flip function

```python
# generate a 'flip'
def flip(num = 1):
    flips = []

    for i in range(num):
        num = np.random.uniform(low=0.0, high=1.0)
        if num > 0.5:
            flips.append('H')            # should be doing yield here if you know 'generators'
        else:
            flips.append('T')
    return flips
```

```python
flips = flip(10)
print(flips)
```

```
['T', 'T', 'T', 'T', 'H', 'T', 'H', 'H', 'H', 'H']
```

]:

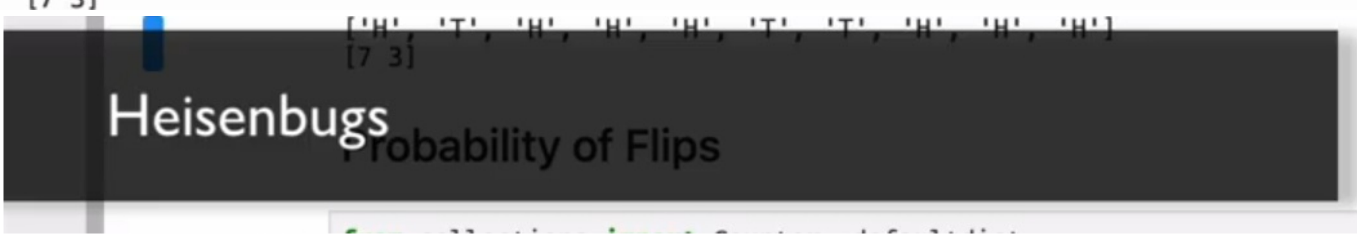
```python
values, counts = np.unique(flips, return_counts=True)
```

you need to reproduce the problem and experiment

do not hard code made the code predicatble

```python
def flip(num = 1):
    flips = []

    for i in range(num):
        num = np.random.uniform(low=0.0, high=1.0)
        if num > 0.5:
            flips.append('H')          # should be doing yield here if you know 'generators'
        else:
            flips.append('T')
    return flips


# Flip
flips = flip(10)
values, counts = np.unique(flips, return_counts=True)

# print values/stats
# print(flip())
print(flips)
print(counts)
```

```
['H', 'T', 'H', 'H', 'H', 'T', 'T', 'H', 'H', 'H']
[7 3]
```



Reproducable Randomness

Seeds not to take the real world instead take from the zero

```
# computers are 'deterministic'. You can not do 'random' in computers!
# So, you start with some 'seed' then do deterministic things
# this is called pseudo-randomness

# sometimes you want to suppress this!

np.random.seed(0)   # random numbers and seed
```

must import seed before start the experiment

```
]: import matplotlib
   import matplotlib.pyplot as plt
   %matplotlib inline

   import numpy as np

   import seaborn as sns
   sns.set(color_codes=True)
   sns.set_style("white")     # See more styling options here: https://seaborn.pydata.org/tutorial/a


   # np.random.seed(0)   # random numbers and seed


   # generate a 'flip'
   def flip(num = 1):
       flips = []

       for i in range(num):
           num = np.random.uniform(low=0.0, high=1.0)
           if num > 0.5:
               flips.append('H')            # should be doing yield here if you know 'generators'
           else:
               flips.append('T')
       return flips


   # Flip
```

Probability of Flips

[ ]:

```python
from collections import Counter, defaultdict


def get_freqs(flips):
    keys = Counter(flips).keys()
    vals = Counter(flips).values()

    # print(keys)
    # print(vals)

    # return dict(zip(keys, vals))     # bug: what if there are no 'H' or no 'T'

    return defaultdict(int, dict(zip(keys, vals)))
```

```python
def get_freqs(flips):
    keys = Counter(flips).keys()
    vals = Counter(flips).values()

    # print(keys)
    # print(vals)

    # return dict(zip(keys, vals))    # bug: what if there are no 'H' or no 'T'

    return defaultdict(int, dict(zip(keys, vals)))
```

making Dictionary forom the counter of values problem what happen there is not tail that leads to key error.

We will use the default dictionary that not lead to the key error.

Probability of Flips

```
from collections import Counter, defaultdict


def get_freqs(flips):
    keys = Counter(flips).keys()
    vals = Counter(flips).values()

    # print(keys)
    # print(vals)

    # return dict(zip(keys, vals))     # bug: what if there are no 'H' or no 'T'

    return defaultdi#ct(int, dict(zip(keys, vals)))
```

if

Experiment: Prob calculated based on 1 flip upto N flips

[ ]:
```
maximum_flips = 1000

probs = []
for num_flips in range(1, maximum_flips):
    flips = flip(num_flips)
    freqs = get_freqs(flips)
    prob_h = freqs['H'] / len(flips)

    probs.append(prob_h)

# print(probs)
```

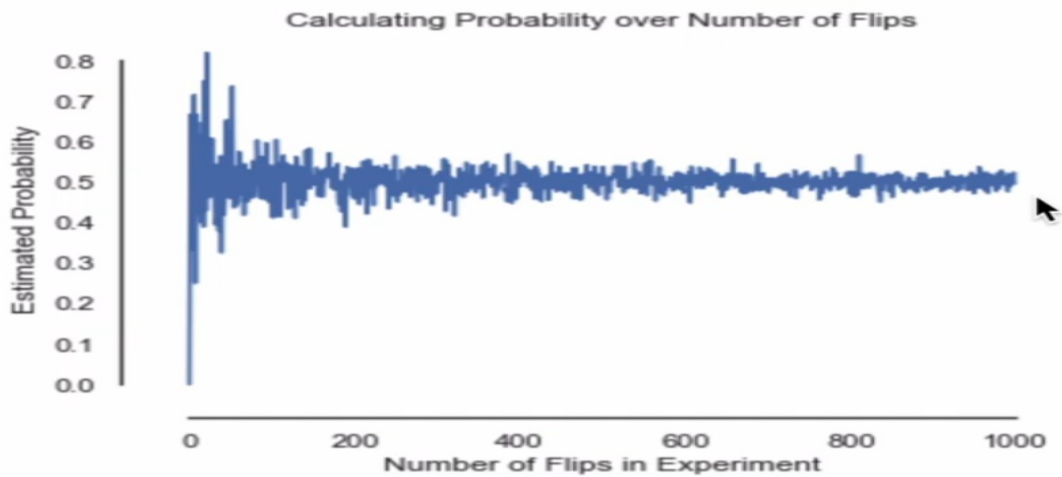
[ ]:
```
print(freqs)
```

i tose the coin 1000 times then probaility of head and tail in represented in the graph

Tas 1 get head
Tas 2 get head two time
Tas 3 get head 3 times

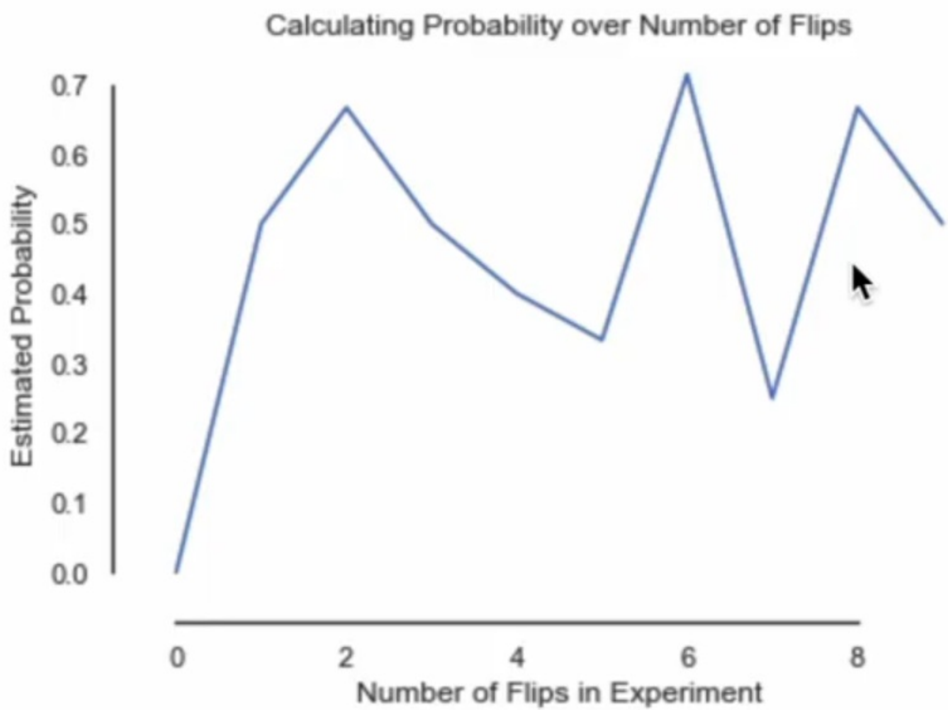
using this experiment we have proof the proabilty of head more tne tail

```
[82]: plt.plot(probs)
      plt.ylabel('Estimated Probability')
      plt.xlabel('Number of Flips in Experiment');
      plt.title("Calculating Probability over Number of Flips")
      sns.despine(offset=10, trim=True);  # move axes away
      plt.show()
```



The proability of last 10 value
form the prabaility

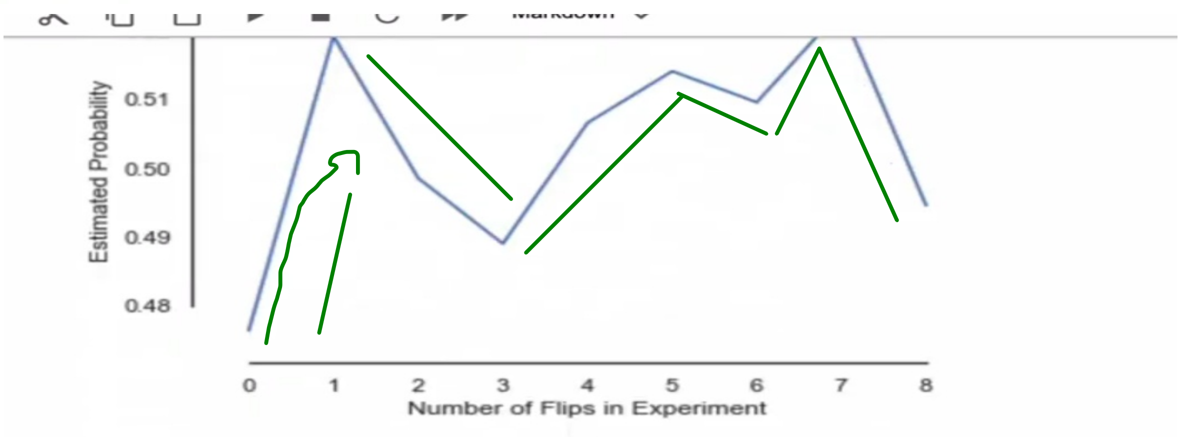
```
[83]: plt.plot(probs[:10])
      plt.ylabel('Estimated Probability')
      plt.xlabel('Number of Flips in Experiment');
      plt.title("Calculating Probability over Number of Flips")
      sns.despine(offset=10, trim=True);  # move axes away
      plt.show()
```



Frist Values of probaility of

the graph

```
plt.plot(probs[maximum_flips-10:])
plt.ylabel('Estimated Probability')
plt.xlabel('Number of Flips in Experiment');
plt.title("Calculating Probability over Number of Flips")
sns.despine(offset=10, trim=True);  # move axes away
plt.show()
```



this is libaray for intractive the grpah more efficeintly

Bokeh For Interactive Plots

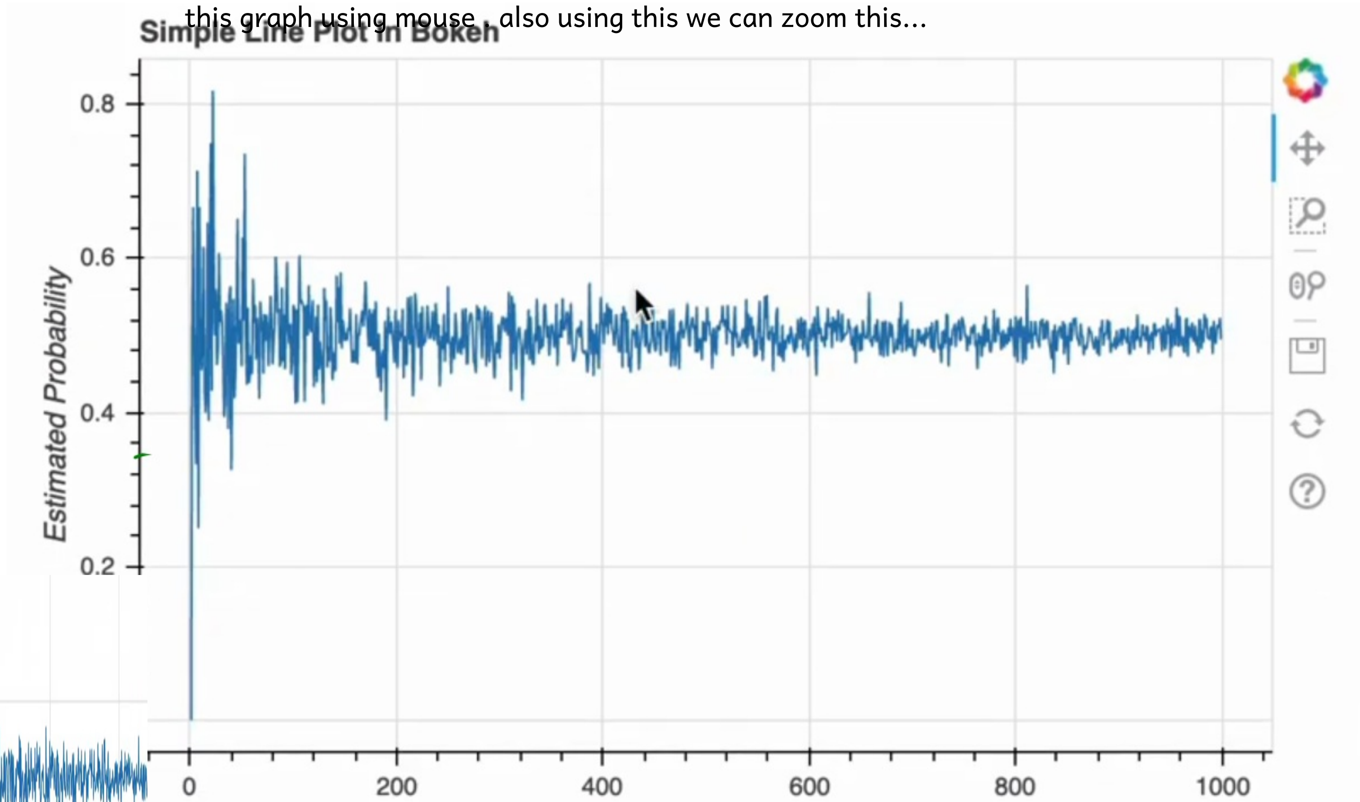
[ ]:
```
!pip install bokeh
```

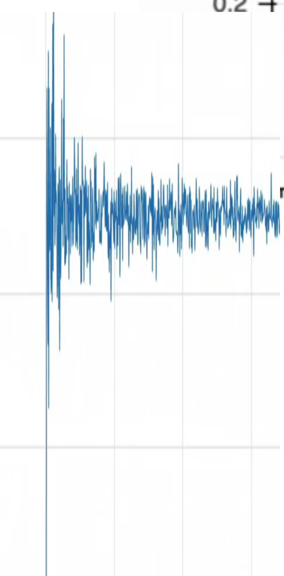
[ ]:
```
from bokeh.io import show, output_notebook
from bokeh.plotting import figure

output_notebook()
```

[ ]:
```
p = figure(title="Simple Line Plot in Bokeh",
           x_axis_label='Number of Flips in Experiment',
           y_axis_label='Estimated Probability',
           plot_width=580, plot_height=380)
```

[ ]:
```
# Add a line renderer with legend and line thickness
x = range(1, maximum_flips)
p.line(x=x, y=probs)

# Show the results
show(p)
```

This is output of graph of bokey libary whic i told the most efficeintly way of representation of graph .you can also pan this graph using mouse , also using this we can zoom this...





Most important Question :
what is chance of head in one flip?
what is chance of head in two flip?
what is chance of head in 100 flip?
what is chance of head in 1000 flip?

Then we do the experiment and calculate one times two times upto 1000 times.
But also we can't experiment more then one times let Election we want to predict who will win in this case you can't do the experiemnt you have to predict from the previos knowledge . Other example Who will the Tournament?

Quantifying Chances

- Flip a coin :

Q: "What are the chances that it will land on its head?" hard

Not: "If I flip it 10 times, how many will be heads?" easy