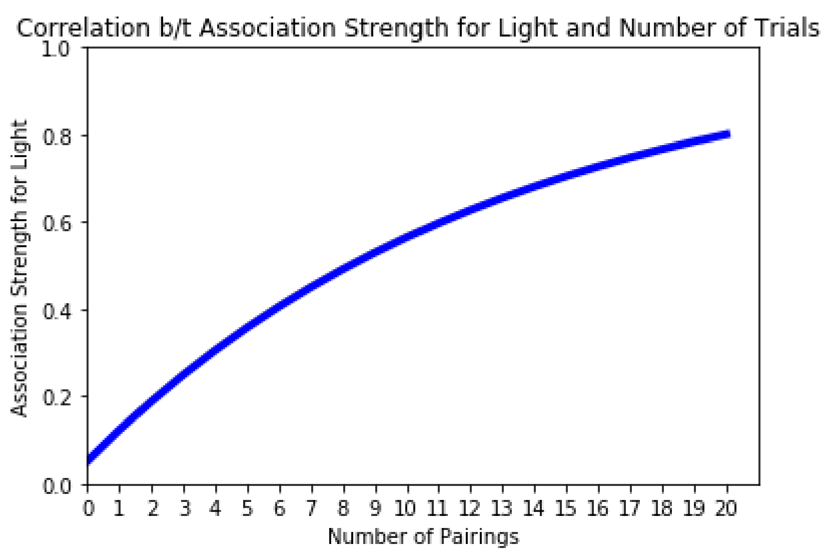
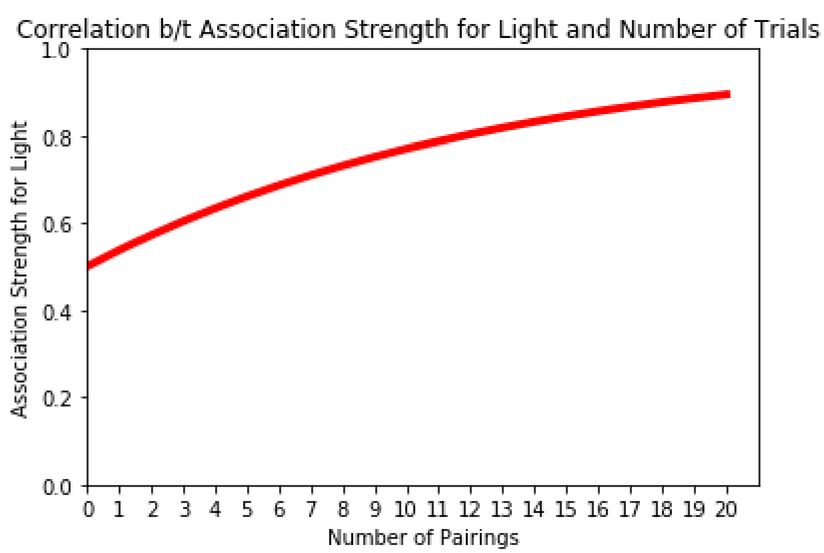
1a

In [13]:
```
import matplotlib.pyplot as plt

def Vx(alpha, beta, lambdah, initial):
    change_Vx = alpha * beta * (lambdah - initial)

    return initial + change_Vx


def trial_runner(alpha, beta, lambdah, initial, num_trials):
    index = 1
    association_list = [initial]
    while index < num_trials:
        next = Vx(alpha, beta, lambdah, initial)
        initial = next
        association_list.append(initial)
        index+=1
    return association_list


x = list(range(0, 21))
y = list(trial_runner(0.75, 0.1, 1.0, 0.05, 21))

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xlim(1, 21)
ax.set_ylim(0, 1)
plt.plot(x, y, color='blue', linewidth=4)
plt.xlabel('Number of Pairings')
plt.ylabel('Association Strength for Light')
plt.title('Correlation b/t Association Strength for Light and Number of
 Trials ')
#plt.scatter(x, y, color='darkgreen', marker='^')
ax.xaxis.set(ticks=range(0,21),)
plt.savefig('foo.png')
plt.show()
```



In [25]:

```
import matplotlib.pyplot as plt

def Vx(alpha, beta, lambdah, initial):
    change_Vx = alpha * beta * (lambdah - initial)

    return initial + change_Vx


def trial_runner(alpha, beta, lambdah, initial, num_trials):
    index = 1
    association_list = [initial]
    while index < num_trials:
        next = Vx(alpha, beta, lambdah, initial)
        initial = next
        association_list.append(initial)
        index+=1
    return association_list


x = list(range(0, 21))
y = list(trial_runner(0.75, 0.1, 1.0, 0.5, 21))

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xlim(1, 21)
ax.set_ylim(0, 1)
plt.plot(x, y, color='red', linewidth=4)
plt.xlabel('Number of Pairings')
plt.ylabel('Association Strength for Light')
plt.title('Correlation b/t Association Strength for Light and Number of
 Trials ')
#plt.scatter(x, y, color='darkgreen', marker='^')
ax.xaxis.set(ticks=range(0,21),)
plt.savefig('foo.png')
plt.show()
```



1b

In [29]:
```
y = list(trial_runner(0.75, 0.1, 1.0, 0.05, 30))
x = list(range(0, 30))
for x in x:
    print(x, y[x])
```

```
0 0.05
1 0.12125000000000001
2 0.18715625000000002
3 0.24811953125000002
4 0.30451056640625
5 0.35667227392578127
6 0.4049218533813477
7 0.4495527143777466
8 0.49083626079941556
9 0.5290235412394594
10 0.5643467756464999
11 0.5970207674730125
12 0.6272442099125365
13 0.6552008941690962
14 0.681060827106414
15 0.704981265073433
16 0.7271076701929254
17 0.747574594928456
18 0.766506500308821 8
19 0.7840185127856601
20 0.8002171243267356
21 0.8152008400022305
22 0.8290607770020632
23 0.8418812187269085
24 0.8537401273223904
25 0.8647096177732111
26 0.8748563964402203
27 0.8842421667072038
28 0.8929240042041635
29 0.9009547038888512
```

In [23]:
```
### It takes 20 trials to reach .8 if the initial association is 0.05.
```

1c

In [5]:
```
x = list(range(0, 16))
y = list(trial_runner(1.17, 0.1, 1.0, 0, 16))
for x in x:
    print(x, y[x])
```

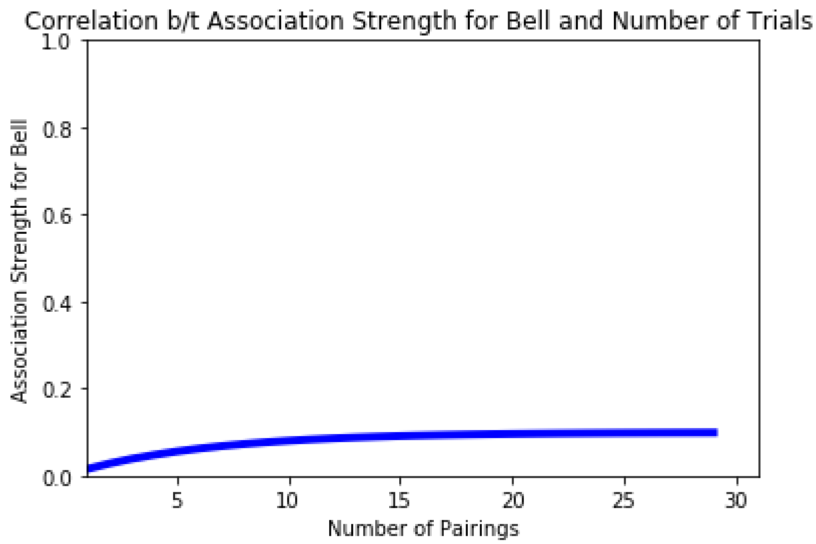
```
0 0
1 0.11699999999999999
2 0.22031099999999998
3 0.31153461299999996
4 0.39208506327899995
5 0.46321111087535694
6 0.5260154109029401
7 0.5814716078272961
8 0.6304394297115025
9 0.6736780164352567
10 0.7118576885123317
11 0.7455703389563888
12 0.7753386092984913
13 0.8016239920105678
14 0.8248339849453313
15 0.8453284087067275
```

In [6]:
```
### By using the numerical trial & error, I was able to approximate the
 salience value to be 1.17 that surpasses 0.80 association by 13th trial
s, assuminng the initial association is 0.
```

2

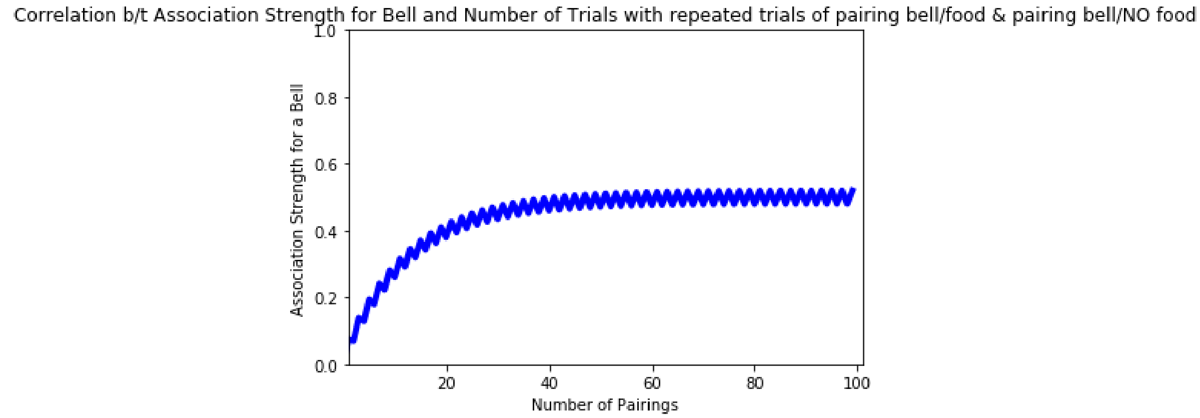
In [15]:

```python
import matplotlib.pyplot as plt

def Vax(alpha, beta, lambdah, a, x):
    change_Vx = alpha * beta * (lambdah - (a+x))

    return a + change_Vx


def trial_runner(alpha, beta, lambdah, a, x, num_trials):
    index = 1
    association_list = [(a, x)]
    while index < num_trials:
        new_a = Vax(alpha, beta, lambdah, a, x)
        new_x = Vax(alpha, beta, lambdah, x, a)
        association_list.append((new_a, new_x))
        a = new_a
        x = new_x
        index+=1
    return association_list


x = list(range(0, 30))
y = [y for (x, y) in trial_runner(0.75, 0.1, 1, 0.8, 0, 30)]



fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xlim(1, 31)
ax.set_ylim(0, 1)
plt.plot(x, y, color='blue', linewidth=4)
plt.xlabel('Number of Pairings')
plt.ylabel('Association Strength for Bell')
plt.title('Correlation b/t Association Strength for Bell and Number of T
rials ')

#plt.scatter(x, y, color='darkgreen', marker='^')
#ax.xaxis.set(ticks=range(0,30),)
plt.savefig('foo.png')
plt.show()
```


Correlation b/t Association Strength for Bell and Number of Trials
Association Strength for Bell
1.0
0.8
0.6
0.4
0.2
0.0
5
10
15
20
25
30
Number of Pairings

3a

In [19]:
```
import matplotlib.pyplot as plt

def Vx(alpha, beta, lambdah, initial):
    change_Vx = alpha * beta * (lambdah - initial)

    return initial + change_Vx

def Vex(alpha, beta, lambdah, x):
    change_Vx = alpha * beta * (-x)

    return x + change_Vx

def l_e_trial_runner(alpha, beta, lambdah, initial, num_trials):
    index = 1
    association_list = [initial]
    while index < num_trials:
        if index % 2 == 1:
            initial = Vx(alpha, beta, lambdah, initial)
        else:
            initial = Vex(alpha, beta, lambdah, initial)
        association_list.append(initial)
        index+=1
    return association_list

x = list(range(0, 100))
y = list(l_e_trial_runner(0.75, 0.1, 1, 0, 100))

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xlim(1, 101)
ax.set_ylim(0, 1)
plt.plot(x, y, color='blue', linewidth=4)
plt.xlabel('Number of Pairings')
plt.ylabel('Association Strength for a Bell')
plt.title('Correlation b/t Association Strength for Bell and Number of T
rials with repeated trials of pairing bell/food & pairing bell/NO food')
#plt.scatter(x, y, color='darkgreen', marker='^')
#ax.xaxis.set(ticks=range(0,101),)
plt.savefig('foo.png')
plt.show()

#for x in x:
 #   print(x, y[x])
```

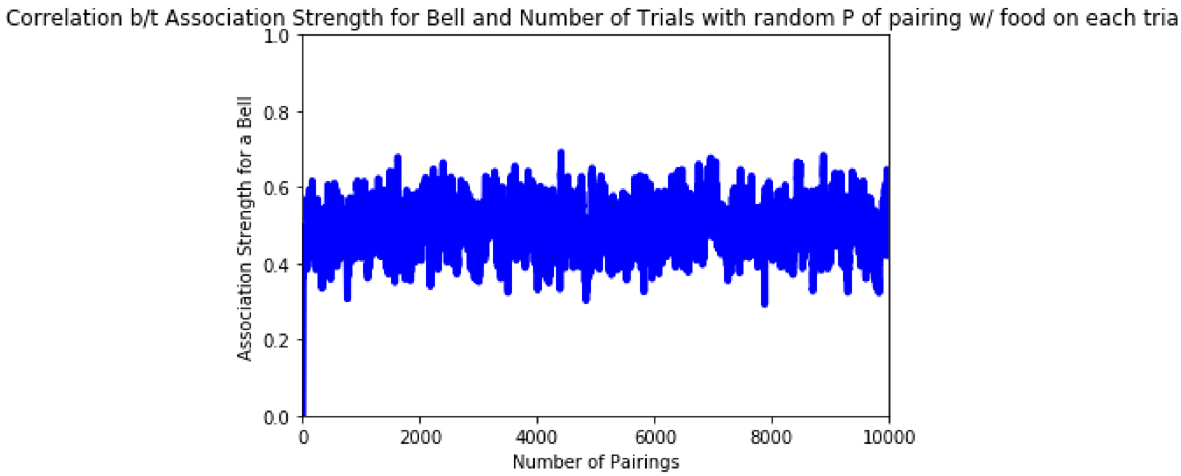


In [9]:
```
### As trials continue, the associaciation strength gradually increases and plateaus around 0.5 of association strength since 0.5 is where the learning formula and extinction formula cancel each other out, hence the association difference is close to 0, which is exactly why the line plateau towards more trials.
```

3b

In [20]:

```
import matplotlib.pyplot as plt
import random

def VxVex(alpha, beta, lambdah, initial, p):
    change_VxVex = p * alpha * beta * (lambdah - initial) + (1-p) * alph
a * beta * (-initial)

    return initial + change_VxVex


def trial_runner_with_random_p(alpha, beta, lambdah, initial, num_trials
):
    index = 1
    association_list = [initial]

    while index < num_trials:
        p = random.random()
        result = VxVex(alpha, beta, lambdah, initial, p)
        initial = result
        association_list.append(initial)
        index+=1
    return association_list

x = list(range(0, 10000))
y = list(trial_runner_with_random_p(0.75, 0.1, 1, 0, 10000))

#for x in x:
 #   print(x, y[x], p_list[x])

fig = plt.figure()
ax = fig.add_subplot(111)
ax.set_xlim(1, 10000)
ax.set_ylim(0, 1)
plt.plot(x, y, color='blue', linewidth=4)
plt.xlabel('Number of Pairings')
plt.ylabel('Association Strength for a Bell')
plt.title('Correlation b/t Association Strength for Bell and Number of T
rials with random P of pairing w/ food on each trial')

#plt.scatter(x, y, color='darkgreen', marker='^')
#ax.xaxis.set(ticks=range(0,101),)
plt.savefig('foo.png')
plt.show()

#for x in x:
 #   print(x, y[x])
```



In [11]:
```
### At a computational level, even if probability of paring with bell &
 food and probability of pairing with bell & no food, the averages of ea
ch probability is 0.5. That is why the association strength oscillates a
round 0.5.
```

4

In [12]:
```
### Value of Salience means how strong signals captures subject's attent
ion. The value of learning rate is how fast subjects learn. The reason w
hy we think they are different factors is that even if salience does hav
e a positive correlation with learning rate, it is not an identical fact
or. For example, a sound of bell on a subject will have a stronger salie
nce on a subject than a sound of a clock ticking. Just because it captur
ed attention of a subject in a higher rate does not necessarily mean tha
t their learning process was in place.

#In an instance of experiment, monkeys are presumed to learn slower than
 humans, which means that their learning rate will be lower than humans,
 which in turn means that their salience can be presumed to be lower tha
n humans as well. In order to disentangle the salience and learning rat
e, we can have a setting where salience for both monkeys and humans are
 kept same, but humans will demonstrate to have a higher learning rate.
 Therefore, salience and learning rate is not an identical factor.
```