Assignment #1

February 13, 2022

[4]:
```
from platform import python_version
print(python_version())
```

```
3.10.2
```

[2]:
```
import pandas as pd
print(pd.__version__)
```

```
1.4.0
```

[2]:
```
import numpy as num
print(num.__version__)
```

```
1.22.2
```

[5]:
```
pv = 1000
rate = 0.08
n = 40

for i in range(0,n):
    pv = (pv * rate) + pv

pv = "${:,.2f}".format(pv)
print (pv)
```

```
$21,724.52
```

[44]:
```
def happyHome(cats, dogs):
    cat_count = cats
    dog_count = dogs

    if cat_count > 0:
        has_cats = True
    elif cat_count <= 0:
        has_cats = False
    if dog_count > 0:
        has_dogs = True
    elif dog_count <= 0:
```

```
        has_dogs = False

    if has_cats == True and has_dogs == False:
        print('Happy Home because cats =', has_cats, 'and no dogs')
    elif has_dogs == True and has_cats == False:
        print('Happy Home because dogs =', has_dogs, 'and no cats')
    else:
        print(False, 'Not a Happy Home')


happyHome(0,2)
happyHome(3,0)
happyHome(1,2)
happyHome(-1,3)
happyHome(2,-3)
happyHome(-2,-1)
```

```
Happy Home because dogs = True and no cats
Happy Home because cats = True and no dogs
False Not a Happy Home
Happy Home because dogs = True and no cats
Happy Home because cats = True and no dogs
False Not a Happy Home
```

[25]:
```
first10 = list(range(1,11))
second10 = list(range(11,21))
firstprimes = []
secondprimes = []

print(first10)
print(second10)

prime = [True for i in range(21)]
p = 2

while (p * p <= 20):
    if (prime[p] == True):
        for i in range(p * p, 21, p):
            prime[i] = False
    p += 1

for p in range(2, 21):
    if prime[p] and p <= 10:
        firstprimes.append(p)
    elif prime[p] and p > 10:
        secondprimes.append(p)
```

```
first10 = [x for x in first10 if x in firstprimes]
second10 = [x for x in second10 if x in secondprimes]

print(first10)
print(second10)

first10.append(second10)
print(first10)

first10.extend(second10)
print(first10)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
[11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
[2, 3, 5, 7]
[11, 13, 17, 19]
[2, 3, 5, 7, [11, 13, 17, 19]]
[2, 3, 5, 7, [11, 13, 17, 19], 11, 13, 17, 19]
```

[17]:
```
money1 = {'USD':1.00, 'BTC':51013.93, 'EUR':1.131735}
money1['CAD'] = 0.79

def currencyconversion(base, convert):
    exchange = money1[base] * (1 / money1[convert])
    print('One', base, '=', exchange, convert)

currencyconversion('EUR', 'CAD')
currencyconversion('BTC', 'USD')
currencyconversion('EUR', 'BTC')
currencyconversion('USD', 'CAD')
currencyconversion('CAD', 'BTC')

currencyconversion('EUR', 'YEN')
```

```
One EUR = 1.4325759493670884 CAD
One BTC = 51013.93 USD
One EUR = 2.2184822851327077e-05 BTC
One USD = 1.2658227848101264 CAD
One CAD = 1.5485966284111026e-05 BTC
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
Input In [17], in <module>
     11 currencyconversion('USD', 'CAD')
     12 currencyconversion('CAD', 'BTC')
---> 14 currencyconversion('EUR', 'YEN')
```

```
Input In [17], in currencyconversion(base, convert)
      4 def currencyconversion(base, convert):
----> 5     exchange = money1[base] * (1 / money1[convert])
      6     print('One', base, '=', exchange, convert)

KeyError: 'YEN'
```

[30]:
```
x1 = 5
x2 = 9
y1 = 12
y2 = 45

point1 = (x1, y1)
point2 = (x2, y2)

magnitude = ((point1[0]**2+point1[1]**2)+(point2[0]**2+point2[1]**2))**0.5

print(magnitude)
```

```
47.69696007084728
```