for larger number
# price = input("how much did it cost? ")
# if float(price) >= 1.00:
#     tax = 0.07
#     print(tax)
# else:
#     tax = 0
# print(tax)

# more elegant version 3
# price = input("how much did it cost? ")
# if float(price) >= 1.00:
#     tax = 0.07
# else:
#     tax = 0
# print(tax)
# boolean ifs
# gpa = .85
# lowest_grade = .7
# prize_winner = False
#
# if (gpa >= .85 and lowest_grade >= .7):
#     prize_winner = True
# else:
#     prize_winner = False



# some time later in the code check
# we don't use prize_winner == True <- c-ish syntax is
frowned upon in Python
# don't use prize_winner == True

# if prize_winner:
#     print("Special award needs to be printed")
# else:
#     print("no prize needed")


# string comparisons hold lots of potential
# error sources
# try input ireland or IRELAND
# case sensitivity!
# my_country = input("Where are you from?")
# if my_country == 'Ireland':
#     print("pot of gold for you")
# else:
#     print("no gold for you")

# better with conversion
# my_country = input("Where are you from? ")
# if my_country.upper() == 'IRELAND':
```

```
#     print("pot of gold for you")
# else:
#     print("no gold for you")

# even better with removing space padding
# my_country = input("Where are you from? ")
# if my_country.upper().strip() == 'IRELAND':
#     print("pot of gold for you")
# else:
#     print("no gold for you")

# else elif and default option else for
# Irish VAT rates
# see what happens if doing or without
vat_bands = ("Intra-Community transactions", "Vessels and Aircraft",
            "Agriculture", "Pharmaceuticals", "Shows",
            "Standard rate")
# my_vat = input("Which category are you in: ").strip()
# # nesting of ifs
# if my_vat in vat_bands:
#     if my_vat in ("Intra-Community transactions","Vessels and Aircraft"):
#         tax = 0
#     elif my_vat == "Agriculture":
#         tax = 0.048
#     elif my_vat in ("Pharmaceuticals", "Shows"):
#         tax = 0.135
#     else:
#         tax = 0.23
#     print(tax)
# else:
#     print("Category does not exist")

# take input string and calculate the number
# of digits and the number of characters
# in the input string
# my_input = input("Enter a sentence: ")
# digit_counter = 0
# char_counter = 0
# #
# for character in my_input:
#     if character.isdigit():
#         digit_counter = digit_counter + 1
#     elif character.isalpha():
#         char_counter = char_counter + 1
#     else:
#         pass # do nothing, we'll just ignore spaces
etc
#
# print("Number of digits: ", digit_counter)
# print("Number of characters: ", char_counter)

# enumerate
# count over iterables
# index = 0
# my_numbers = [1, 2, 3, 4, 5]
# while index < len(my_numbers):
#     print(my_numbers[index])
#     index += 1


# works fine, but now change my_numbers to a
non-sequence object
# like set
# index = 0
# my_numbers = {1, 2, 3, 4, 5}
# while index < len(my_numbers):
#     print(my_numbers[index])
#     index += 1

fruits = ("apple", "banana", "pear")
my_iterator = enumerate(fruits)
print(type(my_iterator))
print(next(my_iterator))

fruits = ("apple", "banana", "pear")
for index, fruit in enumerate(fruits):
   print("index is %d and value is %s " % (index,
fruit))


# manually need to keep track of the iterating variable
# here it is "i" if you don't use enumerate()
fruits = ("apple", "banana", "pear")
i = 0

for fruit in fruits:
   print("index is %d and value is %s " % (i, fruit))
   i += 1
```