





2D Button Arrays










[row][col]

How many rows?

	0	1	2	3	4
0					
1					
2					
3					















[row][col]

4 rows.

	0	1	2	3	4
0	 0	 1	 2	 3	 4
1	 5	 6	 7	 8	 9
2	 10	 11	 12	 13	 14
3	 15	 16	 17	 18	 19
















[row][col]

How many columns?

	0	1	2	3	4
0					
1					
2					
3					




















[row][col]

5
columns.

	0	1	2	3	4
0	 0	 1	 2	 3	 4
1	 5	 6	 7	 8	 9
2	 10	 11	 12	 13	 14
3	 15	 16	 17	 18	 19





[row][col]

What colour of candy is in [2][1]?

	0	1	2	3	4
0	 0	 1	 2	 3	 4
1	 5	 6	 7	 8	 9
2	 10	 11	 12	 13	 14
3	 15	 16	 17	 18	 19



















[row][col]

[2] = row,
[1] = col.
Blue.

	0	1	2	3	4
0	 0	 1	 2	 3	 4
1	 5	 6	 7	 8	 9
2	 10	 11	 12	 13	 14
3	 15	 16	 17	 18	 19

[row][col]

What colour of candy is in [3][1]?

	0	1	2	3	4
0	 0	 1	 2	 3	 4
1	 5	 6	 7	 8	 9
2	 10	 11	 12	 13	 14
3	 15	 16	 17	 18	 19

















[row][col]

[3] = row,
[1] = col.
Purple.

	0	1	2	3	4
0	 0	 1	 2	 3	 4
1	 5	 6	 7	 8	 9
2	 10	 11	 12	 13	 14
3	 15	 16	 17	 18	 19





[row][col]

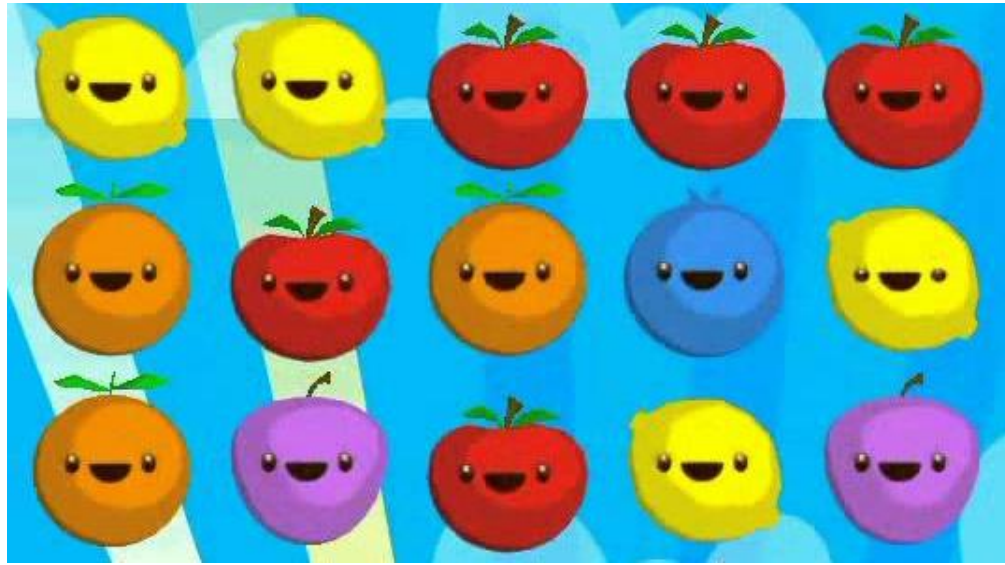
What colour of candy is in [0][3]?
















	0	1	2	3	4
0	 0	 1	 2	 3	 4
1	 5	 6	 7	 8	 9
2	 10	 11	 12	 13	 14
3	 15	 16	 17	 18	 19

[row][col]

[0] = row,
[3] = col.
Red.

	0	1	2	3	4
0	 0	 1	 2	 3	 4
1	 5	 6	 7	 8	 9
2	 10	 11	 12	 13	 14
3	 15	 16	 17	 18	 19



	[0]	[1]	[2]	[3]	[4]
[0]					
[1]					
[2]					





Salad

- ✗ Not Delicious
- ✓ Nutritious



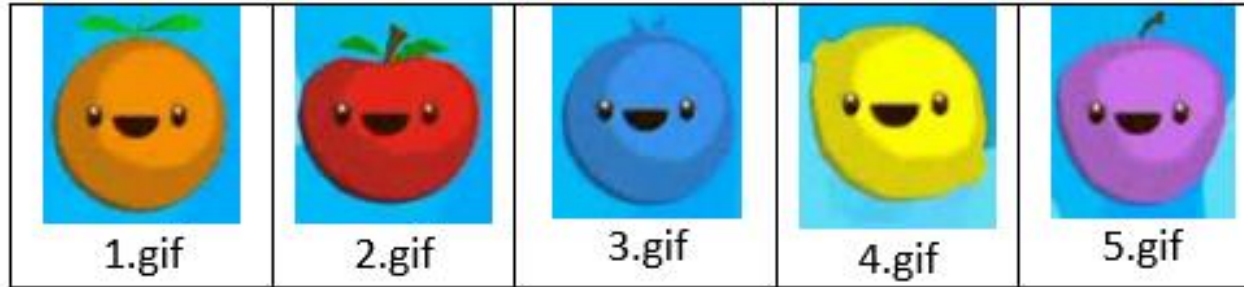
Doughnut

- ✓ Delicious
- ✗ Not nutritious

Trade-off

- A compromise; you must give up one thing to gain another.
- It is not possible to have the best of both worlds.

The Picture Naming Trade-off

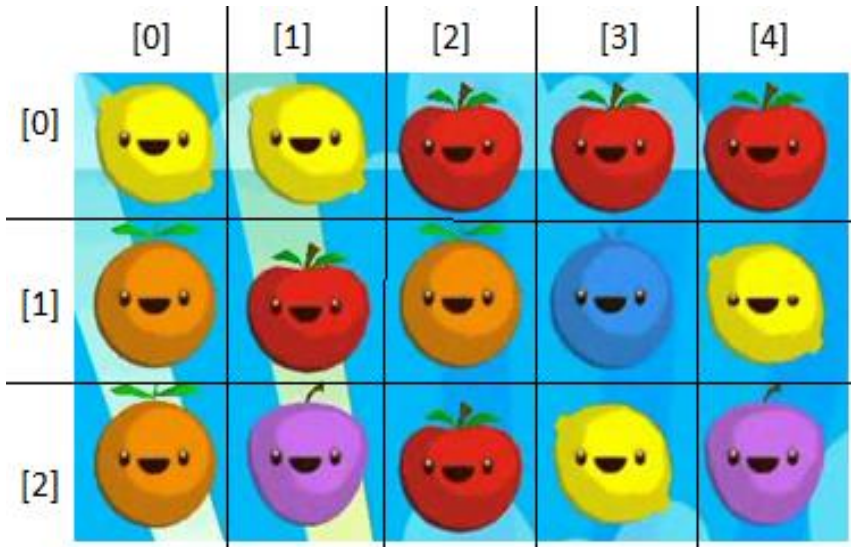
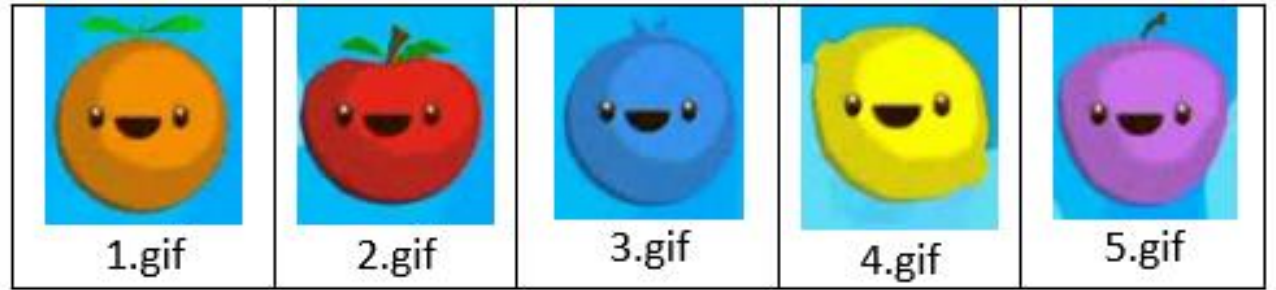


Naming Pictures With a String

- ✓ Meaningful (apple)
- ✗ Strings have to use `.equals()`, it's harder than `==`
- ✗ Strings have double quotes `""`, those are annoying to type
- ✗ Strings are longer to type

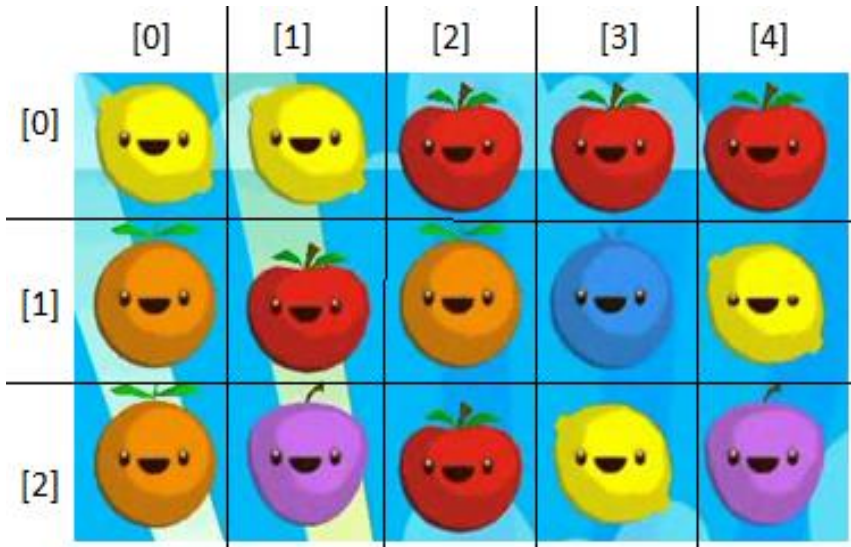
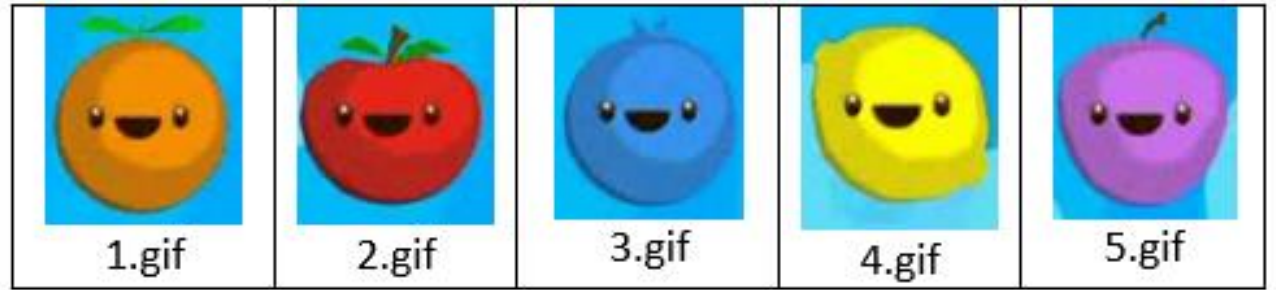
Naming Pictures With an int

- ✗ Not Meaningful (2)
- ✓ Can use `==`, it's easier than `.equals()`
- ✓ No double quotes around anything, easy to type
- ✓ A single number is shorter to type than an entire word



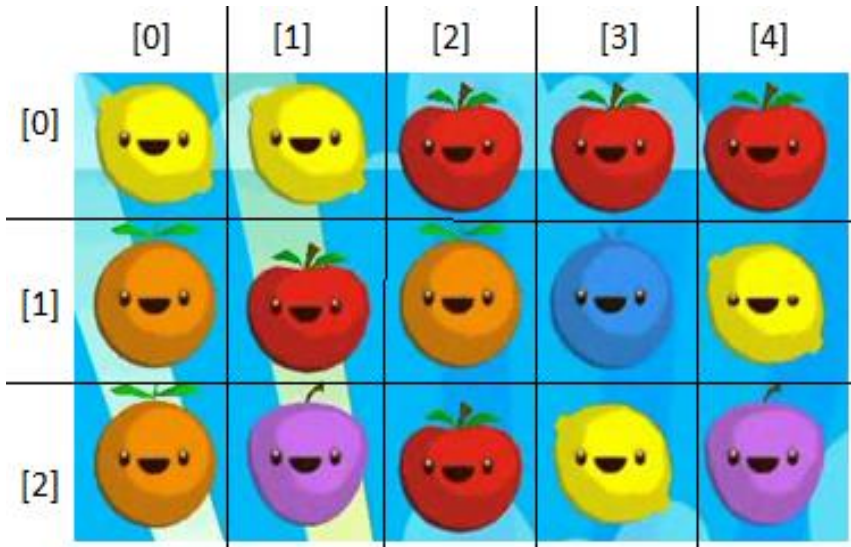
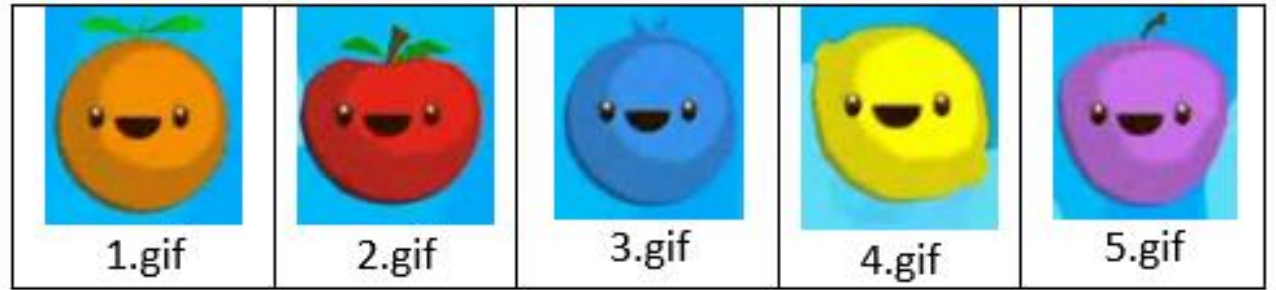
	[0]	[1]	[2]	[3]	[4]
[0]					
[1]					
[2]					

```
int fruit[] [] = {{ 4, 4, 2, __, __},
                  { __, __, __, __, __},
                  { __, __, __, __, __}};
```



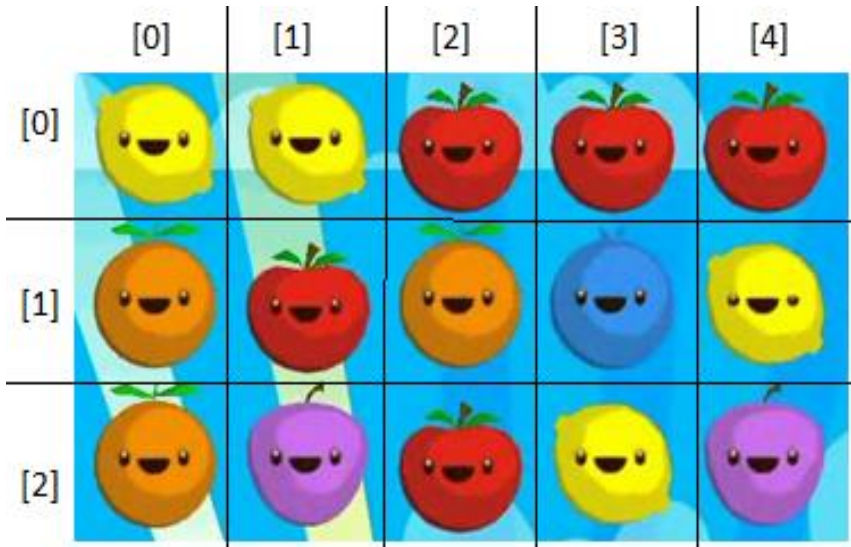
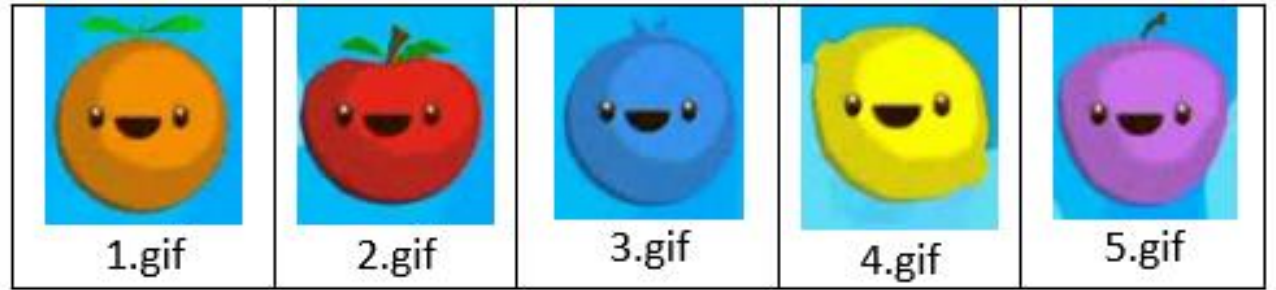
	[0]	[1]	[2]	[3]	[4]
[0]	4	4	2	2	2
[1]					
[2]					

```
int fruit[] [] = {{ 4, 4, 2, __, __},
                  { __, __, __, __, __},
                  { __, __, __, __, __}};
```



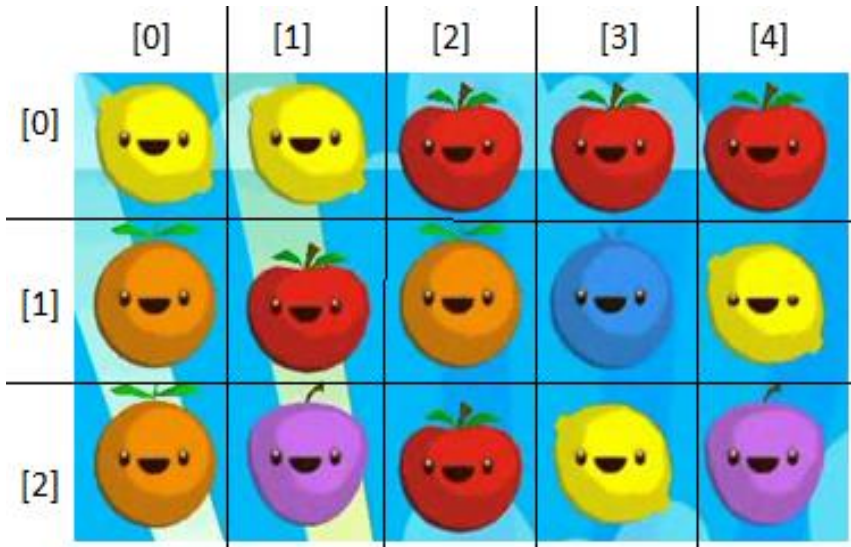
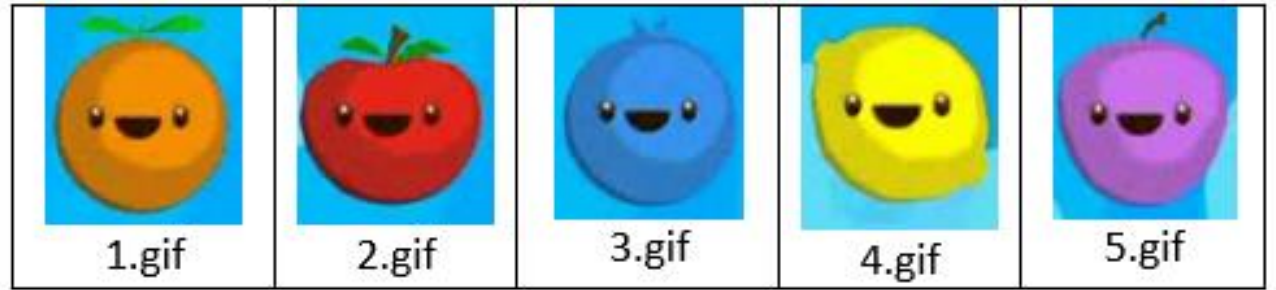
	[0]	[1]	[2]	[3]	[4]
[0]	4	4	2	2	2
[1]	1	2	1	3	4
[2]					

```
int fruit[] [] = {{ 4, 4, 2, __, __},
                  { __, __, __, __, __},
                  { __, __, __, __, __}};
```



	[0]	[1]	[2]	[3]	[4]
[0]	4	4	2	2	2
[1]	1	2	1	3	4
[2]	1	5	2	4	5

```
int fruit[] [] = {{ 4, 4, 2, __, __},
                  { __, __, __, __, __},
                  { __, __, __, __, __}};
```



	[0]	[1]	[2]	[3]	[4]
[0]	4	4	2	2	2
[1]	1	2	1	3	4
[2]	1	5	2	4	5

```
int fruit[] [] = {{ 4, 4, 2, 2, 2 },
                  { 1, 2, 1, 3, 4 },
                  { 1, 5, 2, 4, 5 }};
```

```
public class fruitPop extends Applet implements ActionListener
```

```
{
```

```
int fruit[] [] = {{4, 4, 2, 2, 2},  
                  {1, 2, 1, 3, 4},  
                  {1, 5, 2, 4, 5}};
```

```
int rows = __ ;
```

```
int cols = __ ;
```

```
JButton pics[] =  
    new JButton [rows * cols];
```



```
public void init ()
```

```
{
```

```
Panel grid = new Panel (new GridLayout (rows, cols));
```

```
int m = 0;
```

```
for (int i = 0 ; i < _____ ; i++)
```

```
{
```

```
for (int j = 0 ; j < _____ ; j++)
```

```
{
```

```
pics [m] = new JButton (createImageIcon ( _____ [i] [j] + ".png"));
```

```
pics [m].addActionListener (this);
```

```
pics [m].setActionCommand (m + "");
```

```
grid.add (pics [m]);
```

```
m++;
```

```
}
```

```
}
```

```
add (grid);
```

```
}
```



```
public class fruitPop extends Applet implements ActionListener
```

```
{
```

```
int fruit[] [] = {{4, 4, 2, 2, 2},  
                  {1, 2, 1, 3, 4},  
                  {1, 5, 2, 4, 5}};
```

```
int rows = 3;
```

```
int cols = 5;
```

```
JButton pics[] =  
    new JButton [rows * cols];
```



```
public void init ()
```

```
{
```

```
Panel grid = new Panel (new GridLayout (rows, cols));
```

```
int m = 0;
```

```
for (int i = 0 ; i < _____ ; i++)
```

```
{
```

```
for (int j = 0 ; j < _____ ; j++)
```

```
{
```

```
pics [m] = new JButton (createImageIcon ( _____ [i] [j] + ".png"));
```

```
pics [m].addActionListener (this);
```

```
pics [m].setActionCommand (m + "");
```

```
grid.add (pics [m]);
```

```
m++;
```

```
}
```

```
}
```

```
add (grid);
```

```
}
```



```
public class fruitPop extends Applet implements ActionListener
```

```
{  
    int fruit[] [] = {{4, 4, 2, 2, 2},  
                      {1, 2, 1, 3, 4},  
                      {1, 5, 2, 4, 5}};
```

```
    int rows = 3;
```

```
    int cols = 5;
```

```
    JButton pics[] =  
        new JButton [rows * cols];
```



```
    public void init ()
```

```
    {
```

```
        Panel grid = new Panel (new GridLayout (rows, cols));
```

```
        int m = 0;
```

```
        for (int i = 0 ; i < row ; i++)
```

```
        {
```

```
            for (int j = 0 ; j < col ; j++)
```

```
            {
```

```
                pics [m] = new JButton (createImageIcon ( _____ [i] [j] + ".png"));
```

```
                pics [m].addActionListener (this);
```

```
                pics [m].setActionCommand (m + "");
```

```
                grid.add (pics [m]);
```

```
                m++;
```

```
            }
```

```
        }
```

```
        add (grid);
```

```
    }
```



```
public class fruitPop extends Applet implements ActionListener
```

```
{
```

```
int fruit[] [] = {{4, 4, 2, 2, 2},  
                  {1, 2, 1, 3, 4},  
                  {1, 5, 2, 4, 5}};
```

```
int rows = 3;
```

```
int cols = 5;
```

```
JButton pics[] =  
    new JButton [rows * cols];
```



```
public void init ()
```

```
{
```

```
Panel grid = new Panel (new GridLayout (rows, cols));
```

```
int m = 0;
```

```
for (int i = 0 ; i < row ; i++)
```

```
{
```

```
for (int j = 0 ; j < col ; j++)
```

```
{
```

```
pics [m] = new JButton (createImageIcon (fruit [i] [j] + ".png"));
```

```
pics [m].addActionListener (this);
```

```
pics [m].setActionCommand (m + "");
```

```
grid.add (pics [m]);
```

```
m++;
```

```
}
```

```
}
```

```
add (grid);
```

```
}
```



```
import javax.swing.*; import java.applet.Applet; import java.awt.*; import java.awt.event.*;
public class grid extends Applet implements ActionListener
{
    int row = 4;
    JButton a[] = new JButton [row * row];

    public void init ()
    {
        Panel g = new Panel (new GridLayout (row, row));
        for (int i = 0 ; i < a.length ; i++)
        {
            a [i] = new JButton (" ");
            a [i].addActionListener (this);
            a [i].setActionCommand (" " + i);
            g.add (a [i]);
        }
        add (g);
        resize (180, 150);
    }

    public void actionPerformed (ActionEvent e)
    {
        int n = Integer.parseInt (e.getActionCommand ());
        int x = n / row;
        int y = n % row;
        showStatus ("(" + x + ", " + y + ")");
    }
}
```

[row][col]
[2][1] = blue candy

pos / col = x
 $11 / 5 = 2$

pos % col = y
 $11 \% 5 = 1$



[row][col]
[2][3] = green candy

pos / col = x
 $13 / 5 = 2$

pos % col = y
 $13 \% 5 = 3$



[row][col]
[3][1] = purple candy

pos / col = x
16 / 5 = 3

pos % col = y
16 % 5 = 1

