Announcements

- After Image Survey – it’s part of our commitment to the AP experiment ... your help is REALLY appreciated
Naturally, programs are given sequentially, the declarations at the top.

Braces {} are statement groupers ... they make a sequence of statements into one thing, like the “true clause of an If-statement”.

All statements must end with a semicolon EXCEPT the grouping braces ... they don’t end with a semicolon (OK, it’s a rare inconsistency about computer languages!)

Generally white space doesn’t matter; be neat!
Program Execution

- Keep in mind how a program executes

```java
int next=1;

void setup() {
    size(100,100);
    fill(255, 0,0);
}

void draw() {
    background(0);
    rect(mouseX, mouseY, 25, 25)
}

void mousePressed() {
    if (next == 1) {
        fill(0, 0, 255); // go to blue
    } else {
        fill(255,0,0); // go to red
    }
    next=1-next;
}
```
Art Programs Raise Deep Questions

Mondrian, Pollack, Albers are stars ...
Adding some light to computing ....

Bits of Color

Lawrence Snyder
University of Washington, Seattle
Recall that the screen (and other video displays) use red-green-blue lights, arranged in an array of picture elements, or *pixels*.
Actual Pixels From TFT LCD Display
Combining Colored Light

- The Amazing Properties of Colored Light!

Caution: It doesn’t work like pigment
Colored light seems to violate our grade school rule of green = blue + yellow. What gives?

In pigment, the color we see is the reflected color from white light; the other colors are absorbed.
You know that gray is just different degrees of white as the “light is turned down” till we get to black

Black  = [ 0, 0, 0] 0000 0000 0000 0000 0000 0000
Gray    = [128,128,128] 1000 0000 1000 0000 1000 0000
White   = [255,255,255] 1111 1111 1111 1111 1111 1111

White-gray-black all have same values for RGB
Colors use different combinations of RGB

Husky Purple
Red=160
Green=76
Blue=230
The RGB intensities are binary numbers.
Binary numbers, like decimal numbers, use place notation.

\[ 1101 = 1 \times 1000 + 1 \times 100 + 0 \times 10 + 1 \times 1 \]

\[ = 1 \times 10^3 + 1 \times 10^2 + 0 \times 10^1 + 1 \times 10^0 \]
except that the base is 2 not 10.

\[ 1101 = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 \]

\[ = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \]

1101 in binary is 13 in decimal.
Recall that the place represents a power of the base value

\[ \begin{align*}
    d_7 \times 10^7 & & d_7 \times 2^7 \\
    d_6 \times 10^6 & & d_6 \times 2^6 \\
    d_5 \times 10^5 & & d_5 \times 2^5 \\
    d_4 \times 10^4 & & d_4 \times 2^4 \\
    d_3 \times 10^3 & & d_3 \times 2^3 \\
    d_2 \times 10^2 & & d_2 \times 2^2 \\
    d_1 \times 10^1 & & d_1 \times 2^1 \\
    d_0 \times 10^0 & & d_0 \times 2^0
\end{align*} \]
The Red of HP As A Binary Number

Given a binary number, add up the powers of 2 corresponding to 1s

```
1 \times 2^7 = 1 \times 128 = 128
0 \times 2^6 = 0 \times 64 = 0
1 \times 2^5 = 1 \times 32 = 32
0 \times 2^4 = 0 \times 16 = 0
0 \times 2^3 = 0 \times 8 = 0
0 \times 2^2 = 0 \times 4 = 0
0 \times 2^1 = 0 \times 2 = 0
0 \times 2^0 = 0 \times 1 = 0
```

101000000 = 160
Green of HP As A Binary Number

Given a binary number, add up the powers of 2 corresponding to 1s

\[
\begin{align*}
0 \times 2^7 &= 1 \times 128 &= 0 \\
1 \times 2^6 &= 0 \times 64 &= 64 \\
0 \times 2^5 &= 1 \times 32 &= 0 \\
0 \times 2^4 &= 0 \times 16 &= 0 \\
1 \times 2^3 &= 0 \times 8 &= 8 \\
1 \times 2^2 &= 0 \times 4 &= 4 \\
0 \times 2^1 &= 0 \times 2 &= 0 \\
0 \times 2^0 &= 0 \times 1 &= 0 \\
\hline
\text{=76}
\end{align*}
\]
So Husky purple is (160, 76, 230) which is

```
1010 0000 0100 1100 1110 0110
160    76    230
```

Suppose you decide it’s not “red” enough

- Increase the red by 16 = 1 0000

```
1010 0000
+ 1 0000
1011 0000
```

Adding in binary is pretty much like adding in decimal
Increase by 16 more

\[
\begin{align*}
00110 & 000 & \text{Carries} \\
1011 & 0000 \\
+ & 1 0000 \\
1100 & 0000
\end{align*}
\]

The rule: When the “place sum” equals the radix or more, subtract radix & carry

Check it out online: searching binary addition hits 19M times, and all of the p.1 hits are good explanations
What is 230 (the Blue of HP)? Fill in the Table:

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
<th>6</th>
<th>6</th>
<th>6</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td>6</td>
<td>6</td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

© 2010 Larry Snyder, CSE
Place number to be converted into the table; fill place value row with decimal powers of 2

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Subtract</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Subtract</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Find Binary From Decimal

Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>(\text{Num Being Converted})</th>
<th>230</th>
<th>230</th>
<th>102</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
</tr>
<tr>
<td>Subtract</td>
<td></td>
<td>102</td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Find Binary From Decimal

Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Find Binary From Decimal

Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

© 2010 Larry Snyder, CSE
Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
<th>6</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

© 2010 Larry Snyder, CSE
Find Binary From Decimal

Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
<th>6</th>
<th>6</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
## Find Binary From Decimal

Rule: Subtract PV from the number; a positive result gives new number and "1"; otherwise, "0".

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
<th>6</th>
<th>6</th>
<th>6</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td>6</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Find Binary From Decimal

Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
<th>6</th>
<th>6</th>
<th>6</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>
Rule: Subtract PV from the number; a positive result gives new number and “1”; otherwise, “0”

<table>
<thead>
<tr>
<th>Num Being Converted</th>
<th>230</th>
<th>230</th>
<th>102</th>
<th>38</th>
<th>6</th>
<th>6</th>
<th>6</th>
<th>2</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Place Value</td>
<td>256</td>
<td>128</td>
<td>64</td>
<td>32</td>
<td>16</td>
<td>8</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Subtract</td>
<td>102</td>
<td>38</td>
<td>6</td>
<td></td>
<td>2</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Binary Num</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Read off the result: 0 1110 0110