# Discrete Fourier Transform

Transform into frequency space:

* Find a set of periodic functions (waves, sine curves, wiggles),
  such that when the waves are summed together we get back our
  signal.

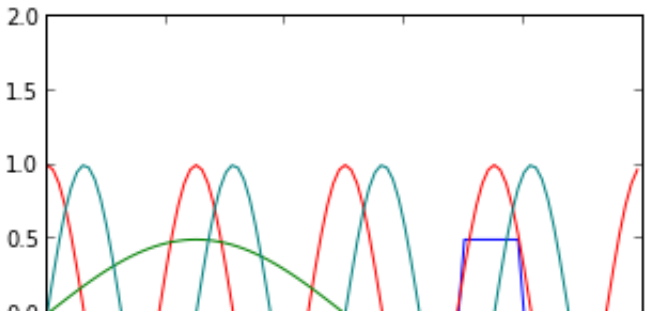
* A wave:
  - frequency (how fast does it wiggle?)
  - amplitude (how tall are the wiggles?)
  - phase     (where do the wiggles start?)

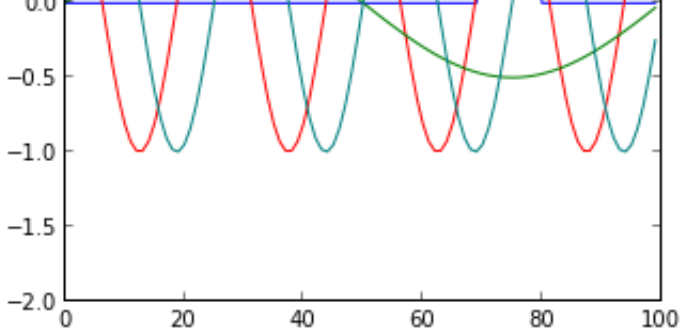
In [150]:
```
np.set_printoptions(precision=2, suppress=True)
X = np.zeros(100)
X [70:80] = 0.5

k = 1
N = 100
n = np.arange(N)
exp1 = e**(1j * 2 * pi * k * n / N) * 0.5
k = 4
exp2 = e**(1j * 2 * pi * k * n / N)

plt.plot(X, 'b', exp2.real, 'r', exp1.imag, 'g', exp2.imag, 'teal')
plt.ylim(-2,2)
```

Out[150]: (-2, 2)






## X_k = Sum(for n = 0 to N) x_n * e ^ (-i * 2 * pi * k * n / N)

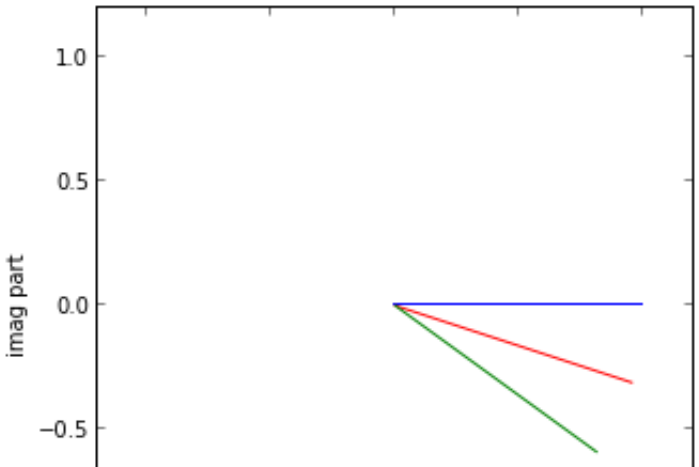
Multiplying every element of X by a coefficient, and then taking the sum.

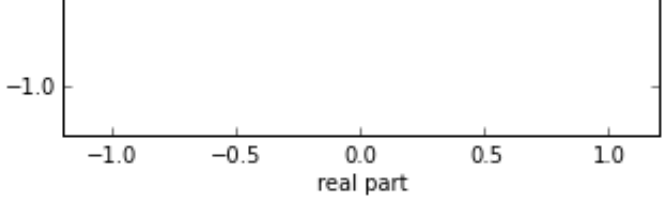
Let's take a closer look at the coefficients!

In [135]:
```
exp = e**(-1j * 2 * pi * 1 * np.arange(100) / 100)
print exp[0:10]
plt.figsize(5,5)
plt.plot([0,exp.real[0]],[0,exp.imag[0]], 'b',
         [0,exp.real[5]],[0,exp.imag[5]], 'r',
         [0,exp.real[10]],[0,exp.imag[10]], 'g')
plt.xlabel('real part')
plt.ylabel('imag part')
plt.xlim(-1.2,1.2)
plt.ylim(-1.2,1.2)
```

```
[ 1.00+0.j    1.00-0.06j  0.99-0.13j  0.98-0.19j  0.97-0.25j  0.95-
0.31j
  0.93-0.37j  0.90-0.43j  0.88-0.48j  0.84-0.54j]
```

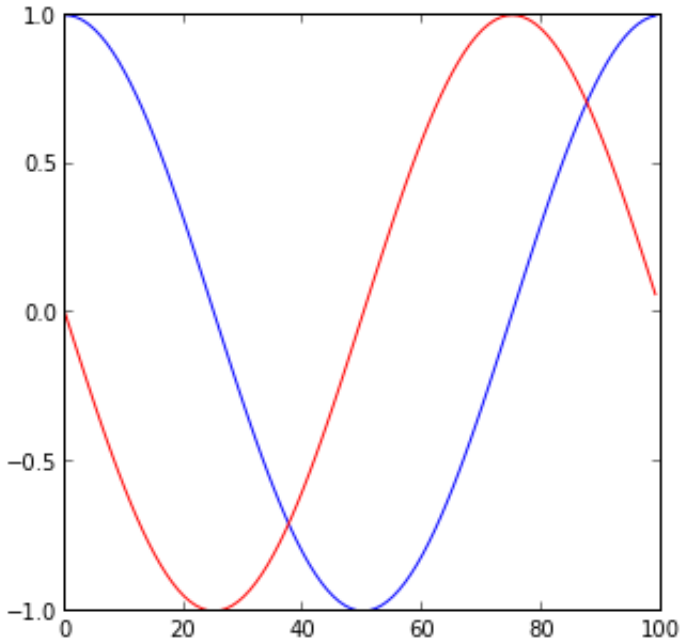
Out[135]: (-1.2, 1.2)





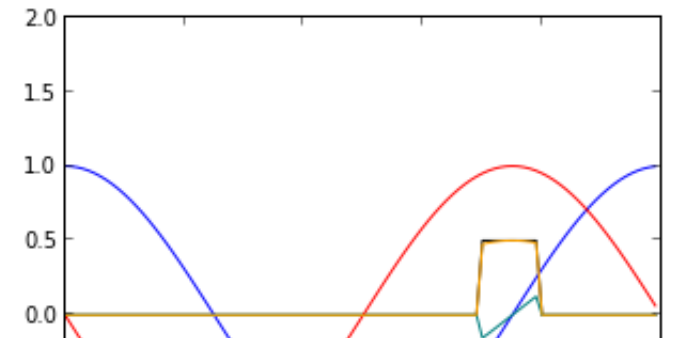

In [136]:
```
plt.plot(exp.real, 'b', exp.imag, 'r')
```

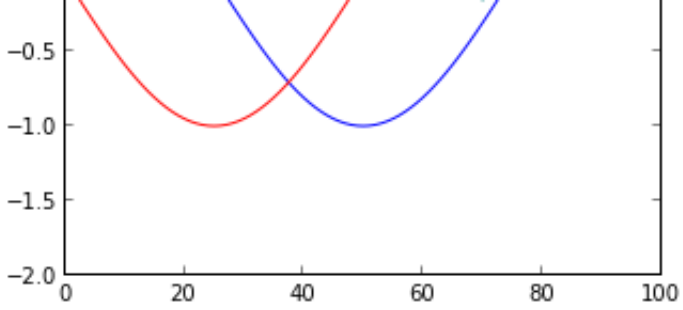
Out[136]:
```
[<matplotlib.lines.Line2D at 0x7914110>,
 <matplotlib.lines.Line2D at 0x7914390>]
```




In [148]:
```
k = 1
N = 100
n = np.arange(N)
exp = e**(-1j * 2 * pi * k * n / N)
resp = X * exp
plt.plot(X, 'k', exp.real, 'b', resp.real, 'teal')
plt.plot(exp.imag, 'r', resp.imag, 'orange')
plt.ylim(-2,2)
```

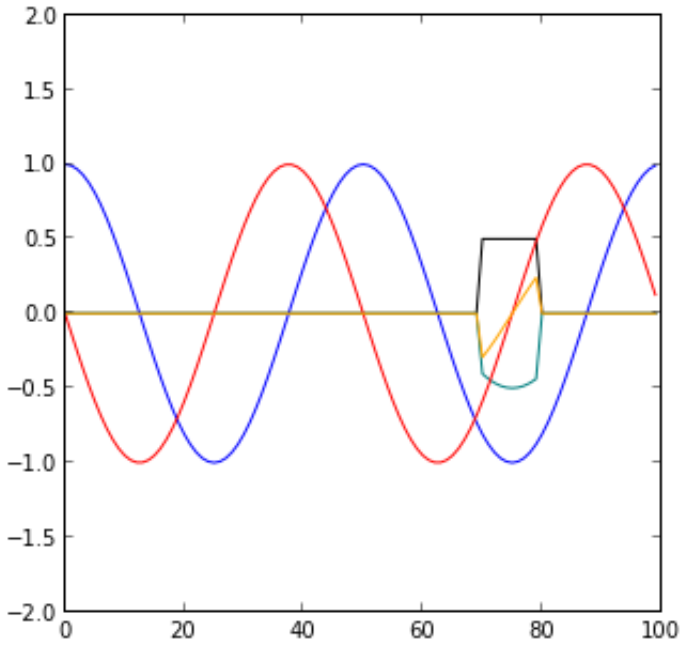
Out[148]:
```
(-2, 2)
```






In [149]:
```
k = 2
N = 100
n = np.arange(N)
exp = e**(-1j * 2 * pi * k * n / N)
resp = X * exp
plt.plot(X, 'k', exp.real, 'b', resp.real, 'teal')
plt.plot(exp.imag, 'r', resp.imag, 'orange')
plt.ylim(-2,2)
```

Out[149]: (-2, 2)



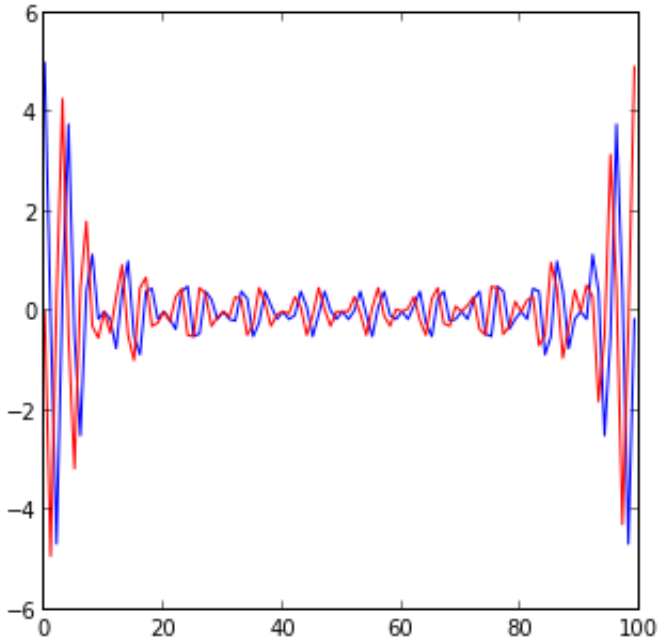

In [140]:
```
resp = np.zeros(100) + 0j

for k in range(100):
    resp[k] = (X * e ** (1j * 2 * pi * k * n / N)).sum()

plt.plot(resp.real, 'b', resp.imag, 'r')
```

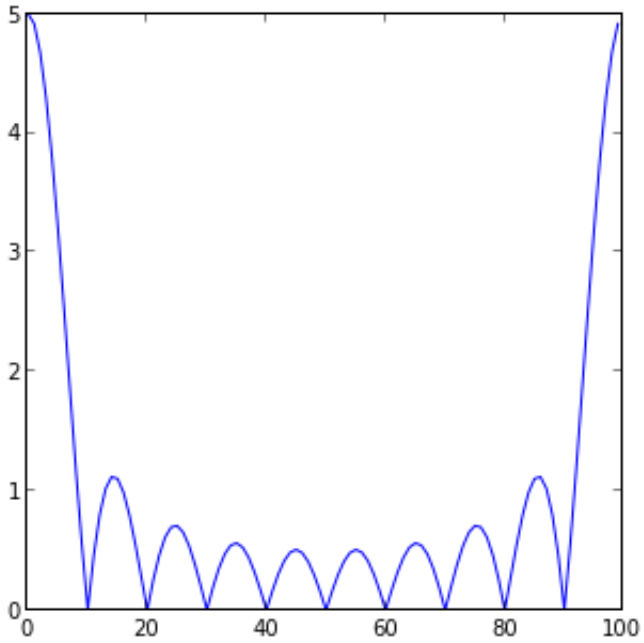
Out[140]: [<matplotlib.lines.Line2D at 0x7934110>,

<matplotlib.lines.Line2D at 0x743f110>]



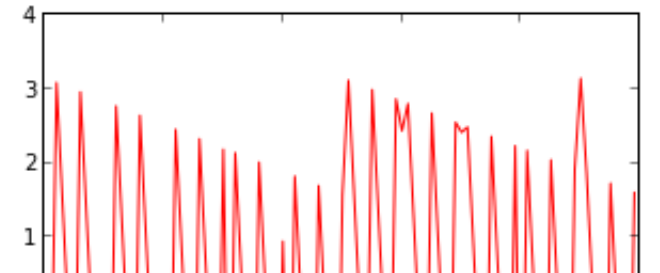

In [141]:
```
plt.plot(np.absolute(resp), 'b')
```

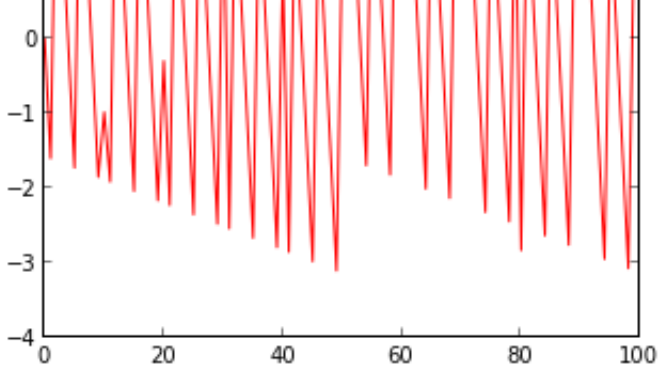
Out[141]: [<matplotlib.lines.Line2D at 0x7b52590>]




In [142]:
```
plt.plot(np.angle(resp), 'r')
```

Out[142]: [<matplotlib.lines.Line2D at 0x7f9db10>]






In [143]:
```
Huzzah!
```

```
  File "<ipython-input-143-7d99cd52236b>", line 1
    Huzzah!
           ^
SyntaxError: invalid syntax
```

In [ ]: