6장 확률과 통계

In [2]:
```python
# 필요 라이브러리 선언
import numpy as np
import scipy.special as scm
import matplotlib.pyplot as plt
```

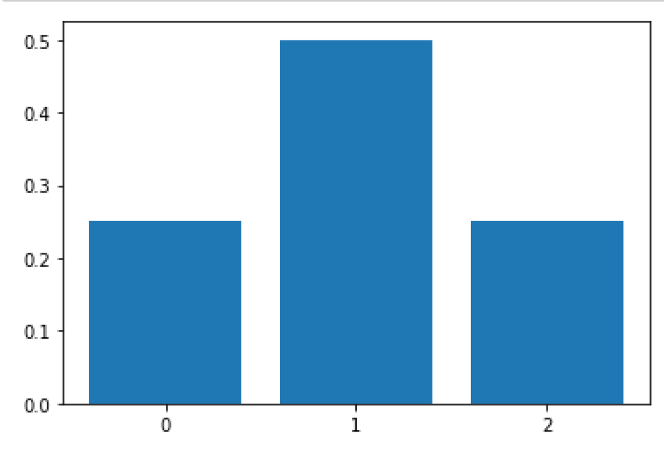
In [3]:
```python
# PDF 출력
from IPython.display import set_matplotlib_formats
set_matplotlib_formats('png', 'pdf')
```

히스토그램 그리기

n=2인 경우

그림 6-1

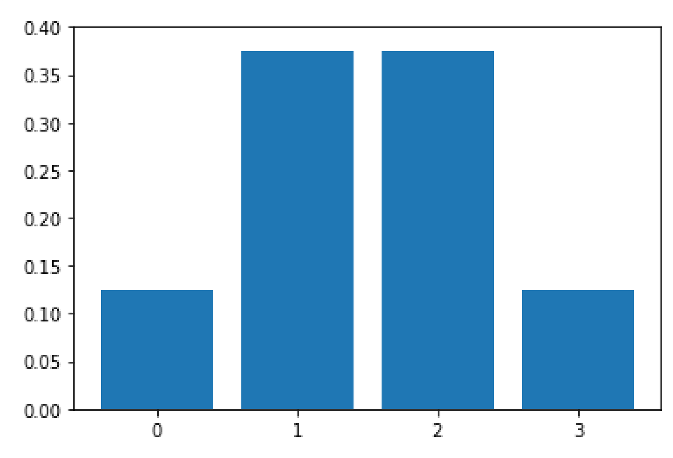
In [4]:
```python
N = 2
M = 2**N
X = range(N + 1)
plt.bar(X, [scm.comb(N, i) / M for i in X])
plt.xticks(X, [str(i) for i in X])
plt.show()
```



n=3인 경우

그림 6-2

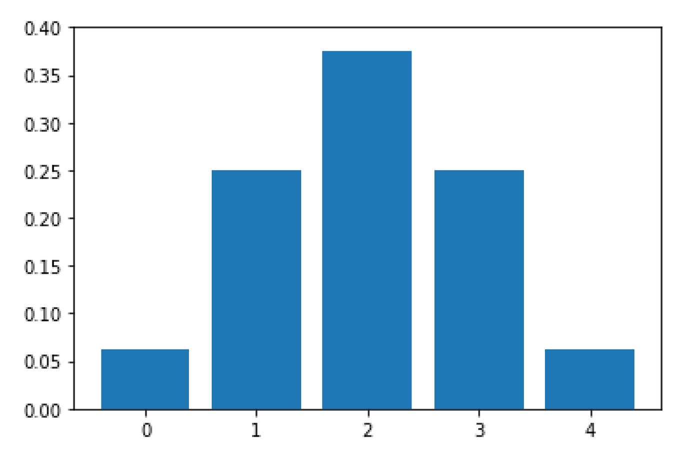
In [5]:
```python
N = 3
M = 2**N
X = range(N + 1)
plt.bar(X, [scm.comb(N, i) / M for i in X])
plt.xticks(X, [str(i) for i in X])
plt.yticks(np.arange(0, 0.405, 0.05))
plt.show()
```



n = 4인 경우

그림 6-3

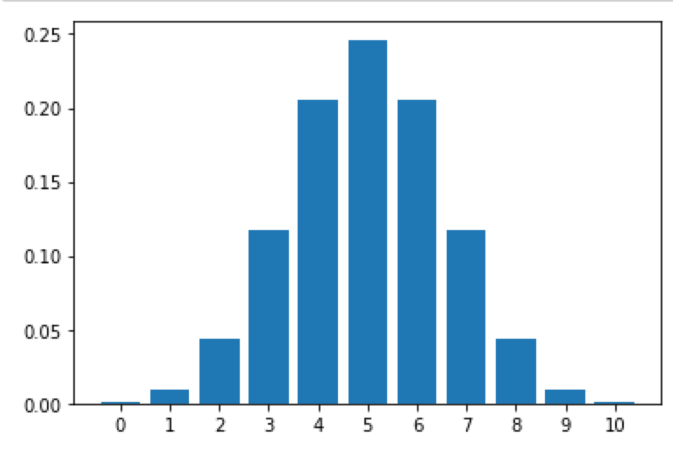
In [6]:
```python
N = 4
M = 2**N
X = range(N + 1)
plt.bar(X, [scm.comb(N, i) / M for i in X])
plt.xticks(X, [str(i) for i in X])
plt.yticks(np.arange(0, 0.405, 0.05))
plt.show()
```



n=10인 경우

그림 6-4

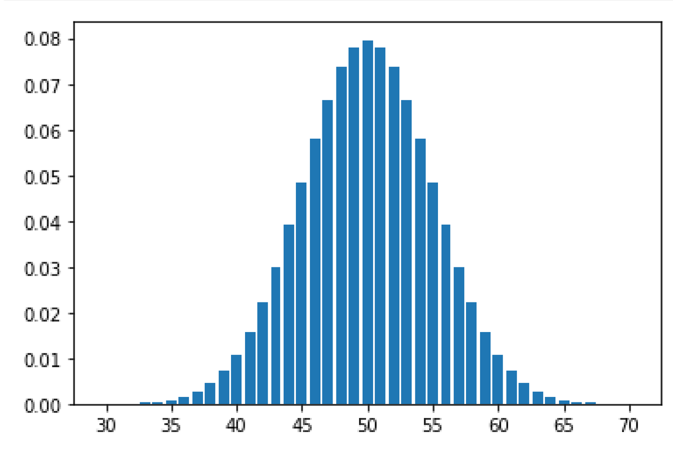
In [7]:
```python
N = 10
M = 2**N
X = range(N + 1)
plt.bar(X, [scm.comb(N, i) / M for i in X])
plt.xticks(X, [str(i) for i in X])
plt.show()
```



n=100인 경우

그림 6-5

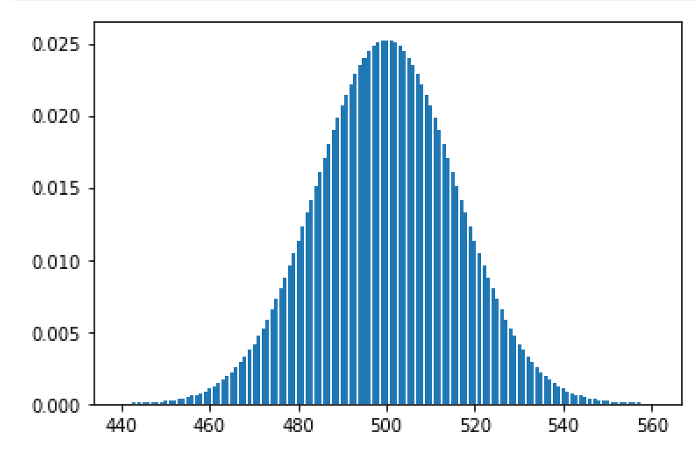
In [8]:
```python
N = 100
M = 2**N
X = range(30, 71)
plt.bar(X, [scm.comb(N, i) / M for i in X])
plt.show()
```



n=1000인 경우

그림 6-6

In [9]:
```python
N = 1000
M = 2**N
X = range(440, 561)
plt.bar(X, [scm.comb(N, i) / M for i in X])
plt.show()
```



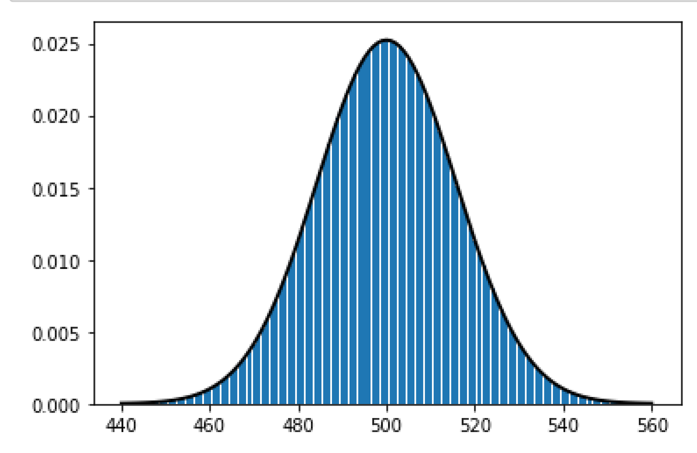
정규분호함수와 히스토그램

그림 6-8

In [11]:
```python
import numpy as np
import scipy.special as scm
import matplotlib.pyplot as plt

# 정규분포함수의 정의
def gauss(x, n):
    m = n / 2
    return np.exp(-(x - m)**2 / m) / np.sqrt(m * np.pi)

# 이항분포의 그래프와 정규분포 그래프를 함께 그린 모습
N = 1000
M = 2**N
X = range(440, 561)
plt.bar(X, [scm.comb(N, i) / M for i in X])
plt.plot(X, gauss(np.array(X), N), c='k', linewidth=2)
plt.show()
```



적분한 결과값

그림 6-10

In [12]:
```python
import numpy as np
from scipy import integrate
def normal(x):
    return np.exp(-((x - 500)**2) / 500) / np.sqrt(500 * np.pi)
integrate.quad(normal, 0, 480)
```

Out[12]:
```
(0.10295160536603419, 1.1220689434463503e-13)
```

시그모이드 함수(sig)와 정규분포함수(std)

그림 6-11

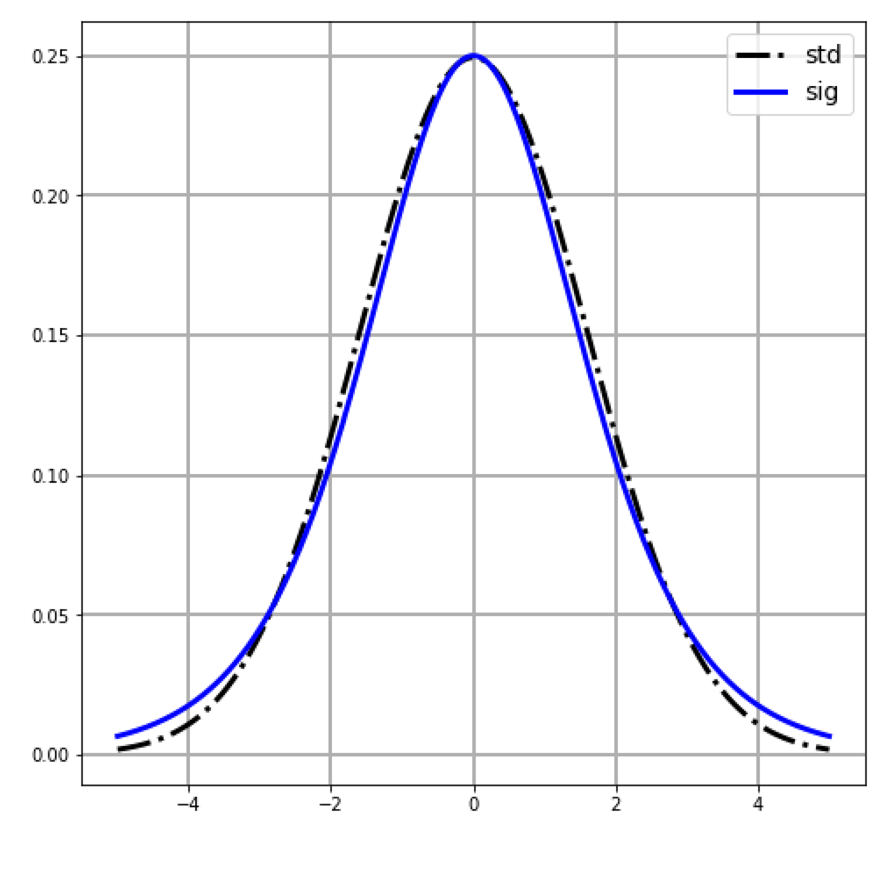
In [13]:
```python
import numpy as np
import matplotlib.pyplot as plt

# 정규분포함수
def std(x, sigma=1):
    return (np.exp(-(x / sigma)**2 / 2)) / (np.sqrt(2 * np.pi) * sigma)

# 시그모이드 함수(확률분포함수)
def sigmoid(x):
    return (1 / (1 + np.exp(x)))

# 좌표 계산
x = np.linspace(-5, 5, 1000)
y_std = std(x, 1.6)
sig = sigmoid(x)
y_sig = sig * (1 - sig)

# 그래프 그리기
plt.figure(figsize=(8, 8))
plt.plot(x, y_std, label='std', c='k', lw=3, linestyle='-.')
plt.plot(x, y_sig, label='sig', c='b', lw=3)
plt.legend(fontsize=14)
plt.grid(lw=2)
plt.show()
```



p를 변수로 한 가능도함수의 그래프

그림 6-12

In [15]:
```python
import numpy as np
import matplotlib.pyplot as plt

def L(p, n, k):
    return ((p**k) * ((1 - p)**(n - k)))

x = np.linspace(0, 1, 1000)
y = L(x, 5, 2)
x0 = np.asarray([0.4, 0.4])
y0 = np.asarray([0, L(0.4, 5, 2)])

plt.figure(figsize=(6, 6))
plt.plot(x, y, c='b', lw=3)
plt.plot(x0, y0, linestyle='dashed', c='k', lw=3)
plt.xticks(size=16)
plt.yticks(size=16)
plt.grid(lw=2)
plt.xlabel('p', fontsize=16)
plt.ylabel('L(p)', fontsize=16)
plt.show()
```